QGIS Desktop 3.40 User Guide

QGIS Project

aug. 16, 2025

Innehåll

1 Komma igång **3**
- 1.1 Installera QGIS . . . 3
  - 1.1.1 Installera från binärfiler . . . 3
  - 1.1.2 Installera från källan . . . 3
  - 1.1.3 Installera på extern media . . . 4
  - 1.1.4 Nedladdning av exempeldata . . . 4
- 1.2 Starta och stoppa QGIS . . . 4
- 1.3 Exempel på session: Ladda raster- och vektorlager . . . 5

2 Arbeta med projektfiler **13**
- 2.1 Introduktion till QGIS-projekt . . . 13
- 2.2 Hantering av brutna filsökvägar . . . 15
- 2.3 Generering av utdata . . . 16

3 QGIS GUI **17**
- 3.1 Menyrad . . . 18
  - 3.1.1 Projekt . . . 18
  - 3.1.2 Redigera . . . 19
  - 3.1.3 Vy . . . 22
  - 3.1.4 Lager . . . 24
  - 3.1.5 Inställningar . . . 27
  - 3.1.6 Tillägg . . . 27
  - 3.1.7 Vektor . . . 27
  - 3.1.8 Raster . . . 29
  - 3.1.9 Databas . . . 31
  - 3.1.10 Webb . . . 31
  - 3.1.11 Nät . . . 31
  - 3.1.12 Bearbetas . . . 32
  - 3.1.13 Hjälp . . . 32
  - 3.1.14 QGIS . . . 32
- 3.2 Paneler och verktygsfält . . . 33
  - 3.2.1 Verktygsfält . . . 33
  - 3.2.2 Paneler . . . 34
- 3.3 Statusrad . . . 34
  - 3.3.1 Locator bar . . . 34
  - 3.3.2 Rapportering av åtgärder . . . 35
  - 3.3.3 Kontrollera kartans canvas . . . 35
  - 3.3.4 Meddelanden . . . 36

4 Bläddrarpanelen **37**
- 4.1 Resurser som kan öppnas/köras från bläddraren . . . 39

4.2 Bläddrarpanelens toppnivåposter . . . 40
4.2.1 Favoriter . . . 40
4.2.2 Spatiala bokmärken . . . 40
4.2.3 Projektets hem . . . 40
4.2.4 Hårddiskar och filsystem . . . 40
4.2.5 Databasposter . . . 41
4.2.6 Kakelplattor och webbtjänster . . . 44
4.3 Resurser . . . 45

5 QGIS-konfiguration **47**
5.1 Alternativ . . . 47
5.1.1 Allmänna inställningar . . . 48
5.1.2 Systeminställningar . . . 50
5.1.3 Inställningar för användarprofiler . . . 52
5.1.4 CRS- och transformationsinställningar . . . 52
5.1.5 Inställningar för datakällor . . . 56
5.1.6 Renderingsinställningar . . . 60
5.1.7 Inställningar för Canvas och Legend . . . 64
5.1.8 Kartverktyg Inställningar . . . 66
5.1.9 Inställningar för höjd . . . 70
5.1.10 3D-inställningar . . . 70
5.1.11 Färginställningar . . . 71
5.1.12 Inställningar för teckensnitt . . . 72
5.1.13 Inställningar för layouter . . . 73
5.1.14 Inställningar för variabler . . . 74
5.1.15 Inställningar för autentisering . . . 75
5.1.16 Nätverksinställningar . . . 76
5.1.17 GPS-inställningar . . . 77
5.1.18 Inställningar för sökare . . . 79
5.1.19 Inställningar för acceleration . . . 81
5.1.20 IDE-inställningar . . . 82
5.1.21 Inställningar för bearbetning . . . 84
5.1.22 Avancerade inställningar . . . 86
5.2 Arbeta med användarprofiler . . . 87
5.2.1 Konceptet . . . 87
5.2.2 Ställa in användarprofil . . . 88
5.3 Projektets egenskaper . . . 89
5.3.1 Allmänna egenskaper . . . 89
5.3.2 Egenskaper för metadata . . . 91
5.3.3 Visa inställningar . . . 92
5.3.4 CRS-egenskaper . . . 92
5.3.5 Transformationer Egenskaper . . . 93
5.3.6 Stilegenskaper . . . 93
5.3.7 Färger egenskaper . . . 94
5.3.8 Egenskaper för datakällor . . . 95
5.3.9 Relationer Egenskaper . . . 97
5.3.10 Variabelegenskaper . . . 98
5.3.11 Egenskaper för makron . . . 98
5.3.12 Egenskaper för QGIS-server . . . 99
5.3.13 Temporala egenskaper . . . 100
5.3.14 Elevation Egenskaper . . . 100
5.3.15 Sensorer Egenskaper . . . 103
5.4 Anpassning . . . 104
5.5 Tangentbordsgenvägar . . . 106
5.6 Kör QGIS med avancerade inställningar . . . 107
5.6.1 Kommandorad och miljövariabler . . . 107
5.6.2 Distribuera QGIS inom en organisation . . . 113

6 Arbeta med projektioner **115**
6.1 Översikt över projektionsstöd . . . 115
6.2 Referenssystem för lagerkoordinater . . . 115
6.3 Referenssystem för projektkoordinater . . . 117
6.4 Väljare för referenssystem för koordinater . . . 119
6.5 Anpassat referenssystem för koordinater . . . 120
6.5.1 Integrera en NTv2-transformation i QGIS . . . 121
6.6 Datumtransformationer . . . 122

7 Visualisering av kartor **125**
7.1 2D-kartvy . . . 125
7.1.1 Utforska kartvyn . . . 126
7.1.2 Styrning av kartrendering . . . 127
7.1.3 Tidsbaserad kontroll på kartbilden . . . 128
7.1.4 Panelen för höjdkontroll . . . 130
7.1.5 Lägga till bokmärken för extenter på kartan . . . 131
7.1.6 Dekorera kartan . . . 133
7.1.7 Verktyg för anteckningar . . . 140
7.1.8 Mätning . . . 147
7.1.9 Ställa in ytterligare kartvyer . . . 149
7.1.10 Exportera kartvyn . . . 151
7.2 3D-kartvy . . . 153
7.2.1 Konfiguration av scen . . . 155
7.2.2 Alternativ för navigering . . . 162
7.2.3 Skapa en animation . . . 163
7.2.4 3D-vektorlager . . . 163
7.3 Höjdprofil Vy . . . 163
7.3.1 Gränssnittet . . . 164
7.3.2 Skapa en höjdprofil . . . 166
7.3.3 Interagera med profilen Vyer . . . 167
7.3.4 Exportera från höjdprofilen . . . 168

8 Allmänna verktyg **169**
8.1 Hjälp med kontext . . . 169
8.2 Paneler . . . 169
8.2.1 Panel för lager . . . 169
8.2.2 Panel för lagerstyling . . . 177
8.2.3 Panel för beställning av lager . . . 179
8.2.4 Översiktspanel . . . 179
8.2.5 Panel för loggmeddelanden . . . 180
8.2.6 Undo/Redo-panel . . . 180
8.2.7 Panel för statistisk sammanfattning . . . 180
8.2.8 Panel för felsöknings-/utvecklingsverktyg . . . 181
8.3 Inbäddning av lager från externa projekt . . . 184
8.4 Interagera med funktioner . . . 185
8.4.1 Välja funktioner . . . 185
8.4.2 Identifiering av funktioner . . . 189
8.5 Spara och dela egenskaper för lager . . . 194
8.5.1 Hantera anpassade stilar . . . 194
8.5.2 Lagra stilar i en fil eller en databas . . . 195
8.5.3 Fil för definition av lager . . . 198
8.6 Dokumentera dina data . . . 198
8.6.1 Metadata . . . 198
8.6.2 Anteckningar om lager . . . 199
8.7 Lagring av värden i variabler . . . 200
8.8 Autentisering . . . 201
8.9 Vanliga widgetar . . . 202
8.9.1 Färgväljare . . . 202

8.9.2 Widget för symboler . . . 206
8.9.3 Väljare för fjärrstyrd eller inbäddad fil . . . 206
8.9.4 Väljare för synlighetsskala . . . 207
8.9.5 Väljare för spatial utsträckning . . . 207
8.9.6 Typsnittsväljare . . . 208
8.9.7 Väljare för enhet . . . 209
8.9.8 Nummerformatering . . . 210
8.9.9 Blandningslägen . . . 211
8.9.10 Inställning av datadefinierad åsidosättning . . . 213

9 Lyft dig själv med uttryck **217**
9.1 Uttryck . . . 217
9.1.1 Strängbyggaren Expression . . . 217
9.1.2 Funktion Redaktör . . . 222
9.2 Lista över funktioner . . . 224
9.2.1 Aggregeringsfunktioner . . . 225
9.2.2 Array-funktioner . . . 235
9.2.3 Färgfunktioner . . . 247
9.2.4 Villkorliga funktioner . . . 256
9.2.5 Konverteringar Funktioner . . . 259
9.2.6 Anpassade funktioner . . . 264
9.2.7 Datum- och tidsfunktioner . . . 264
9.2.8 Fält och värden . . . 275
9.2.9 Funktioner för filer och sökvägar . . . 275
9.2.10 Formulär Funktioner . . . 278
9.2.11 Fuzzy matchningsfunktioner . . . 279
9.2.12 Allmänna funktioner . . . 280
9.2.13 Geometri Funktioner . . . 282
9.2.14 Layoutfunktioner . . . 357
9.2.15 Kartlager . . . 358
9.2.16 Kartor Funktioner . . . 361
9.2.17 Matematiska funktioner . . . 365
9.2.18 Meshes Funktioner . . . 373
9.2.19 Operatörer . . . 375
9.2.20 Bearbetningsfunktioner . . . 384
9.2.21 Rasterfunktioner . . . 384
9.2.22 Funktioner för registrering och attribut . . . 386
9.2.23 Relationer . . . 395
9.2.24 Sensorer Funktioner . . . 395
9.2.25 Strängfunktioner . . . 395
9.2.26 Användaruttryck . . . 406
9.2.27 Variabler . . . 406
9.2.28 Senaste funktionerna . . . 410

10 Stilbiblioteket **411**
10.1 Den stilansvarige . . . 411
10.1.1 Dialogrutan Stilhanterare . . . 411
10.1.2 Inställning av en färgramp . . . 417
10.1.3 Skapa en Legend Patch Shape . . . 419
10.2 Symbolväljaren . . . 421
10.2.1 Symbollagrets träd . . . 422
10.2.2 Konfigurera en symbol . . . 423
10.3 Ställa in en etikett . . . 441
10.3.1 Formatering av etikettexten . . . 443
10.3.2 Konfigurera interaktion med etiketter . . . 452
10.4 Skapa 3D-symboler . . . 462
10.4.1 Punktlager . . . 463
10.4.2 Linjelager . . . 464

10.4.3 Polygonlager . . . 465
10.4.4 Skuggning av texturen . . . 467
10.4.5 Exempel på tillämpning . . . 467

11 Hantera datakälla **469**
11.1 Öppningsdata . . . 469
11.1.1 Bläddrarpanelen . . . 471
11.1.2 DB-chefen . . . 474
11.1.3 Leverantörsbaserade laddningsverktyg . . . 475
11.1.4 Panelen för sökning av metadata för lager . . . 495
11.1.5 QGIS anpassade format . . . 496
11.1.6 QLR - Definitionsfil för QGIS-lager . . . 496
11.1.7 Ansluta till webbtjänster . . . 496
11.2 Skapa lager . . . 503
11.2.1 Skapa nya vektorlager . . . 504
11.2.2 Skapa nya lager från ett befintligt lager . . . 512
11.2.3 Skapa nya DXF-filer . . . 517
11.2.4 Skapa nya lager från urklipp . . . 518
11.2.5 Skapa virtuella lager . . . 519
11.3 Georeferens . . . 521
11.3.1 Georefererande rasterlager . . . 522
11.3.2 Vektorlager för georeferenser . . . 527
11.3.3 Visa och anpassa lagrets egenskaper . . . 528
11.3.4 Konfigurera georeferensen . . . 528
11.4 Utforska dataformat och fält . . . 528
11.4.1 Rasterdata . . . 528
11.4.2 Vektordata . . . 529
11.4.3 Lager som korsar 180° longitud . . . 537

12 Arbeta med vektordata **539**
12.1 Dialogrutan Egenskaper för vektor . . . 539
12.1.1 Informationsegenskaper . . . 540
12.1.2 Källegenskaper . . . 541
12.1.3 Egenskaper för symbologi . . . 544
12.1.4 Etiketter Egenskaper . . . 566
12.1.5 Diagram Egenskaper . . . 578
12.1.6 Masker Egenskaper . . . 583
12.1.7 egenskaper för 3D-vy . . . 584
12.1.8 Fältegenskaper . . . 586
12.1.9 Attribut Formulär Egenskaper . . . 587
12.1.10 Skarvar Egenskaper . . . 597
12.1.11 Egenskaper för hjälplagring . . . 597
12.1.12 Åtgärder Egenskaper . . . 605
12.1.13 Egenskaper för display . . . 610
12.1.14 Rendering av egenskaper . . . 613
12.1.15 Temporala egenskaper . . . 615
12.1.16 Variabler Egenskaper . . . 616
12.1.17 Elevation Egenskaper . . . 616
12.1.18 Egenskaper för metadata . . . 618
12.1.19 Egenskaper för beroenden . . . 618
12.1.20 Förklaringsegenskaper . . . 618
12.1.21 Egenskaper för QGIS-server . . . 619
12.1.22 Egenskaper för digitalisering . . . 621
12.2 Arbeta med attributtabellen . . . 623
12.2.1 Förord: Spatiala och icke-spatiala tabeller . . . 623
12.2.2 Introduktion till gränssnittet för attributtabeller . . . 624
12.2.3 Interagera med funktioner i en attributtabell . . . 629
12.2.4 Redigera attributvärden . . . 633

12.2.5 Utforska funktionsattribut med hjälp av Identify Tool . . . 637
12.2.6 Lagring och hämtning av en extern resurs . . . 638
12.3 Ändrar . . . 640
12.3.1 Inställning av snäpptolerans och sökradie . . . 641
12.3.2 Alternativ för snäppning och digitalisering . . . 641
12.3.3 Topologisk redigering . . . 644
12.3.4 Digitalisering av ett befintligt lager . . . 646
12.3.5 Avancerad digitalisering . . . 656
12.3.6 Digitalisering av form . . . 665
12.3.7 Panelen för avancerad digitalisering . . . 668
12.3.8 Modifieraren för bearbetning av lager på plats . . . 676
12.4 Koppla samman och redigera data i olika lager . . . 678
12.4.1 Sammanfogning av funktioner mellan två lager . . . 678
12.4.2 Inställning av relationer mellan flera lager . . . 680

13 Arbeta med rasterdata **695**
13.1 Dialog för egenskaper för raster . . . 695
13.1.1 Informationsegenskaper . . . 696
13.1.2 Källegenskaper . . . 696
13.1.3 Egenskaper för symbologi . . . 697
13.1.4 Egenskaper för genomskinlighet . . . 707
13.1.5 Egenskaper för histogram . . . 709
13.1.6 Rendering av egenskaper . . . 710
13.1.7 Temporala egenskaper . . . 711
13.1.8 Pyramidegenskaper . . . 713
13.1.9 Elevation Egenskaper . . . 714
13.1.10 Egenskaper för metadata . . . 716
13.1.11 Förklaringsegenskaper . . . 716
13.1.12 Egenskaper för display . . . 717
13.1.13 Egenskaper för QGIS-server . . . 718
13.1.14 Identifiera rasterceller . . . 719
13.2 Rasteranalys . . . 720
13.2.1 Rasterkalkylator . . . 720

14 Arbeta med Mesh-data **725**
14.1 Vad är ett nät? . . . 725
14.2 Format som stöds . . . 727
14.3 Egenskaper för nätdataset . . . 728
14.3.1 Informationsegenskaper . . . 729
14.3.2 Källegenskaper . . . 730
14.3.3 Egenskaper för symbologi . . . 731
14.3.4 Etiketter Egenskaper . . . 737
14.3.5 egenskaper för 3D-vy . . . 739
14.3.6 Rendering av egenskaper . . . 741
14.3.7 Temporala egenskaper . . . 742
14.3.8 Elevation Egenskaper . . . 742
14.3.9 Egenskaper för metadata . . . 744
14.4 Redigera ett mesh-lager . . . 744
14.4.1 Översikt över digitaliseringsverktygen för nät . . . 744
14.4.2 Utforska logiken för tilldelning av Z-värde . . . 745
14.4.3 Välja nätelement . . . 746
14.4.4 Modifiera nätelement . . . 747
14.4.5 Omindexering av maskor . . . 751
14.5 Mesh-kalkylator . . . 751

15 Arbeta med Vector Tiles **755**
15.1 Vad är Vector Tiles? . . . 755
15.2 Format som stöds . . . 756
15.3 Vector Tiles Dataset Egenskaper . . . 757

15.3.1 Informationsegenskaper . . . 757
15.3.2 Källegenskaper . . . 757
15.3.3 Symbologi och etiketter Egenskaper . . . 758
15.3.4 Rendering av egenskaper . . . 759
15.3.5 Egenskaper för metadata . . . 760

16 Arbeta med punktmoln **761**
16.1 Introduktion till punktmoln . . . 761
16.2 Punktmoln Egenskaper . . . 762
16.2.1 Informationsegenskaper . . . 762
16.2.2 Källegenskaper . . . 763
16.2.3 Egenskaper för symbologi . . . 766
16.2.4 egenskaper för 3D-vy . . . 774
16.2.5 Rendering av egenskaper . . . 776
16.2.6 Elevation Egenskaper . . . 776
16.2.7 Egenskaper för metadata . . . 778
16.2.8 Statistik Egenskaper . . . 778
16.3 Virtuellt punktmoln . . . 779

17 Arbeta med 3D-plattor **781**
17.1 Vad är 3D-plattor? . . . 781
17.2 3D Tiles Egenskaper . . . 782
17.2.1 Informationsegenskaper . . . 783
17.2.2 Källegenskaper . . . 783
17.2.3 Egenskaper för symbologi . . . 783
17.2.4 egenskaper för 3D-vy . . . 785
17.2.5 Rendering av egenskaper . . . 785
17.2.6 Elevation Egenskaper . . . 785
17.2.7 Egenskaper för metadata . . . 785

18 Lägga ut kartorna **787**
18.1 Översikt över utskriftslayouten . . . 787
18.1.1 Exempelsession för nybörjare . . . 787
18.1.2 Layoutansvarig . . . 788
18.1.3 Menyer, verktyg och paneler i utskriftslayouten . . . 789
18.2 Layoutobjekt . . . 803
18.2.1 Layoutobjekt Gemensamma alternativ . . . 803
18.2.2 Kartobjektet . . . 809
18.2.3 Objektet 3D-karta . . . 819
18.2.4 Etikettobjektet . . . 820
18.2.5 Det legendariska föremålet . . . 825
18.2.6 Objektet för skalfältet . . . 832
18.2.7 Tabellobjekten . . . 836
18.2.8 Markören, bilden och föremålen med norrpil . . . 845
18.2.9 Höjdprofilens objekt . . . 850
18.2.10 HTML-ramobjektet . . . 851
18.2.11 De formade föremålen . . . 854
18.3 Skapa en utdata . . . 857
18.3.1 Inställningar för export . . . 857
18.3.2 Exportera som bild . . . 858
18.3.3 Exportera som SVG . . . 859
18.3.4 Exportera som PDF . . . 860
18.3.5 Skapa en Atlas . . . 862
18.4 Skapa en rapport . . . 867
18.4.1 Vad är det? . . . 867
18.4.2 Kom igång . . . 868
18.4.3 Layout Rapport Arbetsyta . . . 869
18.4.4 Inställningar för export . . . 884

19 Arbeta med OGC/ISO-protokoll **885**
19.1 WMS/WMTS-klient . . . 886
19.1.1 Översikt över WMS-support . . . 886
19.1.2 Översikt över WMTS-stöd . . . 886
19.1.3 Välja WMS/WMTS-servrar . . . 887
19.1.4 Laddning av WMS/WMTS-lager . . . 889
19.1.5 Mosaiker . . . 891
19.1.6 Använda identifieringsverktyget . . . 892
19.1.7 Visning av egenskaper . . . 893
19.1.8 Visa WMS legendgrafik i innehållsförteckningen och i layouten . . . 896
19.2 WCS-klient . . . 897
19.3 WFS och WFS-T Klient . . . 897
19.4 SensorThings . . . 900
19.4.1 Inställning av anslutning . . . 900
19.4.2 Ladda SensorThings data . . . 902
19.4.3 Arbeta med ett vektorlager från SensorThings . . . 903

20 Arbeta med GPS-data **905**
20.1 Introduktion av GNSS/GPS-data . . . 905
20.1.1 Vad är GPS? . . . 905
20.1.2 Överföra eller ladda GPS-data . . . 906
20.2 GPS-spårning i realtid . . . 908
20.2.1 GPS-verktygsfält . . . 908
20.2.2 GPS-informationspanel . . . 910
20.2.3 Anslut till en Bluetooth GPS för spårning i realtid . . . 912
20.2.4 Exempel på anslutning av GPS-enheter i QGIS . . . 912

21 System för autentisering **915**
21.1 Översikt över autentiseringssystem . . . 915
21.1.1 Autentiseringsdatabas . . . 916
21.1.2 Anpassade autentiseringsdatabaser . . . 916
21.1.3 Huvudlösenord . . . 917
21.1.4 Konfigurationer för autentisering . . . 918
21.1.5 Autentiseringsmetoder . . . 921
21.1.6 Verktyg för Master Password och Auth Config . . . 925
21.1.7 Använda autentiseringskonfigurationer . . . 926
21.1.8 Python-bindningar . . . 927
21.2 Arbetsflöden för användarautentisering . . . 927
21.2.1 HTTP(S)-autentisering . . . 927
21.2.2 Autentisering av databas . . . 928
21.2.3 PKI-autentisering . . . 929
21.2.4 Hantering av dåliga lager . . . 936
21.2.5 Ändra konfigurations-ID för autentisering . . . 937
21.2.6 Stöd för QGIS Server . . . 938
21.2.7 Undantag för SSL-server . . . 938

22 GRASS GIS-integration **943**
22.1 Demo dataset . . . 943
22.2 Ladda GRASS raster- och vektorlager . . . 943
22.3 Importera data till en GRASS LOCATION via dra och släpp . . . 944
22.4 Hantera GRASS-data i QGIS Browser . . . 944
22.5 GRASS Alternativ . . . 944
22.6 Starta tillägget GRASS . . . 944
22.7 Öppna GRASS mapset . . . 945
22.8 GRASS LOCATION och MAPSET . . . 945
22.9 Importera data till en GRASS-lokalisering . . . 945
22.9.1 Skapa en ny GRASS LOCATION . . . 946
22.9.2 Lägga till en ny MAPSET . . . 947
22.10 GRASS-vektordatamodellen . . . 948

22.11 Skapa ett nytt GRASS-vektorlager . . . 948
22.12 Digitalisering och redigering av ett GRASS-vektorlager . . . 949
22.13 GRASS-regionverktyget . . . 951
22.14 GRASS verktygslåda . . . 952
22.14.1 Arbeta med GRASS-moduler . . . 952
22.14.2 Exempel på GRASS-moduler . . . 955
22.14.3 Anpassa GRASS verktygslåda . . . 958

23 Ramverk för bearbetning av QGIS **961**
23.1 Introduktion . . . 961
23.2 Konfigurera ramverket för bearbetning . . . 964
23.2.1 Allmänt . . . 964
23.2.2 Menyer . . . 966
23.2.3 Modeller och manuskript . . . 966
23.2.4 Leverantörer . . . 966
23.3 Verktygslådan . . . 967
23.3.1 Dialogrutan för algoritmen . . . 969
23.3.2 Dataobjekt som genereras av algoritmer . . . 976
23.4 Den historiskt ansvarige . . . 977
23.4.1 Bearbetningshistorik . . . 977
23.4.2 Behandlingsloggen . . . 978
23.5 Modellkonstruktören . . . 979
23.5.1 Gränssnittet för modellkonstruktören . . . 979
23.5.2 Skapa en modell . . . 982
23.5.3 Spara och ladda modeller . . . 992
23.5.4 Redigera en modell . . . 992
23.6 Gränssnittet för batchbehandling . . . 994
23.6.1 Introduktion . . . 994
23.6.2 Parametertabellen . . . 995
23.6.3 Fylla i parametertabellen . . . 996
23.6.4 Exekvering av batchprocessen . . . 997
23.7 Använda bearbetningsalgoritmer från konsolen . . . 997
23.7.1 Anropa algoritmer från Python-konsolen . . . 998
23.7.2 Skapa skript och köra dem från verktygslådan . . . 1002
23.7.3 Skriptkrokar före och efter utförandet . . . 1005
23.8 Använda bearbetning från kommandoraden . . . 1005
23.9 Skriva nya Processing-algoritmer som Python-skript . . . 1007
23.9.1 Förlängning av QgsProcessingAlgorithm . . . 1008
23.9.2 @alg-dekoratorn . . . 1012
23.9.3 Utmatning av algoritm för överlämning . . . 1013
23.9.4 Kommunicera med användaren . . . 1014
23.9.5 Dokumentera dina skript . . . 1014
23.9.6 Flaggor . . . 1014
23.9.7 Bästa praxis för att skriva skriptalgoritmer . . . 1014
23.10 Inmatnings- och utmatningstyper för bearbetningsalgoritmer . . . 1015
23.10.1 Inmatningstyper . . . 1015
23.10.2 Typer av utdata . . . 1017
23.11 Konfigurera externa applikationer . . . 1017
23.11.1 En anmärkning för Windows-användare . . . 1018
23.11.2 En anmärkning om filformat . . . 1018
23.11.3 En anteckning om val av vektorlager . . . 1018
23.11.4 Använda tredjepartsleverantörer . . . 1018

24 Bearbetningsleverantörer och algoritmer **1031**
24.1 QGIS algoritmleverantör . . . 1031
24.1.1 3D-plattor . . . 1031
24.1.2 Kartografi . . . 1033
24.1.3 Databas . . . 1054

24.1.4 Verktyg för filer . . . 1062
24.1.5 GPS . . . 1064
24.1.6 Interpolation . . . 1068
24.1.7 Verktyg för lager . . . 1081
24.1.8 Nät . . . 1084
24.1.9 Verktyg för modellerare . . . 1095
24.1.10 Analys av nätverk . . . 1105
24.1.11 Tomter . . . 1119
24.1.12 Konvertering av punktmoln . . . 1125
24.1.13 Hantering av data från punktmoln . . . 1132
24.1.14 Extrahering av punktmoln . . . 1145
24.1.15 Rasteranalys . . . 1150
24.1.16 Skapande av raster . . . 1209
24.1.17 Rasteranalys av terräng . . . 1228
24.1.18 Rasterverktyg . . . 1239
24.1.19 Vektoranalys . . . 1250
24.1.20 Vektortäckning . . . 1277
24.1.21 Skapande av vektor . . . 1280
24.1.22 Vektor allmänt . . . 1309
24.1.23 Vektorgeometri . . . 1348
24.1.24 Vektoröverlägg . . . 1471
24.1.25 Val av vektor . . . 1491
24.1.26 Vektortabell . . . 1509
24.1.27 Vektorplattor . . . 1526
24.2 Leverantör av GDAL-algoritm . . . 1530
24.2.1 Rasteranalys . . . 1530
24.2.2 Rasterkonvertering . . . 1560
24.2.3 Extrahering av raster . . . 1568
24.2.4 Raster diverse . . . 1577
24.2.5 Rasterprojektioner . . . 1597
24.2.6 Vektoromvandling . . . 1602
24.2.7 Geoprocessning av vektorer . . . 1609
24.2.8 Vektor diverse . . . 1618

25 Tillägg **1631**
25.1 QGIS tillägg . . . 1631
25.1.1 Kärn- och externa tillägg . . . 1631
25.1.2 Dialogrutan för tillägg . . . 1632
25.2 Använda QGIS Core Tillägg . . . 1635
25.2.1 DB Manager Plugin . . . 1635
25.2.2 Tillägget Geometrikontroll . . . 1638
25.2.3 MetaSearch katalogklient . . . 1642
25.2.4 Tillägg för offline-redigering . . . 1649
25.2.5 Tillägget Topologikontroll . . . 1651
25.3 QGIS Python-konsol . . . 1653
25.3.1 Den interaktiva konsolen . . . 1654
25.3.2 Kodredigeraren . . . 1656

26 Bilagor **1659**
26.1 Bilaga C: QGIS-filformat . . . 1659
26.1.1 QGS/QGZ - QGIS-projektets filformat . . . 1659
26.1.2 QLR - QGIS-skiktdefinitionsfilen . . . 1661
26.1.3 QML - QGIS Style File Format . . . 1662
26.2 Bilaga D: Syntax för QGIS R-skript . . . 1663
26.2.1 Inmatningar . . . 1664
26.2.2 Utgångar . . . 1664
26.2.3 Syntaxsammanfattning för QGIS R-skript . . . 1664
26.2.4 Exempel . . . 1666

26.3 Bilaga E: Nätverksanslutningar för QGIS-applikationen 1668

27 Litteratur- och webbreferenser **1671**

KAPITEL 1

Komma igång

Detta kapitel ger en snabb översikt över hur du installerar QGIS, hämtar QGIS exempeldata och kör en första enkel session för att visualisera raster- och vektordata.

1.1 Installera QGIS

QGIS-projektet tillhandahåller olika sätt att installera QGIS beroende på din plattform.

1.1.1 Installera från binärfiler

Standardinstallationsprogram finns tillgängliga för MS Windows och macOS. Binära paket (rpm och deb) eller programvaruarkiv tillhandahålls för många varianter av GNU/Linux.

Mer information och instruktioner för ditt operativsystem finns på https://download.qgis.org.

1.1.2 Installera från källan

Om du behöver bygga QGIS från källkod, se installationsanvisningarna. De distribueras med QGIS källkod i en fil som heter `INSTALL`. Du kan också hitta dem online på https://github.com/qgis/QGIS/blob/release-3_40/INSTALL.md.

Om du vill bygga en viss version och inte den version som är under utveckling ska du ersätta `master` med versionsgrenen (vanligen i formen `release-X_Y`) i länken ovan (installationsanvisningarna kan skilja sig åt).

1.1.3 Installera på extern media

Det är möjligt att installera QGIS (med alla tillägg och inställningar) på ett USB-minne. Detta uppnås genom att definiera ett *–profiles-path*-alternativ som åsidosätter standard *användarprofil*-sökvägen och tvingar **QSettings** att använda den här katalogen också. Se avsnitt *Systeminställningar* för ytterligare information.

1.1.4 Nedladdning av exempeldata

Den här användarhandboken innehåller exempel som baseras på QGIS exempeldataset (även kallat ”Alaska dataset”). Ladda ner exempeldatasetet från https://github.com/qgis/QGIS-Sample-Data/archive/master.zip och packa upp arkivet på en lämplig plats i ditt system.

Alaska-datasetet innehåller alla GIS-data som används för exemplen och skärmdumparna i den här användarhandboken; det innehåller även en liten GRASS-databas. Projektionen för QGIS exempeldataset är Alaska Albers Equal Area med enheterna feet. EPSG-koden är 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]]
```

Om du tänker använda QGIS som en grafisk frontend för GRASS, kan du hitta ett urval av provplatser (t.ex. Spearfish eller South Dakota) på den officiella GRASS GIS-webbplatsen, https://grass.osgeo.org/download/data/.

1.2 Starta och stoppa QGIS

QGIS kan startas som vilket annat program som helst på din dator. Detta innebär att du kan starta QGIS genom att:

- genom att använda program-menyn i , startmenyn i eller dockan i
- dubbelklicka på ikonen i mappen Program eller genvägen på skrivbordet
- dubbelklicka på en befintlig QGIS-projektfil (med tillägget `.qgz` eller `.qgs`). Observera att detta också kommer att öppna projektet.
- skriva `qgis` i en kommandotolk (förutsatt att QGIS har lagts till i din PATH eller att du befinner dig i dess installationsmapp)

För att stoppa QGIS, använd:

- / menyalternativet *Projekt* ► *Avsluta QGIS* eller använd genvägen `Ctrl+Q`
- *QGIS* ► *Avsluta QGIS*, eller använd genvägen `Cmd+Q`
- eller använd det röda korset i det övre högra hörnet av programmets huvudgränssnitt.

1.3 Exempel på session: Ladda raster- och vektorlager

Nu när du har *QGIS installerat* och en *exempeldataset* tillgänglig ska vi demonstrera en första exempelsession. I det här exemplet visualiserar vi ett raster- och ett vektorlager. Vi kommer att använda:

- rasterlagret `landcover` (`qgis_sample_data/raster/landcover.img`)
- och vektorlagret `lakes` (`qgis_sample_data/gml/lakes.gml`)

Där `qgis_sample_data` representerar sökvägen till den utpaketerade datauppsättningen.

1. Starta QGIS så som visas i *Starta och stoppa QGIS*.
2. De data vi ska arbeta med finns i ”Albers Equal Area”, så låt oss ställa in projektets CRS i enlighet med detta:
    1. Klicka på knappen Välj projektion längst ned till höger i QGIS-gränssnittet. Dialogrutan för projektegenskaper öppnas med fliken *CRS* aktiv.
    2. Skriv``2964`` i textfältet *Filter*.
    3. Välj raden med CRS-namnet `NAD27 / Alaska Albers`.

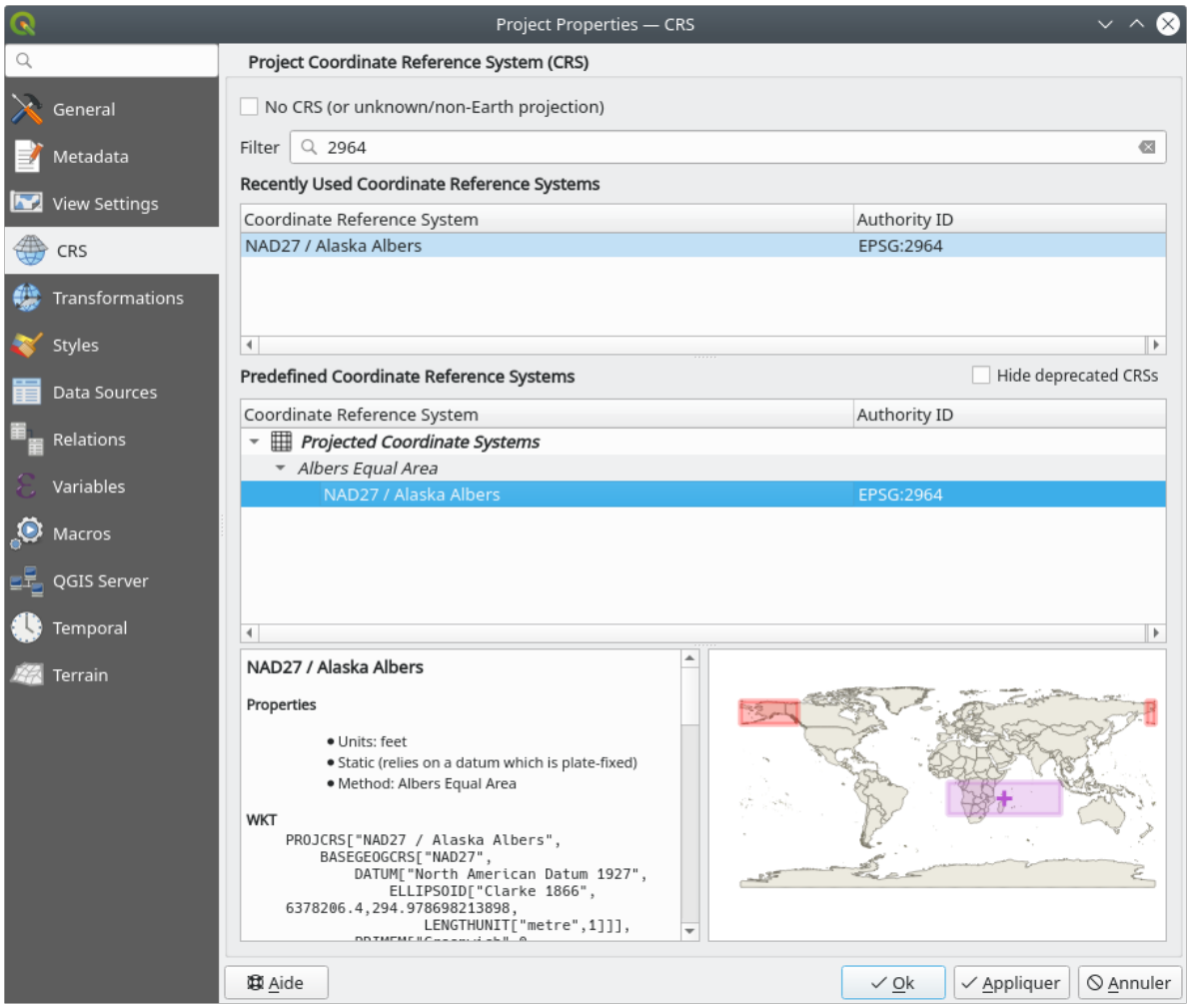


Fig. 1.1: Välj koordinatreferenssystem för data

4. Tryck på *OK*

> **Observera**
>
> Du kan för tillfället ignorera/stänga av det meddelande om ”ballpark transform” som kan visas.

3. Ladda filerna i QGIS:

   1. Klicka på ikonen Open Data Source Manager. Datakällhanteraren öppnas i bläddrarläge.
   2. Bläddra till mappen `qgis_sample_data/raster/`
   3. Välj ERDAS IMG-filen *landcover.img* och dubbelklicka på den. Landcover-lagret läggs till i bakgrunden medan Data Source Manager-fönstret förblir öppet.

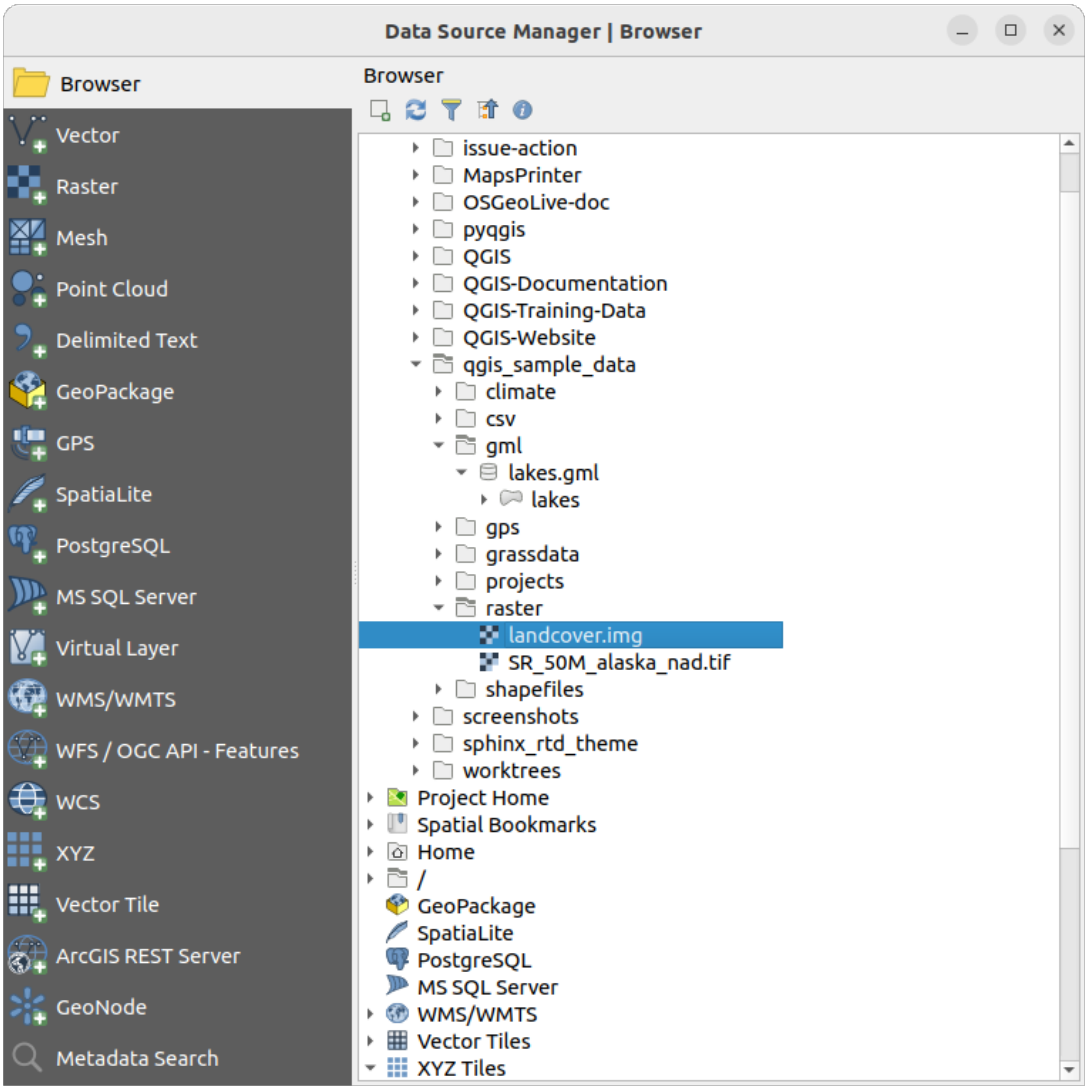


Fig. 1.2: Lägga till data i ett nytt projekt i QGIS

4. För att ladda data från sjöarna, bläddra till mappen `qgis_sample_data/gml/`, och dra och släpp filen *lakes.gml* över QGIS huvuddialog. (Eller bara dubbelklicka som nämnts ovan.)
5. Dialogrutan *Select Items to Add* öppnas och skannar filen. Detta beror på att filformatet `.gml` kan lagra mer än ett lager åt gången.

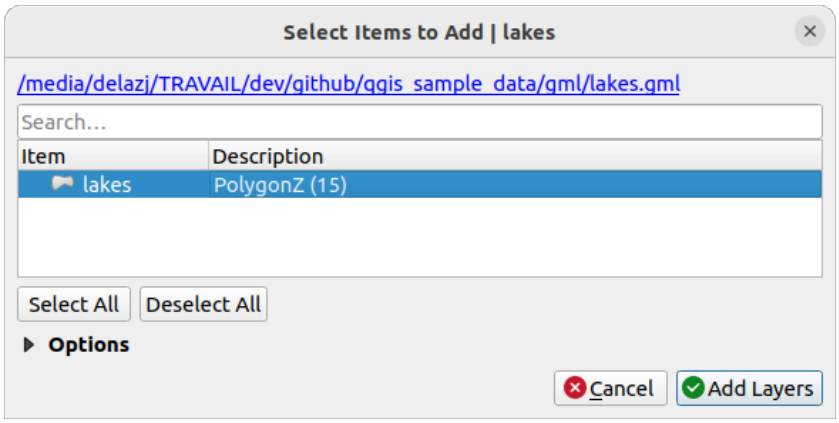


Fig. 1.3: Välj lager i en fil

6. I vårt fall finns det ett enda *lakes*-lager. Markera det och tryck på *Add Layers*.
7. Lagret läggs till i panelen *Layers*

4. Stäng fönstret Datakällhanteraren

I panelen *Layers* kan du se att lagret *lakes* visar Layer has no coordinate reference system set bredvid det. Låt oss justera det.

1. Klicka på ikonen . Dialogrutan *Coordinate Reference System Selector* öppnas.
2. Som tidigare gjort, hitta och välj *NAD27 / Alaska Albers* CRS-post.
3. Klicka på *OK*

Du har nu de två lagren tillgängliga i ditt projekt i några slumpmässiga färger. Låt oss göra några anpassningar på sjölagret.

1. Välj verktyget Zoom In på verktygsfältet *Navigation*
2. Zooma till ett område med några sjöar
3. Dubbelklicka på lagret `lakes` i kartlegenden för att öppna dialogrutan *Egenskaper*
4. För att ändra färg på sjöarna:
    1. Klicka på fliken *Symbology*
    2. Välj blå som fyllnadsfärg.

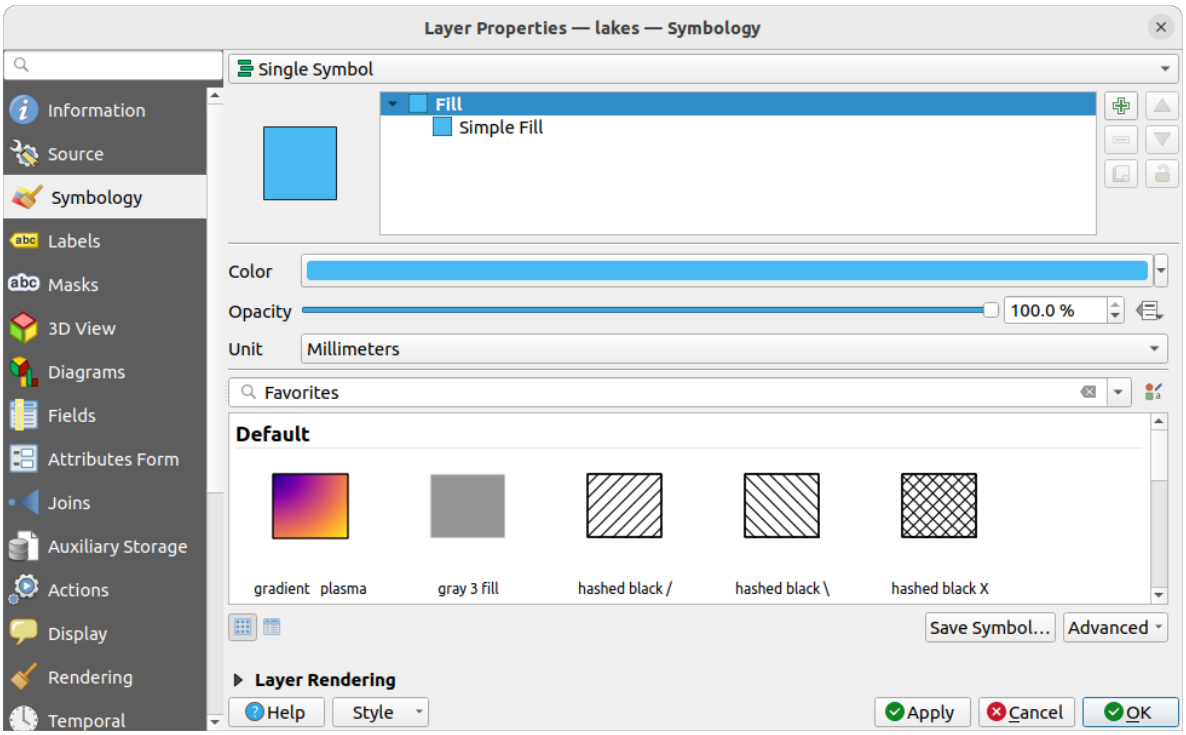


Fig. 1.4: Välja färg på lacer

3. Tryck på *OK*. Sjöar visas nu i blått i kartbilden.

5. För att visa namnet på sjöarna:

   1. Återöppna `lakes`-lagrets *Egenskaper*-dialog
   2. Klicka på fliken *Labels*
   3. Välj *Single labels* i rullgardinsmenyn för att aktivera märkning.
   4. I listan *Label with* väljer du fältet `NAMES`.

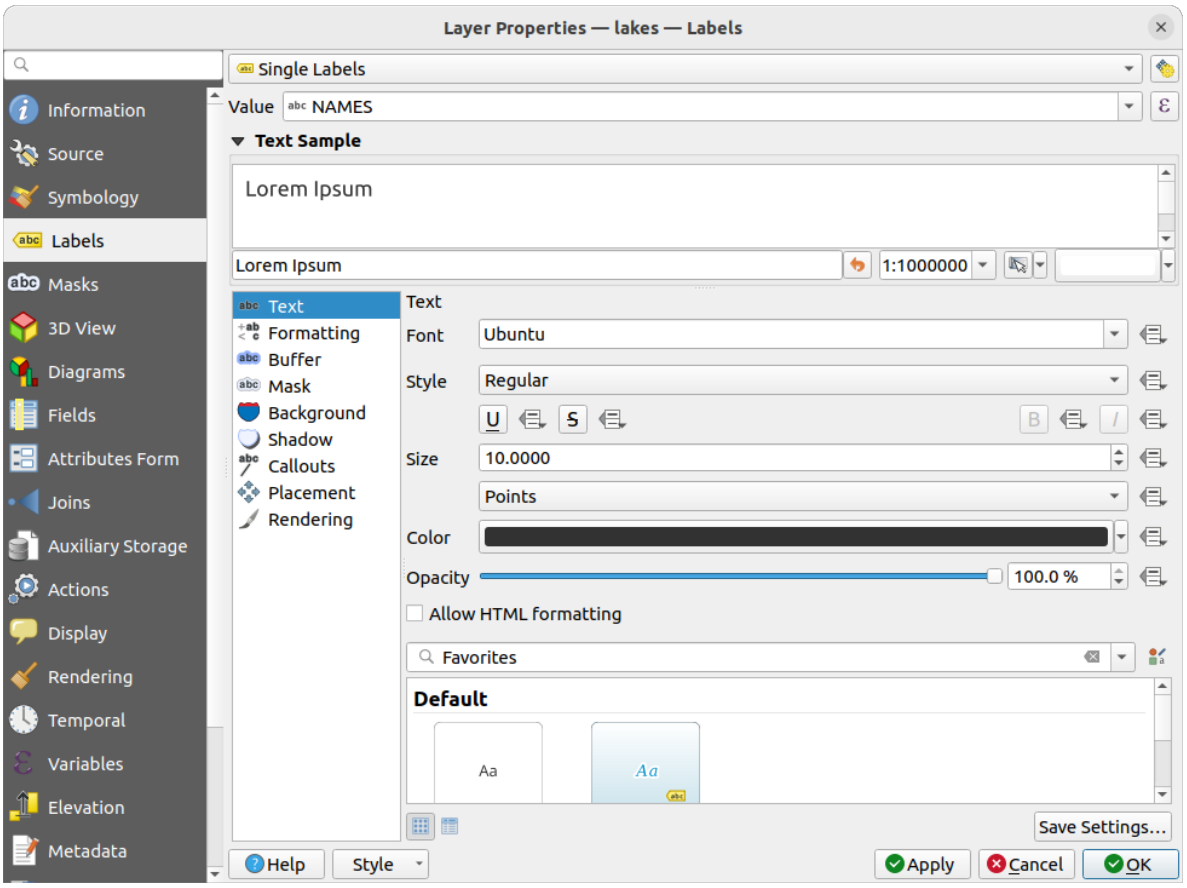


Fig. 1.5: Visning av namn på sjöar

5. Tryck på *Apply*. Namnen kommer nu att laddas över gränserna.

6. Du kan förbättra läsbarheten av etiketterna genom att lägga till en vit buffert runt dem:

   1. Klicka på fliken *Buffer* i listan till vänster
   2. Check *Rita textbuffert*
   3. Välj 3 som buffertstorlek
   4. Klicka på *Apply*
   5. Kontrollera om resultatet ser bra ut och uppdatera värdet om det behövs.
   6. Klicka slutligen på *OK* för att stänga dialogrutan *Lageregenskaper* och tillämpa ändringarna.

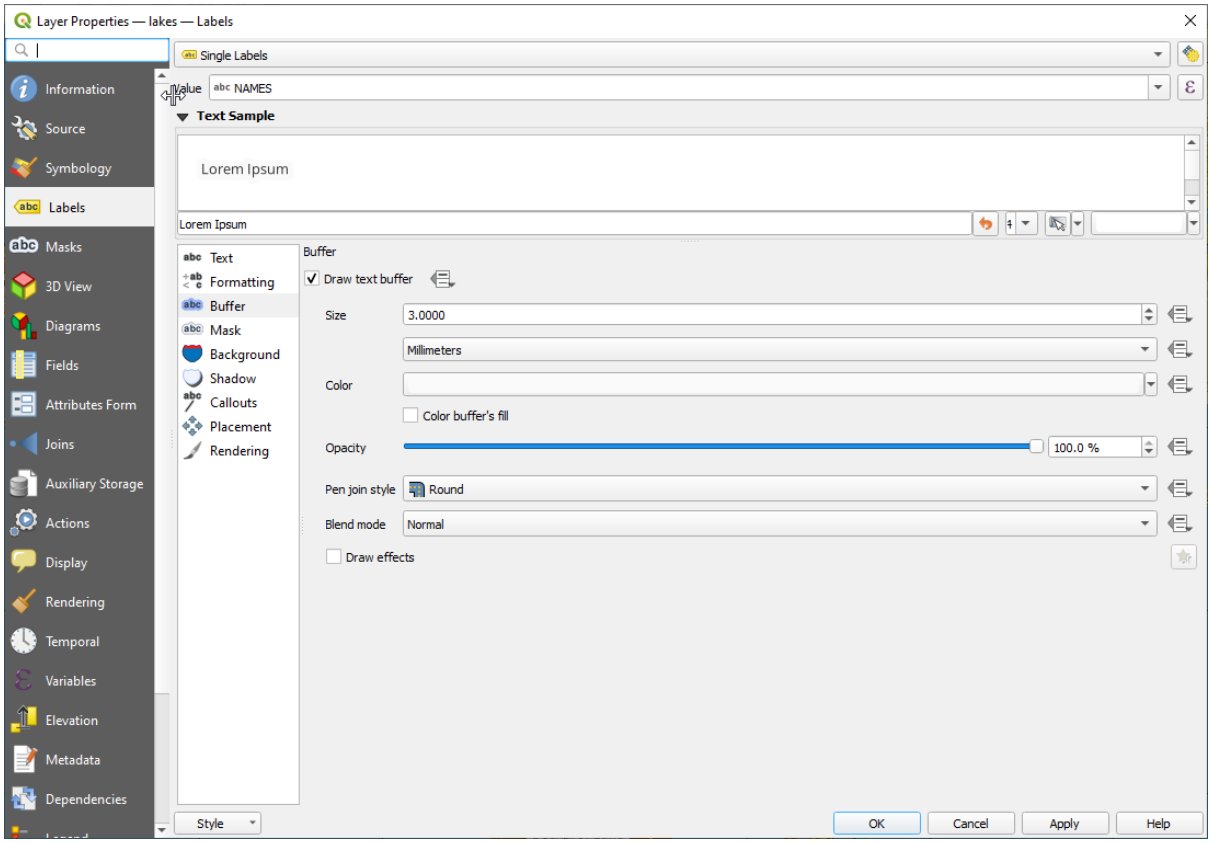


Fig. 1.6: Visning av buffertar runt etiketter

Låt oss nu lägga till några dekorationer för att forma kartan och exportera den från QGIS:

1. Välj *Visa* ► *Dekorationer* ► *Skala Bar*-menyn
2. I dialogrutan som öppnas markerar du alternativet *Enable Scale Bar*
3. Anpassa alternativen i dialogrutan som du vill
4. Tryck på *Apply*
5. På samma sätt kan du från dekorationsmenyn lägga till fler objekt (norrpil, copyright …) till kartduken med anpassade egenskaper.
6. Klicka på *Projekt* ► *Import/Export* ► *Exportera karta till bild…*
7. Tryck på *Save* i den dialogruta som öppnas
8. Välj en filplats, ett format och bekräfta genom att trycka på *Save* igen.
9. Tryck på *Projekt* ► *Spara…* för att spara dina ändringar som en `.qgz` projektfil.

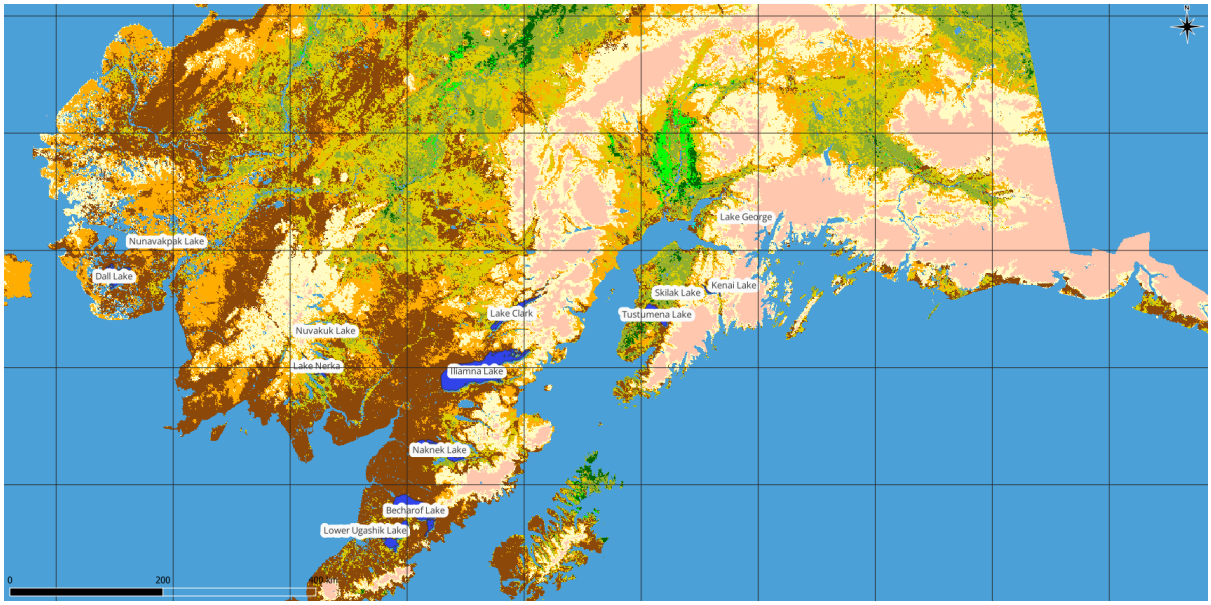


Fig. 1.7: Visar exporterad karta med dekorationer

Nu är det klart! Du kan se hur enkelt det är att visualisera raster- och vektorlager i QGIS, konfigurera dem och generera din karta i ett bildformat som du kan använda i andra program. Låt oss gå vidare och lära oss mer om tillgängliga funktioner, egenskaper och inställningar och hur du använder dem.

Observera

För att fortsätta lära dig QGIS genom steg-för-steg-övningar, följ Utbildningsmanualen.

KAPITEL 2

Arbeta med projektfiler

2.1 Introduktion till QGIS-projekt

Tillståndet för din QGIS-session kallas för ett projekt. QGIS arbetar med ett projekt i taget. En inställning kan vara projektspecifik eller en applikationsomfattande standardinställning för nya projekt (se avsnitt *Alternativ*). QGIS kan spara tillståndet för din arbetsyta i en *QGIS-projektfil* med hjälp av menyalternativen *Projekt* ► *Spara* eller *Projekt* ► *Spara som…*.

Observera

Om projektet har ändrats kommer symbolen * att visas i titelraden och QGIS kommer som standard att fråga dig om du vill spara ändringarna. Detta beteende styrs av inställningen *Prompt to save project and data source changes when required* under *Inställningar* ► *Alternativ* ► *Allmänt*.

Du kan ladda in befintliga projekt i QGIS från Browser-panelen eller genom *Projekt* ► *Öppna…*, *Projekt* ► *Nytt från mall* eller *Project* ► *Öppna tidigare* ►.

Vid start visas en lista över *Project Templates* och *Recent Projects*, inklusive skärmdumpar, namn och filsökvägar (för upp till tio projekt). Listan *Recent Projects* är praktisk för att komma åt nyligen använda projekt. Dubbelklicka på en post för att öppna projektet eller projektmallen. Högerklicka på en post för att *Pin to List*, *Open Directory…* eller *Remove from List*. Om du ser ditt projekt på listan *Recent Projects* men inte kan hitta det i din filhanterare kan du använda alternativet *Open Directory…* för att hitta projekt som kan ha försvunnit, flyttats eller bytt namn. Du kan också gå till *Clear List* om du vill ta bort alla projekt från listan *Recent Projects*. Om du har fäst projekt i listan kommer åtgärden *Clear List* att följas av en meddelanderuta som frågar om de fästade projekten också ska tas bort. Du kan också lägga till ett lager för att skapa ett nytt projekt automatiskt. Listorna kommer sedan att försvinna och ge plats åt kartbilden.

Om du vill rensa din session och börja om på nytt kan du gå till *Projekt* ► *Nytt*. Detta kommer att uppmana dig att spara det befintliga projektet om ändringar har gjorts sedan det öppnades eller senast sparades.

När du öppnar ett nytt projekt visas `Untitled Project` i titelraden tills du sparar det.

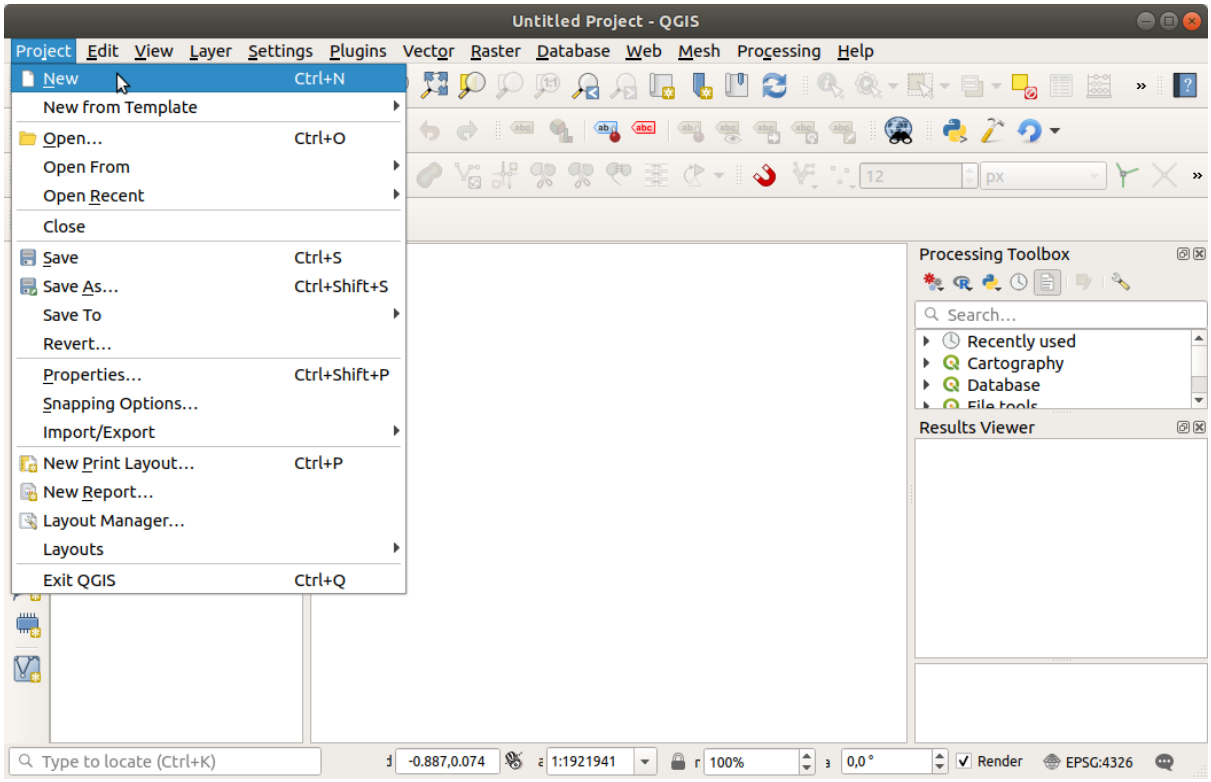


Fig. 2.1: Starta ett nytt projekt i QGIS

Den information som sparas i en projektfil inkluderar:

- Lager läggs till
- Vilka lager som kan efterfrågas
- Egenskaper för lager, inklusive symbolisering och stilar
- Anteckningar om lager
- 2D- och 3D-kartvyer
- Projektion för varje kartvy
- Senast visade omfattning för varje karta
- Utskriftslayouter
- Skriv ut layoutelement med inställningar
- Inställningar för utskriftslayout för atlas
- Inställningar för digitalisering
- Tabell Relationer
- Projektmakron
- Projektets standardstilar
- Tilläggsinställningar
- QGIS Server-inställningar från fliken OWS-inställningar i projektegenskaperna
- Frågor lagrade i DB Manager

Projektfilen sparas i XML-format (se *QGS/QGZ - QGIS-projektets filformat*). Detta innebär att det är möjligt att redigera filen utanför QGIS om du vet vad du gör. Projektfilformatet har uppdaterats flera gånger. Projektfiler från äldre QGIS-versioner kanske inte fungerar korrekt längre.

Observera

Som standard kommer QGIS att varna dig för versionsskillnader. Detta beteende styrs i fliken *Allmänt* i *Inställningar* ► *Alternativ* (*Varna när du öppnar en projektfil som sparats med en äldre version av QGIS*).

När du sparar en projektfil av typen `.qgs` i QGIS skapas en säkerhetskopia av filen i samma katalog som projektfilen, med tillägget `.qgs~`.

Tillägget för QGIS-projekt är `.qgs` men när du sparar från QGIS är standardinställningen att spara i ett komprimerat format med tillägget `.qgz`. Filen `.qgs` är inbäddad i filen `.qgz` (ett zip-arkiv), tillsammans med dess tillhörande SQLite-databas (`.qgd`) för *auxiliary data*. Du kan komma åt dessa filer genom att packa upp filen `.qgz`.

Observera

Mekanismen *Egenskaper för hjälplagring* gör ett zippat projekt särskilt användbart, eftersom det bäddar in hjälpdata.

Projekt kan också sparas/laddas till/från en PostgreSQL-, GeoPackage- eller Oracle-databas med hjälp av följande menyalternativ i Project:

- *Projekt* ► *Öppna från*
- *Projekt* ► *Spara till*

Båda menyalternativen har en undermeny med en lista över extra implementeringar av projektlagring (PostgreSQL, GeoPackage och Oracle). Genom att klicka på åtgärden öppnas en dialog för att välja en GeoPackage-anslutning och projekt, en PostgreSQL-anslutning, schema och projekt eller Oracle-anslutning, ägare och projekt.

Projekt som lagras i GeoPackage, PostgreSQL eller Oracle kan också laddas via QGIS bläddrarpanel, antingen genom att dubbelklicka på dem eller genom att dra dem till kartduken.

2.2 Hantering av brutna filsökvägar

När ett projekt öppnas kan QGIS misslyckas med att nå vissa datakällor på grund av otillgänglig tjänst/databas, eller på grund av en fil som bytt namn eller flyttats. QGIS öppnar då dialogen *Handle Unavailable Layers* och refererar till de otillgängliga lagren. Du kan göra följande:

- Dubbelklicka i fältet *Datakälla*, justera sökvägen för varje lager och klicka på *Apply changes*;
- Välj en rad, tryck på *Browse* för att ange rätt plats och klicka på *Apply changes*;
- Tryck på *Auto-Find* för att bläddra igenom mapparna och försöka att automatiskt fixa alla eller valda brutna sökvägar. Tänk på att sökningen kan ta lite tid. Klicka sedan på *Apply changes*.
- Ignorera meddelandet och öppna ditt projekt med den eller de brutna sökvägarna genom att klicka på *Keep Unavailable Layers*. Lagret visas sedan i panelen *Lager*, men utan data tills du åtgärdar sökvägen med hjälp av ikonen Unavailable layer! bredvid lagret i panelen *Lager* eller *Reparera datakälla…* i lagrets kontextuella meny.

  Med verktyget *Repair Data Source…*, när en lagersökväg har åtgärdats, söker QGIS igenom alla andra trasiga sökvägar och försöker automatiskt åtgärda dem som har samma trasiga filsökväg.
- *Remove Unavailable Layers* från projektet.

Om du startar QGIS från kommandoraden med hjälp av alternativet *--skipbadlayers* kan du hoppa över dialogrutan *Handle Unavailable Layers* vid start.

2.3 Generering av utdata

Det finns flera sätt att generera utdata från din QGIS-session. Vi har redan diskuterat att spara som en projektfil i *Introduktion till QGIS-projekt*. Andra sätt att producera utdatafiler är:

- Skapa bilder: *Projekt* ► *Import/Export* ► *Export Map to Image…* exporterar kartans canvasrendering till ett bildformat (PNG, JPG, TIFF…) med anpassad skala, upplösning och storlek. Det är möjligt att inkludera georeferensinformation i den exporterade bilden, aktivera bara *Lägg till georeferensinformation (inbäddad eller via världsfil)*. Se *Exportera kartvyn* för mer information.
- Exportera till PDF-filer: *Projekt* ► *Import/Export* ► *Exportera karta till PDF…* exporterar kartkanvasen till PDF i anpassad skala, upplösning och med vissa avancerade inställningar (förenkling, georeferenser, …). Se *Exportera kartvyn* för mer information.
- Exportera till DXF-filer: *Project* ► *Import/Export* ► *Export Project to DXF…* öppnar en dialogruta där du kan definiera ”Symbology mode”, ”Symbology scale” och de vektorlager som du vill exportera till DXF. Med hjälp av ”Symbology mode” kan symboler från den ursprungliga QGIS-symbologin exporteras med hög precision (se avsnitt *Skapa nya DXF-filer*).
- Utforma kartor: *Project* ► *New Print Layout…* öppnar en dialogruta där du kan layouta och skriva ut den aktuella kartbilden (se avsnitt *Lägga ut kartorna*).

KAPITEL 3

QGIS GUI

QGIS grafiska användargränssnitt (GUI) visas i figuren nedan (siffrorna 1 till 5 i de gula cirklarna anger viktiga delar av QGIS GUI och diskuteras nedan).

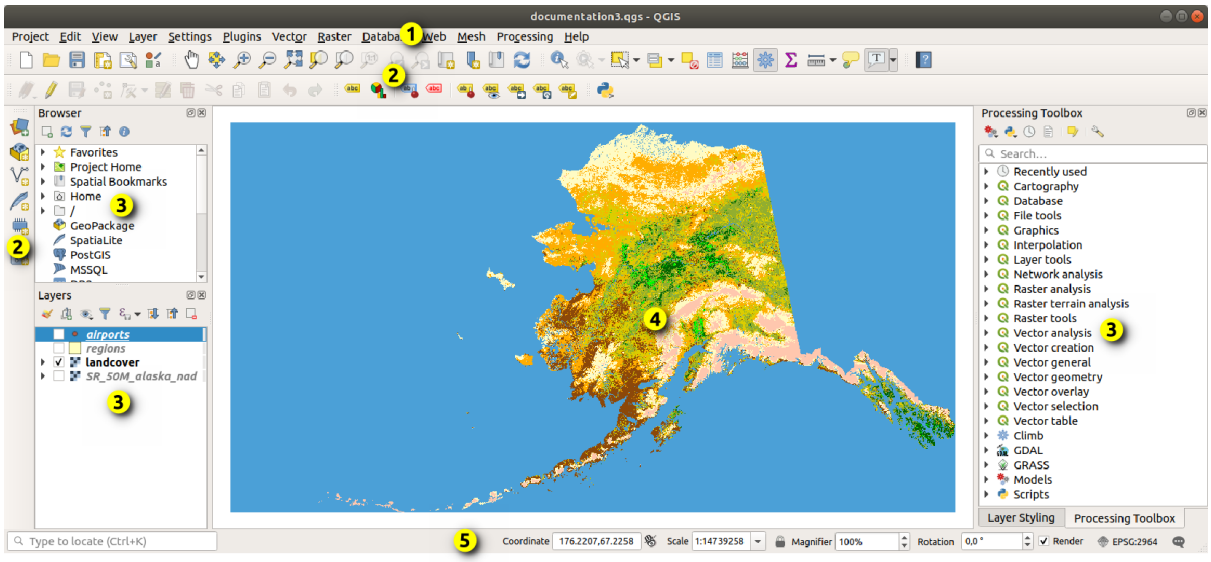


Fig. 3.1: QGIS grafiska gränssnitt med exempel på data från Alaska

Observera

Fönstrets dekorationer (titelraden etc.) kan se olika ut beroende på operativsystem och fönsterhanterare.

QGIS huvudsakliga grafiska gränssnitt (Fig. 3.1) består av fem komponenter/komponenttyper:

1. *Menyrad*
2. *Verktygsrader*
3. *Paneler*
4. *Kartvy*

5. *Statusrad*

Scrolla ner för detaljerade förklaringar av dessa.

3.1 Menyrad

Menyraden ger tillgång till QGIS-funktioner med hjälp av hierarkiska standardmenyer. Menyerna, deras alternativ, tillhörande ikoner och kortkommandon beskrivs nedan. Kortkommandona kan konfigureras om (*Inställningar* ► *Kortkommandon*).

De flesta menyalternativ har ett motsvarande verktyg och vice versa. Menyerna är dock inte organiserade på exakt samma sätt som verktygsfälten. Platserna för menyalternativen i verktygsfälten anges nedan i tabellen. Tillägg kan lägga till nya alternativ i menyerna. För mer information om verktyg och verktygsfält, se *Verktygsfält*.

Observera

QGIS är en plattformsoberoende applikation. Verktygen är i allmänhet tillgängliga på alla plattformar, men de kan vara placerade i olika menyer beroende på operativsystemet. Listorna nedan visar de vanligaste platserna, inklusive kända variationer.

3.1.1 Projekt

Menyn *Project* ger åtkomst till och utgångspunkter för *projektfiler*. Den tillhandahåller verktyg för att:

- Skapa en *Ny* projektfil från början eller använd en annan projektfil som mall (se *Projektfilalternativ* för mallkonfiguration)
- *Öppna…* ett projekt från en fil, ett GeoPackage, PostgreSQL- eller Oracle-databas
- *Stäng* ett projekt eller återställ det till dess senast sparade tillstånd
- *Spara* ett projekt i filformatet `.qgs` eller `.qgz`, antingen som en fil eller i en GeoPackage-, PostgreSQL- eller Oracle-databas
- Exportera kartbilden till olika format eller använd en *print layout* för mer komplexa utdata
- Ange projektegenskaper och snapping-alternativ för geometriredigering.

Table 3.1: Menyalternativen för projekt

Menyalternativ	Genväg	Verktygs	Referens
Ny	`Ctrl+N`	*Projekt*	*Introduktion till QGIS-projekt*
Nytt från mall ►			*Introduktion till QGIS-projekt*
Öppna…	`Ctrl+O`	*Projekt*	*Introduktion till QGIS-projekt*
Öppna från ►			*Introduktion till QGIS-projekt*
► *GeoPackage…*			*Introduktion till QGIS-projekt*
► *PostgreSQL…*			*Introduktion till QGIS-projekt*
► *Oracle…*			*Introduktion till QGIS-projekt*
Öppna senaste ►	`Alt+J` + `R`		*Introduktion till QGIS-projekt*
► *Rensa lista*			*Introduktion till QGIS-projekt*
Stäng			*Introduktion till QGIS-projekt*
Spara	`Ctrl+S`	*Projekt*	*Introduktion till QGIS-projekt*
Spara som…	`Ctrl+Shift+S`	*Projekt*	*Introduktion till QGIS-projekt*
Spara till ►			*Introduktion till QGIS-projekt*
► *Mallar…*			*Introduktion till QGIS-projekt*

continues on next page

Table 3.1 – fortsättning från föregående sida

Menyalternativ	Genväg	Verktygs	Referens
► *GeoPackage…*			*Introduktion till QGIS-projekt*
► *PostgreSQL…*			*Introduktion till QGIS-projekt*
Återgå…			
Egenskaper…	`Ctrl+Shift+P`		*Projektets egenskaper*
Fästningsalternativ…			*Alternativ för snäppning och digitalisering*
Importera/Exportera ►			
► *Exportera karta till bild…*			*Exportera kartvyn*
► *Exportera karta till PDF…*			*Exportera kartvyn*
► *Exportera projekt till DXF…*			*Skapa nya DXF-filer*
► *Importera lager från DWG/DXF…*			*Importera en DXF- eller DWG-fil*
Ny utskriftslayout…	`Ctrl+P`	*Projekt*	*Lägga ut kartorna*
Ny rapport …			*Skapa en rapport*
Layouthanterare…		*Projekt*	*Lägga ut kartorna*
Layouter ►			*Lägga ut kartorna*
Modeller ►			*Modellkonstruktören*
Avsluta QGIS	`Ctrl+Q`		

Under macOS motsvarar kommandot *Avsluta QGIS* kommandot *QGIS* ► *Avsluta QGIS* (`Cmd+Q`).

3.1.2 Redigera

Menyn *Redigera* innehåller de flesta av de inbyggda verktyg som behövs för att redigera lagerattribut eller geometri.

För att aktivera menyalternativen i *Redigera* måste du växla till redigeringsläge genom att klicka på Växla redigering (se *Ändrar* för detaljer).

Table 3.2: Menyalternativen Redigera

Menyalternativ	Genväg	Verktygsrad	Referens
Ångra	`Ctrl+Z`	*Digitalisering*	*Ångra och gör om*
Gör om	`Ctrl+Shift+Z`	*Digitalisering*	*Ångra och gör om*
Klippfunktioner	`Ctrl+X`	*Digitalisering*	*Klipp ut, kopiera och klistra in funktioner*
Kopiera funktioner	`Ctrl+C`	*Digitalisering*	*Klipp ut, kopiera och klistra in funktioner*
Klistra in funktioner	`Ctrl+V`	*Digitalisering*	*Klipp ut, kopiera och klistra in funktioner*
Klistra in funktioner som ►			*Arbeta med attributtabellen*
► *Nytt vektorlager…*			*Arbeta med attributtabellen*
► *Tillfälligt skraplager…*	`Ctrl+Alt+V`		*Arbeta med attributtabellen*
Ta bort markerade		*Digitalisering*	*Radera valda funktioner*

continues on next page

Table 3.2 – fortsättning från föregående sida

Menyalternativ	Genväg	Verktygsrad	Referens
Välj ►			*Välja funktioner*
► *Välj funktion(er)*		*Urval*	*Välja funktioner*
► *Välj funktioner efter polygon*		*Urval*	*Välja funktioner*
► *Välj funktioner med frihand*		*Urval*	*Välja funktioner*
► *Välj funktioner efter radie*		*Urval*	*Välja funktioner*
► *Välj funktioner efter värde…*	`F3`	*Urval*	*Välja funktioner*
► *Välj funktioner efter uttryck…*	`Ctrl+F3`	*Urval*	*Välja funktioner*
► *Avlägsna funktioner från alla lager*	`Ctrl+Alt+A`	*Urval*	*Välja funktioner*
► *Välj bort funktioner från det aktuella aktiva lagret*	`Ctrl+Shift+A`	*Urval*	*Välja funktioner*
► *Välj om funktioner*			*Välja funktioner*
► *Välj alla funktioner*	`Ctrl+A`	*Urval*	*Välja funktioner*
► *Invert Feature Selection*		*Urval*	*Välja funktioner*
Lägg till post	`Ctrl+.`	*Digitalisering*	
Lägg till punktfunktion	`Ctrl+.`	*Digitalisering*	*Lägga till funktioner*
Lägg till linjefunktion	`Ctrl+.`	*Digitalisering*	*Lägga till funktioner*
Lägg till polygonfunktion	`Ctrl+.`	*Digitalisering*	*Lägga till funktioner*
Lägg till anteckning ►			*Verktyg för anteckningar*
► *Form Annotation*		*Annotationer*	*Verktyg för anteckningar*
► *HTML Annotation*		*Annotationer*	*Verktyg för anteckningar*
Ändra attribut ►			
► *Modifiera attribut för valda funktioner*		*Digitalisering*	*Redigera attributvärden*
► *Merge Attributes of Selected Features*		*Avancerad digitalisering*	*Slå samman attribut för utvalda funktioner*
Redigera geometri ►			
► *Flytta funktion(er)*		*Avancerad digitalisering*	*Flytta funktion(er)*
► *Kopiera och flytta funktion(er)*		*Avancerad digitalisering*	*Flytta funktion(er)*
► *Rotera funktion(er)*		*Avancerad digitalisering*	*Rotera funktion(er)*
► *Skala funktion(er)*		*Avancerad digitalisering*	*Skalfunktion*
► *Förenkla funktion*		*Avancerad digitalisering*	*Förenkla funktionen*
► *Lägg till ring*		*Avancerad digitalisering*	*Lägg till ring*
► *Lägg till del*		*Avancerad digitalisering*	*Lägg till del*

continues on next page

Table 3.2 – fortsättning från föregående sida

Menyalternativ	Genväg	Verktygsrad	Referens
► *Fyll ring*		*Avancerad digitalisering*	*Fyllningsring*
► *Ta bort ring*		*Avancerad digitalisering*	*Radera ring*
► *Ta bort del*		*Avancerad digitalisering*	*Ta bort del*
► *Reshape Features*		*Avancerad digitalisering*	*Omforma funktioner*
► *Förskjut kurva*		*Avancerad digitalisering*	*Offset-kurvor*
► *Split Features*		*Avancerad digitalisering*	*Delade funktioner*
► *Split Parts*		*Avancerad digitalisering*	*Delade delar*
► *Merge Selected Features*		*Avancerad digitalisering*	*Slå samman utvalda funktioner*
► *Vertex Tool (Alla lager)*		*Digitalisering*	*Vertex-verktyg*
► *Vertex Tool (Aktuellt lager)*		*Digitalisering*	*Vertex-verktyg*
► *Reverse Line*		*Avancerad digitalisering*	*Omvänd linje*
► *Trim/förläng funktion*		*Avancerad digitalisering*	*Trim/Extend-funktion*
Rotatpunktssymboler		*Avancerad digitalisering*	*Rotera punktsymboler*
Offset Point Symbols		*Avancerad digitalisering*	*Symboler för offsetpunkter*

Verktyg som beror på den valda geometritypen för lagret, dvs. punkt, polylinje eller polygon, aktiveras på motsvarande sätt:

Table 3.3: De geometribaserade ikonerna ”Move feature”

Menyalternativ	Punkt	Polylinje	Polygon
Flytta funktion(er)			
Kopiera och flytta funktion(er)			

3.1.3 Vy

Kartan återges i kartvyer. Du kan interagera med dessa vyer med hjälp av *View*-verktygen. Det kan du till exempel göra:

- Skapa nya 2D- eller 3D-kartvyer bredvid den huvudsakliga kartbilden
- *Zoom eller panorera* till valfri plats
- Fråga efter attribut eller geometri för de visade objekten
- Förbättra kartvyn med förhandsgranskningslägen, anteckningar eller dekorationer
- Öppna valfri panel eller verktygsfält

I menyn kan du också omorganisera själva QGIS-gränssnittet med hjälp av åtgärder som t.ex:

- *Toggle Full Screen Mode*: täcker hela skärmen samtidigt som titelfältet döljs
- *Toggle Panel Visibility*: visar eller döljer aktiverade *panels* - användbart när du digitaliserar objekt (för maximal synlighet på duken) samt för (projicerade/inspelade) presentationer med QGIS huvudduk
- *Toggle Map Only*: döljer paneler, verktygsfält, menyer och statusfält och visar bara kartbilden. Kombinerat med alternativet helskärm gör det att skärmen bara visar kartan

Table 3.4: Menyalternativen Visa

Menyalternativ	Genväg	Verktygsra	Referens
Ny kartvy	`Ctrl+M`	*Map Navigation*	*2D-kartvy*
3D-kartvyer ►			*3D-kartvy*
► *Ny 3D-kartvy*	`Ctrl+Alt+M`	*Map Navigation*	*3D-kartvy*
► *Hantera 3D-kartvyer*			*3D-kartvy*
Panorera karta		*Map Navigation*	*Utforska kartvyn*
Pan Map till urval		*Map Navigation*	*Utforska kartvyn*
Zooma in	`Ctrl+Alt++`	*Map Navigation*	*Utforska kartvyn*
Zooma ut	`Ctrl+Alt+-`	*Map Navigation*	*Utforska kartvyn*
Datafiltrering ►			*2D-kartvy*
► *Temporal Controller*		*Map Navigation*	*Den temporala kontrollpanelen*
► *Elevation Controller*			*Panelen för höjdkontroll*
Identifiera funktioner	`Ctrl+Shift+I`	*Attribut*	*Identifiering av funktioner*
Mätning ►		*Attribut*	*Mätning*
► *Mätlinje*	`Ctrl+Shift+M`	*Attribut*	*Mätning*
► *Mätområde*	`Ctrl+Shift+J`	*Attribut*	*Mätning*
► *Mätbäring*		*Attribut*	*Mätning*
► *Mätvinkel*		*Attribut*	*Mätning*

continues on next page

Table 3.4 – fortsättning från föregående sida

Menyalternativ	Genväg	Verktygsra	Referens
Statistisk sammanfattning		Attribut	Panel för statistisk sammanfattning
Elevation Profile			Höjdprofil Vy
Zoom Full	Ctrl+Shift+F	Map Navigation	Utforska kartvyn
Zoom till urval	Ctrl+J	Map Navigation	Utforska kartvyn
Zooma till lager`		Map Navigation	Utforska kartvyn
Zoom till ursprunglig upplösning (100%)		Map Navigation	Utforska kartvyn
Zoom Last		Map Navigation	Utforska kartvyn
Zooma nästa		Map Navigation	Utforska kartvyn
Dekorationer ►	Alt+V + D		Dekorera kartan
► Rutnät…			Rutnät
► Scale Bar…			Skala Bar
► Bild…			Dekoration av bilder
► North Arrow…			Nordpilen
► Title Label…			Titelmärke
► Copyright Label…			Copyright Etikett
► Layout Extents…			Layout Extents
Förhandsgranskningsläge ►			
► Normal			
► Simulera monokrom			
► Simulera Achromatopsia färgblindhet (gråskala)			
► Simulera protanopisk färgblindhet (inget rött)			
► Simulera färgblindhet med deuteranopi (ingen grön färg)			
► Simulera Tritanopia Färgblindhet (Ingen Blå)			
Visa karttips		Attribut	Egenskaper för display
Nytt spatialt bokmärke…	Ctrl+B	Map Navigation	Lägga till bokmärken för extenter på kartan
Visa spatiala bokmärken	Ctrl+Shift+B	Map Navigation	Lägga till bokmärken för extenter på kartan
Visa spatial bokmärkeshanterare			Lägga till bokmärken för extenter på kartan
Uppdatera	F5	Map Navigation	

continues on next page

Table 3.4 – fortsättning från föregående sida

Menyalternativ	Genväg	Verktygsra	Referens
Lagrets synlighet ►			*Panel för lager*
► *Visa alla lager*	`Ctrl+Shift+U`		*Panel för lager*
► *Dölj alla lager*	`Ctrl+Shift+H`		*Panel för lager*
► *Visa valda lager*			*Panel för lager*
► *Dölj valda lager*			*Panel för lager*
► *Växla valda lager*			*Panel för lager*
► *Toggle Selected Layers Independently*			*Panel för lager*
► *Hide Deselected Layers*			*Panel för lager*
Paneler ►			*Paneler och verktygsfält*
Verktygsfält ►			*Paneler och verktygsfält*
Växla helskärmsläge	`F11`		
Toggle Panel Visibility	`Ctrl+Tab`		
Toggle Endast karta	`Ctrl+Shift+Tab`		

Under Linux KDE finns *Paneler* ►, *Verktygsrader* ► och *Växla helskärmsläge* i menyn *Inställningar*.

3.1.4 Lager

I menyn *Layer* finns en stor uppsättning verktyg för att *skapa* nya datakällor, *lägga till* dem i ett projekt eller *spara modifieringar* till dem. Med hjälp av samma datakällor kan du också:

- *Duplicera* ett lager för att generera en kopia där du kan ändra namn, stil (symbologi, etiketter, …), sammanfogningar, … Kopian använder samma datakälla som originalet.
- *Copy* och *Paste* lager eller grupper från ett projekt till ett annat som en ny instans vars egenskaper kan ändras oberoende av varandra. När det gäller *Duplicate* baseras lagren fortfarande på samma datakälla.
- eller *Embed Layers and Groups…* från ett annat projekt, som skrivskyddade kopior som du inte kan ändra (se *Inbäddning av lager från externa projekt*)

Menyn *Layer* innehåller också verktyg för att konfigurera, kopiera eller klistra in egenskaper för lager (stil, skala, CRS…).

Table 3.5: Menyalternativen för lager

Menyalternativ	Genväg	Verktygsrad	Referens
Hanterare av datakällor	`Ctrl+L`	*Data Source Manager*	*Opening Data*
Skapa Lager ►			*Skapa nya vektorlager*
► *New GeoPackage Layer…*	`Ctrl+Shift+N`	*Data Source Manager*	*Skapa ett nytt GeoPackage-lager*
► *New Shapefile Layer…*		*Data Source Manager*	*Skapa ett nytt Shapefile-lager*
► *New SpatiaLite Layer…*		*Data Source Manager*	*Skapa ett nytt SpatiaLite-lager*
► *Nytt tillfälligt skraplager…*		*Data Source Manager*	*Skapa ett nytt temporärt skrapskikt*

continues on next page

Table 3.5 – fortsättning från föregående sida

Menyalternativ	Genväg	Verktygsrad	Referens
► *Nytt nätlager…*		*Data Source Manager*	*Skapa ett nytt Mesh-lager*
► *Ny GPX-lager…*		*Data Source Manager*	*Skapa ett nytt GPX-lager*
► *Nytt virtuellt lager…*		*Data Source Manager*	*Skapa virtuella lager*
Lägg till lager ►			*Öppningsdata*
► *Lägg till vektorlager……`*	`Ctrl+Shift+V`	*Hantera lager*	*Läsa in ett lager från en fil*
► *Lägg till rasterlager…*	`Ctrl+Shift+R`	*Hantera lager*	*Läsa in ett lager från en fil*
► *Lägg till nätlager…*		*Hantera lager*	*Ladda ett nätlager*
► *Add Delimited Text Layer…*	`Ctrl+Shift+T`	*Hantera lager*	*Importera en avgränsad textfil*
► *Lägg till PostGIS-lager…*	`Ctrl+Shift+D`	*Hantera lager*	*Databasrelaterade verktyg*
► *Add SpatiaLite Layer…*	`Ctrl+Shift+L`	*Hantera lager*	*SpatiaLite-lager*
► *Lägg till MS SQL Server Layer…*		*Hantera lager*	*Databasrelaterade verktyg*
► *Add Oracle Spatial Layer…*		*Hantera lager*	*Databasrelaterade verktyg*
► *Add SAP HANA Spatial Layer…*		*Hantera lager*	*Databasrelaterade verktyg*
► *Lägg till/redigera virtuellt lager…*		*Hantera lager*	*Skapa virtuella lager*
► *Lägg till WMS/WMTS-lager…*	`Ctrl+Shift+W`	*Hantera lager*	*Laddning av WMS/WMTS-lager*
► *Lägg till XYZ-lager…*			*Använda XYZ Tile-tjänster*
► *Lägg till WCS-lager…*		*Hantera lager*	*WCS-klient*
► *Lägg till WFS / OGC API - Features Layer…*		*Hantera lager*	*WFS och WFS-T Klient*
► *Add ArcGIS REST Server Layer…*		*Hantera lager*	*Använda ArcGIS REST-servrar*
► *Add Vector Tile Layer…*			*Använda Vector Tiles-tjänster*
► *Add Point Cloud Layer…*			*Arbeta med punktmoln*
► *Add GPX Layer…*			*Introduktion av GNSS/GPS-data*
Bädda in lager och grupper…			*Inbäddning av lager från externa projekt*
Lägg till från Layer Definition File…			*Fil för definition av lager*
Georeferens…			*Georeferens*
Kopieringsstil			*Spara och dela egenskaper för lager*
Klistra in stil			*Spara och dela egenskaper för lager*
Kopieringslager			

continues on next page

Table 3.5 – fortsättning från föregående sida

Menyalternativ	Genväg	Verktygsrad	Referens
Klistra in lager/grupp			
Öppna attributtabell	F6	*Attribut*	*Arbeta med attributtabellen*
Filtrera attributtabell ►			*Arbeta med attributtabellen*
► *Öppna attributtabell (valda funktioner)*	Skift+F6	*Attribut*	*Arbeta med attributtabellen*
► *Öppna attributtabell (synliga egenskaper)*	Ctrl+F6	*Attribut*	*Arbeta med attributtabellen*
► *Öppna attributtabell (redigerade och nya funktioner)*		*Attribut*	*Arbeta med attributtabellen*
Växla redigering		*Digitalisering*	*Digitalisering av ett befintligt lager*
Spara redigeringar av lager		*Digitalisering*	*Spara redigerade lager*
Aktuella redigeringar ►		*Digitalisering*	*Spara redigerade lager*
► *Spara för valda lager*		*Digitalisering*	*Spara redigerade lager*
► *Rollback för valda lager*		*Digitalisering*	*Spara redigerade lager*
► *Avbryt för valda lager*		*Digitalisering*	*Spara redigerade lager*
► *Spara för alla lager*		*Digitalisering*	*Spara redigerade lager*
► *Rollback för alla lager*		*Digitalisering*	*Spara redigerade lager*
► *Avbryt för alla lager*		*Digitalisering*	*Spara redigerade lager*
Spara som…			*Skapa nya lager från ett befintligt lager*
Spara som lagerdefinitionsfil…			*Fil för definition av lager*
Ta bort lager/grupp	Ctrl+D		
Duplicerade lager			
Ställ in skalans synlighet för lager)			*Väljare för synlighetsskala*
Ställ in CRS för lager	Ctrl+Shift+C		*Referenssystem för lagerkoordinater*
Ställ in projekt CRS från lager			*Referenssystem för projektkoordinater*
Lageregenskaper…			• *vector* • *raster* • *nät* • *point cloud* • *vektorplattor*
Filter…	Ctrl+F		*Frågeverktyg*
Märkning			*Etiketter Egenskaper*
Visa i översikt			*Översiktspanel*
Visa alla i översikten			*Översiktspanel*
Dölj alla från översikten			*Översiktspanel*

3.1.5 Inställningar

Table 3.6: Menyalternativen för Inställningar

Menyalternativ	Referens
Användarprofiler ►	*Arbeta med användarprofiler*
► *standard*	*Arbeta med användarprofiler*
► *Öppna aktiv profilmapp*	*Arbeta med användarprofiler*
► *Ny profil…*	*Arbeta med användarprofiler*
Stilhanterare…	*Den stilansvarige*
Anpassade projektioner…	*Anpassat referenssystem för koordinater*
Tangentbordsgenvägar…	*Tangentbordsgenvägar*
Anpassning av gränssnitt…	*Anpassning*
Alternativ…	*Alternativ*

Under Linux KDE hittar du fler verktyg i menyn *Inställningar* såsom *Paneler* ►, *Verktygsrader* ► och *Växla helskärmsläge*.

3.1.6 Tillägg

Table 3.7: Menyalternativen för tillägg

Menyalternativ	Genväg	Verktygsrad	Referens
Hantera och installera tillägg…			*Dialogrutan för tillägg*
Python-konsol	`Ctrl+Alt+P`	*Tillägg*	*QGIS Python-konsol*

När QGIS startas för första gången är inte alla kärntillägg laddade.

3.1.7 Vektor

Så här ser *Vector*-menyn ut om alla kärntillägg är aktiverade.

Table 3.8: Menyalternativen för Vector

Menyalternativ	Genväg	Verktygsl	Referens
Kontrollera geometrier…			*Tillägget Geometrikontroll*
Topology Checker		*Vektor*	*Tillägget Topologikontroll*
Geoprocessverktyg ►	`Alt+O` + `G`		
► *Buffert…*			*Buffra*
► *Klipp…*			*Klipp*
► *Convex Hull…*			*Konvex skrovform*
► *Skillnad…*			*Skillnad*
► *Dissolve…*			*Lös upp*
► *Snitt…*			*Korsningar*

continues on next page

Table 3.8 – fortsättning från föregående sida

Menyalternativ	Genväg	Verktygsl	Referens
► *Symmetrical Difference…*			*Symmetrisk skillnad*
► *Union…*			*Förening*
► *Eliminate Selected Polygons…*			*Eliminera utvalda polygoner*
Geometriverktyg ►	`Alt+O+E`		
► *Centroids…*			*Centroider*
► *Samla geometrier…*			*Samla in geometrier*
► *Extrahera vertikaler…*			*Extrahera toppar*
► *Multipart till Singleparts…*			*Flera delar till enstaka delar*
► *Polygoner till linjer…*			*Polygoner till linjer*
► *Förenkla…*			*Förenkla*
► *Kontrollera giltighet…*			*Kontrollera giltighet*
► *Delaunay Triangulation…*			*Delaunay-triangulering*
► *Densify by Count…*			*Förtätning efter antal*
► *Lägg till geometriattribut…*			*Lägg till geometriattribut*
► *Linjer till polygoner…*			*Denna algoritm skapar ett vektorlager med ett rutnät som täcker en angiven utsträckning. Elementen i rutnätet kan vara punkter, linjer eller polygoner. Storleken och/eller placeringen av varje elementen i rutnätet bestäms av de horisontella och vertikala avstånden. Koordinatsystemet i utdata måste anges. Rutnätets utsträckning och avståndsvärden måste anges i koordinaterna och enheterna för koordinatsystemet. Den övre vänstra punkten (minX, maxY) används som referenspunkt. Det betyder att, vid den punkten, är ett element garanterat att placeras. Om inte bredden och höjden av den valda utsträckningen är en multipel av de valda avstånden, är detta inte sant för de andra punkterna som definierar den utsträckningen.*
► *Voronoi Polygons…*			*Voronoi-polygoner*
Analysverktyg ►	`Alt+O+A`		
► *Line Intersection…*			*Korsningar mellan linjer*
► *Medelkoordinat(er)…*			*Medelkoordinat(er)*
► *Basic Statistics for Fields…*			*Grundläggande statistik för fält*
► *Räkna punkter i polygon…*			*Räkna punkter i polygonen*
► *Distance Matrix…*			*Avståndsmatris*
► *List Unique Values…*			*Lista unika värden*

continues on next page

Table 3.8 – fortsättning från föregående sida

Menyalternativ	Genväg	Verktygs	Referens
► *Nearest Neighbour Analysis…*			*Analys av närmaste granne*
► *Summa linjelängder…*			*Summera linjelängder*
Datahanteringsverktyg ►	Alt+O + D		
► *Slå samman vektorlager…*			*Sammanfoga vektorlager*
► *Reproject Layer…*			*Återprojicera lagret*
► *Skapa spatialt index…*			*Skapa spatialt index*
► *Join Attributes by Location…*			*Anslut attribut efter plats*
► *Split Vector Layer…*			*Delat vektorlager*
Forskningsverktyg ►	Alt+O + R		
► *Skapa rutnät…*			*Skapa rutnät*
► *Extrahera lagerutsnitt…*			*Extrahera lagrets utsträckning*
► *Random Points in Extent…*			*Slumpmässiga punkter i utbredning*
► *Slumpmässiga punkter i polygoner…*			*Slumpmässiga punkter i polygoner*
► *Slumpmässiga punkter på linjer…*			*Slumpmässiga punkter på linjer*
► *Välj efter plats…*			*Välj efter plats*
► *Välj inom avstånd…*			*Välj inom avstånd*
► *Random Points in Layer Bounds…*			*Slumpmässiga punkter i lagergränser*
► *Random Points Inside Polygons…*			*Slumpmässiga punkter inuti polygoner*
► *Slumpmässigt urval…*			*Slumpmässigt val*
► *Slumpmässigt urval inom delmängder…*			*Slumpmässigt urval inom delmängder*
► *Regular Points…*			*Regelbundna poäng*

Som standard lägger QGIS till *Processing*-algoritmer i *Vector*-menyn, grupperade efter undermenyer. Detta ger genvägar för många vanliga vektorbaserade GIS-uppgifter från olika leverantörer. Om inte alla dessa undermenyer är tillgängliga aktiverar du tillägget Processing i *Tillägg* ► *Manage and Install Tillägg…*.

Observera att listan över algoritmer och deras meny kan ändras/utökas med valfria Processing-algoritmer (läs *Konfigurera ramverket för bearbetning*) eller några externa *tillägg*.

3.1.8 Raster

Så här ser *Raster*-menyn ut om alla kärntillägg är aktiverade.

Table 3.9: Menyalternativen för Raster

Menyalternativ	Genväg	Verkty	Referens
Rasterkalkylator…			*Rasterkalkylator*
Align Rasters…			*Rikta in raster*
Analys ►			
► *Aspekt…*			*Bildförhållande*
► *Fyll nodata…*			*Fylla NoData*

continues on next page

Table 3.9 – fortsättning från föregående sida

Menyalternativ	Genväg	Verkty(	Referens
► *Grid (glidande medelvärde)…*			*Rutnät (glidande medelvärde)*
► *Grid (Data Metrics)…*			*Grid (datamätningar)*
► *Grid (Inverse Distance to a Power)…*			*Grid (omvänt avstånd till en kraft)*
► *Grid (Närmaste granne)…*			*Rutnät (IDW med närmaste granne-sökning)*
► *Nära svart…*			*Nära svart*
► *Terrängskuggning…*			*Terrängskuggning*
► *Proximity (Rasteravstånd)…*			*Närhet (rasteravstånd)*
► *Roughness…*			*Grovhet*
► *Sieve…*			*Sikt*
► *Slope…*			*Lutning*
► *Topographic Position Index (TPI)…*			*Index för topografisk position (TPI)*
► *Terrain Ruggedness Index (TRI)…*			*Index för terrängens robusthet (TRI)*
Projektioner ►			
► *Assign Projection…*			*Tilldela projektion*
► *Extrahera projektion…*			*Extrahera projektion*
► *Warp (Reproject)…*			*Warp (omprojektering)*
Varier ►			
► *Bygg virtuell raster…*			*Skapa virtuell raster*
► *Rasterinformation…*			*Rasterinformation*
► *Slå samman…*			*Sammanfoga*
► *Bygga översikter (pyramider)…*			*Bygga översikter (pyramider)*
► *Tile Index…*			*Kakelindex*
Extraktion ►			
► *Clip Raster by Extent…*			*Klipp raster efter utsträckning*
► *Clip Raster by Mask Layer…*			*Klipp raster genom masklager*
► *Kontur…*			*Söm/kontur*
Konvertering ►			
► *PCT till RGB…*			*PCT till RGB*
► *Polygonize (Raster to Vector)…*			*Polygonisera (raster till vektor)*
► *Rasterize (Vector to Raster)…*			*Rasterize (vektor till raster)*
► *RGB till PCT…*			*RGB till PCT*
► *Översätt (Konvertera format)…*			*Översätt (konvertera format)*

Som standard lägger QGIS till *Processing*-algoritmer i *Raster*-menyn, grupperade efter undermenyer. Detta ger en genväg till många vanliga rasterbaserade GIS-uppgifter från olika leverantörer. Om inte alla dessa undermenyer är tillgängliga aktiverar du tillägget Processing i *Tillägg* ► *Manage and Install Tillägg…*.

Observera att listan över algoritmer och deras meny kan ändras/utökas med valfria Processing-algoritmer (läs *Konfigurera ramverket för bearbetning*) eller några externa *tillägg*.

3.1.9 Databas

Så här ser menyn *Database* ut om alla kärntillägg är aktiverade. Om inga databastillägg är aktiverade kommer det inte att finnas någon *Database*-meny.

Table 3.10: Menyalternativen för Databas

Menyalternativ	Genväg	Verktygsrad	Referens
Offline redigering…	Alt+D + O		*Tillägg för offline-redigering*
► *Konvertera till offlineprojekt…*		*Database*	*Tillägg för offline-redigering*
► *Synkronisera*		*Database*	*Tillägg för offline-redigering*
DB-hanterare…		*Database*	*DB Manager Plugin*

När QGIS startas för första gången är inte alla kärntillägg laddade.

3.1.10 Webb

Så här ser *Web*-menyn ut om alla kärntillägg är aktiverade. Om inga webbplugins är aktiverade kommer det inte att finnas någon *Web*-meny.

Table 3.11: Menyalternativen för webb

Menyalternativ	Genväg	Verktygsrad	Referens
MetaSearch ►	Alt+W + M		*MetaSearch katalogklient*
► *Metasök*		*Web*	*MetaSearch katalogklient*
► *Hjälp*			*MetaSearch katalogklient*

När QGIS startas för första gången är inte alla kärntillägg laddade.

3.1.11 Nät

Menyn *Mesh* tillhandahåller verktyg som behövs för att manipulera *mesh layers*. Tillägg från tredje part kan lägga till objekt i den här menyn.

Table 3.12: Menyalternativen Mesh

Menyalternativ	Genväg	Verktygsrad	Referens
Mesh Calculator…			*Mesh-kalkylator*
Reindexera ytor och vertikaler			*Omindexering av maskor*

3.1.12 Bearbetas

Table 3.13: Menyalternativen för bearbetning

Menyalternativ	Genväg	Verktygsr	Referens
Verktygslåda	`Ctrl+Alt+T`		*Verktygslådan*
Modellkonstruktör…	`Ctrl+Alt+G`		*Modellkonstruktören*
Historia…	`Ctrl+Alt+H`		*Den historiskt ansvarige*
Resultat Viewer	`Ctrl+Alt+R`		*Konfigurera externa applikationer*
Ändra funktioner på plats			*Modifieraren för bearbetning av lager på plats*

När QGIS startas för första gången är inte alla kärntillägg laddade.

3.1.13 Hjälp

Table 3.14: Menyalternativen i Hjälp

Menyalternativ	Genväg	Verktygsrad	Referens
Hjälpens innehåll	`F1`	*Hjälp*	
API Dokumentation			
Tillägg ►			
Rapportera ett problem			
Behöver du kommersiellt stöd?			
QGIS hemsida	`Ctrl+H`		
Kontrollera QGIS-version			
Om			
QGIS Stödjande medlemmar			

3.1.14 QGIS

Den här menyn är endast tillgänglig under macOS och innehåller några OS-relaterade kommandon.

Table 3.15: QGIS-menyalternativen

Menyalternativ	Genväg
Preferenser	
Om QGIS	
Dölj QGIS	
Visa alla	
Dölj andra	
Avsluta QGIS	`Cmd+Q`

Inställningar motsvarar *Inställningar* ► *Alternativ*, *Om QGIS* motsvarar *Hjälp* ► *Om* och *Avsluta QGIS* motsvarar *Projekt* ► *Avsluta QGIS* för andra plattformar.

3.2 Paneler och verktygsfält

Från menyn *Visa* (eller *Inställningar*) kan du aktivera och inaktivera QGIS-widgetar (*Paneler* ►) och verktygsfält (*Verktygsrader* ►). För att (in)aktivera någon av dem högerklickar du på menyraden eller verktygsfältet och väljer det objekt du vill ha. Paneler och verktygsfält kan flyttas och placeras var du vill i QGIS-gränssnittet. Listan kan också utökas med aktivering av *Kärn- eller externa tillägg*.

3.2.1 Verktygsfält

Verktygsfälten ger tillgång till de flesta av funktionerna i menyerna, plus ytterligare verktyg för att interagera med kartan. Varje objekt i verktygsfältet har popup-hjälp tillgänglig. Håll muspekaren över objektet så visas en kort beskrivning av verktygets syfte.

Tillgängliga verktygsfält är:

Table 3.16: QGIS verktygsfält

Verktygsfältets namn	Huvudreferens för verktyg
Avancerad digitalisering	*Avancerad digitalisering*
Annotationer	*Verktyg för anteckningar*
Attribut	*Arbeta med attributtabellen*, *Allmänna verktyg*
Data Source Manager	*Hantera datakälla*
Database	*DB Manager Plugin*
Digitalisering	*Digitalisering av ett befintligt lager*
GPS	*GPS-spårning i realtid*
GRASS	*GRASS GIS-integration*
Hjälp	
Label	*Verktygsfältet för etiketter*
Hantera lager	*Öppningsdata*
Map Navigation	*Utforska kartvyn*
Mesh Digitalisering	*Redigera ett mesh-lager*
Tillägg	*Tillägg*
Projekt	*Arbeta med projektfiler*, *Lägga ut kartorna*, *Stilbiblioteket*
Bearbetningsalgoritmer	*Konfigurera ramverket för bearbetning*
Raster	*Tillägg*
Urval	*Välja funktioner*
Digitalisering av former	*Digitalisering av form*
Snapping	*Inställning av snäpptolerans och sökradie*
Vektor	*Tillägg*
Web	*Tillägg*, *MetaSearch katalogklient*

Observera

Tillägg från tredje part kan utöka standardverktygsfältet med sina egna verktyg eller tillhandahålla ett eget verktygsfält.

Tips

Återställa verktygsfält

Om du av misstag har dolt ett verktygsfält kan du få tillbaka det med hjälp av *Visa* ► *Verktygsrader* ► (eller *Inställningar* ► *Verktygsrader* ►). Om ett verktygsfält (eller någon annan widget) av någon anledning helt försvinner från gränssnittet, hittar du tips för att få tillbaka det på *återställa initialt grafiskt gränssnitt*.

3.2.2 Paneler

QGIS tillhandahåller många paneler. Paneler är speciella widgetar som du kan interagera med (välja alternativ, kryssa i rutor, fylla i värden …) för att utföra mer komplexa uppgifter.

Nedan finns en lista över de standardpaneler som tillhandahålls av QGIS:

Table 3.17: QGIS-paneler

Panelens namn	Genväg	Referens
Avancerad digitalisering	`Ctrl+4`	*Panelen för avancerad digitalisering*
Browser	`Ctrl+2`	*Bläddrarpanelen*
Browser (2)		*Bläddrarpanelen*
Debugging/Development Tools	`F12`	*Panel för felsöknings-/utvecklingsverktyg*
Elevation Profile		
Geometrivalidering		*Egenskaper för digitalisering*
GPS-information	`Ctrl+0`	*GPS-spårning i realtid*
GRASS-verktyg		*GRASS GIS-integration*
Lagerordning	`Ctrl+9`	*Panel för beställning av lager*
Layer Styling	`Ctrl+3`	*Panel för lagerstyling*
Lager	`Ctrl+1`	*Panel för lager*
Logga meddelanden		*Panel för loggmeddelanden*
Översikt	`Ctrl+8`	*Översiktspanel*
Processing Toolbox		*Verktygslådan*
Resultatsvisare		*Verktygslådan*
Avklippning och digitaliseringsalternativ		*Inställning av snäpptolerans och sökradie*
Spatial Bookmark Manager	`Ctrl+7`	*Lägga till bokmärken för extenter på kartan*
Statistik	`Ctrl+6`	*Panel för statistisk sammanfattning*
Temporal styrenhet		*Den temporala kontrollpanelen*
Tegelskala		*Mosaiker*
Undo/Redo	`Ctrl+5`	*Undo/Redo-panel*
Vertex Editor		*Vertexredigerarens panel*

3.3 Statusrad

Statusfältet ger dig allmän information om kartvyn och bearbetade eller tillgängliga åtgärder och erbjuder dig verktyg för att hantera kartvyn.

3.3.1 Locator bar

På vänster sida av statusfältet finns sökfältet, en widget för snabbsökning, som hjälper dig att hitta och köra alla funktioner eller alternativ i QGIS:

1. Klicka i textwidgeten för att aktivera sökfältet för lokalisering eller tryck på `Ctrl+K`.
2. Skriv en text som är kopplad till det objekt du söker (namn, tagg, nyckelord, …). Som standard returneras resultat för de aktiverade lokaliseringsfiltren, men du kan begränsa sökningen till ett visst omfång genom att prefixera din text med prefixet *locator filters*, dvs. om du skriver `l cad` returneras endast de lager vars namn innehåller `cad`.

   Filtret kan också väljas med ett dubbelklick i den meny som visas när du öppnar locator-widgeten.
3. Klicka på ett resultat för att utföra motsvarande åtgärd, beroende på vilken typ av objekt det är.

Tips

Begränsa sökningen till ett eller flera fält i det aktiva lagret

Som standard körs en sökning med filtret ”active layer features” (`f`) genom hela attributtabellen för lagret. Du kan begränsa sökningen till ett visst fält med hjälp av prefixet `@`. T.ex. `f @name sal` eller `@name sal` returnerar endast de funktioner vars ”name”-attribut innehåller ”sal”. Autokomplettering av text är aktiv när du skriver och förslaget kan tillämpas med hjälp av `Tab`-nyckeln.

En mer avancerad kontroll av de fält som efterfrågas är möjlig från lagret *Fields* tab. Läs *Fältegenskaper* för mer information.

Sökningen hanteras med hjälp av trådar, så att resultaten alltid blir tillgängliga så snabbt som möjligt, även om långsamma sökfilter är installerade. De visas också så snart de påträffas av ett filter, vilket innebär att t.ex. ett filsökningsfilter kommer att visa resultaten ett efter ett när filträdet skannas. Detta säkerställer att användargränssnittet alltid är responsivt, även om det finns ett mycket långsamt sökfilter (t.ex. ett som använder en onlinetjänst).

Observera

Lokaliseringsverktyget Nominatim kan bete sig annorlunda (ingen sökning med autokomplettering, fördröjning av hämtning av resultat, …) med avseende på OpenStreetMaps Nominatim användningspolicy.

Tips

Snabb åtkomst till lokalisatorns konfigurationer

Klicka på ikonen i widgeten Locator i statusfältet för att visa en lista över filter som du kan använda och en *Configure*-post som öppnar fliken *Locator* i menyn *Inställningar* ► *Alternativ*….

3.3.2 Rapportering av åtgärder

I området bredvid sökfältet visas vid behov en sammanfattning av de åtgärder du har utfört (t.ex. att välja funktioner i ett lager, ta bort lager, panorera avstånd och riktning) eller en lång beskrivning av det verktyg du håller muspekaren över (gäller inte alla verktyg).

Vid långdragna operationer, t.ex. insamling av statistik i rasterlager, exekvering av bearbetningsalgoritmer eller rendering av flera lager i kartvyn, visas en förloppsindikator i statusfältet.

3.3.3 Kontrollera kartans canvas

Alternativet *Coordinate* visar musens aktuella position och följer den medan den rör sig över kartvyn. Du kan ställa in enheterna (och precisionen) i *Projekt* ► *Egenskaper…* ► *Allmänt*. Klicka på den lilla knappen till vänster om textrutan för att växla mellan alternativet Coordinate och alternativet *Extents* som visar koordinaterna för de aktuella nedre vänstra och övre högra hörnen av kartvyn i kartenheter.

Bredvid koordinatdisplayen hittar du displayen *Scale*. Den visar skalan på kartvyn. Det finns en skalväljare som gör att du kan välja mellan *predefined and custom scales*.

Tryck på knappen på höger sida av skalans display för att låsa skalan så att du kan använda förstoringsglaset för att zooma in eller ut. Med förstoringsglaset kan du zooma in på en karta utan att ändra kartskalan, vilket gör det lättare att justera positionerna för etiketter och symboler exakt. Förstoringsnivån uttrycks i procent. Om *Magnifier* har en nivå på 100% förstoras inte den aktuella kartan, dvs. den återges i korrekt skala i förhållande till skärmens upplösning (DPI). Ett standardförstoringsvärde kan definieras i *Inställningar* ► *Alternativ* ► *Rendering* ► *Renderingsbeteende*,

vilket är mycket användbart för högupplösta skärmar för att förstora små symboler. Dessutom finns det en inställning i *Inställningar* ► *Alternativ* ► *Canvas & Legend* ► *DPI* som styr om QGIS ska respektera varje bildskärms fysiska DPI eller använda det övergripande systemets logiska DPI.

Till höger om förstoringsverktyget kan du ange en aktuell medursrotation för kartvyn i grader.

På höger sida av statusfältet finns kryssrutan *Render* som kan användas för att tillfälligt avbryta renderingen av kartvyn (se avsnitt *Styrning av kartrendering*).

Till höger om funktionen *Rendera* finns knappen *EPSG:code* som visar det aktuella projektets CRS. Om du klickar på den öppnas dialogrutan *Projektegenskaper* och du kan projicera om kartvyn eller justera någon annan projektegenskap.

Tips

Beräkna rätt skala för din kartbild

När du startar QGIS är standard CRS `WGS 84 (EPSG 4326)` och enheterna är grader. Detta innebär att QGIS kommer att tolka alla koordinater i ditt lager som angivna i grader. För att få korrekta skalvärden kan du antingen manuellt ändra denna inställning på fliken *General* under *Projekt* ► *Egenskaper…* (t.ex. till meter), eller så kan du använda ikonen EPSG:code som du ser ovan. I det senare fallet ställs enheterna in på det som projektprojektionen anger (t.ex. `+units=us-ft`).

Observera att CRS val vid uppstart kan ställas in i *Inställningar* ► *Alternativ* ► *CRS Hantering*.

3.3.4 Meddelanden

Knappen Meddelanden bredvid den öppnar *Log Messages Panel* som innehåller information om underliggande processer (QGIS-start, laddning av tillägg, bearbetningsverktyg …)

KAPITEL 4

Bläddrarpanelen

- *Resurser som kan öppnas/köras från bläddraren*
- *Bläddrarpanelens toppnivåposter*
  - *Favoriter*
  - *Spatiala bokmärken*
  - *Projektets hem*
  - *Hårddiskar och filsystem*
  - *Databasposter*
  - *Kakelplattor och webbtjänster*
- *Resurser*

QGIS-bläddrarpanelen är ett utmärkt verktyg för att bläddra, söka, inspektera, kopiera och ladda QGIS-resurser. Endast resurser som QGIS vet hur de ska hanteras visas i bläddraren.

Med hjälp av Bläddrar-panelen kan du lokalisera, inspektera och lägga till data, enligt beskrivningen i *Bläddrarpanelen*. Dessutom stöder Browser-panelen dra och släpp av många QGIS-resurser, t.ex. projektfiler, Python-skript, Processing-skript och Processing-modeller.

Python-skript, Processing-skript och Processing-modeller kan också öppnas för redigering i en extern editor och i den grafiska modellen.

Du kan dra och släppa lager från panelen *Lager* till panelen *Bläddrare*, t.ex. till ett GeoPackage eller en PostGIS-databas.

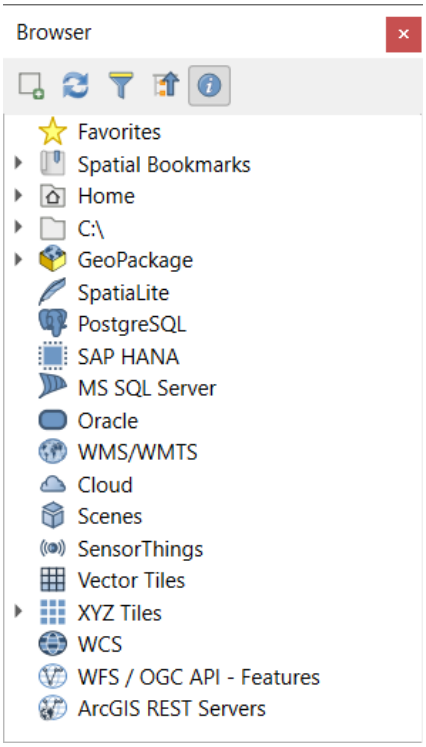


Fig. 4.1: Bläddrarpanelen

Bläddrarpanelen (Fig. 4.1) är organiserad som en expanderbar hierarki med några fasta toppnivåposter som organiserar de resurser som hanteras av bläddraren. Nodposter expanderas genom att klicka på ▶ till vänster om postens namn. En gren kollapsas genom att klicka på ▼. Med knappen Collapse All kollapsar alla poster på den översta nivån.

I *Inställningar* ► *Interface Customization* är det möjligt att inaktivera resurser. Om du t.ex. inte vill visa Python-skript i bläddraren kan du avmarkera posten *Bläddrare* ► *py*, och om du vill bli av med din hemmapp i webbläsaren kan du avmarkera posten *Bläddrare* ► *special:Home*.

Ett filter (Filter Browser) kan användas för att söka baserat på postnamn (både bladposter och nodposter i hierarkin). Med hjälp av rullgardinsmenyn Options bredvid filtertextfältet kan du

- växla *Skiftlägeskänslig* sökning
- ställa in *Filter pattern syntax* till en av
  - *Normal*
  - *Jokertecken*
  - *Reguljära uttryck*

Widgeten *Egenskaper*, som visar användbar information om vissa poster/resurser, kan aktiveras/inaktiveras med hjälp av knappen Aktivera/inaktivera egenskapswidget. När den är aktiverad öppnas den längst ner i bläddrarpanelen, som visas i Fig. 4.2.

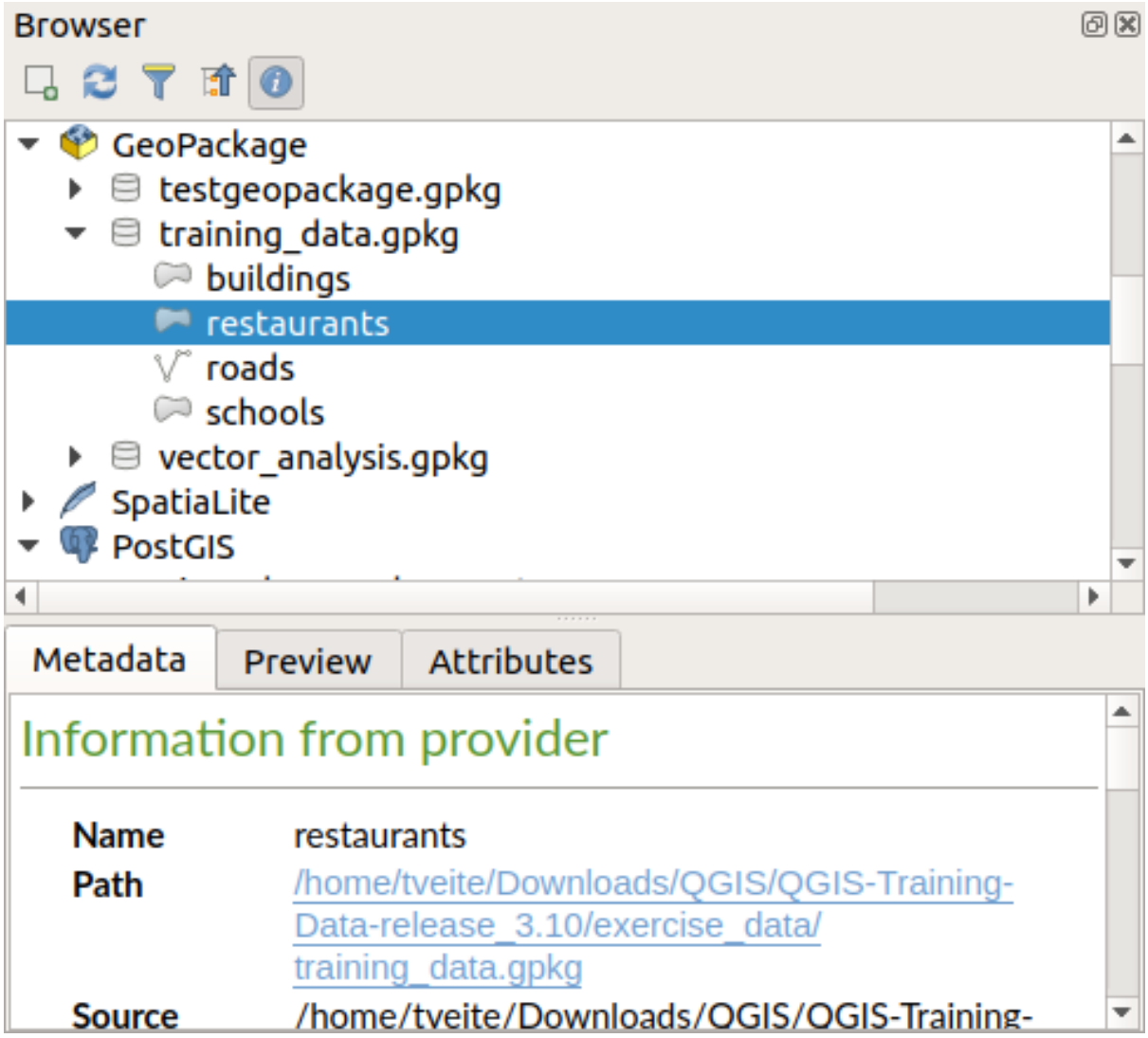


Fig. 4.2: Widgeten för egenskaper

En andra bläddrarpanel kan öppnas genom att aktivera *Bläddrare (2)*-panelen i *Visa* ► *Paneler*. Att ha två bläddrarpaneler kan vara användbart när man kopierar lager mellan resurser som är placerade djupt ner i olika grenar av bläddrarhierarkin.

4.1 Resurser som kan öppnas/köras från bläddraren

Mycket kan åstadkommas i Browser-panelen

- Lägg till vektor-, raster- och mesh-lager på kartan genom att dubbelklicka, dra på kartbilden eller klicka på knappen Add Selected Layers (efter att du har valt lager)
- Kör Python-skript (inklusive Processing-algoritmer) genom att dubbelklicka eller dra på kartbilden
- Kör modeller genom att dubbelklicka eller dra dem på kartbilden
- *Extrahera symboler…* från QGIS Project-filer med hjälp av snabbmenyn
- Öppna filer med deras standardprogram (*Öppna <file type> Externt…* i snabbmenyn). Exempel: HTML-filer, kalkylblad, bilder, PDF-filer, textfiler, …
- Kopiera poster

- Byt namn på och ta bort (flera) lager (snabbmeny: *Manage* ►)
- Öppna ett filutforskarfönster och välj filen direkt *Visa i filer*

Resursspecifika åtgärder listas för de olika resursgrupperna sorterade under de överordnade posterna som listas nedan.

4.2 Bläddrarpanelens toppnivåposter

4.2.1 Favoriter

Ofta använda platser i filsystemet kan märkas som favoriter. De som du har taggat kommer att visas här.

Förutom de åtgärder som beskrivs under *Home* kan du via snabbmenyn *Rename Favorite…* och *Remove Favorite*.

4.2.2 Spatiala bokmärken

Det är här du hittar dina spatiala bokmärken, organiserade i *Project Bookmarks* och *User Bookmarks*.

Från snabbmenyn på översta nivån kan du skapa ett bokmärke (*New Spatial Bookmark…*), *Show the Spatial Bookmark Manager*, *Import Spatial Bookmarks…* och *Export Spatial Bookmarks.…*

För bokmärkesgrupper kan du *Exportera spatiala bokmärken…*, skapa ett bokmärke (*New Spatial Bookmark…*), *Rename Bookmark Group*, och *Delete Bookmark Group*.

För bokmärkesposter kan du använda *Zoom to Bookmark*, *Edit Spatial Bookmark…* och *Delete Spatial Bookmark*

4.2.3 Projektets hem

När projektfilen har sparats är *Project home* en mapp som innehåller data och annat innehåll (skript, modeller, text, …) som kan användas i det aktuella projektet. Den visas i panelen *Browser* och gör att du snabbt kan komma åt data och andra filer i projektet.

Standardinställningen är projektets filmapp, men den kan ändras genom *Projekt* ► *Egenskaper…* -> *Allmänt* ► *Projekthem* eller genom att högerklicka på objektet *Projekthem* i bläddrarpanelen och välja *Ställ in projekthem.…* –> Allmänt –> Projekthem`, eller genom att högerklicka på objektet *Projekthem* i panelen Bläddrare och välja *Ställ in projekthem.…* Att anpassa den mappen är särskilt användbart i sammanhang där QGIS-projekt inte lagras i rotmappen för ett organisatoriskt ”projekt”, tillsammans med dataset.

4.2.4 Hårddiskar och filsystem

De följande objekten i *Browser*-panelen beror på vilket operativsystem som används och gäller toppnivåposterna i dess filsystem.

De är huvudsakligen:

- Mappen *Home*, som pekar på den aktuella användarens hemmapp
- på Unix-baserade maskiner, rotmappen `/`
- de anslutna enheterna, antingen lokalt eller i nätverket. Beroende på operativsystem listas de direkt (t.ex. `C:\`, `D:\`) eller via posten `/Volumes`.

Från den kontextuella menyn i var och en av dessa mappar eller enheter kan du:

- uppdatera innehållet
- skapa ett *New* ► underobjekt som är en *Directory*, *GeoPackage* eller ESRI *Shapefile* format dataset
- dölja katalogen (*Hide from Browser*)
- *Set color*: anpassa mappikonens färg, vilket underlättar snabb navigering i komplexa mappstrukturer

- aktivera *Scanning*:
  - *Övervaka för ändringar*: gör det möjligt att manuellt styra om en viss katalog ska övervakas och uppdateras automatiskt. Den här inställningen gäller för den valda katalogen och alla underkataloger. Det innebär att du manuellt kan välja övervakning av nätverksenheter om du vet att det inte är något problem, eller manuellt välja bort övervakning av stora kataloger som du av andra skäl inte vill ska övervakas. Som standard övervakas inte fjärr- eller nätverksenheter automatiskt.
  - *Snabbskanning av denna katalog*
- öppna katalogen i din filhanterare (*Öppna katalog…*)
- öppna katalogen i ett terminalfönster (*Öppna i terminal…*)
- inspektera *Egenskaper…* eller den överordnade *Katalogegenskaper…*

4.2.5 Databasposter

Beroende på ditt operativsystem och installerade drivrutiner kan du ha tillgång till olika databastyper som du kan använda i QGIS. Nedan listas de olika posterna i den kontextuella menyn på varje nivå i dataset-trädet.

Nivå	Kontextmeny	Typ av databas					
		GeoPackage ([1])	SpatiaLite	PostGIS	SAP HANA	MS SQL Server	Oracle
Toppmeny	Skapa en *Ny anslutning…* till en befintlig databas	✓	✓	✓	✓	✓	✓
	Skapa databas…	✓	✓				
	Skapa databas och lager…	✓					
	Spara anslutningar. detaljer till en fil			✓	✓	✓	
	Ladda anslutningar.			✓	✓	✓	
Anslutning / databas	*Uppdatera* en anslutning			✓	✓	✓	✓
	Redigera anslutning… inställningar			✓	✓	✓	✓
	Duplicerad anslutning			✓	✓	✓	✓
	Ta bort anslutning…	✓	✓	✓	✓	✓	✓

continues on next page

Table 4.1 – fortsättning från föregående sida

	Ny fältdomän ► Ny intervalldom	☑					
	Ny fältdomän ► Ny domän för kodade värden	☑					
	Ny fältdomän ► Ny global domän	☑					
	Ta bort <database_n	☑	☑				
	Kompakt databas (VACUUM)	☑					
	Skapa ett *Nytt schema…*			☑	☑	☑	
	Skapa en *Ny tabell…*	☑	☑	☑	☑		
	Utför SQL… fråga	☑	☑	☑	☑		
Schema	*Uppdatera* ett schema			☑	☑	☑	
	Schemaopera ► Byt namn på schema…			☑	☑	☑	
	Schemaopera ► Radera schema…			☑	☑	☑	
	Skapa en *Ny tabell…*			☑	☑		
	Utför SQL… fråga			☑	☑		
Tabell / lager	*Tabelloperati ► Byt namn på tabell…*			☑	☑	☑	
	Table Operations ► Truncate Table…			☑		☑	

continues on next page

Table 4.1 – fortsättning från föregående sida

	Utför SQL… fråga	☑	☑	☑		
	Exportera lager ► Till fil…	☑	☑	☑	☑	☑
	Manage ► Byt namn på lager <lager_namn	☑	☑			
	Hantera ► Ta bort lager <layer_name	☑	☑	☑	☑	☑
	Hantering ► Ta bort valda lager	☑	☑	☑	☑	☑
	Hantera ► Lägg till lager i projektet	☑	☑	☑	☑	☑
	Manage ► Lägg till valda lager i projektet	☑	☑	☑	☑	☑
	Öppna *Lageregenska* dialogrutan	☑	☑	☑	☑	☑
	Öppna dialogrutan *Filegenskape*	☑				
	Öppna med datakällhant	☑	☑			
Fält	*Lägg till nytt fält…*	☑	☑	☑	☑	
Fält	*Set Field Domain*	☑				
	Rename Field	☑				
	Set Alias…	☑				
	Set Comment…	☑		☑		
	Ta bort fält…	☑	☑	☑	☑	

Observera

[1] De olika posterna kan också vara tillgängliga för vissa vektorfilformat som stöds av GDAL <https://gdal.org/en/latest/drivers/vector/index.html>`_, t.ex. ESRI File Geodatabase, FlatGeobuf, GeoParquet, NetCDF, … när de är kompatibla.

När vektorlager laddas in i QGIS upptäcks fält med *Field Domains* (t.ex. de som definieras i en GeoPackage- eller ESRI File Geodatabase) automatiskt. Dessa domäner är begränsningar på databasnivå, vilket innebär att de verkställs av själva databasen och gäller för olika program, inte bara QGIS.

4.2.6 Kakelplattor och webbtjänster

Nivå	Kontextmeny	Typ av tjänster WMS / WMTS (*Reference*)	XYZ-plattor (*Reference*)	WCS (*Referenc*	WFS / OGC API - Funktioner (*Reference*)	ArcGIS REST-servrar (*Reference*)
Toppm	Skapa en *Ny anslutning…*	☑	☑	☑	☑	☑
	Spara anslutningar… detaljer till en fil	☑	☑	☑	☑	☑
	Ladda anslutningar…	☑	☑	☑	☑	☑
Anslut	*Refresh* anslutning	☑	☑	☑	☑	☑
	Redigera anslutning… inställningar	☑	☑	☑	☑	☑
	Duplicerad anslutning	☑	☑	☑	☑	☑
	Ta bort anslutning…	☑	☑	☑	☑	☑
	Visa serviceinformatio i webbläsaren					☑
Tabell / lager	*Exportera lager ► Till fil…*	☑	☑	☑	☑	☑
	Lägg till lager i projektet	☑	☑	☑	☑	☑
	Öppna *Lageregenskaper.* dialogrutan	☑	☑	☑	☑	☑
	Visa serviceinformatio i webbläsaren					☑

Nivå	Kontextmeny	Typ av tjänster Vektorplattor (*Reference*)	Scen (*Reference*)	Moln (*Reference*)	SensorThings (*Reference*)
Toppmeı	Skapa en *Ny anslutning…*		☑	☑	☑
	Skapa en *Ny generisk anslutning…*	☑			
	Skapa en *Ny ArcGIS Vector Tile Service Connection…*	☑			
	Spara anslutningar… detaljer till en fil	☑	☑	☑	☑
	Ladda anslutningar…	☑	☑	☑	☑
Anslutni	*Redigera anslutning…* inställningar	☑	☑	☑	☑
	Duplicerad anslutning	☑	☑	☑	☑
	Ta bort anslutning…	☑	☑	☑	☑
Tabell / lager	*Exportera lager ► Till fil…*				☑
	Lägg till lager i projektet	☑	☑		☑
	Öppna *Lageregenskaper…*-dialogrutan	☑	☑		☑

4.3 Resurser

- Projektfiler. Med snabbmenyn för QGIS-projektfiler kan du:
  - öppna det (*Öppna projekt*)
  - extract symbols (*Extract Symbols…*) - öppnar stilhanteraren där du kan exportera symboler till en XML-fil, lägga till symboler i standardstilen eller exportera som PNG eller SVG.
  - inspektera egenskaper (*File Properties…*)

  Du kan expandera projektfilen för att se dess lager. Kontextmenyn för ett lager erbjuder samma åtgärder som på andra ställen i bläddraren.
- QGIS-skiktdefinitionsfiler (QLR). Följande åtgärder är tillgängliga från snabbmenyn:
  - exportera det (*Exportera lager ► Till fil*)
  - lägga till det i projektet (*Add Layer to Project*)
  - inspektera egenskaper (*Lageregenskaper…*)
- Bearbetning av modeller (.model3). Följande åtgärder är tillgängliga från snabbmenyn:
  - *Kör modell…*)
  - *Ändra modell…*)
- QGIS mallar för utskriftskompositör (QPT). Följande åtgärd är tillgänglig från snabbmenyn:
  - (*Ny layout från mall*)
- Python-skript (.py). Följande åtgärder är tillgängliga från snabbmenyn:
  - (*Kör skript…*)
  - (*Öppna i extern editor*)

- Erkända rasterformat. Följande åtgärder är tillgängliga från snabbmenyn:
  - ta bort den (*Delete File <dataset name>*)
  - exportera det (*Exportera lager ► Till fil*)
  - lägga till det i projektet (*Add Layer to Project*)
  - inspektera egenskaper (*Lageregenskaper…*, *File Properties…*)

  För vissa format kan du också *Öppna <file type> Externt…*
- Erkända vektorformat. Följande åtgärder är tillgängliga från snabbmenyn:
  - ta bort den (*Delete File <dataset name>*)
  - exportera det (*Exportera lager ► Till fil*)
  - lägga till det i projektet (*Add Layer to Project*)
  - inspektera egenskaper (*Lageregenskaper…*, *File Properties…*)

  För vissa format kan du också *Öppna <file type> Externt…*

KAPITEL 5

QGIS-konfiguration

QGIS är mycket konfigurerbart. Genom menyn *Settings* finns det olika verktyg för att:

- *Style Manager…*: skapa och hantera *symboler, stilar och färgramper*.
- *Anpassade projektioner…*: skapa dina egna *koordinatreferenssystem*.
- *Tangentbordsgenvägar…*: definiera din egen uppsättning av *tangentbordsgenvägar*. De kan också åsidosättas under varje QGIS-session genom *projektegenskaper* (tillgängligt under *Projekt*-menyn).
- *Interface Customization…*: konfigurera *applikationsgränssnittet*, dölja dialogrutor eller verktyg som du kanske inte behöver.
- *Alternativ…*: ange globala *options* som ska tillämpas i olika delar av programmet. Dessa inställningar sparas i de aktiva inställningarna för *användarprofil* och tillämpas som standard när du öppnar ett nytt projekt med den här profilen.

5.1 Alternativ

Vissa grundläggande alternativ för QGIS kan väljas med hjälp av dialogrutan *Options*. Välj menyalternativet *Settings* ► *Options*. Du kan ändra alternativen efter dina behov. Vissa av ändringarna kan kräva en omstart av QGIS innan de börjar gälla.

De flikar där du kan anpassa dina alternativ beskrivs nedan.

Observera

Tillägg kan bädda in sina inställningar i dialogrutan Alternativ

Endast kärninställningarna presenteras nedan, men observera att listan kan utökas genom att *installerade tillägg* implementerar sina egna alternativ i standarddialogrutan Alternativ. Detta undviker att varje plugin har sin egen konfigurationsdialog med extra menyalternativ bara för dem…

5.1.1 Allmänna inställningar

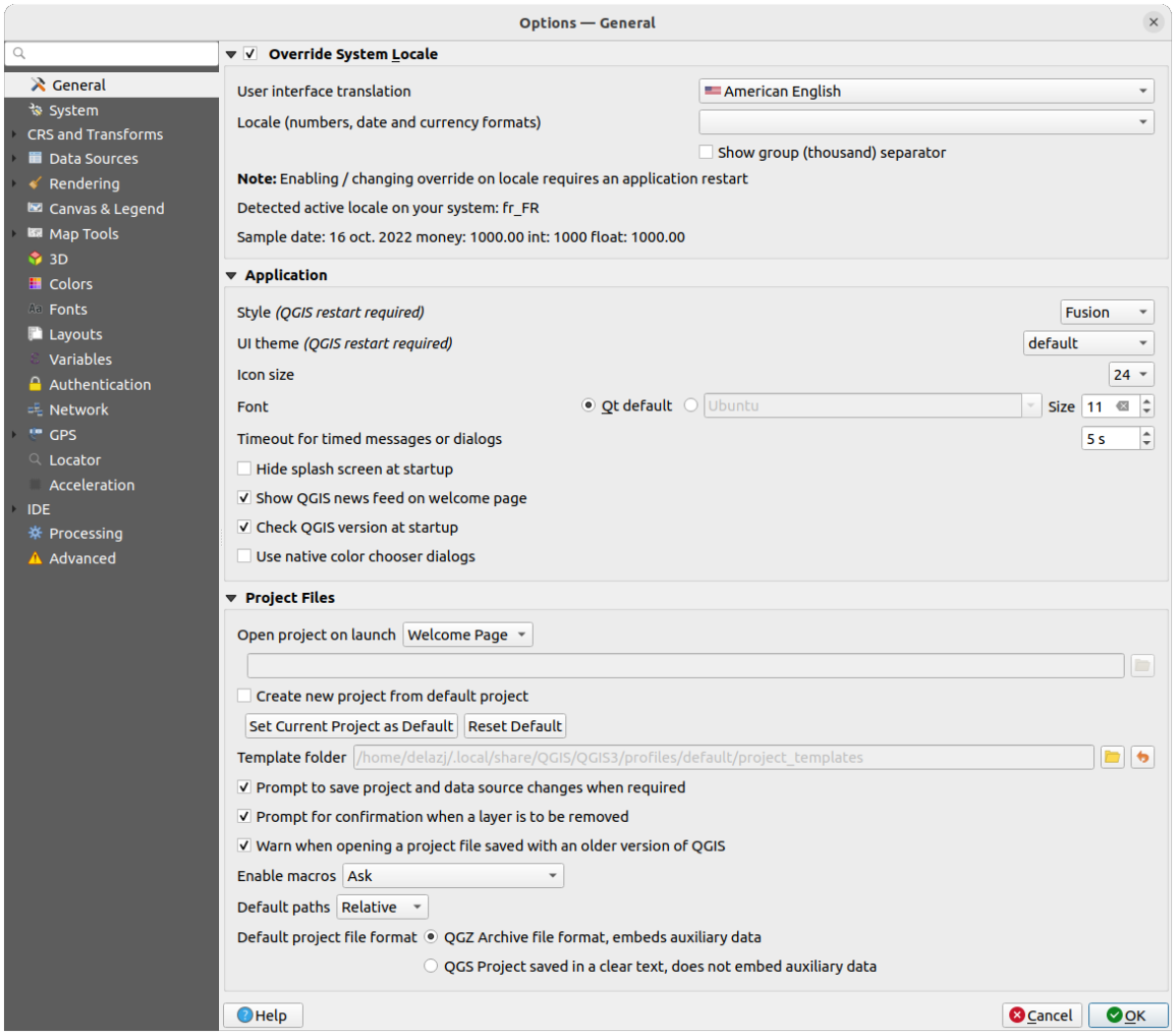


Fig. 5.1: Allmänna inställningar

Åsidosätta systemets lokala språk

Som standard förlitar sig QGIS på operativsystemets konfiguration för att ställa in språk och manipulera numeriska värden. Om du aktiverar den här gruppen kan du anpassa beteendet.

- Välj från *User interface translation* det språk som ska användas för GUI
- Välj i *Locale (tal-, datum- och valutaformat)* det system som datum och numeriska värden ska matas in och återges i
- *Visa separator för grupp (tusen)*

En sammanfattning av de valda inställningarna och hur de ska tolkas visas längst ned i ramen.

Du kan inte ändra inloggningsnyckelns applikation eftersom det skulle göra inloggningsnyckeln ogiltig

- Välj *Style (QGIS restart required)*, dvs. widgetarnas utseende och placering i dialogrutor. Möjliga värden beror på ditt operativsystem.
- Definiera *UI theme (QGIS restart krävs)* . Det kan vara 'default', 'Night Mapping' eller 'Blend of Grey'
- Definiera *ikonens storlek*

- Definiera *Font* och dess *Size*. Teckensnittet kan vara *Qt default* eller ett användardefinierat
- Ändra *Timeout för tidsinställda meddelanden eller dialogrutor*
- *Dölj startskärm vid uppstart*
- *Visa QGIS nyhetsflöde på välkomstsidan*: visar ett samlat QGIS nyhetsflöde på välkomstsidan, vilket ger dig ett direkt sätt att bli medveten om projektnyheter (datum och sammanfattning för användar-/utvecklarmöten, communityundersökningar, meddelanden om releaser, olika tips …)
- *Kontrollera QGIS-versionen vid start* för att hålla dig informerad om en nyare version släpps
- *Use native color chooser dialogs* (se *Färgväljare*)

Projektfiler

- *Öppet projekt vid lansering*
  - ”Välkomstsida” (standard): kan visa flödet ”Nyheter”, projektmallar och de senaste projekten (med miniatyrbilder) för *användarprofilen*. Inget projekt öppnas som standard.
  - ”New”: öppnar ett nytt projekt, baserat på standardmallen
  - ”Senaste”: öppnar det senast sparade projektet igen
  - och ”Specific”: öppnar ett visst projekt. Använd knappen … för att definiera vilket projekt som ska användas som standard.
- *Skapa nytt projekt från standardprojekt*. Du har möjlighet att trycka på *Set current project as default* eller på *Reset default*. Du kan bläddra igenom dina filer och definiera en katalog där du hittar dina användardefinierade projektmallar. Detta kommer att läggas till i *Projekt* ► *Nytt från mall*. Om du först aktiverar *Create new project from default project* och sedan sparar ett projekt i mappen för projektmallar.
- *Fråga om att spara ändringar i projekt och datakällor när så krävs* för att undvika att ändringar du gjort går förlorade.
- *Fråga efter bekräftelse när ett lager ska tas bort*
- *Varning när du öppnar en projektfil som sparats med en äldre version av QGIS*. Du kan alltid öppna projekt som skapats med en äldre version av QGIS, men när projektet har sparats kan försök att öppna det med en äldre version misslyckas på grund av funktioner som inte finns tillgängliga i den versionen.
- *Aktivera projektets inbäddade Python-kod* . Det här alternativet hanterar exekvering av makron som skrivs för att utföra en åtgärd på projekthändelser, samt anpassade Python-funktioner som ska användas som uttryck. Du kan välja mellan ”Aldrig”, ”Fråga”, ”Endast för den här sessionen” och ”Alltid (rekommenderas inte)”.
- *Default paths*: anger om sökvägar till filer och lager som används i nya projekt ska lagras som ”Absoluta” eller ”Relativa” i förhållande till projektfilen. Denna inställning kan skrivas över på projektnivå.
- *Förvalt projektfilformat*
  - *QGZ Archive filformat, bäddar in hjälpdata* (se *hjälpdata*)
  - *QGS-projekt sparas i klartext, bäddar inte in hjälpdata*: hjälpdata lagras i en separat `.qgd`-fil tillsammans med projektfilen.

5.1.2 Systeminställningar

SVG-vägar

Lägg till eller ta bort *Sökväg(ar) för att söka efter SVG-symboler (Scalable Vector Graphic)*. Dessa SVG-filer är sedan tillgängliga för att symbolisera eller märka funktionerna eller dekorera din kartkomposition.

Läs även *Väljare för fjärrstyrd eller inbäddad fil* för olika sätt att hänvisa till svg-filer i en QGIS-sökväg.

Tilläggssökvägar

Lägg till eller ta bort *Sökväg(ar) för att söka efter ytterligare C++ plugin-bibliotek*.

Dokumentationsvägar

Lägg till eller ta bort *Dokumentationssökväg(ar)* som ska användas för QGIS-hjälp. Som standard läggs en länk till den officiella användarhandboken online som motsvarar den version som används till (t.ex. `https://docs.qgis.org/$qgis_short_version/$qgis_locale/docs/user_manual/`). Du kan dock lägga till andra länkar och prioritera dem från topp till botten: varje gång du klickar på en *Hjälp*-knapp i en dialogruta kontrolleras den översta länken och om ingen motsvarande sida hittas, försöker man med nästa, och så vidare.

Observera

Dokumentation versionshanteras och översätts endast för QGIS Long Term Releases (LTR), vilket innebär att om du kör en vanlig version (t.ex. QGIS 3.0) kommer hjälpknappen som standard att öppna nästa LTR-manualsida (dvs. 3.4 LTR), som kan innehålla beskrivningar av funktioner i nyare versioner (3.2 och 3.4). Om ingen LTR-dokumentation finns tillgänglig används *testing*-dokumentet, med funktioner från nyare versioner och utvecklingsversioner.

Inställningar

Den hjälper dig *Återställ användargränssnittet till standardinställningarna (omstart krävs)* om du har gjort någon *customization*.

Miljö-URL:erna har sparats

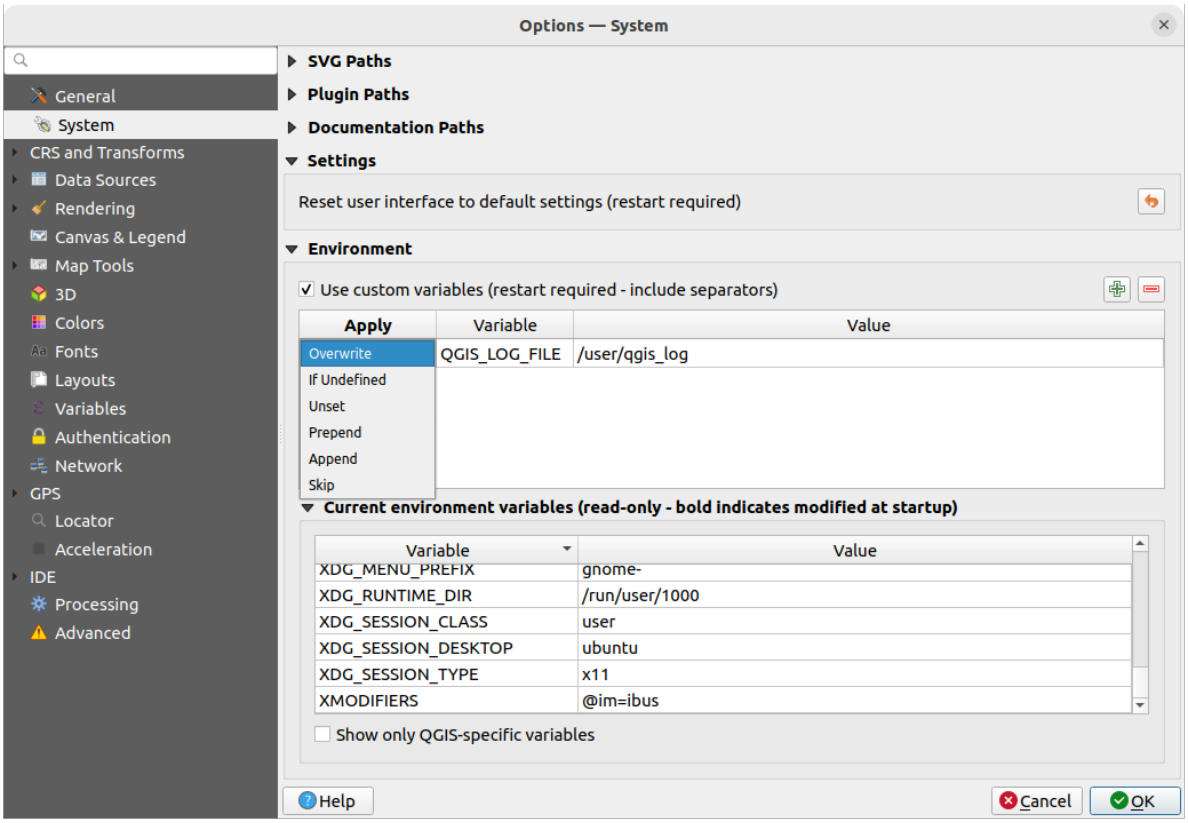


Fig. 5.2: Variabler för systemmiljö

Systemets miljövariabler kan visas och många av dem kan konfigureras i gruppen **Environment**. Detta är användbart för plattformar, t.ex. Mac, där en grafisk gränssnittsapplikation inte nödvändigtvis ärver användarens shell-miljö. Det är också användbart för att ställa in och visa miljövariabler för de externa verktygsuppsättningar som styrs av Processing toolbox (t.ex. SAGA, GRASS) och för att aktivera felsökningsutmatning för specifika delar av källkoden.

Om du markerar kryssrutan *Use custom variables (restart required - include separators)* kan du lägga till och ta bort miljövariablerna Add och Remove. För varje nytt objekt kan du konfigurera namnet på en *Variabel*, dess *Värde* och metoden *Apply* som ska användas, bland annat:

- *Overwrite*: ersätter alla tidigare värden för variabeln
- *Om odefinierad*: använd detta värde för variabeln om den inte redan är definierad på en högre nivå (t.ex. OS- eller applikationsnivå)
- *Unset*: tar bort variabeln från miljön (parametern *Value* används inte)
- *Prepend*: prepend värdet till variabelns redan existerande värde
- *Append*: lägger till värdet till variabelns befintliga värde
- *Skip*: objektet sparas i listan för framtida referens men används inte

Redan definierade miljövariabler visas i *Aktuella miljövariabler*, och det är möjligt att filtrera dem genom att aktivera kryssrutan *Visa endast QGIS-specifika variabler*.

5.1.3 Inställningar för användarprofiler

Observera

Om du vill ha mer information om hur du hanterar användarprofiler kan du läsa det särskilda avsnittet på *Arbeta med användarprofiler*.

5.1.4 CRS- och transformationsinställningar

Observera

För mer information om hur QGIS hanterar lagerprojektion, läs det dedikerade avsnittet på *Arbeta med projektioner*.

CRS Hantering

På fliken *CRS Handling* kan du konfigurera vilken CRS som ska användas för ett nytt projekt eller lager.

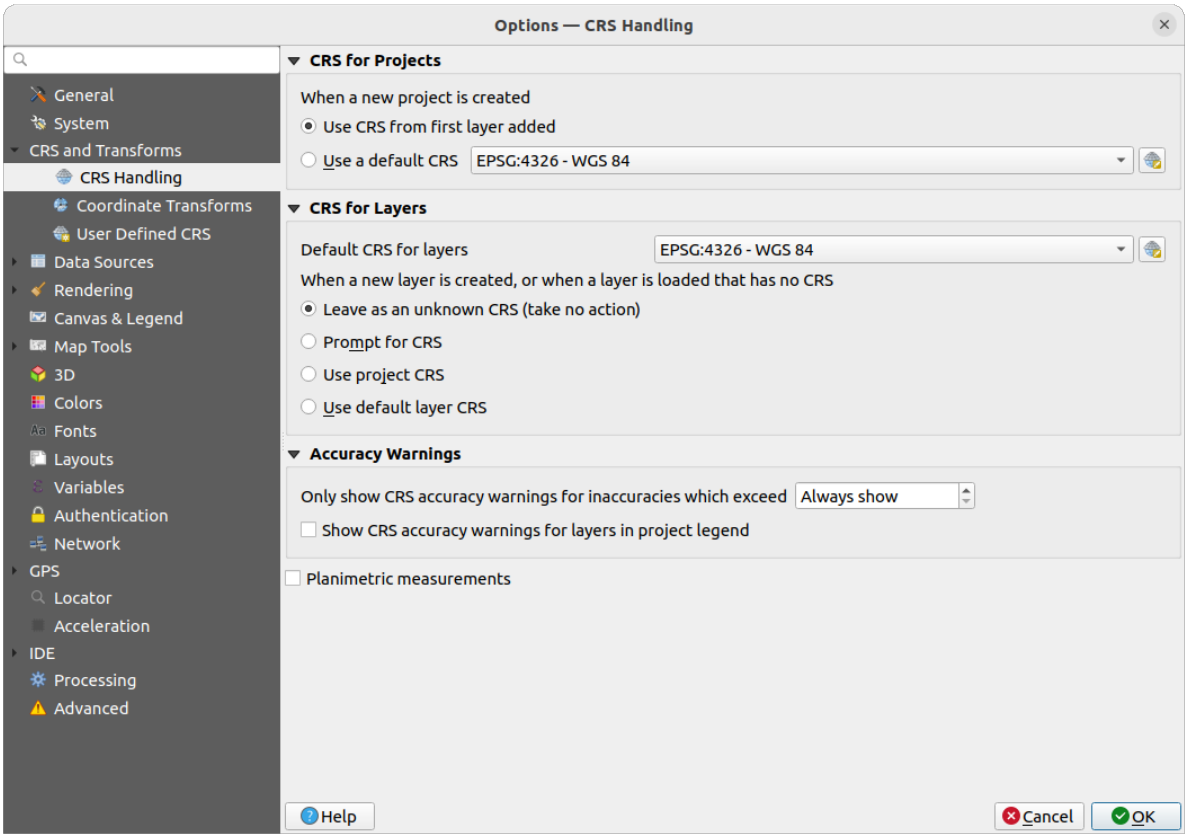


Fig. 5.3: CRS-inställningar

CRS för projekt

Det finns ett alternativ för att automatiskt ställa in CRS för nya projekt:

- *Använd CRS från det första lagret som läggs till*: CRS för projektet kommer att ställas in på CRS för det första lagret som laddas in i det

- *Use a default CRS*: ett förvalt CRS används som standard i alla nya projekt och lämnas oförändrat när lager läggs till i projektet.

Valet kommer att sparas för användning i efterföljande QGIS-sessioner. Projektets koordinatreferenssystem kan fortfarande åsidosättas från fliken *Projekt* ► *Egenskaper…* ► *CRS* fliken.

CRS för lager

Default CRS for layers: välj ett CRS som ska användas när du skapar ett lager

Du kan också definiera vilken åtgärd som ska vidtas när ett nytt lager skapas eller när ett lager utan CRS laddas.

- *Lämna som okänd CRS (vidta inga åtgärder)*
- *Uppmaning till CRS`*
- *Använda CRS i projektet*
- *Använd standardlager CRS`*

Varning för felaktigheter

Visa endast CRS-noggrannhetsvarningar för felaktigheter som överstiger ett visst avstånd: inträffar när du uttryckligen skapar eller ändrar en dataset och väljer ett CRS baserat på en datumensemble med lägre noggrannhet. Standardinställningen är att ”alltid visa” varningen om någon felaktighet. Kräver en QGIS-version som använder minst PROJ 8.0.

Visa varning för CRS-noggrannhet för lager i projektlegenden: Om markerad kommer alla lager med ett CRS med noggrannhetsproblem (dvs. ett dynamiskt CRS utan tillgänglig koordinatepok eller ett CRS baserat på en datumensemble med inneboende felaktighet som överstiger den gräns som användaren har ställt in) att ha varningsikonen i panelen *Layers* som visar att det är ett lager med låg noggrannhet.

Detta är utformat för användning inom teknik, BIM, kapitalförvaltning och andra områden där felaktigheter i mätar-/submeternivå potentiellt kan vara mycket farliga eller dyra!

Planimetriska mätningar: anger standardvärdet för egenskapen *planimetriska mätningar* för nyskapade projekt.

Koordinattransformationer

Fliken *Coordinate Transforms* hjälper dig att ställa in koordinattransformationer och operationer som ska tillämpas när du laddar ett lager till ett projekt eller återprojicerar ett lager.

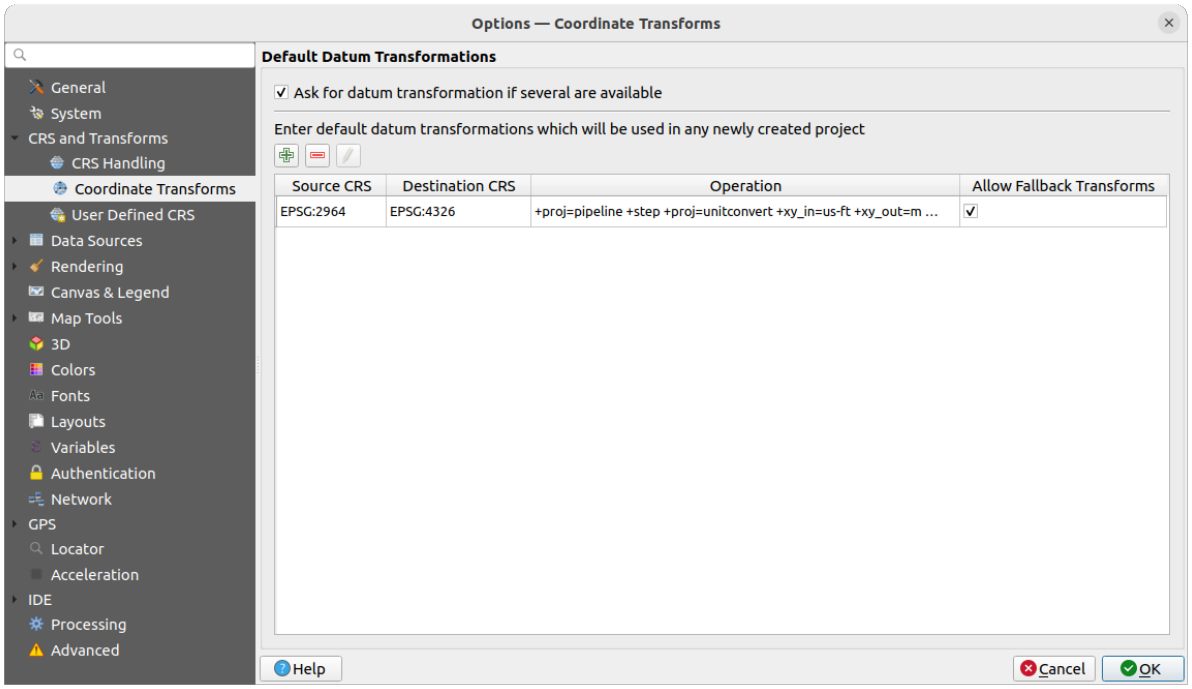


Fig. 5.4: Inställningar för transformationer

Transformationer av standarddatum

Här kan du styra om lager ska omprojiceras till en annan CRS:

- bearbetas automatiskt med hjälp av QGIS standardinställningar för transformationer;
- och/eller mer styrda av dig med anpassade preferenser som t.ex:
  - *Begära datumtransformation om flera är tillgängliga*
  - en fördefinierad lista med datumtransformationer som ska tillämpas som standard. Se *Datumtransformationer* för mer information.

Du kan använda transformationerna Add, Remove eller Edit, som kommer att användas i alla nyskapade projekt.

Användardefinierad CRS

Fliken *User Defined CRS* hjälper dig att definiera en anpassad CRS som måste överensstämma med ett WKT- eller Proj-strängformat.

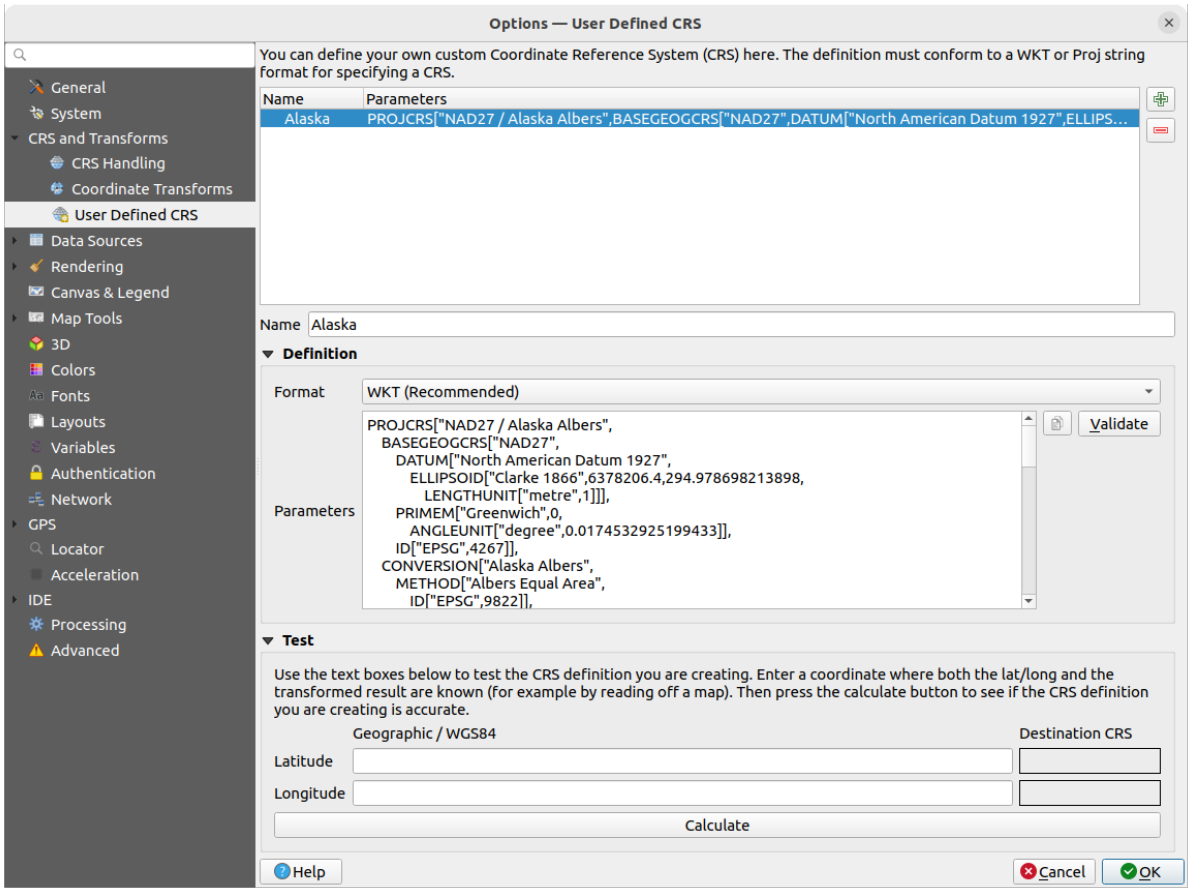


Fig. 5.5: Användardefinierad CRS

Ange ett *Namn* och använd Lägg till ny CRS. Om du vill ta bort en befintlig CRS kan du använda Remove CRS.

Definition

- ***Format***
  - WKT (rekommenderas)
  - Proj String (Legacy - rekommenderas ej)
- ***Parametrar***
  - Kopiera parametrar från en befintlig CRS.
  - *Validate* testar om ditt uttryck är korrekt.

Test

Här kan du testa din skapade CRS-definition med hjälp av latitud och longitud. Använd en känd koordinat för att kontrollera om din definition är korrekt.

5.1.5 Inställningar för datakällor

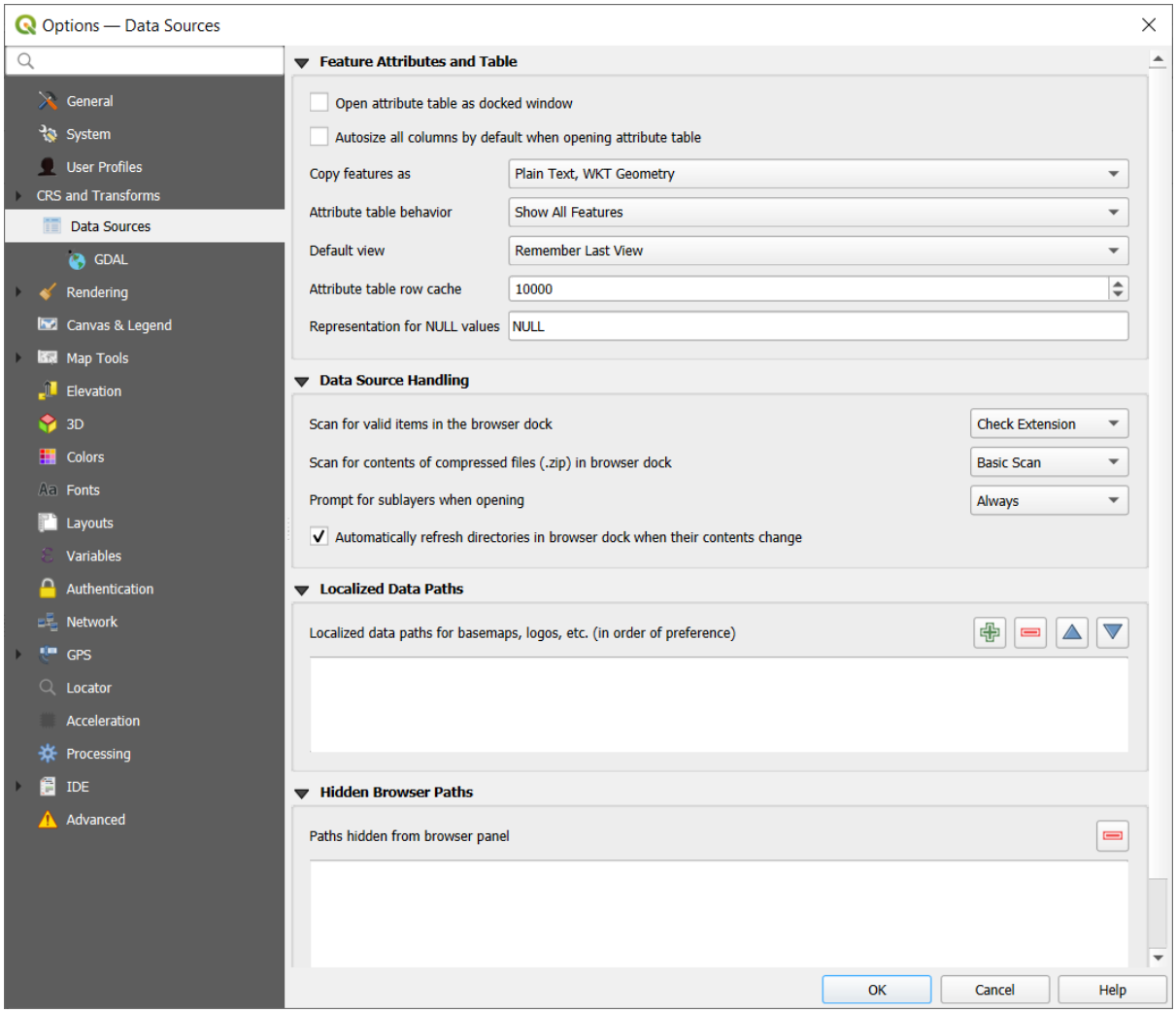


Fig. 5.6: Inställningar för datakällor

Förhållande till attribut och tabell

- *Öppna attributtabell som dockat fönster*
- *Automatisera alla kolumner som standard när attributtabellen öppnas*
- *Copy features as*: definierar det format som ska användas för data när funktioner klistras in i andra program.
  - *Enklare text, ingen geometri*: attribut kopieras som text men geometrin hoppas över
  - *Plain text, WKT geometry*: attribut kopieras som text och geometrin returneras i WKT
  - *Plain text, WKB geometry*: attribut kopieras som text och geometrin returneras i WKB (hex-format)
  - *GeoJSON*: attribut och geometri formateras som GeoJSON-data
- *Attribute table behavior* : Ställ in filter på attributtabellen vid öppningen. Det finns tre möjligheter: ”Visa alla funktioner”, ”Visa valda funktioner” och ”Visa funktioner som är synliga på kartan”.
- *Default view*: definierar visningsläget för attributtabellen vid varje öppning. Det kan vara ”Kom ihåg senaste vyn”, ”Tabellvy” eller ”Formulärvy”.
- *Radcache för attributtabell* 1,00. Denna radcache gör det möjligt att spara de senast inlästa N attributraderna så att det går snabbare att arbeta med attributtabellen. Cachen raderas när attributtabellen stängs.

- *Representation för NULL-värden*. Här kan du definiera ett värde för datafält som innehåller ett NULL-värde.

Tips

Förbättra öppningen av attributtabellen för stora datamängder

När du arbetar med lager med många poster kan det gå långsamt att öppna attributtabellen eftersom dialogrutan begär alla rader i lagret. Om du ställer in *Attributtabellens beteende* till **Visa funktioner som är synliga på kartan** kommer QGIS att begära endast funktionerna i den aktuella kartbilden när tabellen öppnas, vilket möjliggör en snabb dataladdning.

Observera att data i denna attributtabellinstans alltid kommer att vara knutna till den kartutbredning som den öppnades med, vilket innebär att om du väljer **Show All Features** i en sådan tabell visas inte nya funktioner. Du kan dock uppdatera uppsättningen av visade funktioner genom att ändra canvasutbredningen och välja alternativet **Show Features Visible On Map** i attributtabellen.

Hantering av datakällor

- *Sök efter giltiga objekt i bläddrarens docka* . Du kan välja mellan ”Kontrollera filtillägg” och ”Kontrollera filinnehåll”.
- *Sök efter innehåll i komprimerade filer (.zip) i bläddrarens docka* definierar hur detaljerad widgetinformationen längst ned i bläddrarpanelen ska vara när du söker efter sådana filer. ”Nej”, ”Grundläggande skanning” och ”Fullständig skanning” är möjliga alternativ.
- *Fråga efter underlager när du öppnar*. Vissa raster stöder sublayers — de kallas subdataset i GDAL. Ett exempel är netCDF-filer — om det finns många netCDF-variabler, ser GDAL varje variabel som en underdatamängd. Med alternativet kan du styra hur sublayers ska hanteras när en fil med sublayers öppnas. Du har följande valmöjligheter:
  - ”Alltid”: Fråga alltid (om det finns befintliga underlager)
  - ”Om det behövs”: Fråga om lagret inte har några band, men har underlager
  - ”Aldrig”: Aldrig prompt, laddar inte något
  - ”Ladda alla”: Aldrig fråga, men ladda alla underlager
- *Automatisk uppdatering av kataloger i bläddrarens docka när deras innehåll ändras*: Låter dig manuellt välja bort övervakning av kataloger i *Bläddrare*-panelen som standard (t.ex. för att undvika potentiella långsamheter på grund av nätverkslatens).

Lokaliserade datasökvägar

Det är möjligt att använda lokaliserade sökvägar för alla typer av filbaserade datakällor. Det är en lista med sökvägar som används för att abstrahera datakällans plats. Om t.ex. `C:\my_maps` listas i de lokaliserade sökvägarna kommer ett lager som har `C:\my_mapsmy_countryortho.tif` som datakälla att sparas i projektet med `localized:my_countryortho.tif`.

Sökvägarna är listade i prioritetsordning, med andra ord kommer QGIS först att leta efter filen i den första sökvägen, sedan i den andra osv.

Dolda sökvägar i bläddraren

Denna widget listar alla mappar som du har valt att dölja från *Browser panel*. Om du tar bort en mapp från listan blir den tillgänglig i *Browser*-panelen.

GDAL-inställningar

GDAL är ett datautbytesbibliotek för geospatiala data som stöder ett stort antal vektor- och rasterformat. Det tillhandahåller drivrutiner för att läsa och (ofta) skriva data i dessa format. Fliken *GDAL* exponerar drivrutinerna för raster- och vektorformat med deras funktioner.

GDAL raster- och vektordrivrutiner

På flikarna *Raster Drivers* och *Vector Drivers* kan du definiera vilken GDAL-drivrutin som ska användas för att läsa och/eller skriva filer, eftersom det i vissa fall finns mer än en GDAL-drivrutin tillgänglig.

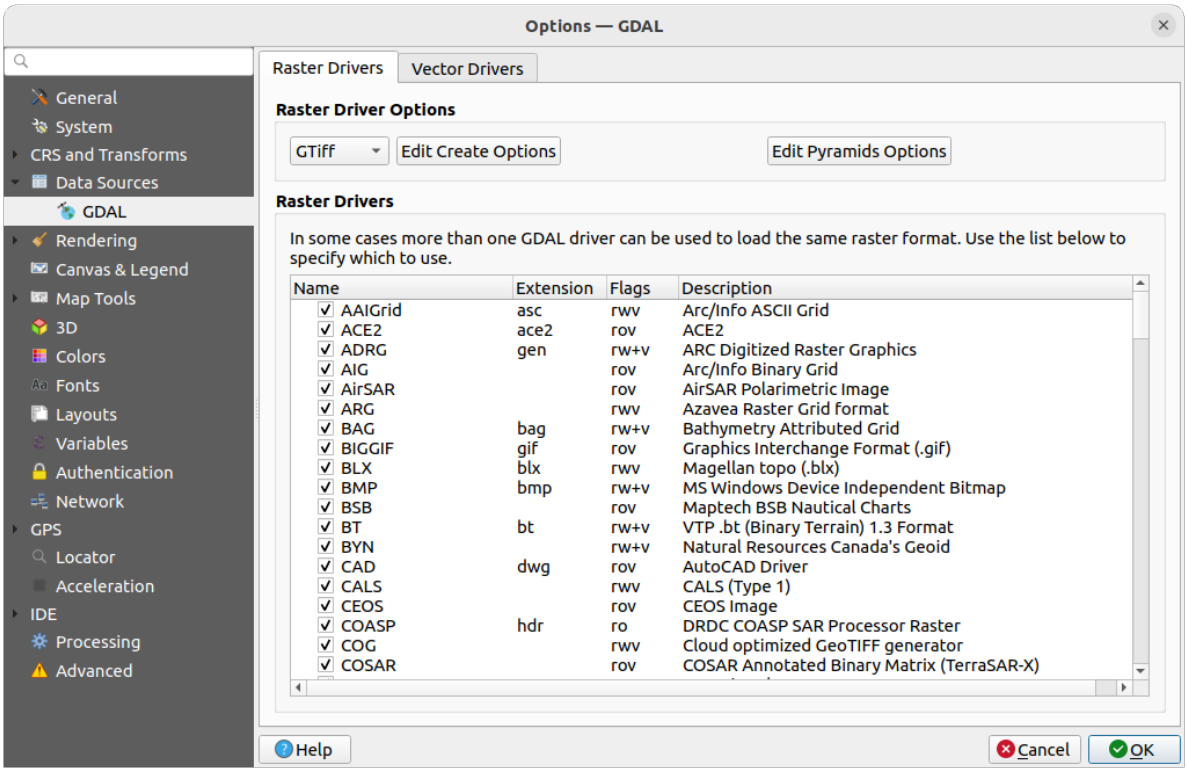


Fig. 5.7: GDAL-inställningar - Raster-drivrutiner

> **Tips**
>
> Om du dubbelklickar på en rasterdrivrutin som tillåter läs- och skrivåtkomst (`rw+ (v)`) öppnas dialogrutan *Edit Create options* för anpassning.

Alternativ för rosterdrivrutin

Denna ram innehåller sätt att anpassa beteendet hos rasterdrivrutiner som stöder läs- och skrivåtkomst:

- *Edit create options*: Här kan du redigera eller lägga till olika profiler för filtransformation, dvs. en uppsättning fördefinierade kombinationer av parametrar (typ och nivå av komprimering, blockstorlek, översikt, kolorimetri, alfa…) som ska användas vid utmatning av rasterfiler. Parametrarna beror på drivrutinen.

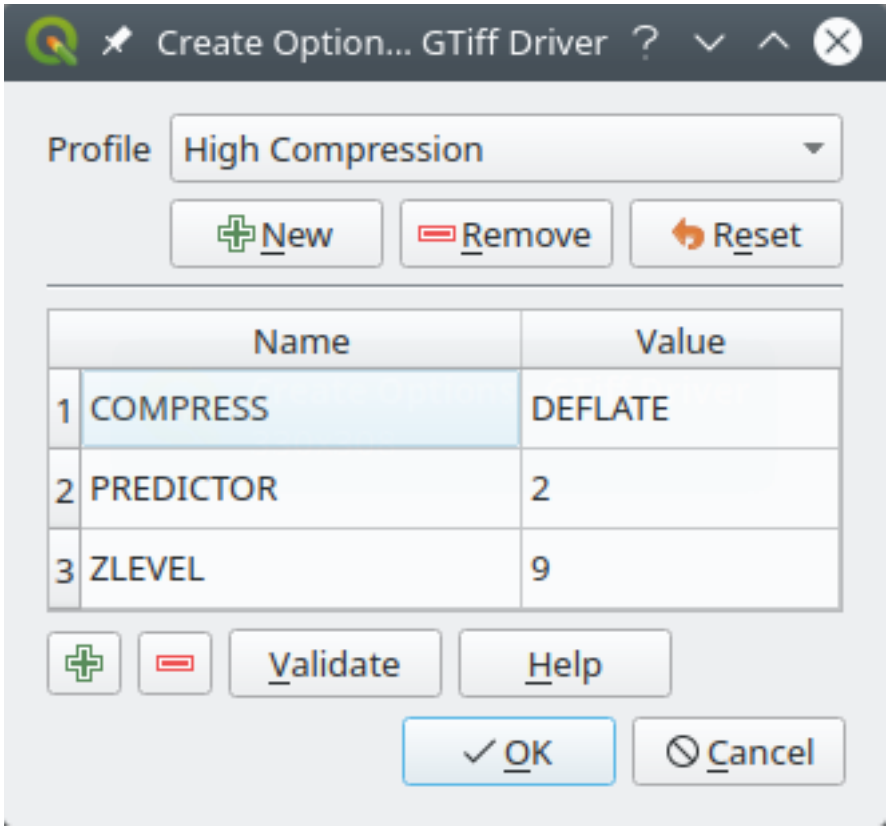


Fig. 5.8: Exempel på profil för att skapa alternativ (för GeoTiff)

I den övre delen av dialogrutan listas de aktuella profilerna och du kan lägga till nya eller ta bort någon av dem. Du kan också återställa profilen till dess standardparametrar om du har ändrat dem. Vissa drivrutiner (t.ex. GeoTiff) har ett urval av profiler som du kan arbeta med.

Längst ner i dialogrutan:

- Med knappen kan du lägga till rader som fylls i med parameternamn och värde
- Knappen raderar den valda parametern
- Klicka på knappen *Validate* för att kontrollera att de alternativ för skapande som angetts för det angivna formatet är giltiga
- Använd knappen *Help* för att hitta de parametrar som ska användas, eller se dokumentationen för GDAL-rasterdrivrutiner <https://gdal.org/en/latest/drivers/raster/index.html>`_.

- *Ändra alternativ för pyramider*

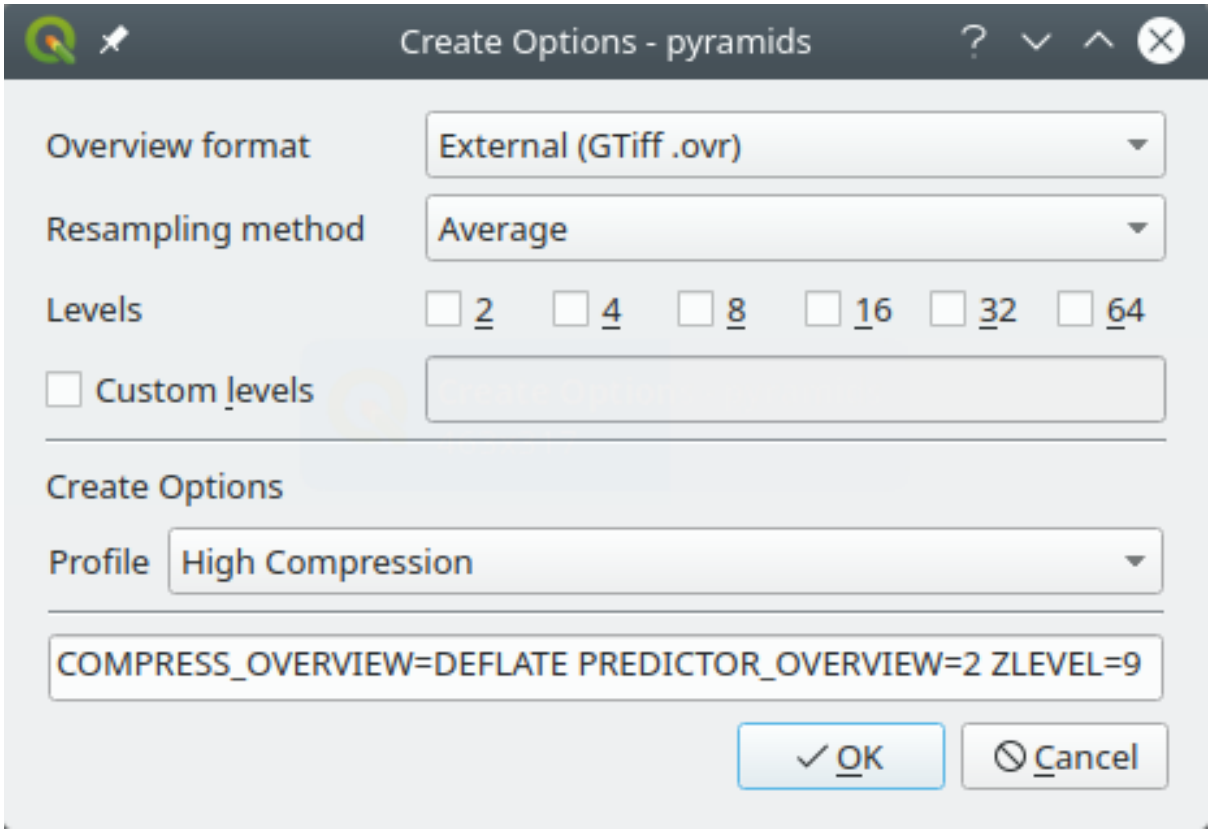


Fig. 5.9: Prov på pyramidernas profil

5.1.6 Renderingsinställningar

Fliken *Rendering* innehåller inställningar för att styra rendering av lager i kartbilden.

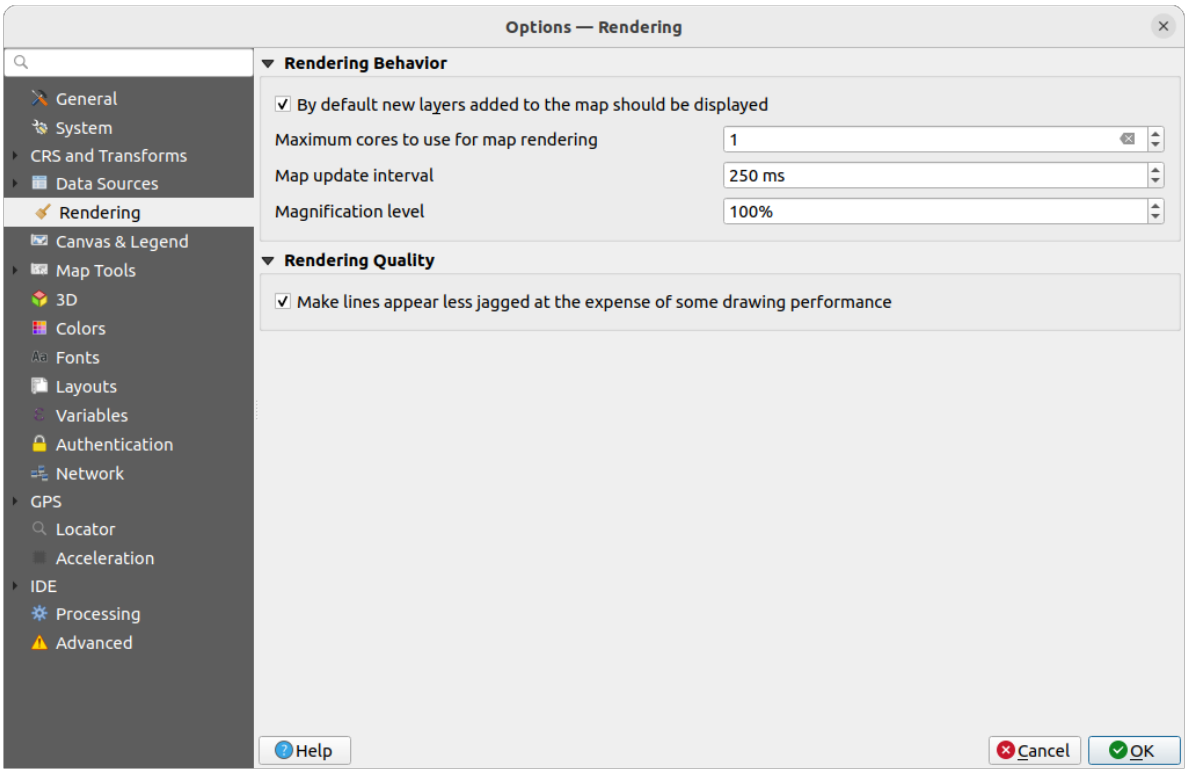


Fig. 5.10: Renderingsinställningar

Rendering av beteende

- *Som standard ska nya lager som läggs till på kartan visas*: att avmarkera det här alternativet kan vara praktiskt när du laddar flera lager för att undvika att varje nytt lager renderas i canvas och saktar ner processen
- Ställ in *Maximala kärnor att använda för kartrendering*
- Kartan renderas i bakgrunden på en separat bild och vid varje *Map update interval* (standardvärde 250 ms) kommer innehållet från den här (off-screen) bilden att användas för att uppdatera den synliga skärmrepresentationen. Men om renderingen avslutas snabbare än denna tid, kommer den att visas direkt.
- *förstoringsgrad* (se *förstoringsgrad*)

Renderingskvalitet

- *Får linjer att se mindre ojämna ut på bekostnad av viss ritprestanda*

Inställningar för vektorrendering

Fliken *Vector* innehåller specifika inställningar för rendering av vektorlager.

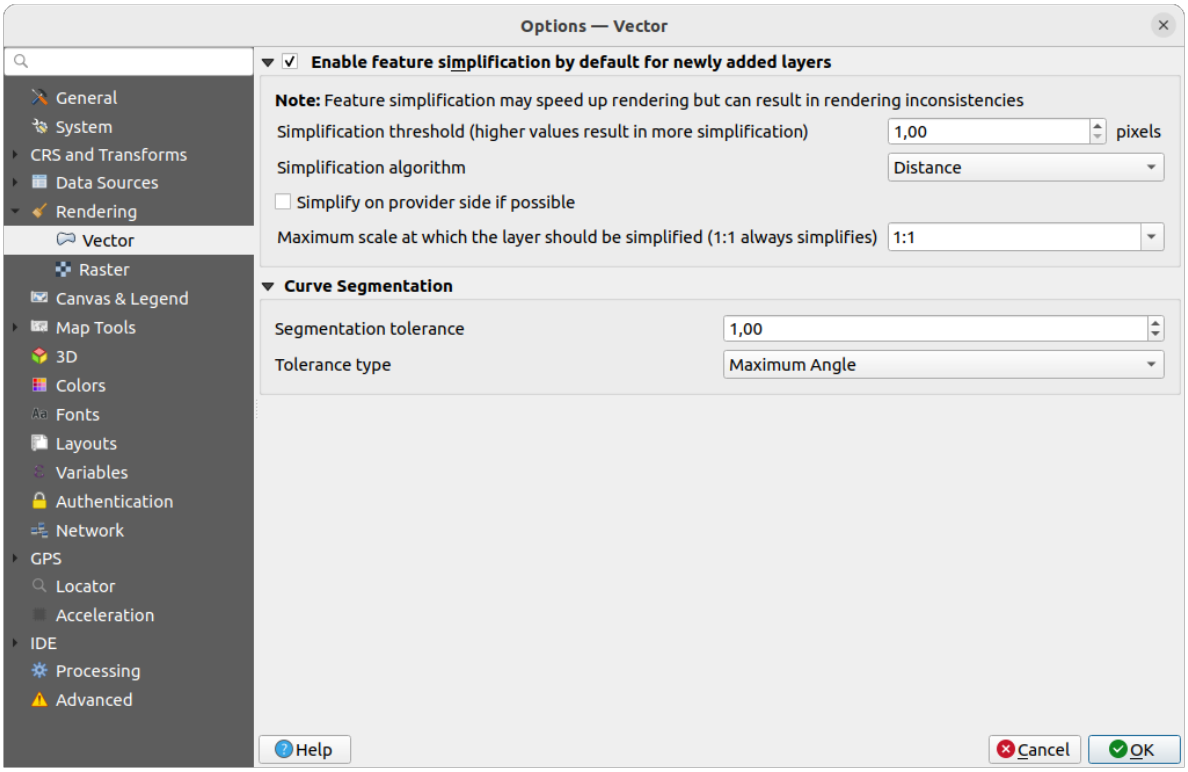


Fig. 5.11: Inställningar för vektorrendering

- *Aktivera förenkling av funktioner som standard för nytillkomna lager*: du förenklar funktionernas geometri (färre noder) och som ett resultat visas de snabbare. Tänk på att detta kan orsaka inkonsekvenser i renderingen. Tillgängliga inställningar är:
  - *Tröskelvärde för förenkling (högre värden ger mer förenkling)*
  - *Simplifikationsalgoritm*: Detta alternativ utför en lokal ”on-the-fly”-förenkling av objekt och påskyndar geometrirenderingen. Det ändrar inte den geometri som hämtas från dataleverantörerna. Detta är viktigt när du har uttryck som använder objektets geometri (t.ex. beräkning av area) - det säkerställer att dessa beräkningar görs på den ursprungliga geometrin, inte på den förenklade. För detta ändamål tillhandahåller QGIS tre algoritmer: ’Distance’ (standard), ’SnapToGrid’ och ’Visvalingam’.
  - *Förenkla på leverantörssidan om möjligt*: geometrierna förenklas av leverantören (PostGIS, Oracle …) och till skillnad från förenklingen på den lokala sidan kan geometribaserade beräkningar påverkas
  - *Maximala skalan vid vilken lagret ska förenklas (1:1 förenklar alltid)*

> **Observera**
>
> Förutom den globala inställningen kan förenklingen av funktioner ställas in för varje specifikt lager från dess *Lageregenskaper* ► *Rendering*-meny.

- *Kurvsegmentering*
  - *Segmenteringstolerans*: denna inställning styr hur cirkelbågar återges. **Ju mindre** maximal vinkel (mellan de två på varandra följande hörnen och kurvans mittpunkt, i grader) eller maximal skillnad (avstånd mellan

segmentet för de två hörnen och kurvlinjen, i kartenheter), desto **mer raka linjesegment** kommer att användas vid renderingen.

- *Toleranstyp*: det kan vara *Maximal vinkel* eller *Maximal skillnad* mellan approximation och kurva.

Inställningar för rasterrendering

Fliken *Raster* innehåller specifika inställningar för rendering av rasterlager.

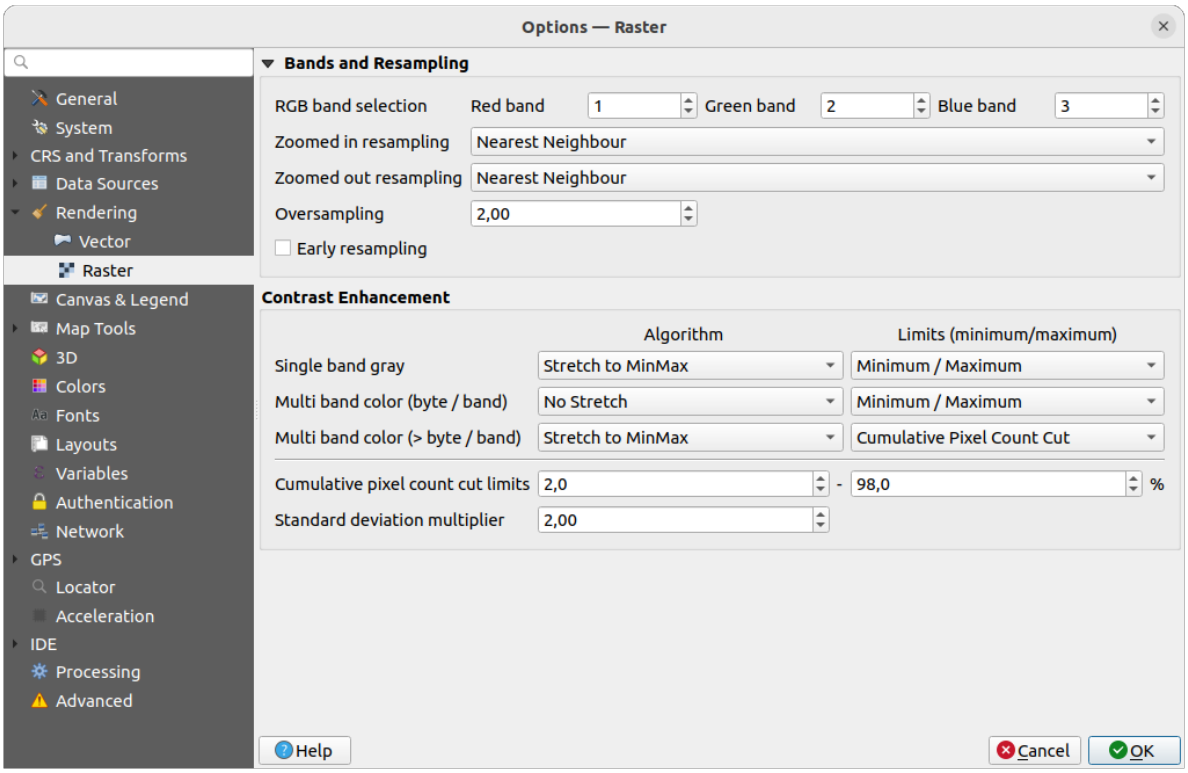


Fig. 5.12: Inställningar för rasterrendering

Under *Band och omsampling*:

- Med *RGB band selection* kan du definiera numret för det röda, gröna och blå bandet.
- Metoderna *Zoomed in resampling* och *Zoomed out resampling* kan definieras. För *Zoomed in resampling* kan du välja mellan tre omsamplingsmetoder: 'Nearest neighbour', 'Bilinear (2x2 kernel)' och 'Cubic (4x4 kernel)'. För *Zoomed out resampling* kan du välja mellan 'Nearest Neighbour' och 'Average'. Du kan också ställa in värdet för *Oversampling* (mellan 0,0 och 99,99 - ett stort värde innebär mer arbete för QGIS - standardvärdet är 2,0).
- *Early resampling*: gör det möjligt att beräkna rasterrenderingen på leverantörsnivå där källans upplösning är känd, och säkerställer en bättre zoom i rendering med QGIS anpassad styling. Mycket praktiskt för kakelraster som laddats med en *tolkningsmetod*. Alternativet kan också ställas in på lagernivå (*Symbology* properties)

alternativen *Contrast Enhancement* kan tillämpas på *Single band gray*, *Multi band color (byte/band)* eller *Multi band color (>byte/band)*. För var och en kan du ställa in:

- den *Algoritm* som ska användas, vars värden kan vara 'No stretch', 'Stretch to MinMax', 'Stretch and Clip to MinMax' eller 'Clip to MinMax'
- den *Limits (minimum/maximum)* som ska tillämpas, med värden som "Cumulative pixel count cut", "Minimum/Maximum", "Mean +/- standard deviation".

Alternativen för *Kontrastförbättring* inkluderar också:

- *Kumulativa gränser för pixelantalet*
- *Standardavvikelsemultiplikator*

5.1.7 Inställningar för Canvas och Legend

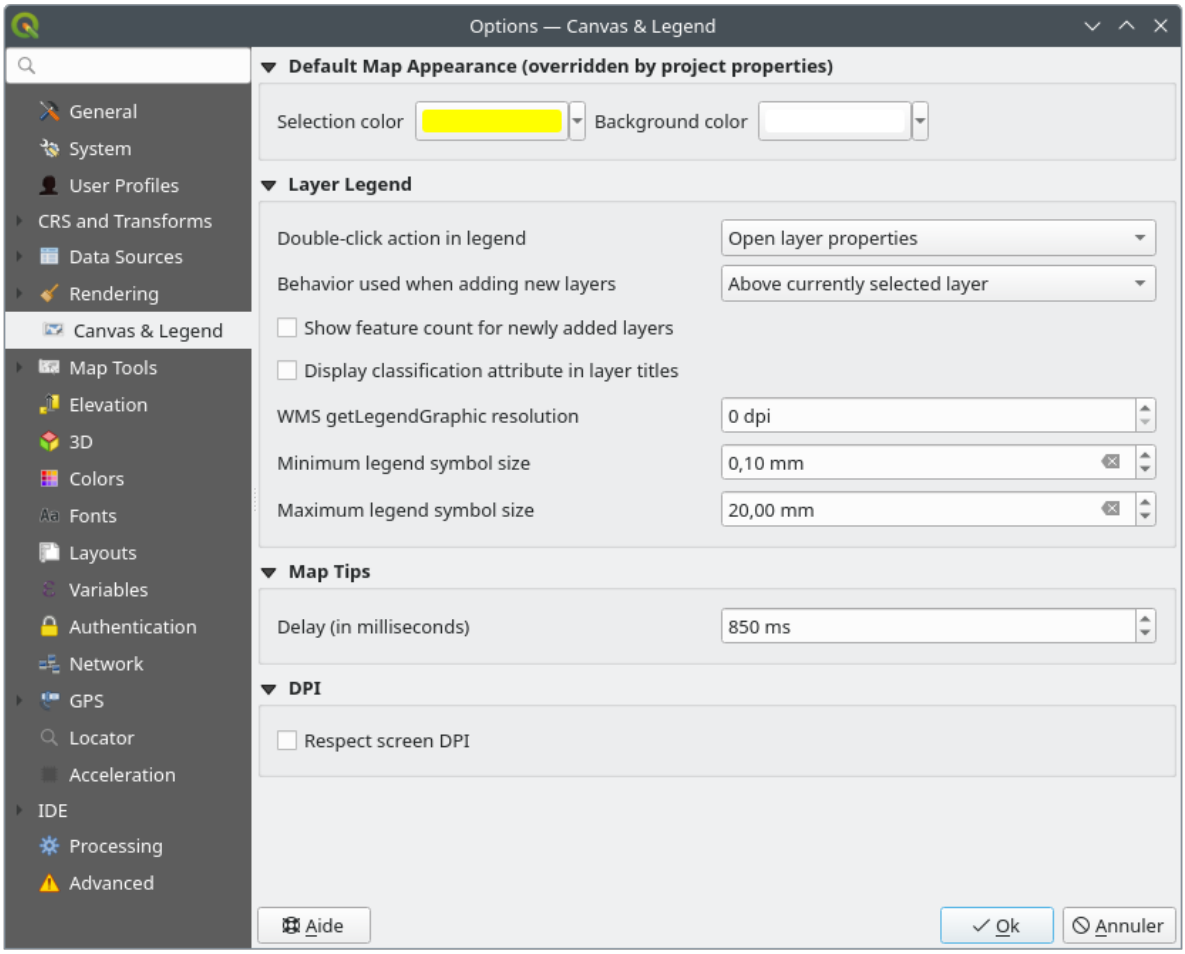


Fig. 5.13: Inställningar för Canvas och Legend

Dessa egenskaper låter dig ställa in:

- **Default kartutseende (åsidosätts av projektegenskaper)**: *Selection color* och *Background color*.
- **Layer förklaring** interaktion:
  - *Dubbelklicksåtgärd i förklaring*: om ett dubbelklick på ett lager ska antingen *Öppna lageregenskaper* (standard), *Öppna attributtabell* eller *Öppna lagerstylingdocka*.
  - *Behavior used when adding new layers*: avgör var lager placeras i panelen *Layers* när de laddas in i QGIS. Det kan vara:
    - *Över aktuellt valt lager*
    - *Alltid högst upp i lagerträdet*
    - *Optimalt index inom aktuell lagerträdsgrupp*: Till skillnad från de andra alternativen som sorterar de nya lagren bland dem och placerar dem som en stapel på önskad plats, utvidgar det här alternativet sorteringslogiken till hela lagerträdet (eller den aktiva gruppen) och infogar nya lager på ett ”optimalt” sätt genom att se till att punktlager ligger ovanpå punktlager, följt av linjelager ovanpå linjer, följt av polygonlager osv.

  - *Visa antal funktioner för nyligen tillagda lager*: visar i panelen *Lager* antalet funktioner bredvid lagrets namn. Eventuellt antal funktioner för klasser visas också. Du kan högerklicka på ett lager för att slå på/av antalet funktioner.
  - *Visa namn på klassificeringsattribut* i panelen Lager, t.ex. när du använder en kategoriserad eller regelbaserad rendering (se *Egenskaper för symbologi* för mer information).
  - den *WMS getLegendGraphic Resolution*
  - *Minimum* och *Maximum förklaring symbol size* för att styra symbolstorleken som visas i panelen *Lager*
- *Fördröjning* i millisekunder för visning av lagers *karttips*
- Om QGIS ska *Respect screen DPI*: Om detta är aktiverat kommer QGIS att försöka visa kanvasen med fysiskt korrekt skala på skärmen, beroende på skärmens fysiska DPI. Symbologi med angiven visningsstorlek kommer också att återges korrekt, t.ex. kommer en 10 mm symbol att visas som 10 mm på skärmen. Etiketternas teckensnittsstorlekar på skärmen kan dock skilja sig från dem i QGIS användargränssnitt eller andra program. Om den här inställningen är avstängd kommer QGIS att använda operativsystemets logiska DPI, som kommer att överensstämma med andra program på systemet. Canvasskalan och symbologistorleken kan dock vara fysiskt felaktiga på skärmen. I synnerhet på skärmar med hög dpi kommer symbologin sannolikt att visas för liten.

  För bästa upplevelse rekommenderas att du aktiverar *Respect screen DPI*, särskilt när du använder flera eller olika bildskärmar och förbereder visuellt högkvalitativa kartor. Om du avaktiverar *Respect screen DPI* genereras utdata som kan vara mer lämpliga för kartläggning som endast är avsedd för användning på skärmen, särskilt när teckenstorlekar ska matcha andra applikationer.

Observera

Rendering i layouter påverkas inte av inställningen *Respect screen DPI*; den respekterar alltid den angivna DPI:n för målets utdataenhet. Observera också att den här inställningen använder den fysiska skärmens DPI som rapporteras av operativsystemet, vilket kanske inte är korrekt för alla skärmar.

5.1.8 Kartverktyg Inställningar

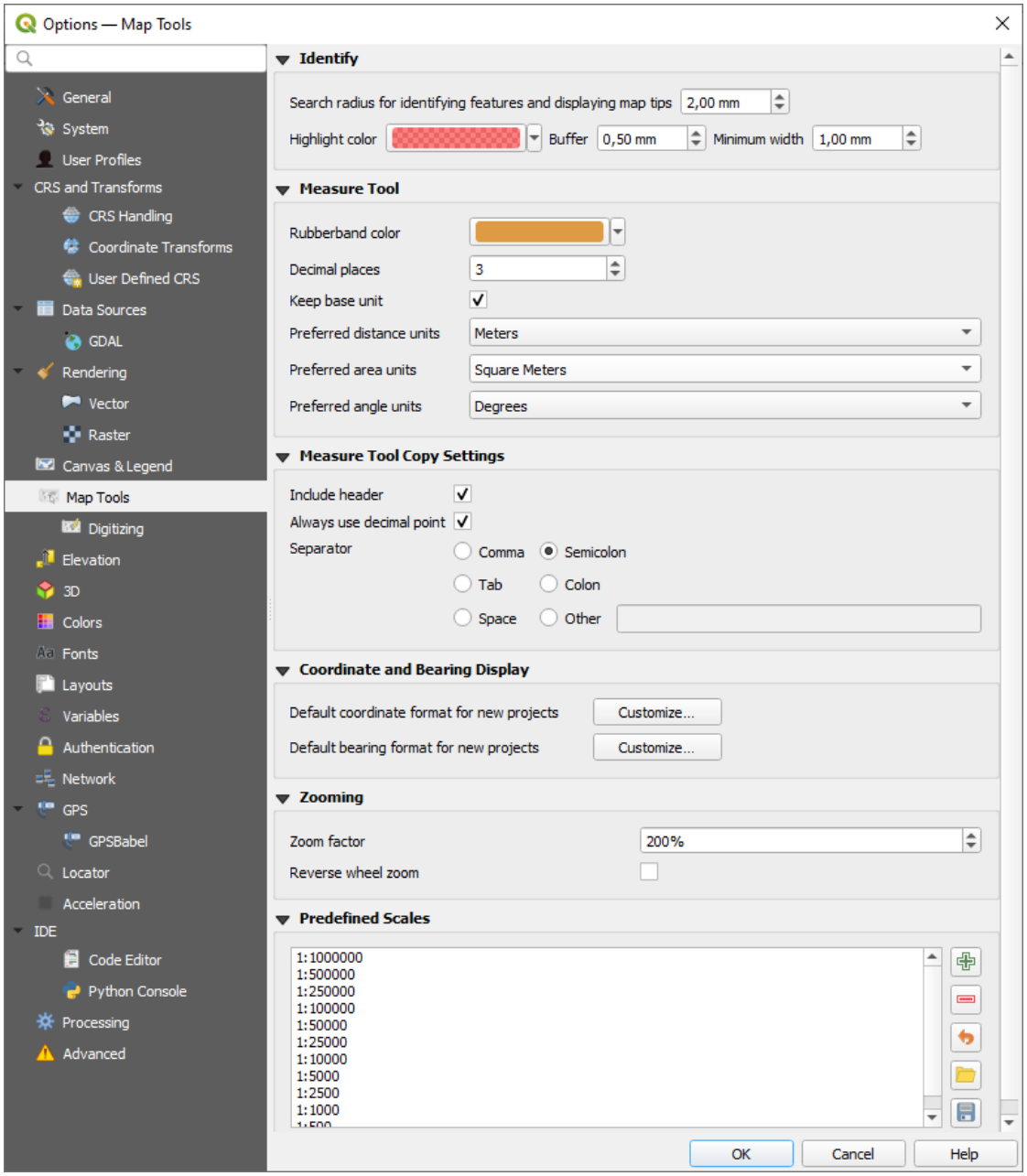


Fig. 5.14: Inställningar för kartverktyg

På den här fliken finns några alternativ för hur *Identifiera verktyg* ska fungera.

- *Sökradie för identifiering av funktioner och visning av karttips* är ett toleransavstånd inom vilket identifieringsverktyget visar resultat så länge du klickar inom toleransen.
- *Highlight color* ger dig möjlighet att välja med vilken färg funktioner som identifieras ska markeras.
- *Buffer* bestämmer ett buffertavstånd som ska återges från konturen av den identifierade markeringen.
- *Minsta bredd* bestämmer hur tjock konturen av ett markerat objekt ska vara.

Mätverktyg

- Definiera *Rubberband color* för mätverktyg

- Definiera *Decimaltecken*
- *Keep base unit* för att inte automatiskt konvertera stora tal (t.ex. meter till kilometer)
- *Önskade avståndsenheter*: alternativen är 'Meter', 'Kilometer', 'Feet', 'Yard', 'Miles', 'Nautical Miles', 'Centimeter', 'Millimeter', 'Inches', 'Degrees' eller 'Map Units'
- *Önskade ytenheter*: alternativen är 'Square meters', 'Square kilometers', 'Square feet', 'Square yards', 'Square miles', 'Hectares', 'Acres', 'Square nautical miles', 'Square centimeters', 'Square millimeters', 'Square inches', 'Square degrees' eller 'Map Units'
- *Önskade vinkelenheter*: alternativen är 'grader', 'radianer', 'gon/gradianer', 'bågminuter', 'bågsekunder', 'varv/varv', milliradianer (SI-definition) eller mil (NATO/militär definition)

Vissa av dessa alternativ kan åsidosättas på *projektnivå*.

Mätverktyg Kopieringsinställningar

Med dessa inställningar kan du styra hur knappen *Copy* fungerar i fönstret Measure Line. Du kan välja att *Include header* för att behålla kolumnnamnen, och du kan också välja *Separator* efter eget val. Du kan också välja *Always use decimal point* för att behålla dina koordinater kopierade till urklipp med en punkt som decimalavgränsare, även om dina språkinställningar i QGIS-alternativen är inställda på att använda ett kommatecken. Om du inte har markerat alternativet *Använd alltid decimalpunkt* kan du i det här fallet inte välja kommatecken som fältavgränsare.

Visning av koordinater och bäring

I detta avsnitt beskrivs olika sätt att *Configure*:

- *Default coordinate format for new projects*, som visas i rutan *Coordinates* i QGIS statusfält och i avsnittet *Derived* i verktyget Identify features's resultat
- *Default bearing format for new projects*, som visas i statusfältet för kartans panoreringsriktning och av verktyget Measure bearing.

Dessa alternativ kan åsidosättas på *projektnivå*.

Panorering och zoomning

- Definiera en *Zoom factor* för zoomverktyg eller hjulmus
- *Reverse wheel zoom* gör att du kan justera mushjulets rullningsriktning.

Fördefinierade skalor

Här hittar du en lista över fördefinierade skalor som ska visas som standard i de skalrelaterade rullgardinsmenyerna, t.ex. statusfältet *Scale*, väljaren för synlighetsskalor eller sekundära inställningar för 2D-kartvyn, … Med knapparna och kan du lägga till eller ta bort dina personliga skalor. Du kan också importera eller exportera skalor från/till en "XML"-fil. Observera att du fortfarande har möjlighet att ta bort dina ändringar och återställa till den fördefinierade listan.

I dialogrutan för projektegenskaper kan du också ställa in en egen lista över skalor, som åsidosätter den globala listan i widgetarna.

Inställningar för digitalisering

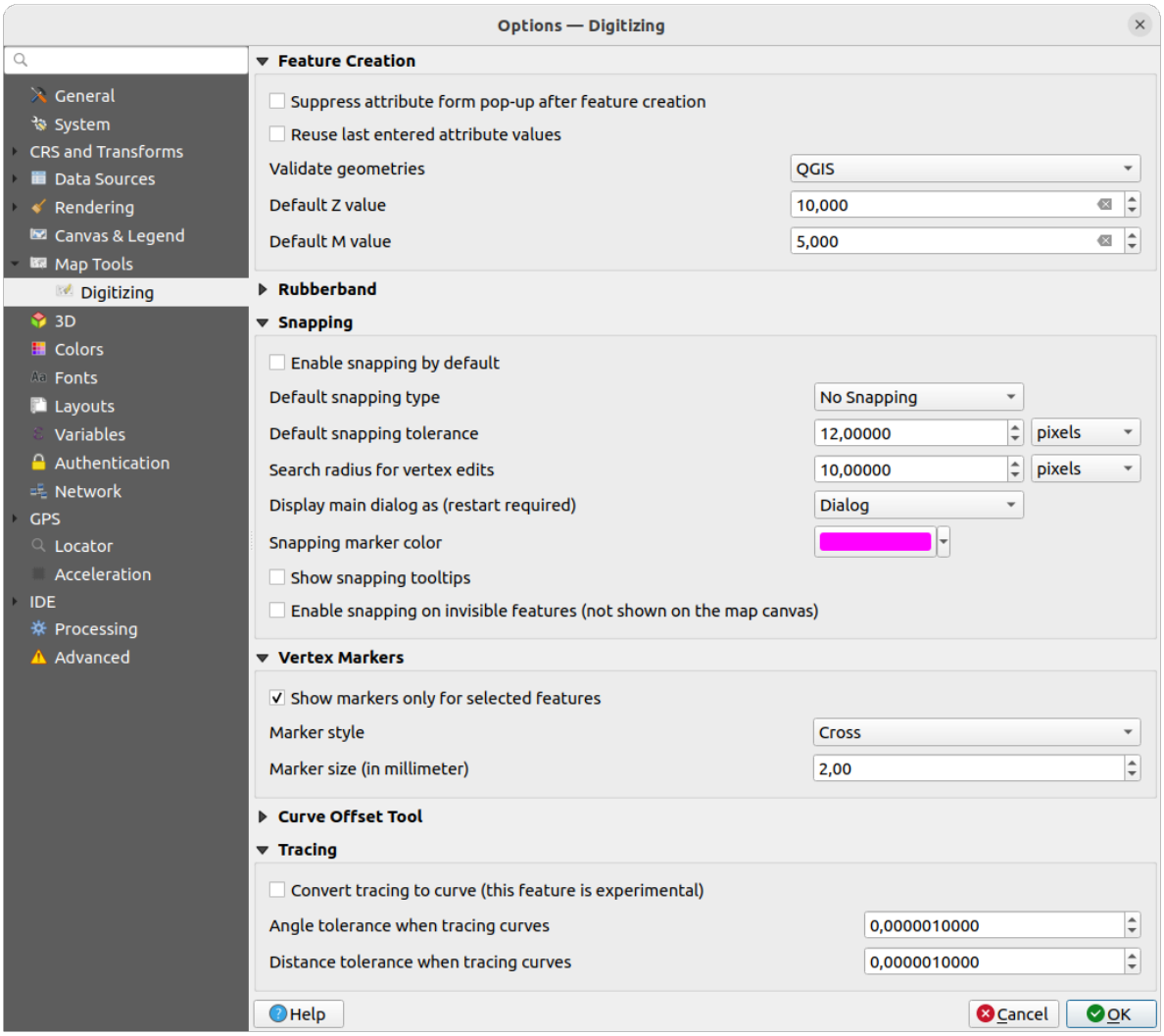


Fig. 5.15: Inställningar för digitalisering

Den här fliken hjälper dig att konfigurera allmänna inställningar när du *redigerar vektorlager* (attribut och geometri).

Skapande av funktioner

- *Suppress attribute form pop-up after feature creation*: detta val kan åsidosättas i varje dialogruta för lageregenskaper.
- *Återanvänd senast angivna attributvärden*: kom ihåg det senast använda värdet för varje attribut och använd det som standard för nästa funktion som digitaliseras. Fungerar per lager. Detta beteende kan också kontrolleras per fält (se *Konfigurera fältets beteende*).
- *Validera geometrier*. Redigering av komplexa linjer och polygoner med många noder kan resultera i mycket långsam rendering. Detta beror på att standardvalideringsprocedurerna i QGIS kan ta mycket tid. För att påskynda renderingen är det möjligt att välja GEOS geometrivalidering (från och med GEOS 3.3) eller att stänga av den. GEOS geometrivalidering är mycket snabbare, men nackdelen är att endast det första geometriproblemet kommer att rapporteras.

  Observera att rapporteringen av geometrifel kan variera beroende på valet (se *Olika typer av felmeddelanden och deras innebörd*)
- *Default Z value* att använda när man skapar nya 3D-funktioner.

Rubberband

- Definiera Rubberband *Line width*, *Line color* och *Fill color*.
- *Aktualisera inte gummiband under vertexredigering*.

Snapping

- *Enable snapping by default* aktiverar snapping när ett projekt öppnas
- Definiera *Default snap mode* (’Vertex’, ’Segment’, ’Centroid’, ’Middle of segments’, ’Line endpoints’, ’Area’)
- Definiera *Default snapping tolerance* i kartenheter eller pixlar
- Definiera *Sökradie för vertexredigeringar* i kartenheter eller pixlar
- *Display main dialog as (restart required)*: anger om dialogrutan för Advanced Snapping ska visas som ”Dialog” eller ”Dock”.
- *Färg på snäppmarkör*
- *Visa verktygstips för snäppning*, t.ex. namnet på det lager vars funktion du håller på att snäppa. Användbart när flera funktioner överlappar varandra.
- *Aktivera snapping på osynliga funktioner (visas inte på kartbilden)*

Vertexmarkörer

- *Visa markörer endast för valda funktioner*
- Definiera vertex *Markeringsstil* (’Kors’ (standard), ’Halvgenomskinlig cirkel’ eller ’Ingen’)
- Definiera toppunkt *Markeringsstorlek (i millimeter)*

Verktyg för kurvförskjutning

De följande alternativen hänvisar till verktyget Offset Curve i *Avancerad digitalisering*. Genom de olika inställningarna är det möjligt att påverka formen på linjeförskjutningen.

- *Join style*: anger om ”runda”, ”miter” eller ”fasade” fogar ska användas vid avfasning av hörn
- *Quadrant segments*: styr antalet linjesegment som ska användas för att approximera en kvartscirkel när man skapar avrundade offsets
- *Miter limit*: Ställer in det maximala avståndet från offsetgeometrin som ska användas när man skapar en geringad fogning som en faktor av offsetavståndet (gäller endast för geringad fogning)
- *End cap style*: styr hur radavslut hanteras; de kan vara ’Runda’, ’Platta’ eller ’Fyrkantiga’.

Spårning

Genom att aktivera kryssrutan *Konvertera spårning till kurva* kan du skapa kurvsegment när du digitaliserar. Tänk på att din dataleverantör måste stödja den här funktionen.

5.1.9 Inställningar för höjd

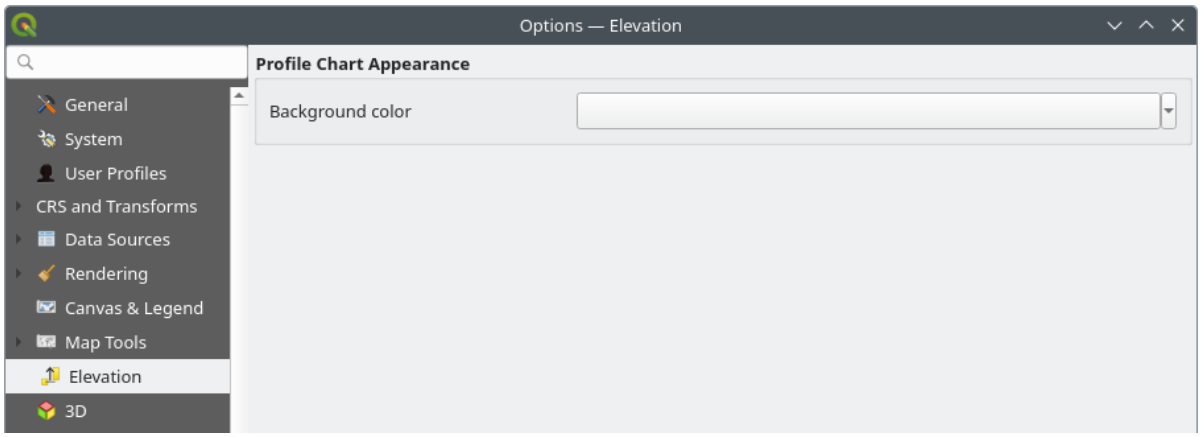


Fig. 5.16: Inställningar för höjd

I menyn *Elevation* kan du ange en specifik färg som ska användas som *Bakgrundsfärg* för *höjdprofiler*. Detta kan göra diagrammet mer läsbart för vissa datamängder, t.ex. punktmoln med RGB-färg, där standardbakgrundsfärgen är för lik punktfärgerna för att vara lätt att urskilja. Om detta inte ändras kommer höjdprofilerna att fortsätta att visas med systemets standardbakgrundsfärg.

5.1.10 3D-inställningar

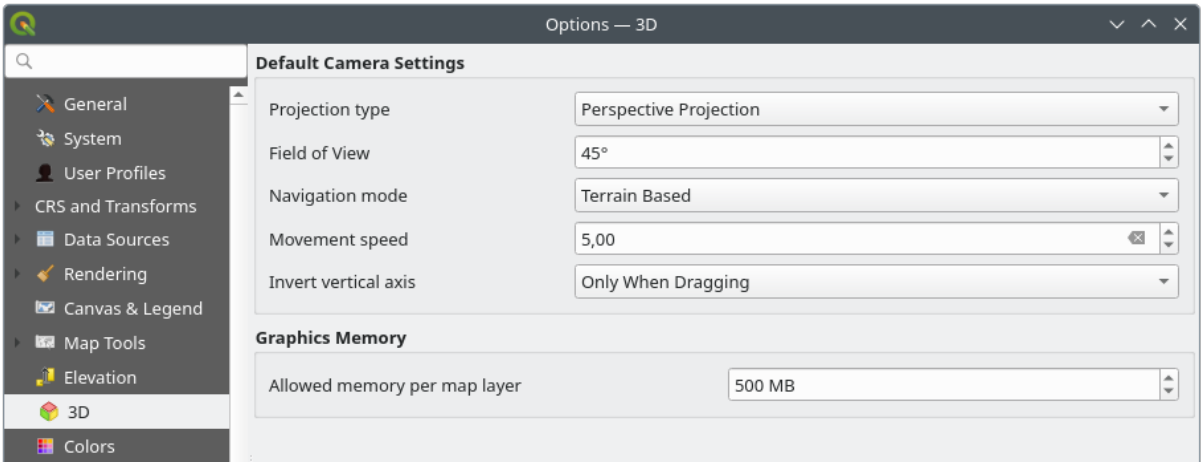


Fig. 5.17: 3D-inställningar

3D-menyn hjälper dig att konfigurera vissa standardinställningar som ska användas för alla *3D Map-vyer*. Dessa kan referera till:

- *Förvalda kamerainställningar*:
  - *Projiceringstyp*: gör det möjligt att visa 3D-scenen i en:
    - *Perspektivisk projektion* (standard): Parallella linjer ser ut att mötas på långt håll. Objekt verkar krympa ju längre bort från kameran de befinner sig.
    - eller en *Orthogonal projektion*: Parallella linjer ser parallella ut. Föremål ser ut att vara lika stora oavsett avstånd.

  - Kamerans *Field of view*: endast relevant vid perspektivprojektion, anger det aktuella vertikala synfältet i grader och avgör hur stor del av scenen som är synlig för kameran. Standardvärdet är 45°.
  - *Navigeringsläge*: ger olika sätt att interagera med 3D-scenen. Tillgängliga lägen är:
    * *Terrängbaserad*: kameran följer runt en fast position på terrängens yta när scenen navigeras.
    * *Gångläge (första person)*

    Beroende på det valda läget skiljer sig *navigationskommandona* åt.
  - *Bevegelsehastighet*
  - *Invertera vertikal axel*: Bestämmer om rörelser med vertikal axel ska inverteras från sitt normala beteende. Påverkar endast rörelser i *Walk mode*. Den kan ställas in på:
    * *Någonsin*
    * *Bara när man drar*: gör att den vertikala rörelsen inverteras endast när man utför en klick-och-drag-kamerarotation
    * och *Always*: gör att rörelserna inverteras både när du klickar och drar och när kameraröreIsen är låst till markören (via en ~-tangenttryckning)

- Under *Graphics memory* kan du med alternativet *Allowed memory per layer* ställa in konfigurationen för GPU-minnesgränsen för varje lager. Detta är användbart för användare som använder stora 3D-scener som tömmer de tillgängliga GPU-minnesresurserna. När en gräns nås visas också en varning, vilket bör hjälpa till vid felsökning av stora scener.

5.1.11 Färginställningar

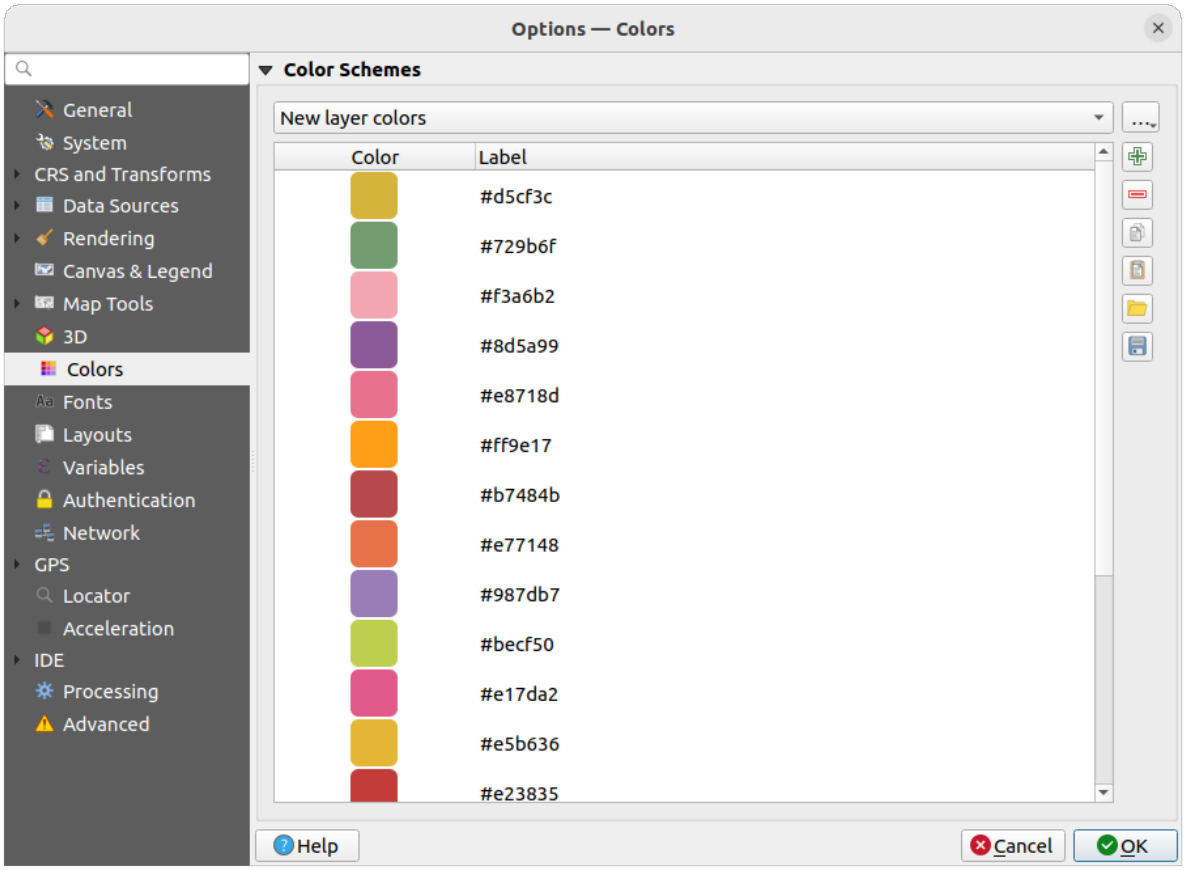


Fig. 5.18: Färginställningar

Med den här menyn kan du skapa eller uppdatera paletter med färger som används i hela programmet i *color selector widget*. Du kan välja mellan:

- *Recent colors* visar nyligen använda färger
- *Standardfärger*, standardpaletten med färger
- *Projektfärger*, en uppsättning färger som är specifika för det aktuella projektet (se *Stilegenskaper* för mer information)
- *New layer colors*, en uppsättning färger som ska användas som standard när nya lager läggs till i QGIS
- eller anpassade paletter som du kan skapa eller importera med hjälp av knappen … bredvid kombinationsrutan för paletter.

Som standard kan paletterna *Recent colors*, *Standard colors* och *Project colors* inte tas bort och är inställda på att visas i rullgardinsmenyn för färgknapparna. Anpassade paletter kan också läggas till i den här widgeten tack vare alternativet *Show in Color Buttons*.

För alla paletter kan du hantera listan med färger med hjälp av verktygen bredvid ramen, t.ex:

- *Lägg till* eller *Ta bort* färg
- *Kopiera* eller *Klistra in* färg
- *Importera* eller *Exportera* uppsättningen färger från/till `.gpl`-filen.

Dubbelklicka på en färg i listan för att justera eller byta ut den i dialogrutan *Color Selector*. Du kan också byta namn på den genom att dubbelklicka i kolumnen *Label*.

5.1.12 Inställningar för teckensnitt

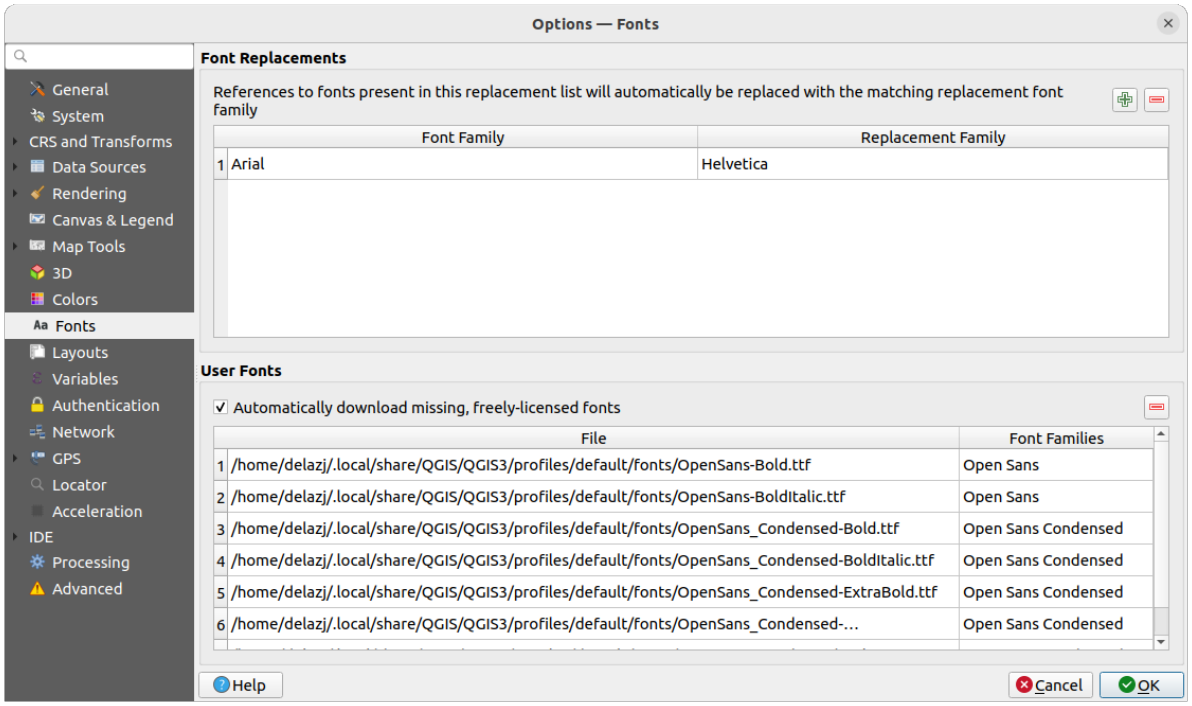


Fig. 5.19: Inställningar för typsnitt

Fliken *Fonts* ger stöd för att hantera teckensnitt som används i olika projekt:

- *Font Replacements*: ger möjlighet att fylla i en lista med automatiska teckensnittsersättningar som ska tillämpas när projekt eller stilar laddas, vilket ger bättre stöd för projekt och stilar som ska användas i olika operativsystem (t.ex. ersätta ”Arial” med ”Helvetica”).
- *User Fonts*: Gör att du kan placera TTF- eller OTF-teckensnitt i undermappen `fonts` i *användarprofilen*. Dessa teckensnitt kan laddas automatiskt när QGIS startas. Detta är ett sätt att använda teckensnitt utan att de måste installeras på operativsystemnivå, vilket ofta är blockerat i företagsmiljöer. Panelen listar alla installerade användartypsnitt och låter dig hantera (d.v.s. ta bort) tidigare installerade användartypsnitt.

  Det är också möjligt att markera kryssrutan *Automatisk nedladdning av saknade, fritt licensierade teckensnitt*: Om du t.ex. öppnar ett projekt eller en stil, eller försöker ladda ett vektorlager som refererar till teckensnitt som inte är tillgängliga för närvarande, konsulteras en hårdkodad lista över fritt licensierade teckensnitt som kan laddas ner via URL för att avgöra om det är möjligt att automatiskt ladda ner teckensnittet till användarens profilkatalog (med meddelande om teckensnittslicensen).

5.1.13 Inställningar för layouter

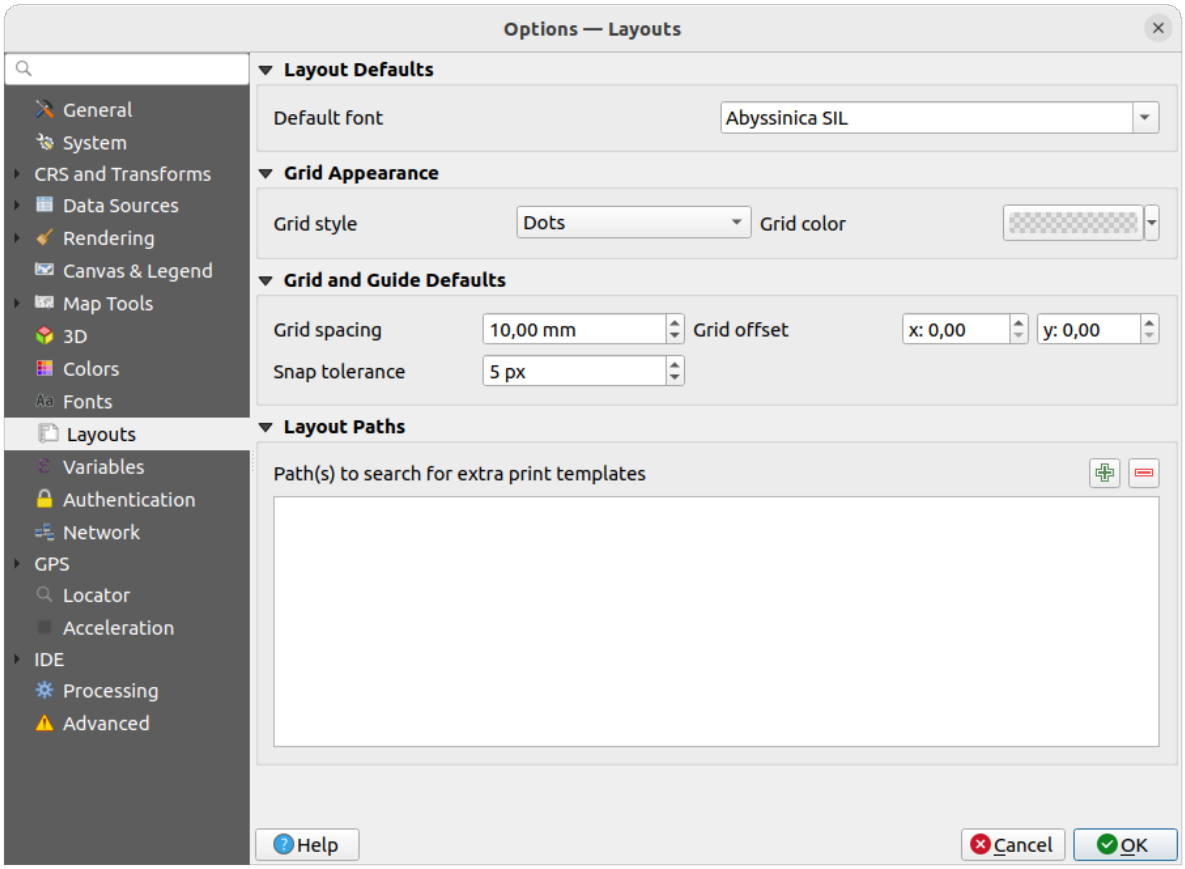


Fig. 5.20: Inställningar för layouter

Standardvärden för sammansättning

Du kan definiera *Default font* som används inom *print layout*.

Grid utseende

- Definiera *Grid style* (’Solid’, ’Prickar’, ’Kors’)
- Definiera *Grid färg*

Standardvärden för rutnät och guider

- Definiera *Grid spacing*
- Definiera *Grid offset* för X och Y
- Definiera *Snap tolerans*

Layoutvägar

- Definiera *Sökväg(ar) efter extra utskriftsmallar*: en lista över mappar med anpassade layoutmallar som ska användas när du skapar en ny.

5.1.14 Inställningar för variabler

Fliken *Variables* innehåller en lista över alla variabler som är tillgängliga på global nivå.

Det gör det också möjligt för användaren att hantera variabler på global nivå. Klicka på knappen för att lägga till en ny anpassad variabel på global nivå. På samma sätt kan du välja en anpassad variabel på global nivå i listan och klicka på knappen för att ta bort den.

Mer information om variabler finns i avsnittet *Lagring av värden i variabler*.

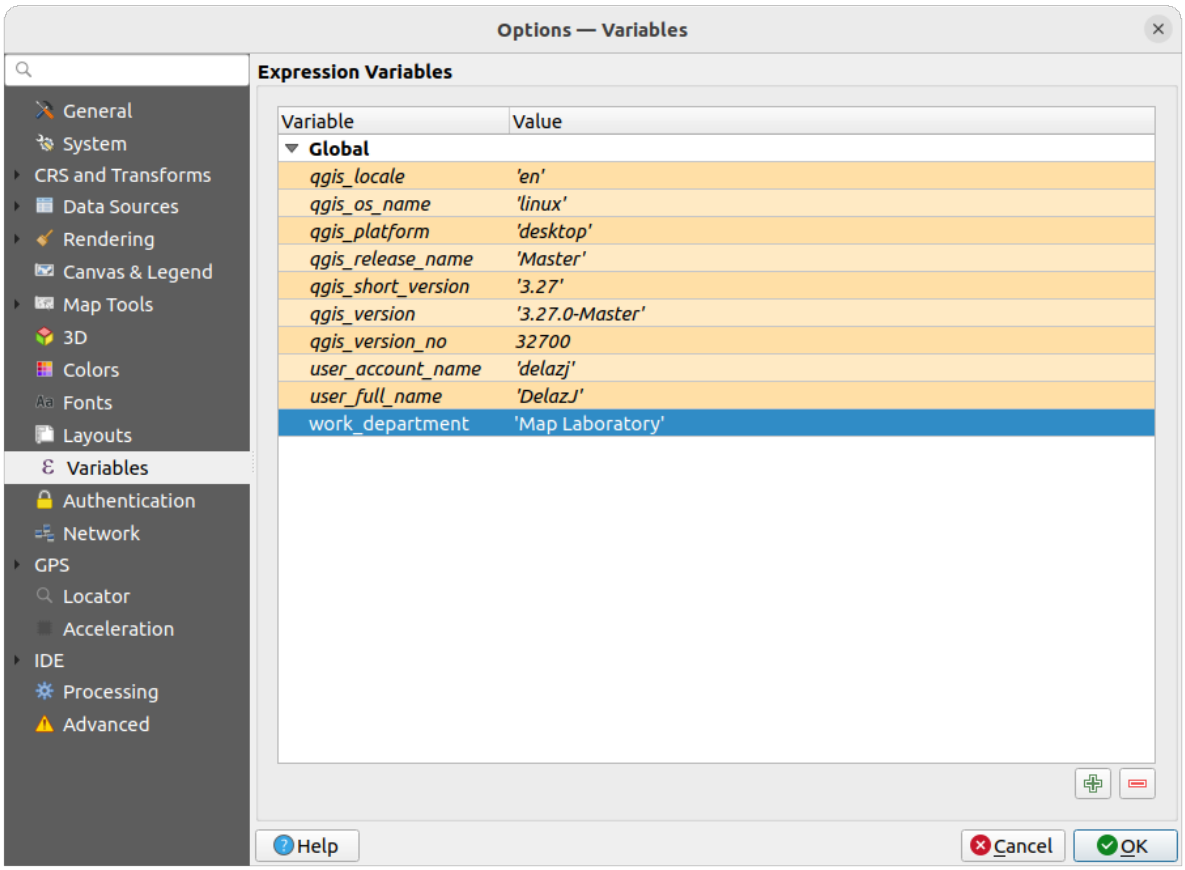


Fig. 5.21: Inställningar för variabler

5.1.15 Inställningar för autentisering

På fliken *Authentication* kan du ställa in autentiseringskonfigurationer och hantera PKI-certifikat. Se *System för autentisering* för mer information.

För att hantera autentiseringar kan du använda listan med verktyg bredvid ramen, t.ex:

- Lägg till ny autentiseringskonfiguration
- Ta bort vald autentiseringskonfiguration
- Ändra vald konfiguration för autentisering

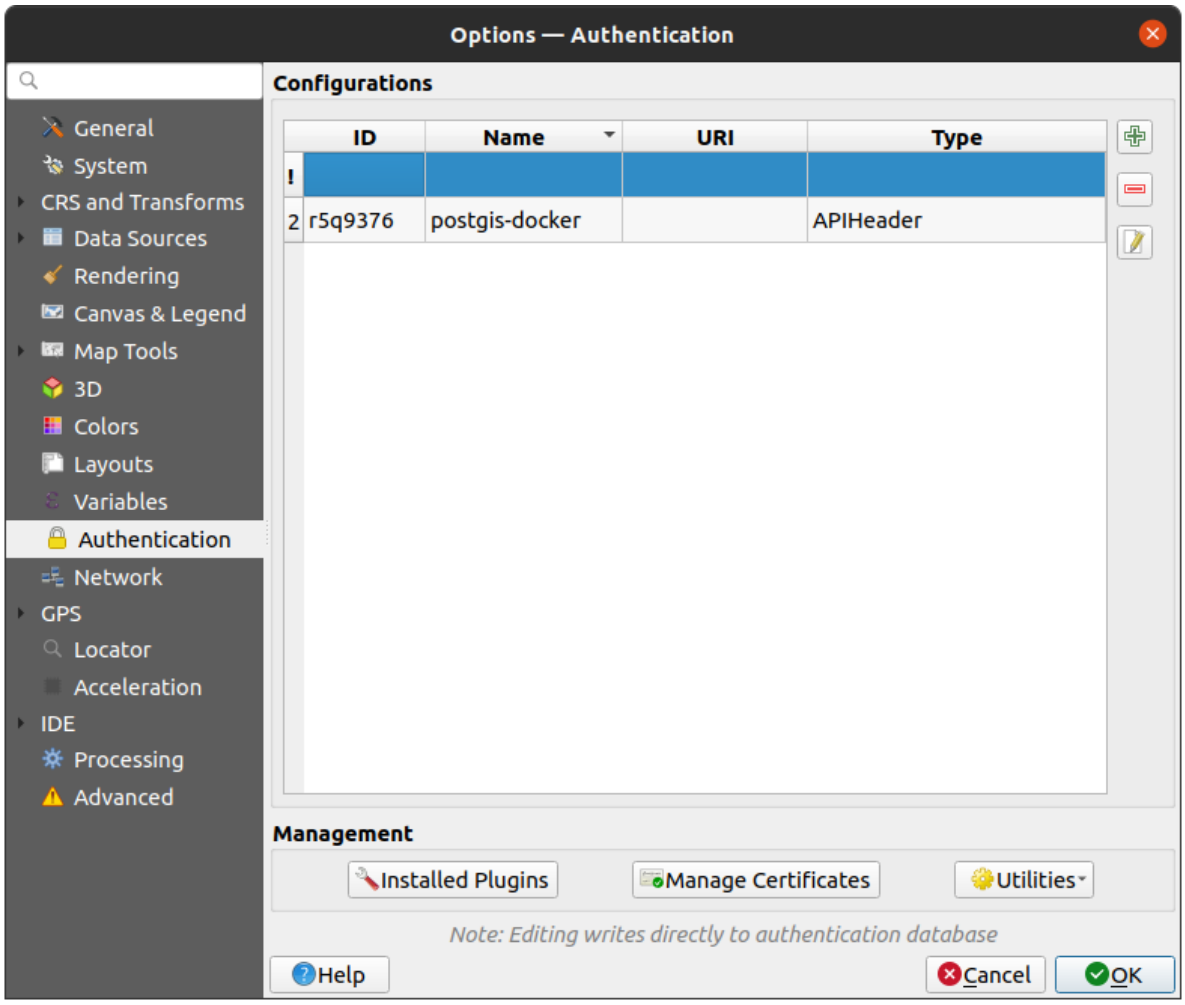


Fig. 5.22: Inställningar för autentisering

5.1.16 Nätverksinställningar

Generellt

- Definiera *Timeout för nätverksförfrågningar (ms)* - standard är 60000
- Definiera *Förvalt utgångsdatum för WMS-kapacitet (timmar)* - standard är 24
- Definiera *Förvalt utgångsdatum för WMS-C/WMTS-brickor (timmar)* - standard är 24
- Definiera *Max retry i händelse av fel på tile eller feature request*
- Definiera *User-Agent prefix* som standard är `Mozilla/5.0`. Detta värde kommer att läggas till både QGIS och OS-versionen, t.ex. `Mozilla/5.0 QGIS/33801/Ubuntu 22.04.4 LTS` för att forma användaragenten.

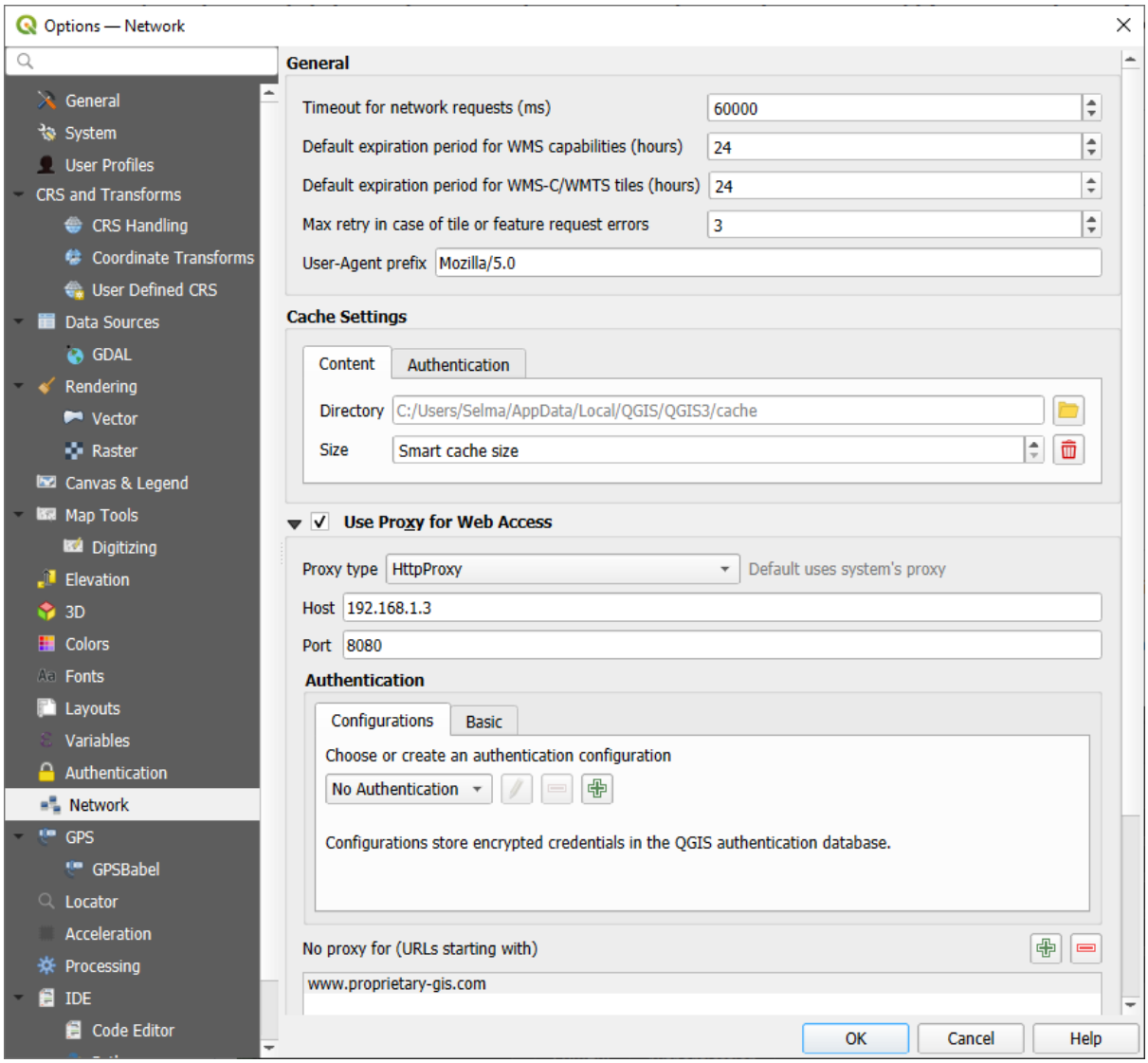


Fig. 5.23: Nätverks- och proxyinställningar

Cache-inställningar

Definierar *Directory* och en *Size* för cachen som anges i megabyte. Du kan rensa värdet för att aktivera *Smart cache size*, som anger den maximala cachestorleken baserat på tillgängligt diskutrymme. Erbjuder även verktyg för att *automatiskt rensa cacheminnet för anslutningsautentisering vid SSL-fel (rekommenderas)*.

Proxy för webbåtkomst

- *Använd proxy för webbåtkomst*
- Ställ in *Proxy type* efter dina behov och definiera 'Host' och 'Port'. Tillgängliga proxytyper är:
  - *Default Proxy*: Proxy bestäms baserat på systemets proxy
  - *Socks5Proxy*: Generisk proxy för alla typer av anslutningar. Stöder TCP, UDP, bindning till en port (inkommande anslutningar) och autentisering.
  - *HttpProxy*: Implementeras med kommandot "CONNECT", stöder endast utgående TCP-anslutningar; stöder autentisering.
  - *HttpCachingProxy*: Implementeras med hjälp av vanliga HTTP-kommandon och är endast användbar i samband med HTTP-förfrågningar.
  - *FtpCachingProxy*: Implementeras med hjälp av en FTP-proxy och är endast användbar i samband med FTP-begäranden.

Legitimationsuppgifter för proxy anges med hjälp av *authentication widget*.

Att utesluta vissa webbadresser kan läggas till i textrutan under proxyinställningarna (se Fig. 5.23). Ingen proxy kommer att användas om måladressen börjar med en av strängarna som anges i textrutan.

Om du behöver mer detaljerad information om de olika proxyinställningarna kan du läsa manualen för det underliggande QT-bibliotekets dokumentation på https://doc.qt.io/archives/qt-5.9/qnetworkproxy.html#ProxyType-enum

Tips

Använda proxyservrar

Att använda proxyservrar kan ibland vara knepigt. Det är bra att prova sig fram med ovanstående proxytyper för att kontrollera om de fungerar i ditt fall.

5.1.17 GPS-inställningar

Dialogrutan *GPS* hjälper dig att konfigurera anslutningar och egenskaper för GPS-enheter i QGIS. Den ger också inställningar för GPS-spårning och digitalisering av data.

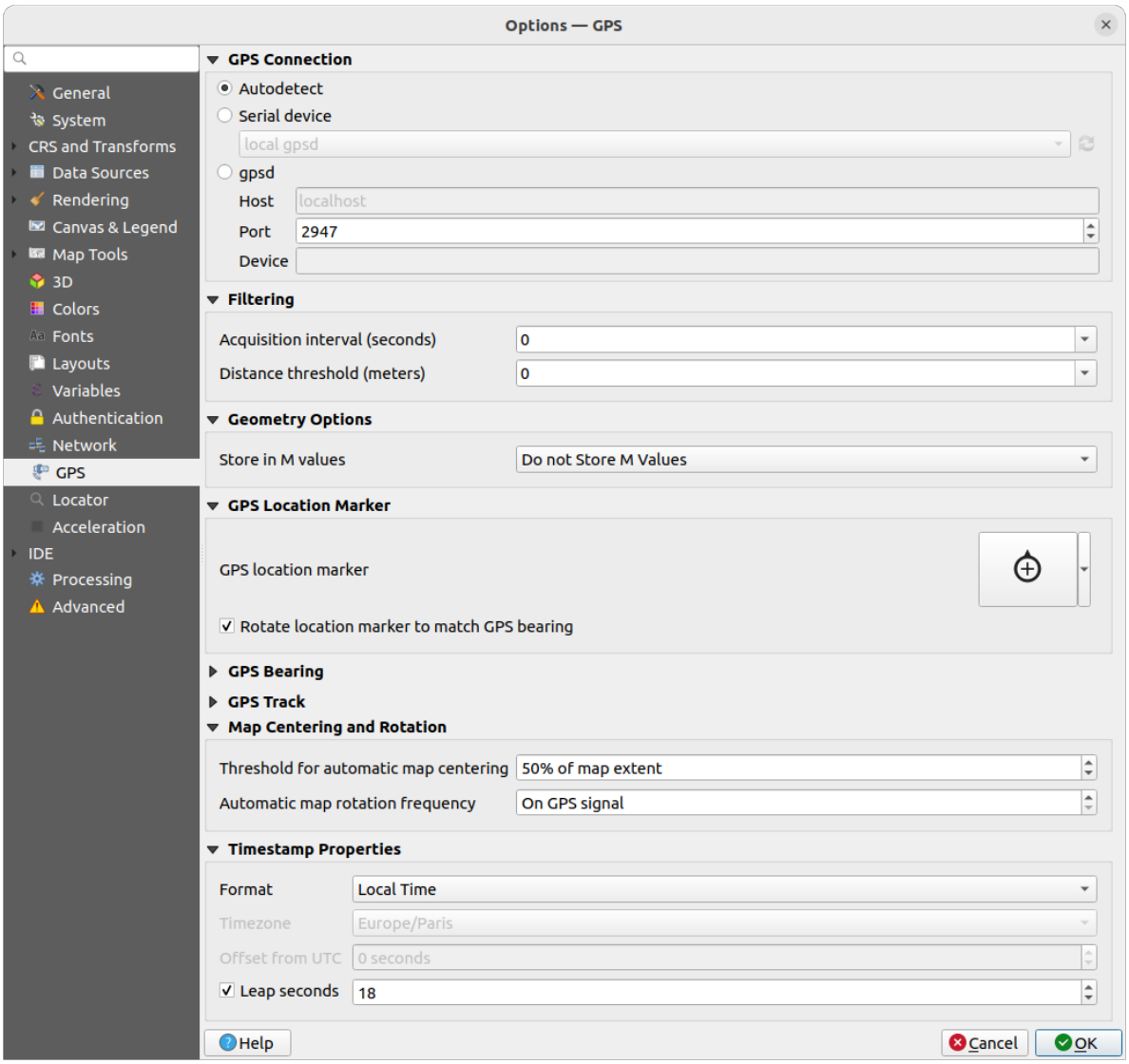


Fig. 5.24: GPS-inställningar

Du kan ange:

- *GPS Connection*: tillhandahåller olika sätt att ansluta enheten till QGIS
  - *Autodetect*
  - *Seriell enhet* (omladdning krävs om en ny GPS-enhet ansluts)
  - *gpsd* (väljer *Host* och *Port* som din GPS är ansluten till och anger ett namn på *Device*)

  Om det uppstår problem med anslutningen kan du försöka växla från ett alternativ till ett annat.
- *Filtrering*: Du kan ställa in en *Acquisition interval (seconds)* och/eller en *Distance threshold (meters)* parametrar för att hålla markören fortfarande aktiv när mottagaren är i statiska förhållanden.
- Under *Geometry Options* kan man med alternativet *Store in M values* skapa geometrier med M-värden från de inbyggda GPS-loggningsverktygen. Detta gäller både funktioner som digitaliserats från GPS-loggar och från *Logga till Geopackage/SpatiaLite*-funktionaliteten… Alternativen omfattar lagring av tidsstämplar (som ms sedan epok), markhastighet, höjder, bäringar och noggrannhetskomponenter som m-värden.
- *GPS platsmarkör*
  - Konfigurera en symbol för *GPS platsmarkör*, som anger aktuell GPS-position

  - *Rotera platsmarkören för att matcha GPS-lagringen*: om markörsymbolen ska roteras för att matcha GPS-riktningen

- *GPS Bäring*:
  - konfigurera en *Bearing line style* med hjälp av QGIS egenskaper för linjesymboler
  - anger om *Beräkna bäring från färdriktning* ska vara avmarkerad: Om markerad ignoreras den bäring som rapporteras av GPS-enheten och bäringen beräknas istället med hjälp av vinkeln mellan de två föregående GPS-positionerna.
- *GPS Track*: ange symbol som ska användas för *Track line style*
- *Map Centering and Rotation*: definierar när kartbilden uppdateras i enlighet med användarens förflyttning på fältet:
  - *Tröskelvärde för automatisk kartcentrering*: definierar den minsta förskjutningen av GPS-positionen från kartans mitt för att utlösa en automatisk *kartcentrering* om den är aktiverad.
  - *Automatisk kartrotationsfrekvens*: definierar hur ofta kartrotationen för att matcha GPS-pejlingen kan ske; det kan vara på en anpassad varaktighet eller *På GPS-signal*.
- *Timestamp Properties* för att konfigurera hur tidsvärden visas och lagras i data. Parametrar inkluderar *Format* som kan vara:
  - *Lokal tid*
  - *UTC*`
  - *UTC med förskjutning*, för att ta hänsyn till sommartidsförskjutningar eller andra komplexa tidszonsfrågor
  - en specifik *Tidszon*

  Dessutom kan *Leap seconds*-korrektion tillämpas genom att lägga till sekunderna till GPS-tidsstämpeln.

GPSBabel

GPSBabel konverterar waypoints, spår och rutter mellan populära GPS-mottagare som Garmin eller Magellan och kartprogram som Google Earth eller Basecamp. Bokstavligen talat stöds hundratals GPS-mottagare och program. QGIS förlitar sig på GPSBabel för att interagera med dessa enheter och *manipulera deras data*.

För mer information om hur du gör, se *Laddning till eller från en enhet*.

5.1.18 Inställningar för sökare

På fliken *Locator* kan du konfigurera *Locator bar*, en widget för snabbsökning som finns i statusfältet och som hjälper dig att göra sökningar i programmet. Den innehåller några standardfilter (med prefix) som kan användas:

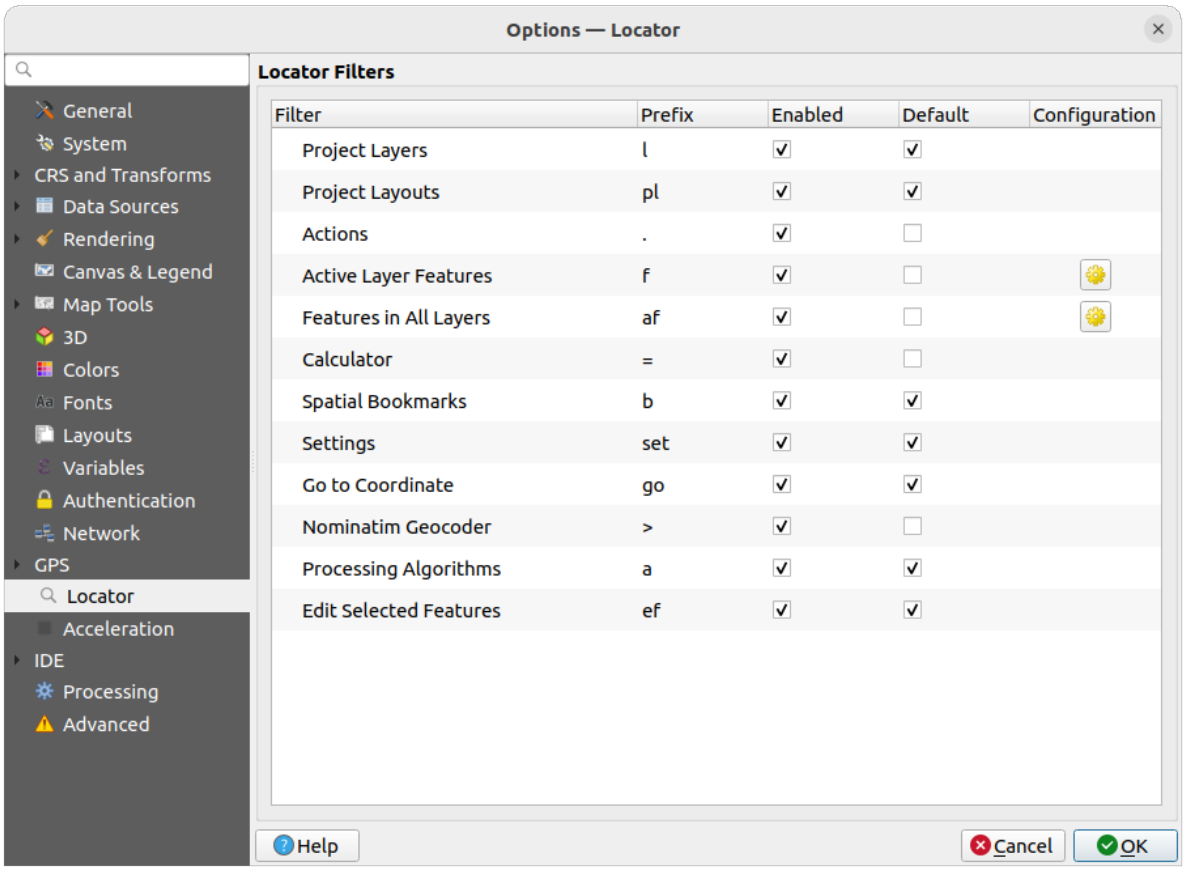


Fig. 5.25: Inställningar för sökare

- *Project Layers* (`l`): söker efter och väljer ett lager i panelen *Layers*.
- *Project Layouts* (`pl`): hittar och öppnar en utskriftslayout.
- Åtgärder (`.`): hittar och utför en QGIS-åtgärd; åtgärder kan vara vilket verktyg eller meny som helst i QGIS, öppna en panel…
- *Active Layer Features* (`f`): söker efter matchande attribut i valfritt fält från det aktuella aktiva lagret och zoomar till den valda funktionen. Tryck på för att konfigurera det maximala antalet resultat.
- *Features in All Layers* (`af`): söker efter matchande attribut i *display name* för varje *searchable layers* och zoomar till den valda funktionen. Tryck på för att konfigurera det maximala antalet resultat och det maximala antalet resultat per lager.
- *Calculator* (`=`): gör det möjligt att utvärdera valfritt QGIS-uttryck och, om det är giltigt, ge möjlighet att kopiera resultatet till urklippet.
- *Spatial Bookmarks* (`b`): hittar och zoomar till bokmärkets utsträckning.
- *Inställningar* (`set`): bläddrar i och öppnar dialogrutor med egenskaper för projekt och program.
- *Go to Coordinate* (`go`): panorerar kartbilden till en plats som definieras av ett komma- eller mellanslagsseparerat par av x- och y-koordinater eller en formaterad URL (t.ex. OpenStreetMap, Leaflet, OpenLayer, Google Maps, …). Koordinaten förväntas vara i WGS 84 (`epsg:4326`) och/eller CRS för kartbilden.
- *Nominatim Geocoder* (`>`): geokodar med hjälp av geokodningstjänsten Nominatim från OpenStreetMap Foundation.
- Bearbetningsalgoritmer (`a`): söker och öppnar en dialogruta för bearbetningsalgoritmer.
- *Edit Selected Features* (`ef`): ger snabb åtkomst och kör en kompatibel *modify-in-place* bearbetningsalgoritm på det aktiva lagret.

I dialogrutan kan du göra följande:

- anpassa filtret *Prefix*, dvs. det nyckelord som ska användas för att utlösa filtret
- ange om filtret är *Enabled*: filtret kan användas i sökningarna och en genväg finns tillgänglig i menyn i sökfältet
- ange om filtret är *Default*: en sökning som inte använder ett filter returnerar resultat från endast standardfilterkategorierna.
- Vissa filter ger möjlighet att konfigurera antalet resultat i en sökning.

Uppsättningen av standardfilter för lokalisering kan utökas med tillägg, t.ex. för OSM nominatim-sökningar, direkt databassökning, lagerkatalogsökningar, …

5.1.19 Inställningar för acceleration

Inställningar för OpenCL-acceleration.

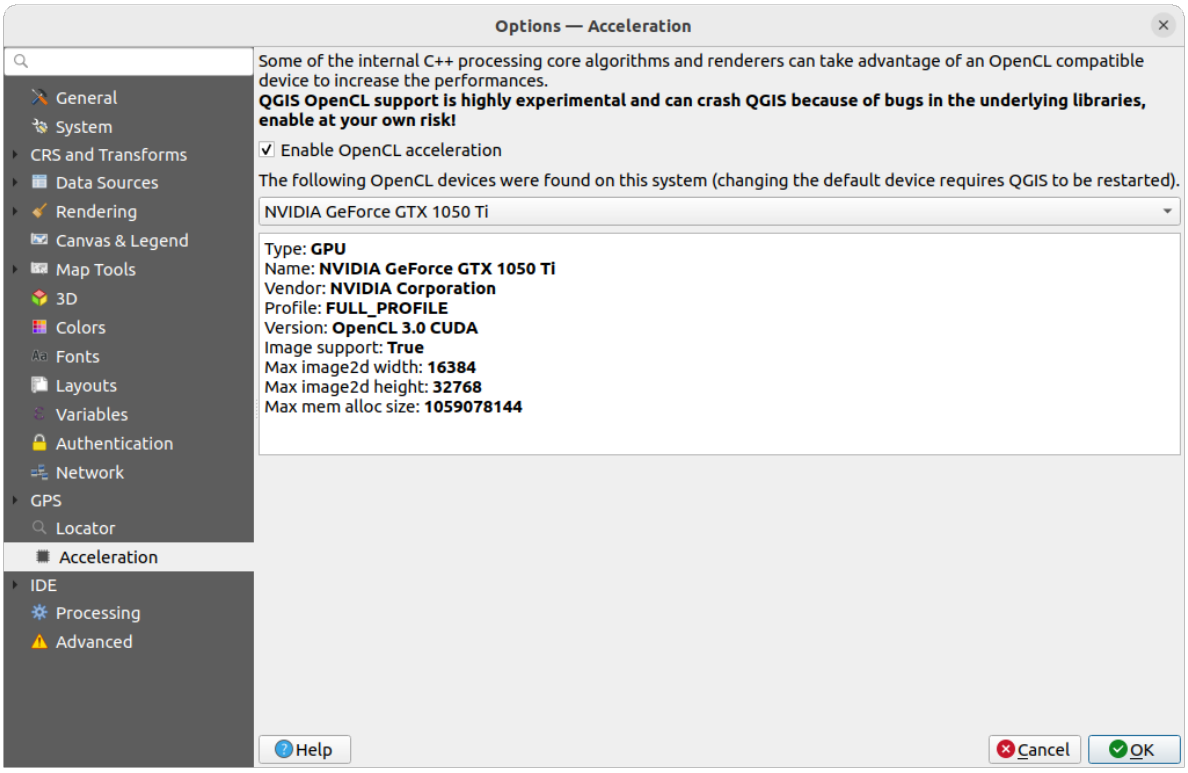


Fig. 5.26: Inställningar för acceleration

Beroende på din maskinvara och programvara kan du behöva installera ytterligare bibliotek för att aktivera OpenCL-acceleration.

5.1.20 IDE-inställningar

Under *GitHub access token* kan du generera en personlig token som gör att du kan dela kodsnuttar från Python-kodredigeraren. Mer information om GitHub-autentisering

Inställningar för kodredigeraren

På fliken *Code Editor* kan du styra utseendet och beteendet för kodredigeringswidgets (Pythons interaktiva konsol och redigerare, uttryckswidget och funktionsredigerare, …).

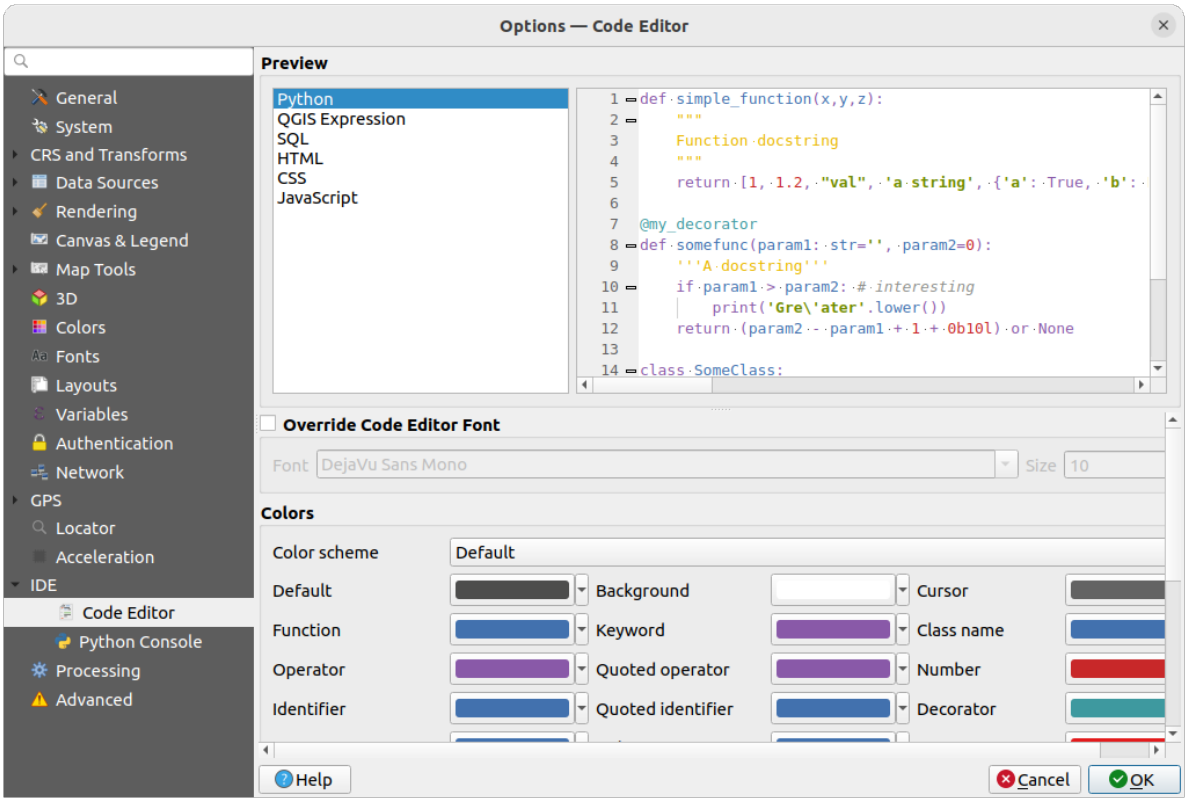


Fig. 5.27: Inställningar för kodredigeraren

Längst upp i dialogrutan finns en widget som ger en förhandsgranskning av de aktuella inställningarna i olika kodningsspråk (Python, QGIS expression, HTML, SQL, JavaScript). Ett bekvämt sätt att justera inställningarna.

- Markera *Overriva teckensnitt för kodredigerare* för att ändra standard *Teckensnittsfamilj* och *Storlek*.
- Under gruppen *Colors* kan du:
  - välj ett *Färgschema*: fördefinierade inställningar är `Default`, `Solarized Dark` och `Solarized Light`. Ett `Custom` schema aktiveras så snart du ändrar en färg och kan återställas genom att välja ett fördefinierat schema.
  - ändra *färg* för varje element i kodskrivning, t.ex. färgerna som ska användas för kommentarer, citat, funktioner, bakgrund, …

Python-inställningar

Inställningarna för *Python* hjälper dig att hantera och styra Python-redigerarnas beteende (*interactive console*, *code editor*, *project macros*, *custom expressions*, …). Den kan också nås med hjälp av knappen Options… från:

- verktygsfältet *Python-konsol*
- den kontextuella menyn för widgeten *Python-konsol*
- och den kontextuella menyn i kodredigeraren.

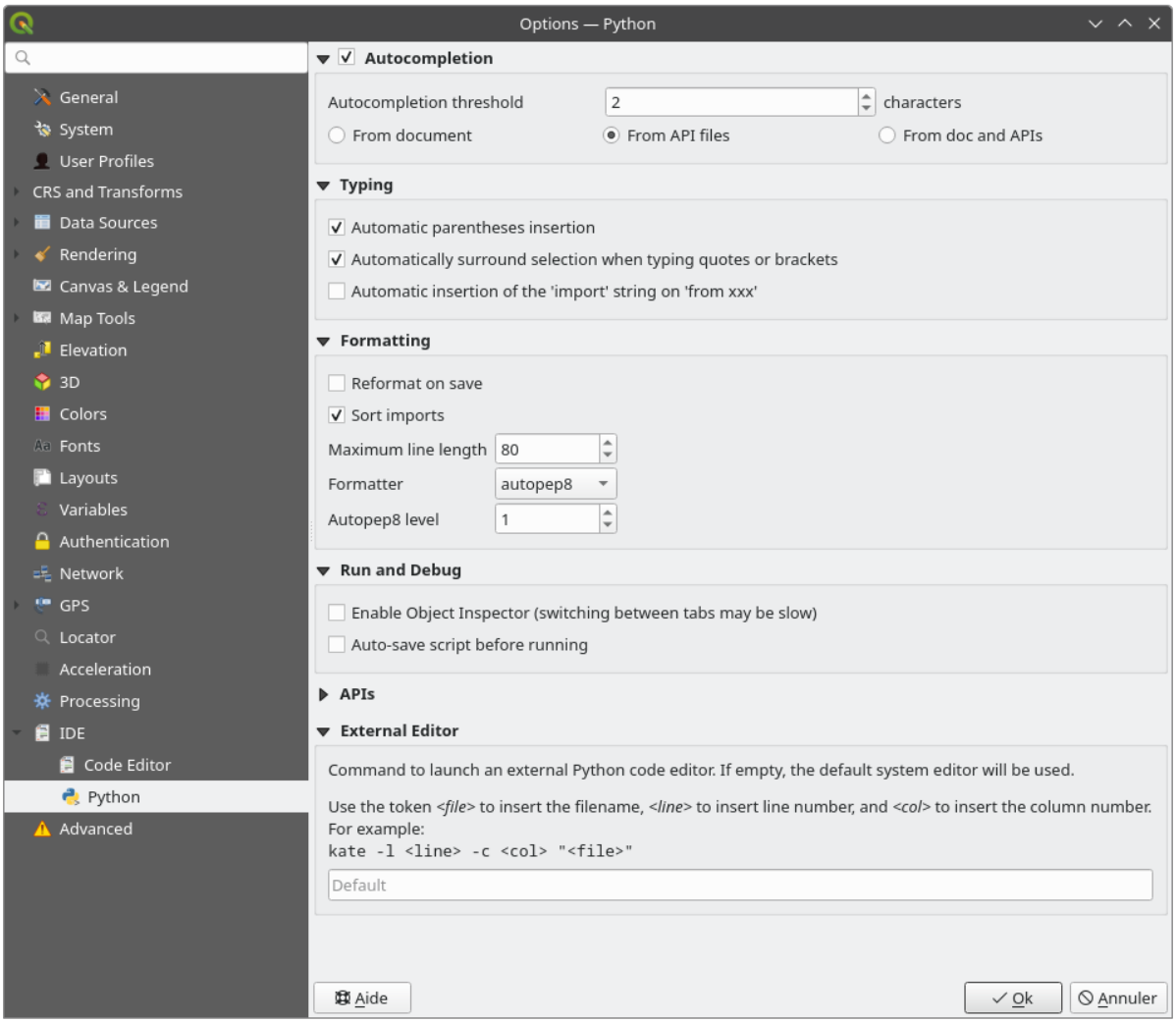


Fig. 5.28: Python-inställningar

Du kan ange:

- *Autokomplettering*: Aktiverar kodkomplettering. Du kan få autokomplettering från det aktuella dokumentet, de installerade API-filerna eller båda.
  - *Tröskel för autokomplettering*: Ställer in tröskeln för visning av autokompletteringslistan (i tecken)
- under *Typning*
  - *Automatisk infogning av parenteser*: Om ingen text är markerad och ett öppningstecken (parentes, citattecken, hakparentes, …) skrivs in, infogas det matchande slutteckenet precis efter markören. Observera att det här beteendet är inaktiverat om markören befinner sig i en sträng eller kommentar.

  - *Automatiskt omge markeringen när du skriver in citattecken eller parenteser*: När ett inledande tecken skrivs in omsluts den markerade texten av det inledande/avslutande paret. Markeringen förblir densamma, så det är möjligt att citera ett markerat ord och omsluta det med parenteser bara genom att skriva `"` sedan `(`.

    Specialfall för flerradigt urval med citattecken och dubbla citattecken: urvalet omges av trippla enkla/dubbla citattecken.

  - *Automatiskt införande av strängen 'import' vid 'from xxx'*: Aktiverar infogning av 'import' när import anges

- under *Formatting* kan du lägga till automatiserade verktyg för att formatera om den kod du skriver:
  - *Reformat on save*: formatering tillämpas precis innan skriptet sparas
  - *Sort imports*: sorterar 'import'-uttalanden med hjälp av biblioteket isort
  - *Maximal radlängd*: styr hur formateraren ska slå om raderna och styr redigeringslinjalen
  - *Formatter* - verktyg som stöds är *autopep8* och *black*, med dedikerade alternativ:
    - *Autopep8-nivå* - mer information finns på autopep8 avancerad användning
    - *Normalize quotes*: ersätter alla enkla citattecken med dubbla citattecken om möjligt
- under *Kör och felsök*
  - *Aktivera Object Inspector (det kan gå långsamt att växla mellan flikar)*
  - *Auto-sparar skript innan det körs*: Sparar skriptet automatiskt när det körs. Denna åtgärd lagrar en temporär fil (i den temporära systemkatalogen) som raderas automatiskt efter körning.

För *APIs* kan du ange:

- *Använder förinstallerade API-filer*: Du kan välja om du vill använda de förinstallerade API-filerna. Om detta inte är markerat kan du lägga till API-filer och du kan också välja om du vill använda förberedda API-filer (se nästa alternativ).
- *Använder förberedd API-fil*: Om markerad kommer den valda filen `*.pap` att användas för kodkomplettering. För att generera en förberedd API-fil måste du ladda minst en `*.api`-fil och sedan kompilera den genom att klicka på knappen *Compile APIs…*.

I gruppen *External Editor* kan du ange kommandoradsinstruktioner för att starta en extern Python-kodredigerare, givet ett filnamn, en rad och ett kolumnnummer. Om den är tom kommer systemets standardredigerare att användas.

5.1.21 Inställningar för bearbetning

Fliken *Processing* ger dig allmänna inställningar för verktyg och dataleverantörer som används i QGIS Processing-ramverket. Mer information finns på *Konfigurera ramverket för bearbetning*.

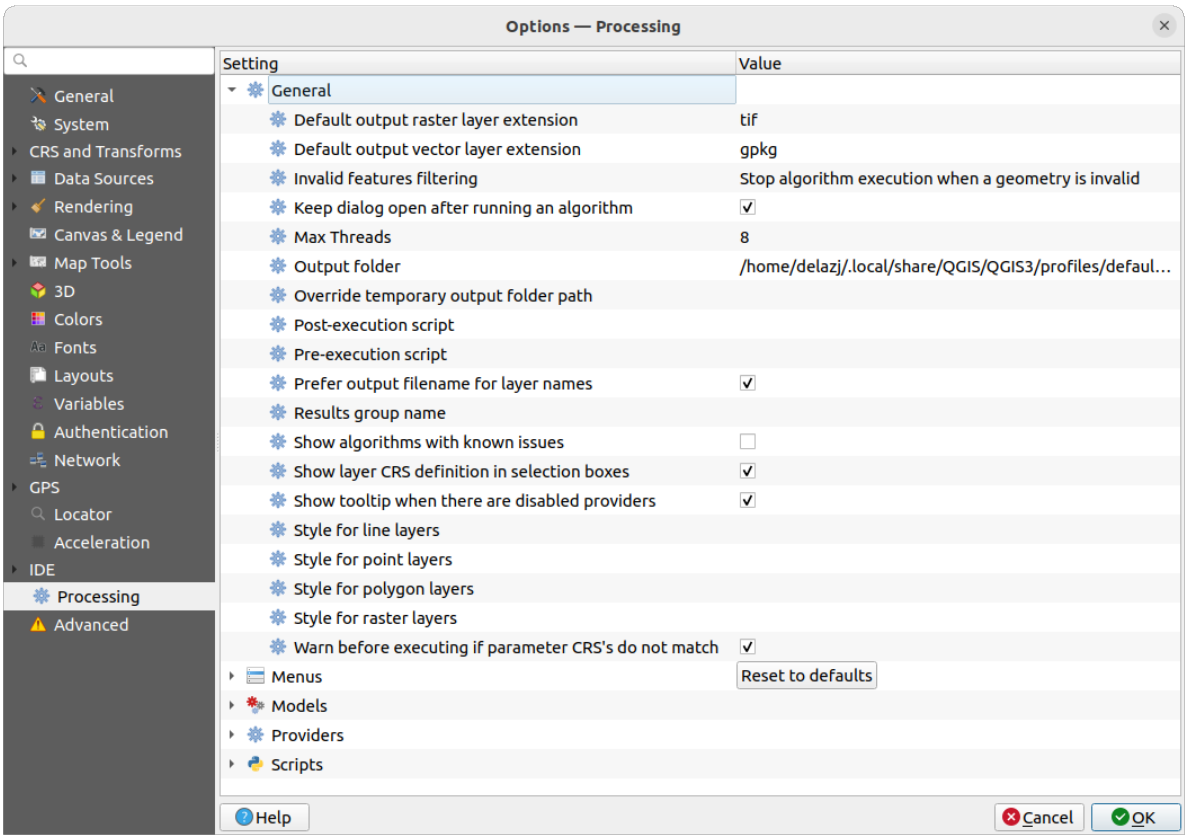


Fig. 5.29: Inställningar för bearbetning

5.1.22 Avancerade inställningar

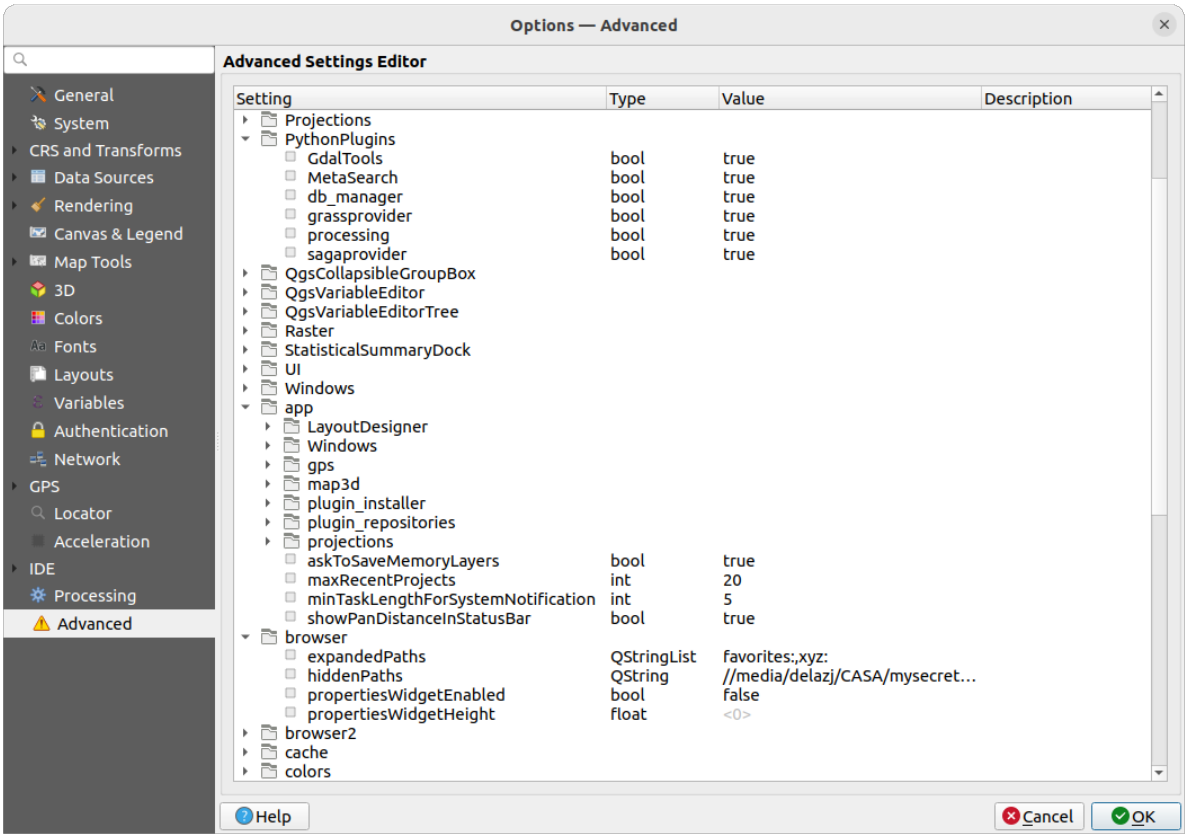


Fig. 5.30: Avancerade inställningar

Alla inställningar relaterade till QGIS (användargränssnitt, verktyg, dataleverantörer, bearbetningskonfigurationer, standardvärden och sökvägar, tillägg-alternativ, uttryck, geometrikontroller …) sparas i en `QGIS/QGIS3.ini`-fil under den aktiva *användarprofilen*-katalogen. Konfigurationer kan delas genom att kopiera den här filen till andra installationer.

I QGIS erbjuder fliken *Advanced* ett sätt att hantera dessa inställningar genom *Advanced Settings Editor*. När du har lovat att vara försiktig fylls widgeten med ett träd med alla befintliga inställningar, och du kan redigera deras värde. Högerklicka på en inställning eller en grupp så kan du ta bort den (för att lägga till en inställning eller grupp måste du redigera filen `QGIS3.ini`). Ändringar sparas automatiskt i filen `QGIS3.ini`.

Varning

Undvik att använda fliken Avancerade inställningar i blindo

Var försiktig när du ändrar objekt i den här dialogen eftersom ändringar tillämpas automatiskt. Om du gör ändringar utan att veta om det kan din QGIS-installation gå sönder på olika sätt.

5.2 Arbeta med användarprofiler

5.2.1 Konceptet

I menyn :menuselection:`Settings --> Användarprofiler` finns funktioner för att ställa in och komma åt användarprofiler. En användarprofil är en enhetlig programkonfiguration som kan lagras i en enda mapp:

- alla *globala inställningar*, inklusive locale, projektioner, autentiseringsinställningar, färgpaletter, genvägar …
- Grafiska gränssnittskonfigurationer och *anpassning*
- gridfiler och andra proj-hjälpfiler installerade för datumtransformation
- installerade *tillägg* och deras konfigurationer
- projektmallar och historik över sparade projekt med förhandsgranskning av bilder
- *bearbetningsinställningar*, loggar, skript, modeller.

Som standard innehåller en QGIS-installation en enda användarprofil med namnet `default`. Men du kan skapa så många användarprofiler du vill:

1. Klicka på *Inställningar* ► *Användarprofiler* ► *Ny profil…*.
2. Du kommer att uppmanas att ange ett profilnamn, vilket skapar en mapp med samma namn under `~/<UserProfiles>/` where:
   - `~` representerar katalogen **HOME**, som på Windows vanligtvis är något i stil med `C:\Users\<username>`.
   - och `<UserProfiles>` representerar den huvudsakliga profilmappen, dvs:
     - `.local/share/QGIS/QGIS3/profiles/`
     - `%AppData%\Roaming\QGIS\QGIS3\profiles\`
     - `Library/Application Support/QGIS/QGIS3/profiles/`

   Användarprofilmappen kan öppnas från QGIS med hjälp av *Open Active Profile Folder*.
3. En ny instans av QGIS startas med en ren konfiguration. Du kan sedan ställa in dina egna konfigurationer.

Om du har mer än en profil i din QGIS-installation visas namnet på den aktiva profilen mellan hakparenteser i programmets titelfält.

Eftersom varje användarprofil innehåller isolerade inställningar, tillägg och historik kan de vara bra för olika arbetsflöden, demos, användare av samma maskin eller testinställningar etc. Och du kan växla från den ena till den andra genom att välja dem i menyn :menuselection:` *Settings –> Användarprofiler*. Du kan också köra QGIS med en specifik användarprofil från *kommandoraden*.

Tips

Kör QGIS under en ny användarprofil för att kontrollera om fel kvarstår

Buggar som du kan stöta på med funktioner i QGIS kan vara relaterade till rester i den aktuella användarprofilen. Att köra QGIS under en annan användarprofil kan hjälpa dig att lösa dem eller kontrollera problemets ursprung. Det allmänna rådet är att starta QGIS under en ny användarprofil, alltså en renare konfiguration, och köra kommandona igen.

Om felet hindrar dig från att skapa en ny användarprofil från menyn :menuselection:` *Settings –> Användarprofiler* kan du antingen:

- Byt namn i filutforskaren på den ”trasiga” användarprofilmappen i mappen `QGIS3/profiles` och starta om QGIS. En ny `standard` användarprofil kommer att skapas och köras.
- Starta QGIS från kommandoraden med hjälp av det nya argumentet *profiltyp*:

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

5.2.2 Ställa in användarprofil

Som standard öppnar QGIS en ny session med profilen för den senast stängda sessionen. Detta, bland andra inställningar, kan anpassas i fliken *Inställningar* ► *Alternativ* ► *Användarprofiler*:

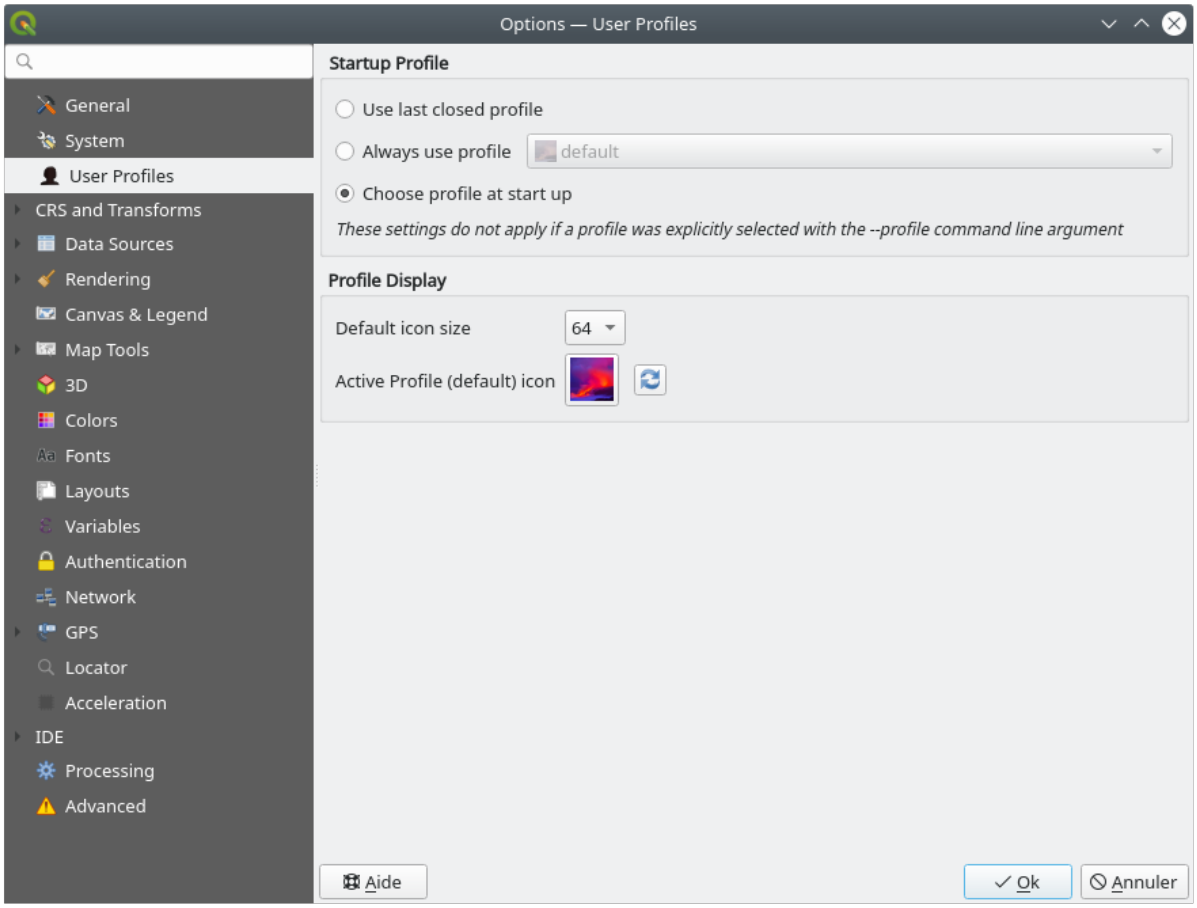


Fig. 5.31: Inställningar för användarprofiler

- *Startup profile*: anger den användarprofil som ska användas när en QGIS-session startas. Den kan vara:
  - *Använd den senast stängda profilen*
  - en specifik användarprofil att välja från rullgardinsmenyn *Använd alltid profil*
  - *Välj profil vid uppstart*: Öppnar dialogrutan *User Profile Selector* med en lista över tillgängliga användarprofiler. Dubbelklicka på en post eller välj en profil och tryck på *OK* för att använda den användarprofilen vid start av sessionen. Det är också möjligt att *Add new profile* till listan: en profil skapas från grunden och körs automatiskt med den inledande sessionen.
- Under *Profilvisning* kan du ställa in:
  - Storleken på ikonerna som ska användas när du väljer en profil i dialogrutan *User Profile Selector*.
  - En specifik ikon som ska visas bredvid den aktuella profilen i menyn *Inställningar* ► *Användarprofiler* eller i dialogrutan *User Profile Selector*. Tryck på Reset profile icon för att ta bort anpassade ändringar.

5.3 Projektets egenskaper

I egenskapsfönstret för projektet under *Projekt* ► *Projektegenskaper* kan du ställa in projektspecifika alternativ. De projektspecifika alternativen skriver över sina motsvarigheter i dialogen *Options* som beskrivs ovan.

5.3.1 Allmänna egenskaper

På fliken *General* kan du använda *General settings*:

- se platsen för projektfilen
- ange mappen för projektets startsida (finns i *Project home* i *Browser*-panelen). Sökvägen kan vara relativ till projektfilens mapp (skriv in den) eller absolut. Projektets hemmapp kan användas för att lagra data och annat innehåll som är användbart för projektet. Praktiskt när dataset och projektfiler inte lagras på samma plats. Om fältet *Project home* inte fylls i används projektfilens mapp som standard.
- ge projektet en titel vid sidan av projektets filväg
- välja den färg som ska användas för funktioner när de väljs
- välj bakgrundsfärg: den färg som ska användas för kartbilden
- anger om sökvägen till lagren i projektet ska sparas som absolut (fullständig) eller som relativ till projektfilens plats. Du kanske föredrar relativ sökväg när både lager och projektfiler kan flyttas eller delas eller om projektet nås från datorer på olika plattformar.
- välj för att undvika artefakter när projektet återges som kartplattor. Observera att om du markerar det här alternativet kan det leda till försämrad prestanda.
- *Memory attribute tables windows and docks between sessions*: Om detta är markerat för ett projekt sparas alla öppnade attributtabeller i projektet och återställs omedelbart när projektet laddas. Detta kan förbättra arbetsflödena när du har skapat ett projekt med en viss uppsättning attributtabellkonfigurationer för dina behov, och det är krångligt att ställa in dessa attributtabeller på nytt.

Att beräkna ytor och avstånd är ett vanligt behov i GIS. Men dessa värden är egentligen knutna till de underliggande projektionsinställningarna. I ramen *Measurements* kan du styra dessa parametrar. Du kan faktiskt välja:

- den *Ellipsoid*, på vilken beräkningar av avstånd, yta och vinkel helt baseras; den kan vara:
  - **None/Planimetric**: returnerade värden är i detta fall kartesiska mätningar. Detta alternativ kan ställas in som standard för nya projekt från menyn *Settings* ► *Options* ► *CRS Handling*
  - en **Custom**: du måste ställa in värden för semi-major- och semi-minor-axlarna.
  - eller en befintlig från en fördefinierad lista (Clarke 1866, Clarke 1880 IGN, New International 1967, WGS 84 …).
- *Enheter för avståndsmätningar* för längd och omkrets, och *Enheter för areamätningar*. Dessa inställningar som är standard för deras motsvarande *globala alternativ* åsidosätter dem i det aktuella projektet. De används i:
  - Bar för uppdatering av fält i attributtabell
  - Beräkningar med fältkalkylator
  - *Identifiera verktyg* härledda värden för längd, omkrets och area
  - *mätdialog*
  - *skalbar dekoration*

> **Observera**
>
> För att säkerställa att måttenheterna (för avstånd och yta) alltid återspeglar enheterna i det aktuella projektets CRS, välj *Map Units* för dessa alternativ.

Med *Coordinate and Bearing display* kan du anpassa visningen av:

- de koordinater som visas i rutan *Coordinates* i QGIS statusfält och i avsnittet *Derived* i verktyget Identify features resultat
- det bäringsvärde som visas i statusfältet för kartans panoreringsriktning och av verktyget Mät bäring.

Tillgängliga parametrar är:

- *Display coordinates using* either:
  - `Map Units`, baserat på projektet CRS
  - `Map Geographic (degrees)`: baseras på projektets CRS om det är av geografisk typ, annars används dess associerade geografiska CRS. Detta är användbart t.ex. för himlakroppar som inte är jordiska.
  - eller `Custom Projection Units`: gör det möjligt att använda vilken CRS du vill för visning av koordinater
- I alternativet *Coordinate CRS* kan du visa eller definiera det CRS som ska användas beroende på visningsläge.
- *Koordinatformat*: du kan konfigurera det som `Decimalgrader`, `Degrader, Minuter` eller `Degrader, Minuter, Sekunder`, och om det ska visas:
  - ☐ *Visa riktningssuffix*
  - ☐ *Visa inledande nollor för minuter och sekunder*
  - ☐ *Visa inledande nollor för grader*
  - ☐ *Visa efterföljande nollor*
- *Koordinatprecision*: antalet decimaler kan vara automatiskt (härlett från typen av CRS) eller ställas in manuellt
- *Koordinatordning*: du kan välja att visa koordinaterna i den ursprungliga ordningen för CRS (`Default`) eller ändra den till antingen `Easting, Northing (Longitude, Latitude)` eller `Northing, Easting (Latitude, Longitude)` ordning
- *Bearing format* möjliga värden är `0 till 180°, med E/W suffix`, `-180 till +180°` eller `0 till 360°`. Antalet *Decimalställen* samt om *Visa nollor i slutet* kan ställas in.

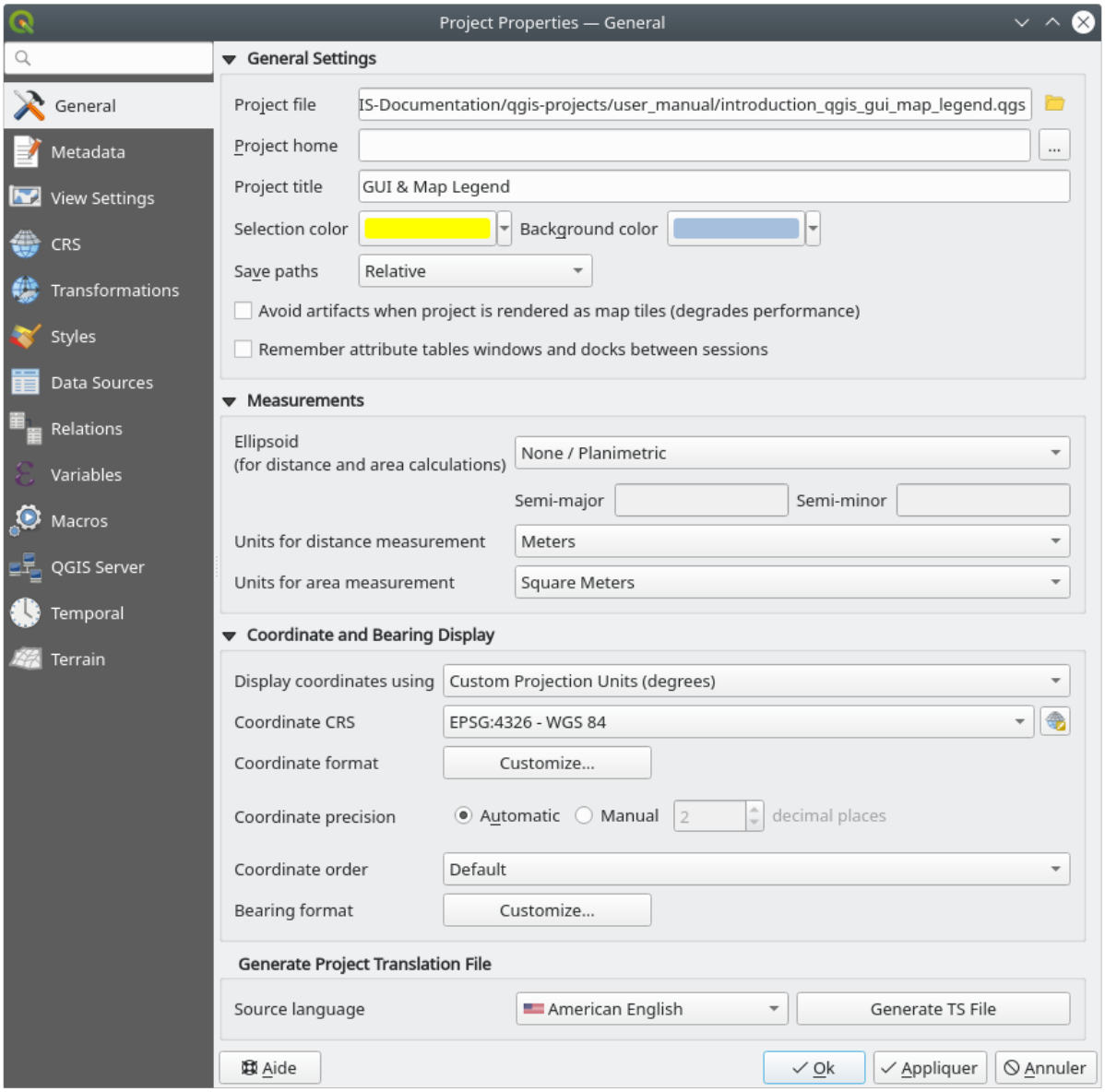


Fig. 5.32: Fliken Allmänt i dialogrutan Egenskaper projekt

5.3.2 Egenskaper för metadata

Fliken *Metadata* gör det möjligt att definiera detaljerade metadata, inklusive (bland annat): författare, skapandedatum, språk, sammanfattningar, kategorier, nyckelord, kontaktuppgifter, länkar, historik. Det finns också en valideringsfunktion som kontrollerar om vissa fält är ifyllda, men den är inte tvingande. Se *vektorlagrets metadataegenskaper* för mer information.

5.3.3 Visa inställningar

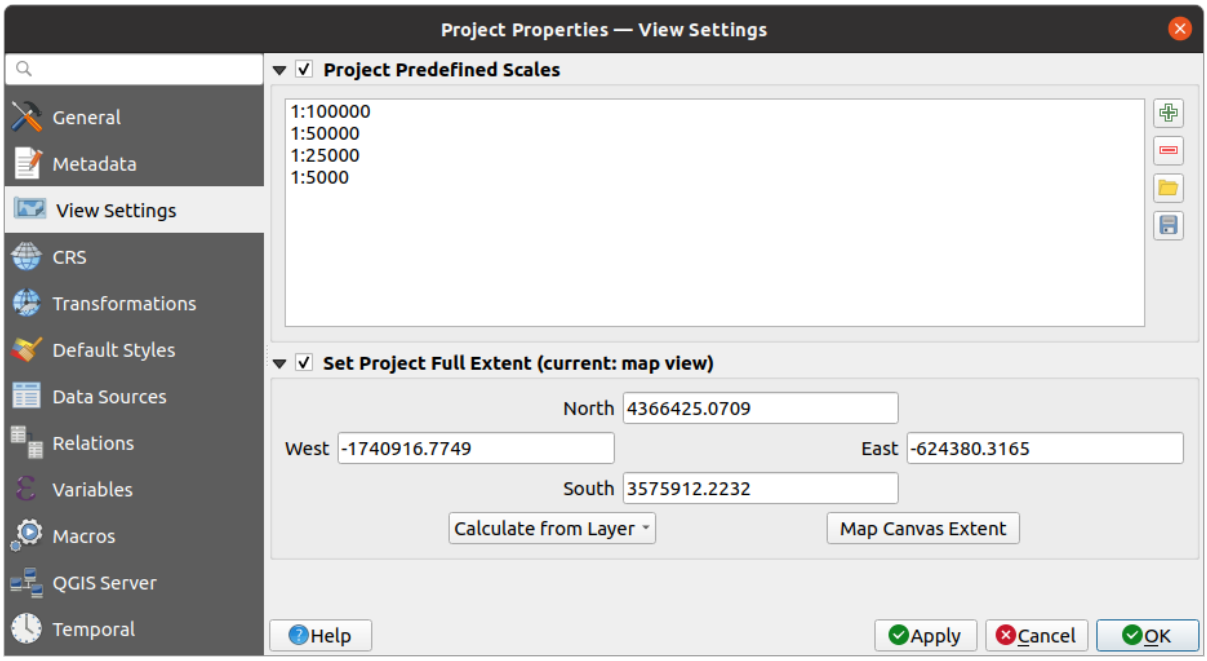


Fig. 5.33: Fliken Visningsinställningar i dialogrutan Egenskaper projekt

Fliken *View Settings* ger dig möjlighet att styra projektkartans canvas. Du kan göra följande:

- set *Project predefined scales*: listan över skalor som ska visas i skalrelaterade rullgardinswidgets, t.ex. statusfältet *Scale*, skalväljaren för synlighet eller inställningar för sekundär 2D-kartvy, … i stället för den globala *predefined scales*.
- *Set Project full Extent*: denna omfattning kommer att användas istället för omfattningen av alla lager när man zoomar till full kartomfattning ( ). Det är användbart när ett projekt innehåller webblager/nationella lager/globala lager men det faktiska intresseområdet för projektet är ett mindre geografiskt område. Koordinaterna för projektets fulla utsträckning kan ställas in med widgeten *extent selector*.

5.3.4 CRS-egenskaper

Observera

För mer information om hur QGIS hanterar projektprojektion, läs det dedikerade avsnittet på *Arbeta med projektioner*.

Fliken *CRS* hjälper dig att ställa in det koordinatreferenssystem som ska användas i projektet. Det kan vara:

- *Ingen CRS (eller okänd/ej jordprojektion)*: lager ritas baserat på deras råkoordinater
- eller ett befintligt koordinatreferenssystem som kan vara *geografiskt*, *projicerat* eller *användardefinierat*. Lager som läggs till i projektet översätts i realtid till detta CRS för att de ska kunna överlagras utan hänsyn till deras ursprungliga CRS.

5.3.5 Transformationer Egenskaper

Fliken *Transformations* hjälper dig att styra inställningarna för reprojektion av lager genom att konfigurera de inställningar för datumtransformation som ska gälla i det aktuella projektet. Som vanligt åsidosätter dessa alla motsvarande globala inställningar. Se *Datumtransformationer* för mer information.

5.3.6 Stilegenskaper

Under fliken *Styles* kan du konfigurera symboler och färger för projektet, vilket gör det möjligt att dela projektet på ett säkert sätt mellan olika maskiner.

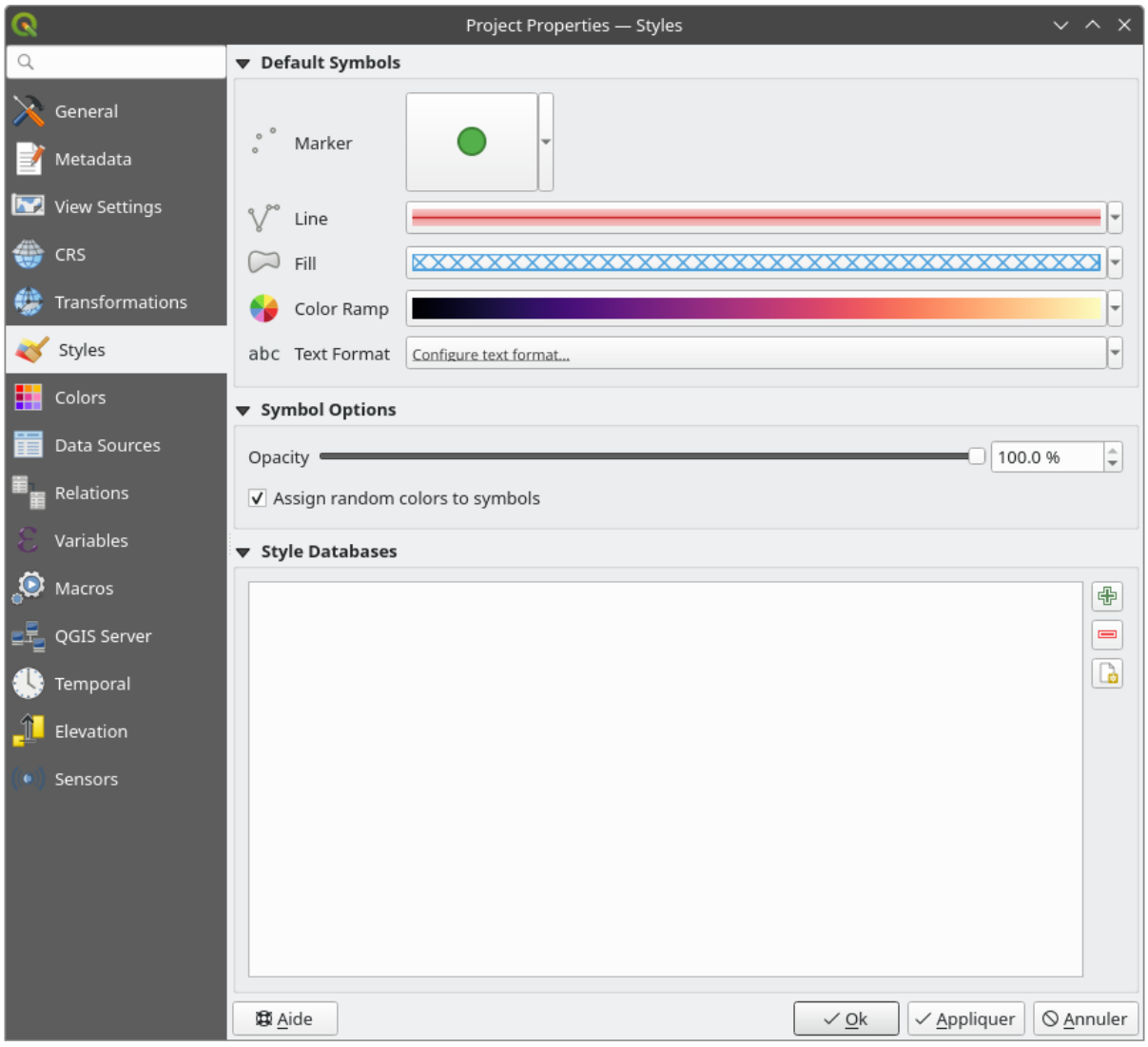


Fig. 5.34: Fliken Stilar

Standardsymboler

Med gruppen *Default Symbols* kan du styra hur nya lager ska ritas i projektet när de inte har en befintlig `.qml`-stil definierad. Du kan ställa in att *Marker*, *Line*, *Fill* ska gälla beroende på lagrets geometrityp samt standard *Color Ramp* och *Text Format* (t.ex. när märkning aktiveras). Alla dessa objekt kan återställas med hjälp av *Clear* i motsvarande rullgardinswidget.

Symbolalternativ

I gruppen *Symbolalternativ* kan du:

- Tillämpa standardvärdet *Opacity* på nya lager
- *Tilldela slumpmässiga färger till symboler*, ändra symbolernas fyllningsfärger och undvik därmed samma rendering för alla lager.

Stildatabas

En stildatabas i QGIS är ett strukturerat arkiv som är utformat för att lagra symboler, textformat och andra stylingelement. Den fungerar som en centraliserad plats där du kan organisera och hantera dina symbologiresurser på ett effektivt sätt. Du kan skapa en dedikerad stildatabas för en specifik kund, med symboler som är skräddarsydda för den kundens behov. Detta säkerställer en ren och organiserad metod utan att störa standardstildatabasen. I miljöer med flera användare är det möjligt att lagra projektspecifika stilar på en gemensam plats. Genom att länka ett projekt till dessa stilar får alla användare inom projektet tillgång till gemensam symbologi, vilket eliminerar behovet av manuell import eller uppdateringar av enskilda lokala stildatabaser. Att lagra symboler i en projektfil är en praktisk lösning som hindrar användare från att överfylla sin globala stildatabas med projektspecifik symbologi. Detta tillvägagångssätt garanterar att andra användare som arbetar med samma projekt omedelbart har tillgång till all nödvändig symbologi när projektet laddas.

I avsnittet *Style Database* kan du välja att *Lägg till* eller *Ta bort* stildatabas eller så kan du *Create new style database*. När du lägger till eller tar bort en stildatabas i det här avsnittet kommer ändringarna automatiskt att återspeglas i *Symbology Properties*.

5.3.7 Färger egenskaper

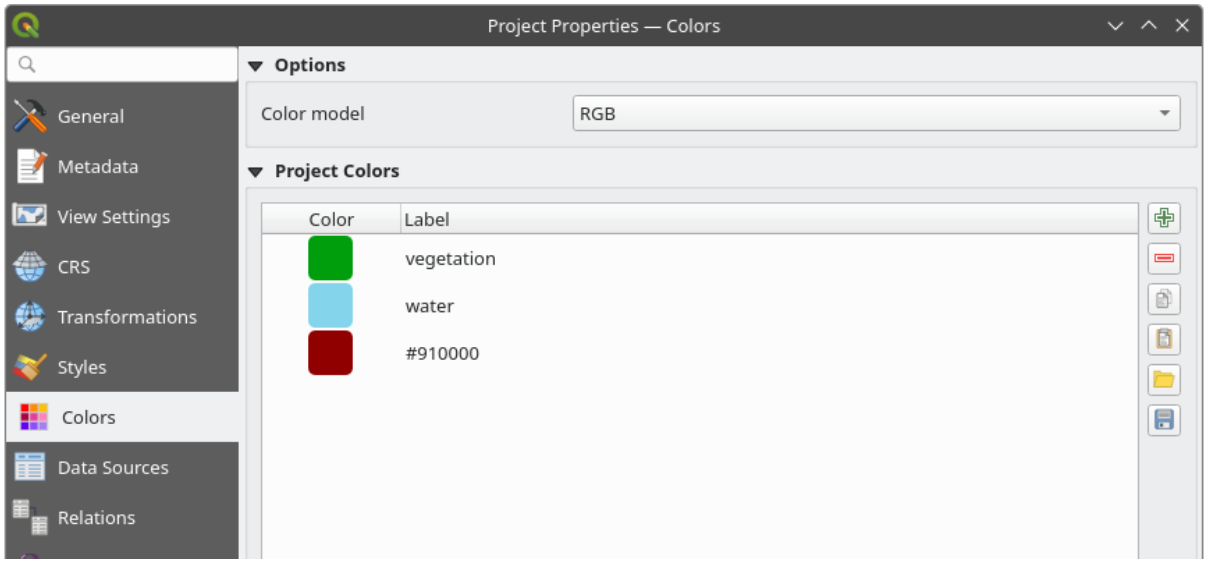


Fig. 5.35: Fliken Färger

På fliken *Colors* kan du välja *Color model*, dvs. om en färg som standard ska definieras som `RGB` eller `CMYK`. Alla färger som definieras i en annan färgmodell än den som anges här kommer att konverteras till denna färgmodell när en layout exporteras.

Det finns också ett extra avsnitt där du kan definiera specifika färger för det pågående projektet. Precis som i *global colors* kan du göra det:

- *Lägg till* eller *Ta bort* färg
- *Kopiera* eller *Klistra in* färg
- *Importera* eller *Exportera* uppsättningen färger från/till `.gpl`-filen.

Dubbelklicka på en färg i listan för att justera eller byta ut den i dialogrutan *Color Selector*. Du kan också byta namn på den genom att dubbelklicka i kolumnen *Label*.

Dessa färger identifieras som *Project colors* och listas som en del av *color widgets*.

Tips

Använd projektfärger för att snabbt tilldela och uppdatera färgwidgets

Projektfärger kan refereras till med hjälp av deras etikett och de färgwidgets som de används i är bundna till dem. Detta innebär att du kan undvika att upprepa samma färg för många egenskaper, och för att undvika en besvärlig uppdatering kan du:

1. Definiera färgen som en projektfärg
2. Klicka på *data defined override widget* bredvid den färgegenskap som du vill ställa in
3. Håll muspekaren över menyn *Color* och välj projektfärgen. Egenskapen tilldelas då uttrycket `project_color('color_label')` och färgwidgeten återspeglar den färgen.
4. Upprepa steg 2 och 3 så ofta som behövs
5. Uppdatera projektfärgen en gång och ändringen återspeglas ÖVERALLT där den används.

5.3.8 Egenskaper för datakällor

På fliken *Datakällor* kan du:

- *Transaktionsläge*, definierar hur ändringar skickas till dataleverantören:
  - *Local Edit Buffer*: redigeringar buffras lokalt och skickas till leverantören när du växlar redigeringsläge för lager eller klickar på *Save edits*.
  - *Automatic Transaction Groups*: på datakällor som stöds (postgres och geopackage-databaser) synkroniseras redigeringstillståndet för alla tabeller som kommer från samma databas och utförs i en transaktion på serversidan. Istället för att buffra redigeringsändringar lokalt skickas de direkt till en transaktion i databasen som bekräftas när du växlar redigeringsläge för lager eller klickar på *Save edits*.
  - *Buffered Transaction Groups*: alla redigerbara lager, oavsett från vilken leverantör, växlas synkront och alla ändringar sparas i en lokal redigeringsbuffert. Spara ändringar utförs inom en enda transaktion på alla lager (per leverantör).

  Observera att du endast kan ändra detta alternativ om inget lager redigeras i projektet.
- *Utvärdera standardvärden på leverantörssidan*: När du lägger till nya funktioner i en PostgreSQL-tabell utvärderas fält med standardvärdebegränsning och fylls i vid formulärröppningen och inte vid commit-ögonblicket. Detta innebär att istället för ett uttryck som `nextval('serial')`, kommer fältet i formuläret *Lägg till funktion* att visa förväntat värde (t.ex. `25`).

- *Memory editable layer status between sessions*: ser till att alla lager som är redigerbara i ett projekt kommer ihåg som sådana när projektet sparas, samt att dessa lager omedelbart görs redigerbara när projektet återställs.
- Konfigurera *Layers Capabilities*, d.v.s:
  - Ange (eller inaktivera) vilka lager som är `identifierbara`, dvs. kommer att svara på *identify tool*. Som standard är lager inställda på att vara sökbara.
  - Ställ in om ett lager ska visas som ”skrivskyddat”, vilket innebär att det inte kan redigeras av användaren, oavsett dataleverantörens kapacitet. Även om det här är ett svagt skydd är det en snabb och praktisk konfiguration för att undvika att slutanvändare ändrar data när de arbetar med filbaserade lager.
  - Definiera vilka lager som är ”sökbara”, dvs. kan sökas med hjälp av *locator widget*. Som standard är lager inställda som sökbara.
  - Definiera vilka lager som ska definieras som ”obligatoriska”. Markerade lager i den här listan är skyddade från att oavsiktligt tas bort från projektet.
  - Definiera vilka lager som är `privata`, dvs. dolda från *Layers*-panelen. Detta är avsett för tillbehörsskikt (baskarta, join, uppslag för värderelationer, troligen aspatiala lager, …) som du fortfarande behöver i ett projekt men som du inte vill ska förorena förklaringsträdet och andra verktyg för val av lager. Om de är inställda som synliga visas de fortfarande i kartbilden och återges i teckenförklaringen för utskriftslayouten. Använd alternativet *Filter förklaring* ► *Show private layers* i *Layers* panelens övre verktygsfält för att tillfälligt aktivera dem för all interaktion.

  Tabellen *Layers Capabilities* innehåller några praktiska verktyg för att:
  - Markera flera celler och tryck på *Toggle Selection* för att ändra deras kryssrutestatus;
  - *Visa endast spatiala lager*, filtrerar bort icke-spatiala lager från lagerlistan;
  - *Filter layers…* och snabbt hitta ett visst lager att konfigurera.
- Under gruppen *Avancerade inställningar* kan du välja *Trust project when data source has no metadata*: För att påskynda projektladdningen genom att hoppa över datakontroller. Användbart i QGIS Server-kontext eller i projekt med enorma databasvyer/materialiserade vyer. Omfattningen av lager kommer att läsas från QGIS-projektfilen (istället för datakällor) och när PostgreSQL-leverantören används kommer primärnyckelns enhetlighet inte att kontrolleras för vyer och materialiserade vyer.

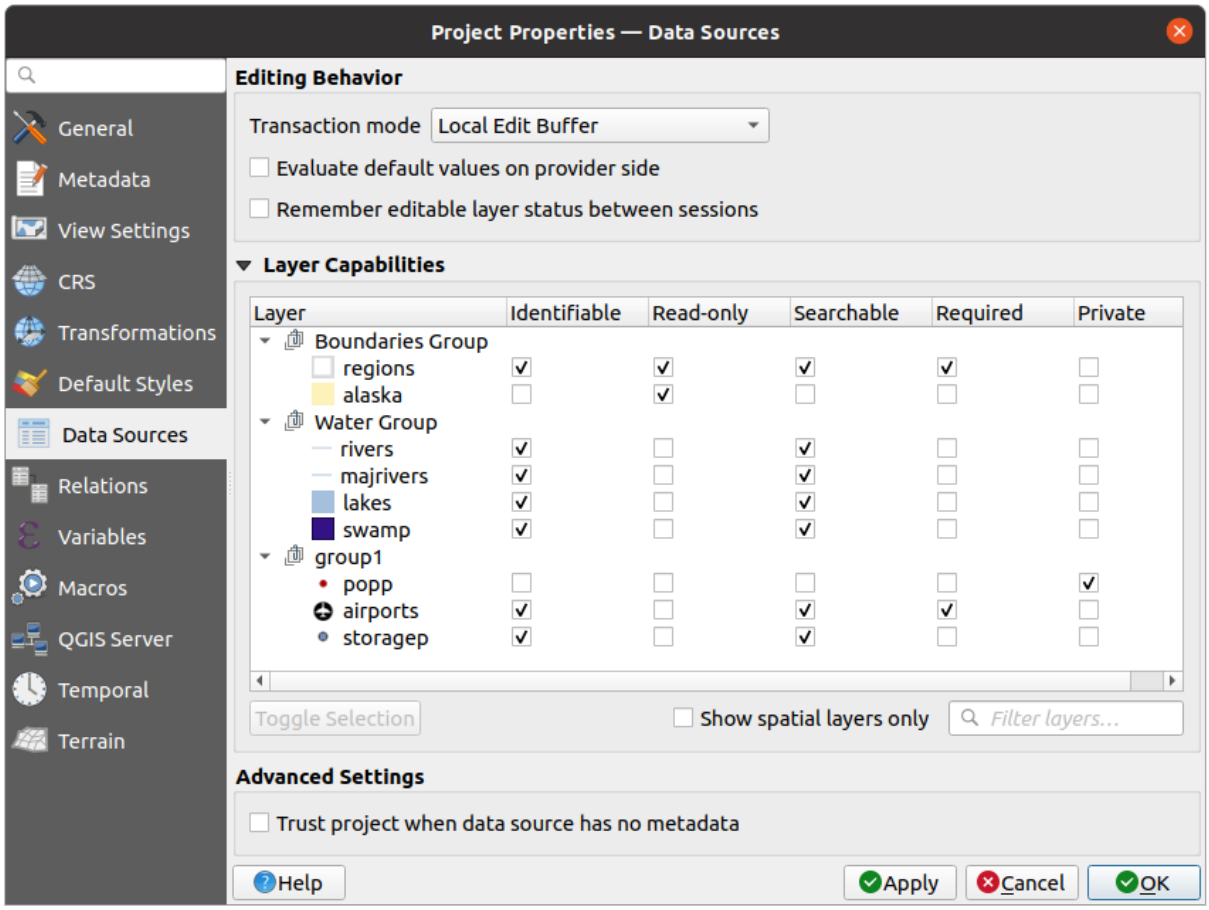


Fig. 5.36: Fliken Datakällor

5.3.9 Relationer Egenskaper

Fliken *Relations* används för att definiera relationer mellan lager. Relationerna kan vara av typen en till en, många till många eller polymorf. De definieras i dialogrutan för projektegenskaper. När det finns relationer för ett lager kommer ett nytt användargränssnittselement i formulärvyn (t.ex. när du identifierar en funktion och öppnar dess formulär) att lista de relaterade enheterna. Detta är ett kraftfullt sätt att uttrycka t.ex. inspektionshistoriken för en rörledningssträcka eller ett vägsegment. Du kan läsa mer om relationsstöd i avsnitt *Inställning av relationer mellan flera lager*.

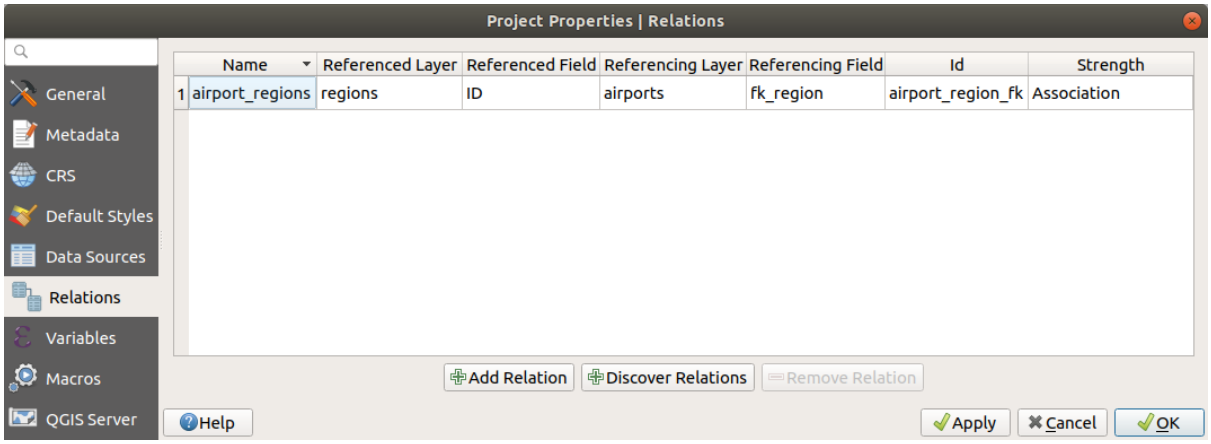


Fig. 5.37: Fliken Relationer

5.3.10 Variabelegenskaper

Fliken *Variables* innehåller en lista över alla variabler som är tillgängliga på projektnivå (inklusive alla globala variabler). Dessutom kan användaren hantera variabler på projektnivå. Klicka på knappen för att lägga till en ny anpassad variabel på projektnivå. På samma sätt kan du välja en anpassad variabel på projektnivå i listan och klicka på knappen för att ta bort den. Mer information om användning av variabler finns i avsnittet Allmänna verktyg *Lagring av värden i variabler*.

5.3.11 Egenskaper för makron

Fliken *Macros* används för att redigera Python-makron för projekt. För närvarande finns bara tre makron tillgängliga: `openProject()`, `saveProject()` och `closeProject()`.

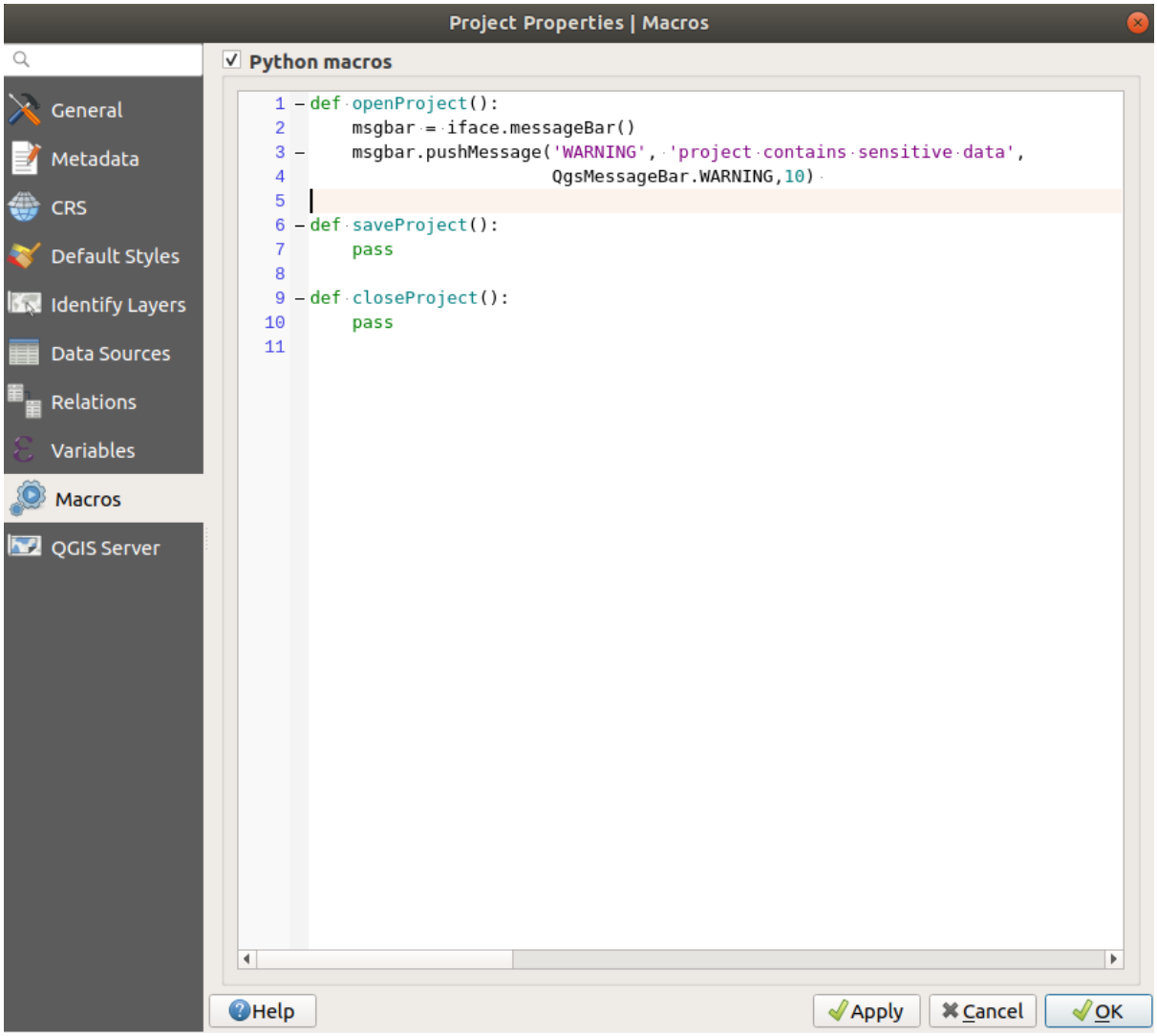


Fig. 5.38: Makroinställningar

5.3.12 Egenskaper för QGIS-server

På fliken *QGIS Server* kan du konfigurera ditt projekt för att kunna publicera det online. Här kan du definiera information om QGIS Server WMS- och WFS-funktioner, omfattning och CRS-restriktioner. Mer information finns i avsnitt Creatingwmsfromproject och efterföljande.

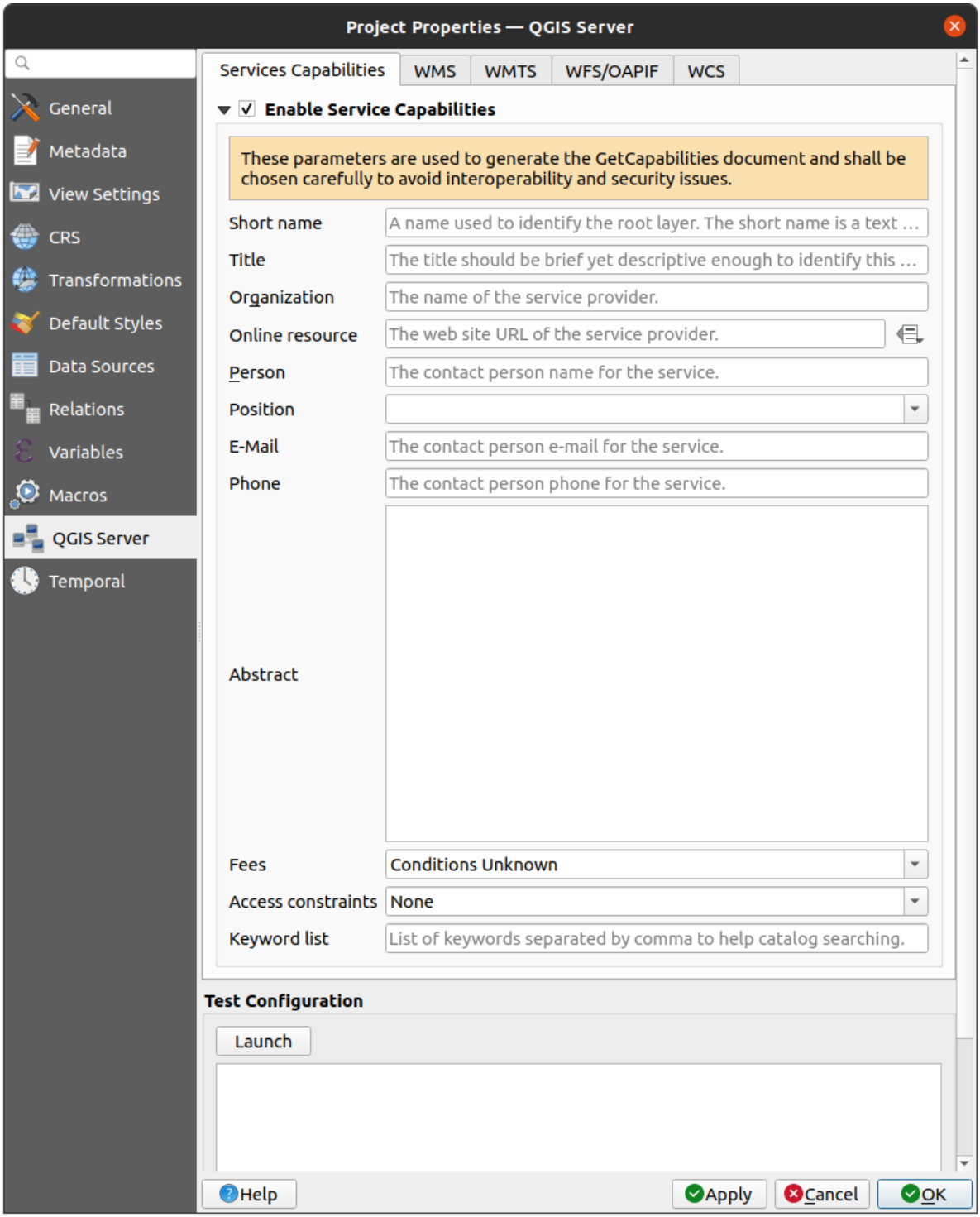


Fig. 5.39: Inställningar för QGIS Server

5.3.13 Temporala egenskaper

Fliken *Temporal* används för att ställa in tidsintervallet för ditt projekt, antingen genom att använda manuella *Startdatum* och *Slutdatum* eller genom att beräkna det från projektets aktuella tidslager. Projektets tidsintervall kan sedan användas i *Temporal controller panel* för att hantera kartans canvas *temporal navigation*.

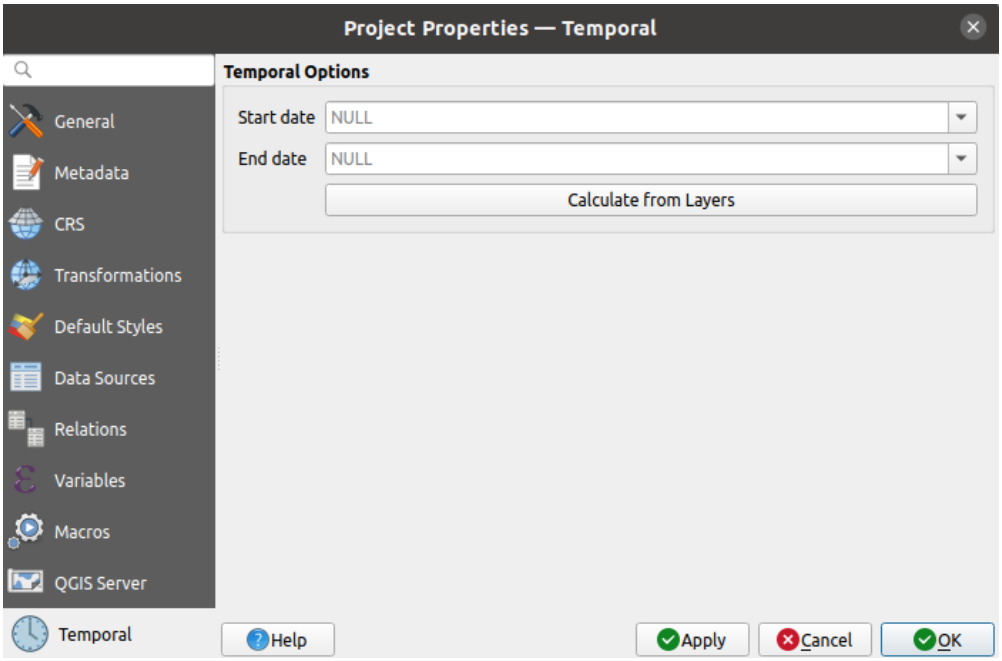


Fig. 5.40: Fliken Projekttid

5.3.14 Elevation Egenskaper

Fliken *Elevation* hjälper dig att konfigurera standardinställningar för terräng och höjd. När en ny *3d-karta* skapas i projektet kommer kartan som standard att använda samma terränginställningar som har definierats för projektet. Projektets höjdinställningar kommer också att respekteras av verktyget *elevation profile*.

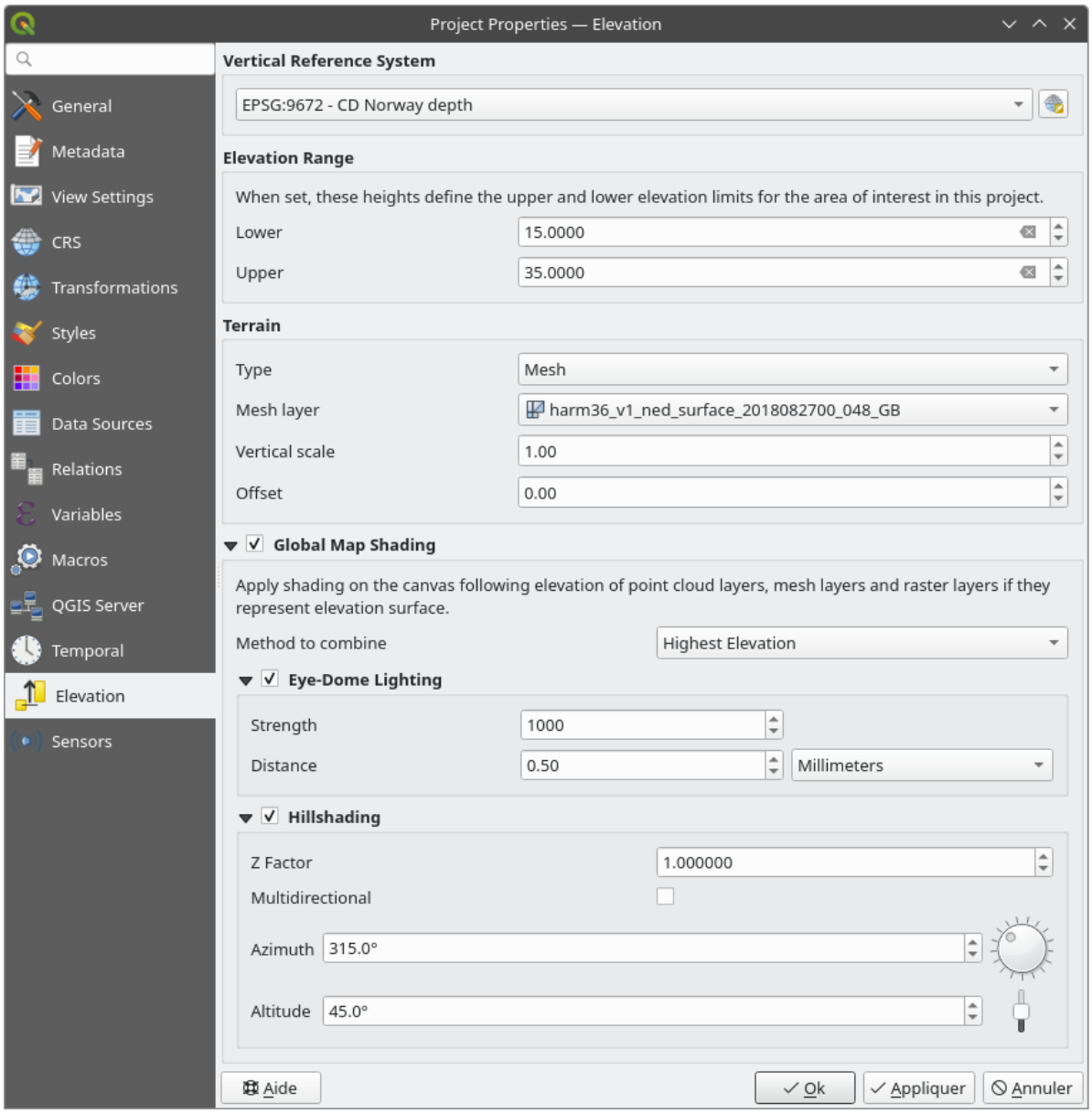


Fig. 5.41: Fliken Projekthöjd

- *Vertikalt referenssystem*: Om CRS för ditt projekt är sammansatt (inklusive en Z-dimension), kommer det vertikala CRS som används för projektet att vara den vertikala komponenten i projektets CRS. Om ditt projekts CRS är horisontellt kan du välja ett specifikt vertikalt CRS genom att klicka på Välj CRS.
- ***Elevation Range*: hjälper dig att definiera *Lower* och *Upper* gränserna**
  för *elevation controller* i projektet.
- *Terrain* och höjdalternativ är tillgängliga för:
  - *Flat terrain* med *Terrain height* inställning
  - *DEM (Raster Layer)*: med inställning för att definiera *Raster layer*, en *Vertical scale*-faktor som ska tillämpas på bandvärden och en vertikal *Offset*
  - *Mesh*: med inställning för att definiera *Mesh-lagret*, en *Vertical scale*-faktor som ska tillämpas på hörnens Z-värde och en vertikal *Offset*

  Dessa inställningar kan skrivas över från 3D-kartan *konfigurationsdialog*.

- Med inställningarna *Global kartskuggning* tillämpar du en global skuggningseffekt på kartan, baserat på höjden för alla aktiva lager som har höjdegenskaper aktiverade, inklusive:
  - *rasterlager*
  - *meshlager*
  - *punktmolnlager*

  Method to combine: höjdvärdet vid en viss position för alla relevanta lager kombineras, och den betraktade höjden väljs beroende på en av dessa metoder:
  - *Högsta höjd* värde
  - *Baserat på lagrets ordning*: höjden på det översta lagret i lagerträdet beaktas.

  Alternativ för skuggning som stöds är:
  - *Eye-dome lighting*: tillämpar skuggningseffekter på kartans canvas för en bättre djupåtergivning. Följande parametrar kan kontrolleras:
    - *Strength*: ökar kontrasten, vilket ger bättre djupseende
    - *Distance*: representerar avståndet mellan de använda pixlarna och mittpixeln och gör att kanterna blir tjockare.
  - *Hillshading*, formar några reliefer på kartan med hjälp av skuggning (nivåer av grått):
    - *Z-faktor*: Skalningsfaktor för höjdvärdet
    - *Multidirectional*: Ange om multidirektionell hillshading ska användas
    - *Azimuth*: Ljuskällans azimut
    - *Altitud*: Ljuskällans höjdvinkel

> **Observera**
>
> En genväg till egenskaperna för *Global map shading* finns via panelen *Layer Styling*.

5.3.15 Sensorer Egenskaper

Fliken *Sensors* används för att konfigurera sensorer och växla deras anslutningsstatus. När sensorerna är aktiva samlar de passivt in data i bakgrunden och gör sina senaste data tillgängliga för uttryck och pythonskript.

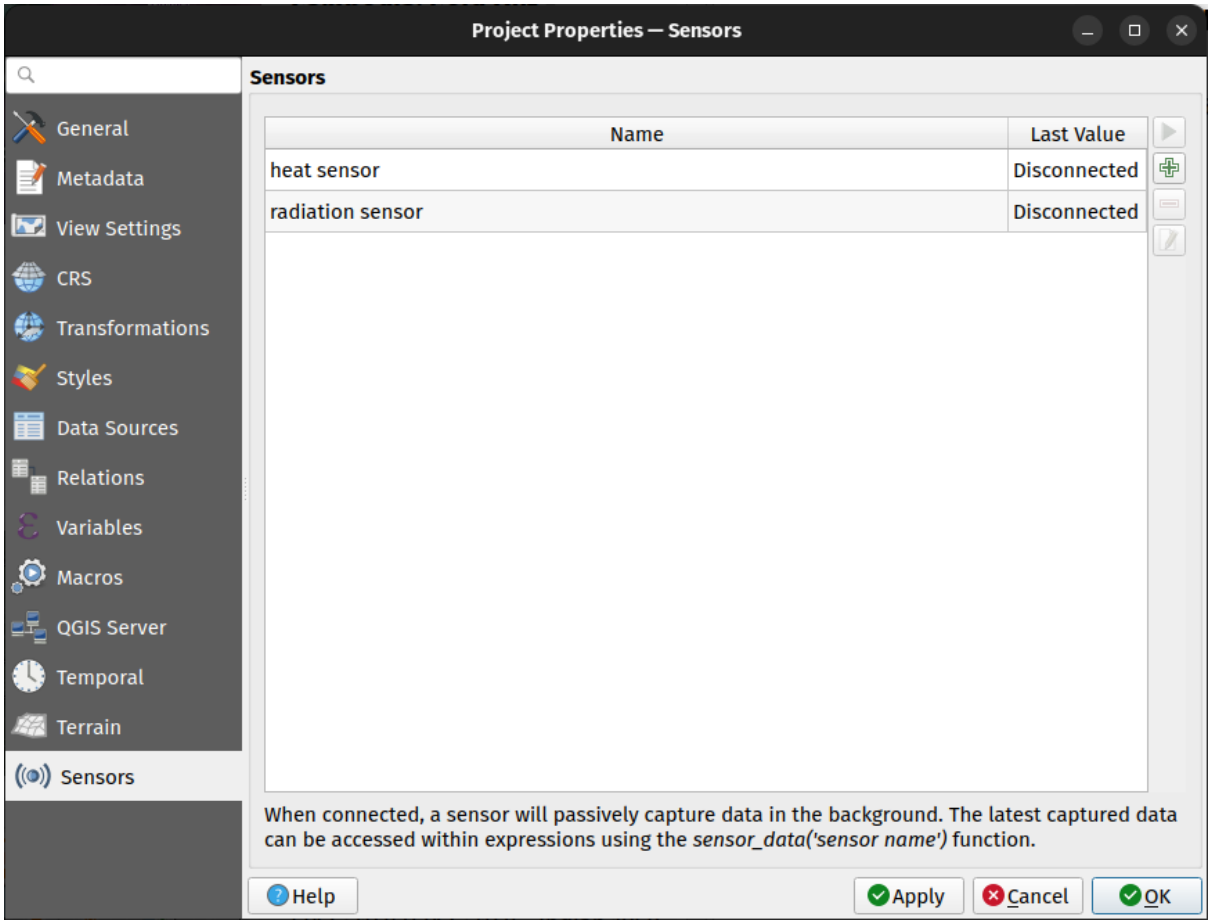


Fig. 5.42: Fliken Projektsensorer

För att lägga till en ny sensor klickar du på knappen . En underpanel för inställningar öppnas och du kan konfigurera den:

- *Sensornamn*: används för att hämta sensorvärden i uttryck och python-skript;
- *Sensortyp*: TCP, UDP, serieport, etc.; och
- ytterligare typspecifika detaljer (t.ex. värdnamn och port).

  För sensorer med serieport finns kombinationsrutor med systemets tillgängliga serieportar och anslutningens baudhastighet tillgängliga för val. En *Data frame delimiter* kan användas för att analysera returnerade data. Om det finns flera dataramar i den sista datapunkten som skickas från serieporten returneras alla dataramar. Du kan sedan använda ett uttryck för att ytterligare dela upp sensorns sista datavärde i enskilda ramar (t.ex. med hjälp av funktionen *string_to_array*).

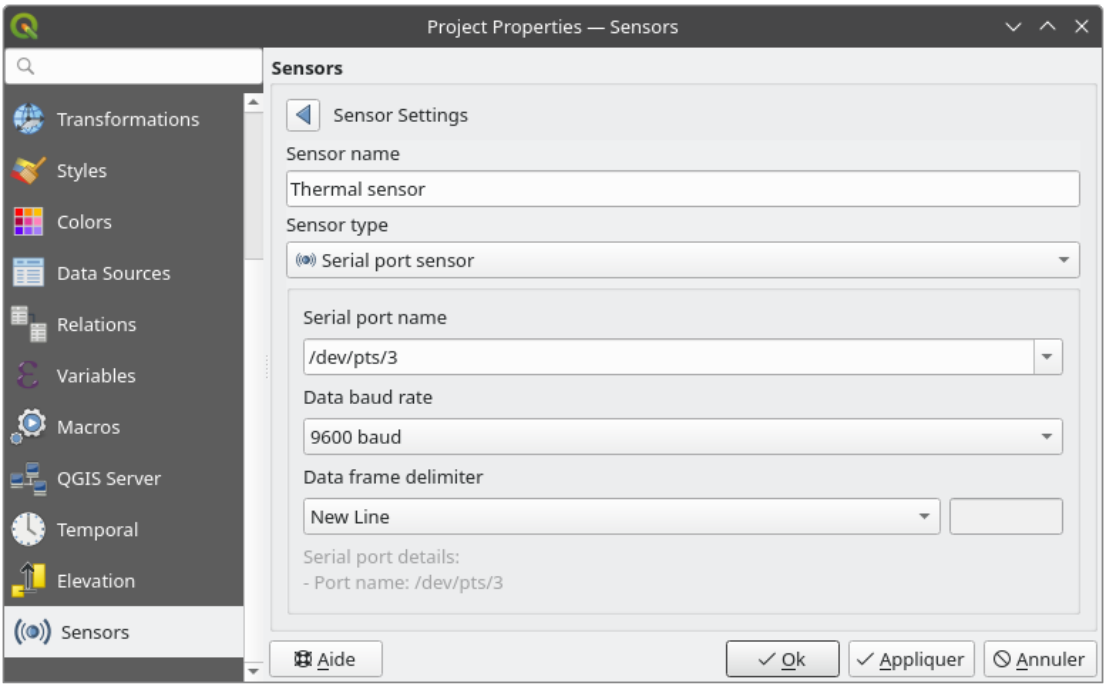


Fig. 5.43: Underpanel för sensorinställningar

När en sensor har konfigurerats kan du använda knappen Start för att ansluta sensorn. När sensorn är aktiv visas de senast insamlade uppgifterna i kolumnen ”Senaste värde” i tabellen *Last value*.

5.4 Anpassning

Med dialogrutan *Customization* kan du (av)aktivera nästan alla element i QGIS användargränssnitt. Detta kan vara mycket användbart om du vill förse dina slutanvändare med en ”lätt” version av QGIS, som bara innehåller de ikoner, menyer eller paneler som de behöver.

Observera

Innan dina ändringar tillämpas måste du starta om QGIS.

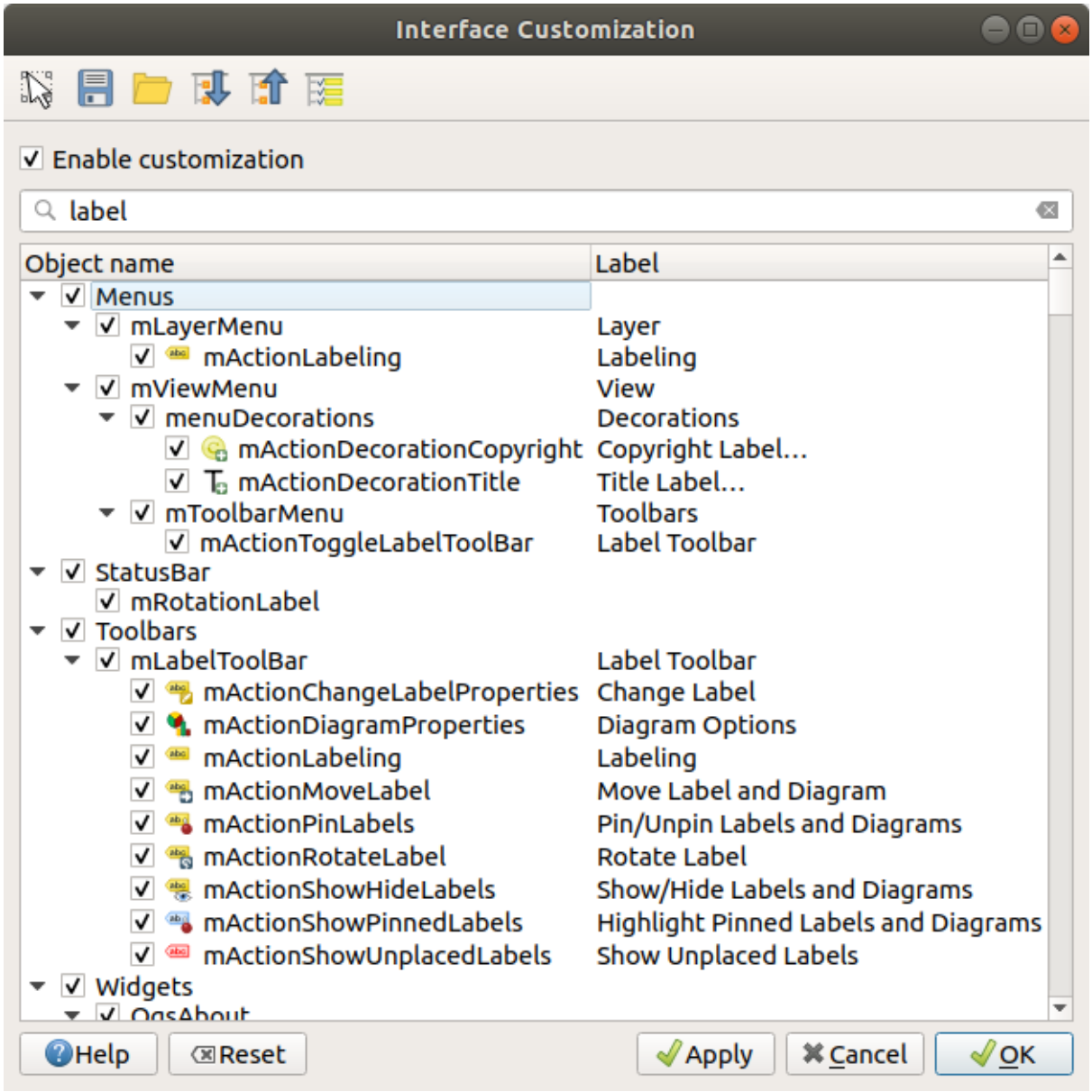


Fig. 5.44: Dialogrutan Anpassning

Att kryssa i kryssrutan *Enable customization* är det första steget på vägen mot QGIS-anpassning. Detta aktiverar verktygsfältet och widgetpanelen där du kan avmarkera och därmed inaktivera vissa GUI-objekt.

Det konfigurerbara objektet kan vara:

- en **Menu** eller några av dess undermenyer från *Menyrad*
- en hel **Panel** (se *Paneler och verktygsfält*)
- **Statusfältet** som beskrivs i *Statusrad* eller några av dess objekt
- ett **verktygsfält**: hela fältet eller några av dess ikoner
- eller valfri **widget** från valfri dialogruta i QGIS: etikett, knapp, kombinationsruta…

Med Switch to catching widgets in main application kan du klicka på ett objekt i QGIS-gränssnittet som du vill ska vara dolt och QGIS avmarkerar automatiskt motsvarande post i dialogrutan Anpassning. Du kan också använda rutan *Sök* för att hitta objekt med hjälp av deras namn eller etikett.

När du har ställt in din konfiguration klickar du på *Apply* eller *OK* för att validera dina ändringar. Denna konfiguration blir den som används som standard av QGIS vid nästa start.

Ändringarna kan också sparas i en `.ini`-fil med hjälp av Save To File-knappen. Detta är ett praktiskt sätt att dela ett gemensamt QGIS-gränssnitt mellan flera användare. Klicka bara på Load from File från destinationsdatorn för att importera filen `.ini`. Du kan också köra *kommandoradsverktyg* och spara olika inställningar för olika användningsfall.

Tips

Återställer enkelt fördefinierade QGIS

Den ursprungliga konfigurationen av QGIS grafiska gränssnitt kan återställas med någon av metoderna nedan:

- avmarkera alternativet *Aktiver anpassningar* i dialogrutan Anpassning eller klicka på knappen Check All
- tryck på knappen *Reset* i ramen **Settings** under menyn *Inställningar* ► *Alternativ*, fliken *System*
- starta QGIS i en kommandotolk med följande kommandorad `qgis --nocustomization`
- sätta värdet på variabeln *UI* ► *Anpassning* ► *Aktiverad* till `false` under *Inställningar* ► *Alternativ*-menyn, *Avancerat*-fliken (se *varning*).

I de flesta fall måste du starta om QGIS för att ändringen ska tillämpas.

5.5 Tangentbordsgenvägar

QGIS tillhandahåller standardtangentbordsgenvägar för många funktioner. Du hittar dem i avsnitt *Menyrad*. Dessutom kan du med menyalternativet *Settings* ► *Keyboard Shortcuts…* ändra standardtangentbordsgenvägarna och lägga till nya för QGIS-funktioner.

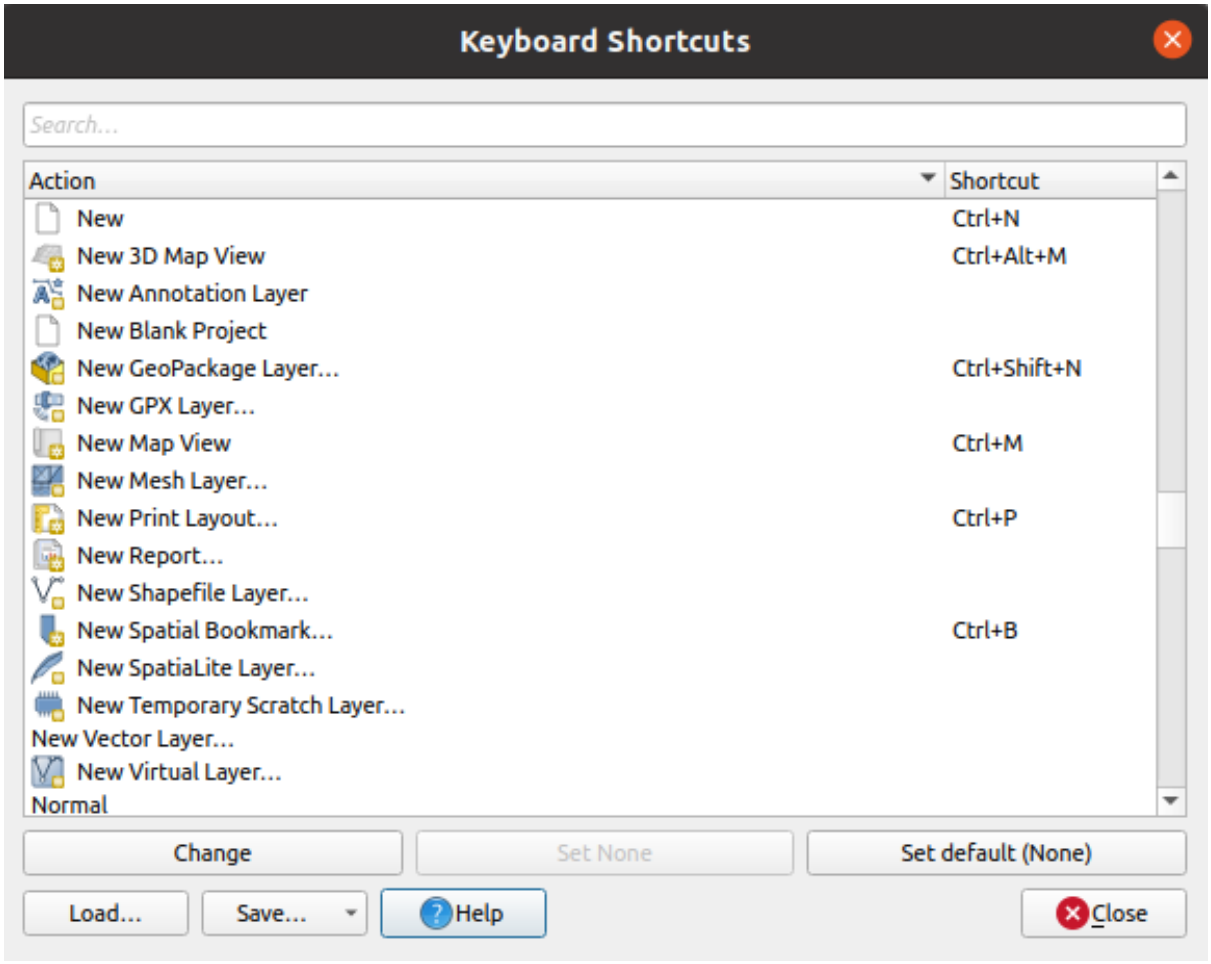


Fig. 5.45: Definiera alternativ för genvägar

Konfigurationen är mycket enkel. Använd sökrutan längst upp i dialogrutan för att hitta en viss åtgärd, markera den i listan och klicka på :

- *Ändra* och tryck på den nya kombinationen som du vill tilldela som ny genväg
- *Set None* för att ta bort alla tilldelade genvägar
- eller *Set Default* för att återställa genvägen till dess ursprungliga och förvalda värde.

Fortsätt enligt ovan för alla andra verktyg som du vill anpassa. När du är klar med konfigurationen är det bara att *Close* dialogrutan för att ändringarna ska tillämpas. Du kan också *Spara* ändringarna antingen som en `.XML`-fil med endast användarens genvägar eller med alla genvägar eller som en `.PDF`-fil med alla genvägar och *Ladda* in dem i en annan QGIS-installation.

5.6 Kör QGIS med avancerade inställningar

5.6.1 Kommandorad och miljövariabler

Vi har sett att *starta QGIS* görs på samma sätt som för alla andra program på ditt operativsystem. QGIS tillhandahåller kommandoradsalternativ för mer avancerade användningsfall (i vissa fall kan du använda en miljövariabel istället för kommandoradsalternativet). För att få en lista över alternativen, ange `qgis --help` på kommandoraden, som returnerar:

```
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
```

Tips

Exempel på användning av kommandoradsargument

Du kan starta QGIS genom att ange en eller flera datafiler på kommandoraden. Om du till exempel befinner dig i katalogen `qgis_sample_data` kan du starta QGIS med ett vektorlager och en rasterfil som ska läsas in vid start med följande kommando: `qgis ./raster/landcover.img ./gml/lakes.gml`

`--version`

Detta alternativ returnerar QGIS versionsinformation.

`--snapshot`

Med det här alternativet kan du skapa en ögonblicksbild i PNG-format från den aktuella vyn. Det här är praktiskt när du har många projekt och vill skapa ögonblicksbilder från dina data, eller när du behöver skapa ögonblicksbilder av samma projekt med uppdaterade data.

För närvarande genereras en PNG-fil med 800x600 pixlar. Storleken kan justeras med hjälp av argumenten `--width` och `--height`. Filnamnet kan läggas till efter `--snapshot`. Till exempel:

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

`--width`

Detta alternativ anger bredden på den ögonblicksbild som ska skickas ut (används med `--snapshot`).

`--height`

Detta alternativ anger höjden på den ögonblicksbild som ska sändas ut (används med `--snapshot`).

`--lang`

QGIS väljer rätt lokalisering baserat på din lokala språkversion. Om du vill ändra språk kan du ange en språkkod. Till exempel, `qgis --lang it` startar QGIS med italiensk lokalisering.

`--project`

Det är också möjligt att starta QGIS med en befintlig projektfil. Lägg bara till kommandoradsalternativet `--project` följt av ditt projektnamn så öppnas QGIS med alla lager i den angivna filen laddade.

`--extent`

Använd det här alternativet om du vill börja med en specifik kartutbredning. Du måste lägga till avgränsningsrutan för din utsträckning i följande ordning, åtskilda med ett kommatecken:

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

Detta alternativ är förmodligen mer meningsfullt i kombination med alternativet `--project` för att öppna ett specifikt projekt i önskad omfattning.

--nologo

Detta alternativ döljer sprutskärmen när du startar QGIS.

--noversioncheck

Hoppa över att söka efter en ny version av QGIS vid start.

--noplugins

Om du har problem med tillägg vid uppstart kan du undvika att ladda dem vid uppstart med det här alternativet. De kommer fortfarande att vara tillgängliga från Tillägg Manager efteråt.

--nocustomization

Om du använder det här alternativet kommer inte någon befintlig *anpassning av grafiskt gränssnitt* att tillämpas vid start. Detta innebär att alla dolda knappar, menyalternativ, verktygsfält och så vidare kommer att visas när QGIS startas. Detta är inte en permanent förändring. Anpassningen kommer att tillämpas igen om QGIS startas utan det här alternativet.

Det här alternativet är användbart för att tillfälligt ge åtkomst till verktyg som har tagits bort genom anpassning.

--skipbadlayers

Med detta alternativ kan du undvika att QGIS visar dialogrutan *Handle Unavailable Layers* vid start. Projektfilen laddas, med saknade lager som hålls otillgängliga. Mer information om ämnet finns på *Hantering av brutna filsökvägar*.

--customizationfile

Med det här alternativet kan du definiera en fil för anpassning av användargränssnittet som ska användas vid start.

--globalsettingsfile

Motsvarande miljövariabel är `QGIS_GLOBAL_SETTINGS_FILE`.

Med det här alternativet kan du ange sökvägen till en fil med globala inställningar (`.ini`), även kallad standardinställningar. Inställningarna i den angivna filen ersätter de ursprungliga inline-standardinställningarna, men användarprofilernas inställningar kommer att ställas in ovanpå dessa.

QGIS letar efter filen med globala standardinställningar i följande ordning och endast den först hittade filen används:

- sökväg som anges av kommandoradsparametern
- sökväg som definieras av miljövariabeln
- appDataLocation-mappen, där beständiga programdata kan lagras; den hanteras av användaren eller systemadministratören och berörs inte av installationsprogrammet och kräver ingen ytterligare installation som att skicka kommandoradsparametrar eller miljövariabler för inställningar. Beroende på operativsystem är det:
  - `$HOME/.local/share/QGIS/QGIS3/`
  - `C:\Users\<username>\%AppData%\Roaming\QGIS\QGIS3\`
  - `$HOME/Library/Application Support/QGIS/QGIS3/`
- installationskatalogen, dvs. `din_QGIS_package_path/resources/qgis_global_settings.ini`.

För närvarande finns det inget sätt att ange en fil att skriva inställningar till; därför kan du skapa en kopia av en originalinställningsfil, byta namn på den och anpassa den.

Genom att ange filsökvägen för `qgis_global_setting.ini` till en delad mapp i nätverket kan en systemadministratör ändra globala inställningar och standardvärden i flera maskiner genom att endast redigera en fil.

`--authdbdirectory`

Det här alternativet liknar `--globalsettingsfile`, men definierar sökvägen till den katalog där autentiseringsdatabasen ska lagras och laddas.

`--code`

Detta alternativ kan användas för att köra en given python-fil direkt efter att QGIS har startat.

Till exempel, när du har en pythonfil med namnet `load_alaska.py` med följande innehåll:

```
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)
```

Förutsatt att du befinner dig i den katalog där filen `load_alaska.py` finns, kan du starta QGIS, ladda rasterfilen `landcover.img` och ge lagret namnet ”Alaska” med följande kommando:

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

`--py-args`

Detta alternativ gör det möjligt att skicka argument till skriptet som tillhandahålls via parametern *–code* och mer allmänt till varje pythonkörning. Efter `--py-args` är det inte möjligt att skicka någon annan parameter än – till QGIS. Alla argument efter `--py-args` till `--` överförs till python-tolken och ignoreras av QGIS.

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

I ovanstående kod kommer `test.py` att ha detta innehåll i `sys.argv`: [’/home/user/test.py’, ’–specialScriptArguments’, ’a text arg’, ’and another arg’]. `layer1` och `layer2` kommer normalt att hanteras av QGIS som lager att ladda.

`--defaultui`

Vid laddning återställs **permanent** användargränssnittet (UI) till standardinställningarna. Detta alternativ återställer panelernas och verktygsfältets synlighet, position och storlek. Om det inte ändras igen kommer standardinställningarna för användargränssnittet att användas i följande sessioner.

Observera att det här alternativet inte har någon effekt på *anpassning av grafiskt gränssnitt*. Objekt som döljs av GUI-anpassning (t.ex. statusfältet) kommer att förbli dolda även om alternativet `--defaultui` används. Se även alternativet `--nocustomization`.

--hide-browser

Vid laddning döljs *Browser*-panelen från användargränssnittet. Panelen kan aktiveras genom att högerklicka på ett utrymme i verktygsfälten eller genom att använda *Visa* ► *Paneler* (*Inställningar* ► *Paneler* i Linux KDE).

Om den inte aktiveras igen kommer panelen Browser att förbli dold i följande sessioner.

--dxf-*

Dessa alternativ kan användas för att exportera ett QGIS-projekt till en DXF-fil. Flera alternativ är tillgängliga:

- *–dxf-export*: DXF-filnamnet som lagren ska exporteras till;
- *–dxf-extent*: omfattningen av den slutliga DXF-filen;
- *–dxf-symbology-mode*: flera värden kan användas här: `none` (ingen symbologi), `symbollayer` (symbolskiktssymbologi), `feature` (funktionssymbologi);
- *–dxf-scale-denom*: skalans nämnare för symbologin;
- *–dxf-encoding*: filens kodning;
- *–dxf-map-theme*: välj ett *mapptema* från konfigurationen av lagerträdet.

--take-screenshots

Tar skärmdumpar för användardokumentationen. Kan användas tillsammans med `--screenshots-categories` för att filtrera vilka kategorier/avsnitt av dokumentationen som skärmdumpar ska skapas för (se QgsAppScreenShots::Categories).

--profile

Laddar QGIS med hjälp av en specifik profil från användarens profilmapp. Om den namngivna profilen inte finns kommer den att skapas. Det här alternativet har företräde framför *inställning för start av användarprofil*.

--profiles-path

Med det här alternativet kan du välja en sökväg för att ladda och spara profilerna (användarinställningar). Profilerna skapas i mappen `{path}\profiles`, som innehåller inställningar, installerade tillägg, bearbetningsmodeller och skript osv.

Med det här alternativet kan du t.ex. spara alla dina tillägg och inställningar på ett USB-minne eller dela inställningarna mellan olika datorer med hjälp av en fildelningstjänst.

Motsvarande miljövariabel är `QGIS_CUSTOM_CONFIG_PATH`.

--version-migration

Om inställningar från en äldre version hittas (t.ex. mappen `.qgis2` från QGIS 2.18), importerar detta alternativ dem till standardprofilen för QGIS.

--openclprogramfolder

Med det här alternativet kan du ange en alternativ sökväg för dina OpenCL-program. Detta är användbart för utvecklare när de testar nya versioner av programmen utan att behöva ersätta de befintliga.

Motsvarande miljövariabel är `QGIS_OPENCL_PROGRAM_FOLDER`.

--

Detta alternativ gör det möjligt att skicka följande argument som filer att öppna i QGIS. De kan vara raster- eller vektorfiler som stöds, QGIS-skiktdefinitionsfiler eller projektfiler.

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

Ovanstående kod startar QGIS, öppnar filen `project` och laddar in `layer1` i den.

5.6.2 Distribuera QGIS inom en organisation

Om du behöver distribuera QGIS inom en organisation med en anpassad konfigurationsfil måste du först kopiera/klistra in innehållet i standardinställningsfilen som finns i `your_QGIS_package_path/resources/qgis_global_settings.ini`. Den här filen innehåller redan några standardavsnitt som identifieras av ett block som börjar med `[]`. Vi rekommenderar att du behåller dessa standardvärden och lägger till dina egna avsnitt längst ned i filen. Om ett avsnitt dupliceras i filen kommer QGIS att ta det sista avsnittet uppifrån och ned.

Du kan ändra `allowVersionCheck=false` för att inaktivera QGIS versionskontroll.

Om du inte vill visa migreringsfönstret efter en nyinstallation behöver du följande avsnitt:

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

Om du vill lägga till en anpassad variabel i det globala omfånget:

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

Många inställningar kan göras med hjälp av `.INI`-filen, t.ex. WMS/WMTS, PostGIS-anslutningar, proxyinställningar, karttips, standardvärden för algoritmparametrar… Om du t.ex. vill konfigurera standardvärden för hela organisationen för vissa algoritmer skulle det vara praktiskt om du kunde konfigurera det i den förberedda INI-filen för alla användare. Om användarna sedan öppnar verktygslådan för att köra en algoritm kommer standardvärdena att finnas där. För att ställa in standardvärdet, t.ex. för parametern `GRID_SIZE` för algoritmen `native:intersection`, kan du skriva följande i filen `.INI`:

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

Alla inställningar kan inte förberedas från QGIS grafiska gränssnitt, men för att upptäcka möjligheterna med inställningarna i filen `.INI` föreslår vi att du ställer in den konfiguration du vill ha i QGIS Desktop och sedan söker efter den i filen `.INI` som finns i din profil med hjälp av en textredigerare.

Slutligen måste du ställa in miljövariabeln `QGIS_GLOBAL_SETTINGS_FILE` till sökvägen för din anpassade fil.

Dessutom kan du distribuera filer som Python-makron, färgpaletter, layoutmallar, projektmallar … antingen i QGIS-systemkatalogen eller i QGIS-användarprofilen.

- Layoutmallar måste distribueras i katalogen `composer_templates`.
- Projektmallar måste distribueras i katalogen `project_templates`.
- Anpassade Python-makron måste distribueras i katalogen `python`.

KAPITEL 6

Arbeta med projektioner

Ett koordinatreferenssystem, eller CRS, är en metod för att associera numeriska koordinater med en position på jordens yta. QGIS har stöd för cirka 7 000 standard-CRS:er, alla med olika användningsområden, för- och nackdelar! Att välja ett lämpligt referenssystem för dina QGIS-projekt och data kan vara en komplex uppgift, men lyckligtvis hjälper QGIS dig att vägleda dig genom detta val och gör arbetet med olika CRS så transparent och korrekt som möjligt.

6.1 Översikt över projektionsstöd

QGIS har stöd för cirka 7 000 kända CRS. Dessa standard-CRS baseras på de som definieras av European Petroleum Search Group (EPSG) och Institut Geographique National de France (IGNF), och görs tillgängliga i QGIS genom det underliggande projektionsbiblioteket ”Proj”. Vanligtvis identifieras dessa standardprojektioner genom användning av en kombination av auktoritet och kod, där auktoriteten är ett organisationsnamn som ”EPSG” eller ”IGNF” och koden är ett unikt nummer som är associerat med ett specifikt CRS. Exempelvis är det vanliga CRS-systemet för latitud/longitud WGS 84 känt genom identifieraren ”EPSG:4326” och CRS-systemet för webbkartor är ”EPSG:3857”.

Anpassade, användarskapade CRS lagras i en CRS-databas för användare. Se avsnitt *Anpassat referenssystem för koordinater* för information om hur du hanterar dina anpassade koordinatreferenssystem.

6.2 Referenssystem för lagerkoordinater

För att korrekt kunna projicera data till ett specifikt CRS-mål måste antingen dina data innehålla information om dess koordinatreferenssystem eller så måste du manuellt tilldela lagret rätt CRS. För PostGIS-lager använder QGIS den identifierare för spatial referens som angavs när PostGIS-lagret skapades. För data som stöds av GDAL förlitar sig QGIS på att det finns ett erkänt sätt att ange CRS. För Shapefile-formatet är detta till exempel en fil som innehåller en ESRI Well-Known Text (WKT)-representation av lagrets CRS. Denna projektionsfil har samma basnamn som `.shp`-filen och ett `.prj`-tillägg. Till exempel skulle `alaska.shp` ha en motsvarande projektionsfil med namnet `alaska.prj`.

När ett lager laddas in i QGIS försöker QGIS att automatiskt bestämma rätt CRS för lagret. I vissa fall är detta inte möjligt, t.ex. när ett lager har tillhandahållits utan att denna information har sparats. Du kan konfigurera QGIS beteende när det inte automatiskt kan fastställa rätt CRS för ett lager:

1. Öppna *Inställningar* ► *Alternativ…* ► *CRS*

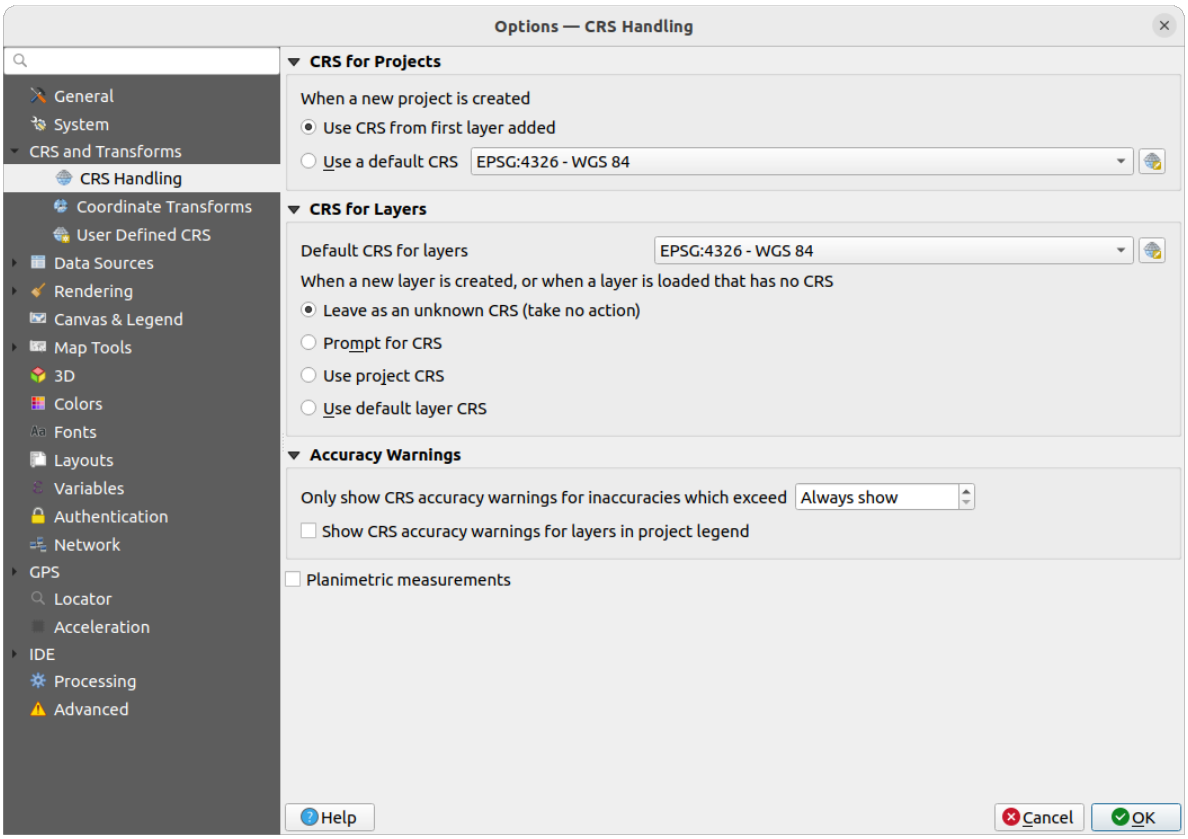


Fig. 6.1: Fliken CRS i dialogrutan Alternativ i QGIS

2. Under gruppen *CRS för lager*, ställ in åtgärden att göra *när ett nytt lager skapas, eller när ett lager laddas som inte har någon CRS*. Ett av följande:

   - *Lämna som okänd CRS (ingen åtgärd)*: det kommer inte att finnas någon uppmaning att välja en CRS när ett lager utan CRS laddas, utan valet av CRS skjuts upp till en senare tidpunkt. Praktiskt när man laddar många lager samtidigt. Sådana lager kommer att kunna identifieras i *Layers*-panelen genom ikonen bredvid dem. De kommer också att vara utan referens, med koordinater från lagret som behandlas som rent numeriska, icke-jordiska värden, dvs. samma beteende som alla lager får när *ett projekt är inställt på att inte ha någon CRS*.
   - *Prompt for CRS*: du uppmanas att manuellt välja CRS. Det är mycket viktigt att välja rätt, eftersom ett felaktigt val placerar ditt lager i fel position på jordens yta! Ibland beskriver medföljande metadata rätt CRS för ett lager, i andra fall måste du kontakta den ursprungliga författaren till data för att fastställa rätt CRS att använda.
   - *Använda CRS i projektet*
   - *Use default layer CRS*, enligt inställningarna i *Default CRS for layers* ovan.

Tips

Att tilldela samma CRS till flera lager som inte har några CRS eller har fel CRS i en och samma operation:

1. Välj lager i panelen *Layers* (lager)
2. Tryck på `Ctrl+Shift+C`. Du kan också högerklicka på ett av de markerade lagren eller gå till *Layer* ► *Set CRS of layer(s)*
3. Hitta och välj rätt CRS att använda

4. Tryck sedan på *OK*. Du kan bekräfta att den har ställts in korrekt på fliken *Source* i dialogen för lagrets egenskaper.

Observera att om du ändrar CRS i den här inställningen ändras inte den underliggande datakällan på något sätt, utan det ändrar bara hur QGIS tolkar de råa koordinaterna från lagret i det aktuella QGIS-projektet.

6.3 Referenssystem för projektkoordinater

Varje projekt i QGIS har också ett associerat koordinatreferenssystem. Projektets CRS avgör hur data projiceras från dess underliggande råkoordinater till den platta karta som återges i din QGIS-kartkanvas.

QGIS stöder ”on the fly”-transformering av CRS för både raster- och vektordata. Detta innebär att oavsett underliggande CRS för vissa kartlager i ditt projekt kommer de alltid att automatiskt omvandlas till det gemensamma CRS som definierats för ditt projekt. Bakom kulisserna återprojicerar QGIS på ett transparent sätt alla lager som finns i ditt projekt till projektets CRS, så att de alla återges i rätt position i förhållande till varandra!

Det är viktigt att välja ett lämpligt CRS för dina QGIS-projekt. Om du väljer ett olämpligt CRS kan det leda till att dina kartor ser förvrängda ut och dåligt återspeglar de verkliga relativa storlekarna och positionerna för funktioner. När man arbetar i mindre geografiska områden finns det vanligtvis ett antal standard-CRS som används inom ett visst land eller administrativt område. Det är viktigt att undersöka vilka CRS som är lämpliga eller standardval för det område du kartlägger, och se till att ditt QGIS-projekt följer dessa standarder.

Som standard startar QGIS varje nytt projekt med en global standardprojektion. Detta standard-CRS är `EPSG:4326` (även känt som ”WGS 84”), och det är ett globalt latitud/longitudbaserat referenssystem. Detta standard-CRS kan ändras via inställningen *CRS för nya projekt* i fliken *CRS* under *Inställningar* ► *Alternativ…* (se Fig. 6.1). Det finns ett alternativ för att automatiskt ställa in projektets CRS så att det matchar CRS för det första lagret som laddas in i ett nytt projekt, eller så kan du välja ett annat standard-CRS som ska användas för alla nyskapade projekt. Detta val kommer att sparas för användning i efterföljande QGIS-sessioner.

Projektets CRS kan också ställas in via fliken *CRS* i dialogrutan *Projekt* ► *Egenskaper…*. Den kommer också att visas längst ned till höger i QGIS statusfält.

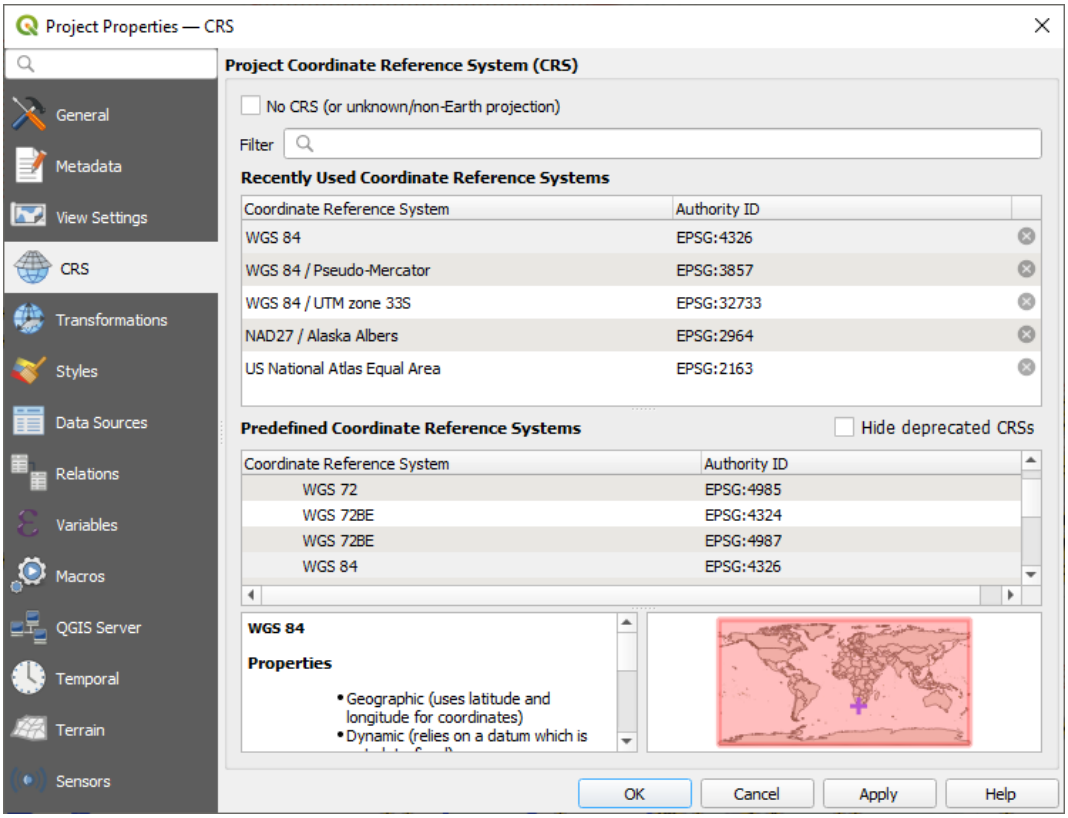


Fig. 6.2: Dialog för projektegenskaper

Tillgängliga alternativ är:

- *Ingen CRS (eller okänd/ej jordprojektion)*: Om du markerar den här inställningen inaktiveras ALL projektionshantering i QGIS-projektet, vilket gör att alla lager och kartkoordinater behandlas som enkla kartesiska 2D-koordinater, utan någon relation till positioner på jordens yta. Den kan användas för att gissa ett lagers CRS (baserat på dess råkoordinater eller när QGIS används för andra ändamål än jorden, t.ex. rollspelskartor, byggnadskartor eller mikroskopiska saker. I det här fallet:
  - Ingen reprojektion görs vid rendering av lagren: objekten ritas bara med sina råkoordinater.
  - Ellipsoiden är utestängd och tvingas till `None/Planimetric`.
  - Enheterna för avstånd och yta samt koordinatvisningen är låsta och tvingade till ”unknown units”; alla mätningar görs i okända kartenheter och ingen konvertering är möjlig.
- eller ett befintligt koordinatreferenssystem som kan vara *geografiskt*, *projicerat* eller *användardefinierat*. En förhandsgranskning av CRS-utbredningen på jorden visas för att hjälpa dig att välja rätt. Lager som läggs till i projektet översätts i realtid till detta CRS för att kunna läggas över dem oavsett deras ursprungliga CRS. Användning av enheter och ellipsoidinställning är tillgänglig och meningsfull och du kan utföra beräkningar i enlighet med detta.

När du väljer ett nytt CRS för ditt QGIS-projekt kommer måttenheterna automatiskt att ändras på fliken *General* i dialogrutan *Projektegenskaper* (*Projekt* ► *Egenskaper…*) för att matcha det valda CRS:et. Till exempel definierar vissa CRS sina koordinater i fot istället för meter, så om du ställer in ditt QGIS-projekt på ett av dessa CRS kommer ditt projekt också att ställa in att mäta med fot som standard.

Tips

Ställa in projektets CRS från ett lager

Du kan tilldela en CRS till projektet med hjälp av en CRS för lager:

1. I panelen *Layers* högerklickar du på det lager där du vill välja CRS
2. Välj *Set project CRS from Layer*.

Projektets CRS omdefinieras med hjälp av lagrets CRS. Kartans omfattning och koordinatvisning uppdateras i enlighet med detta och alla lager i projektet översätts direkt till det nya projekt-CRS:et.

6.4 Väljare för referenssystem för koordinater

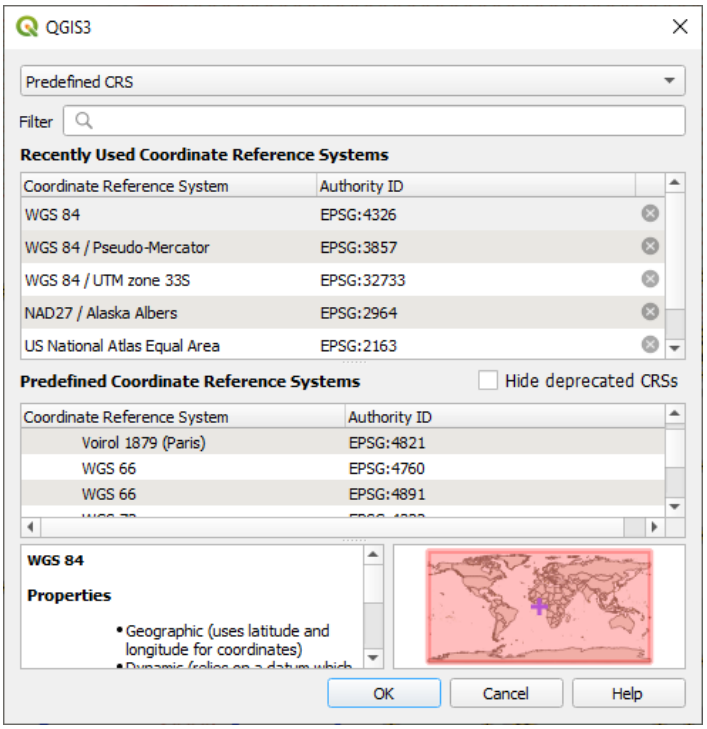


Fig. 6.3: CRS-väljare

Denna dialog hjälper dig att tilldela ett koordinatreferenssystem till ett projekt eller ett lager, förutsatt att det finns en uppsättning projektionsdatabaser. Objekt i dialogrutan är:

- **Filter**: Om du känner till EPSG-koden, identifieraren eller namnet för ett koordinatreferenssystem kan du använda sökfunktionen för att hitta det. Ange EPSG-koden, identifieraren eller namnet.
- **Nyligen använda koordinatreferenssystem**: Om du har vissa CRS som du ofta använder i ditt dagliga GIS-arbete visas dessa i den här listan. Klicka på ett av dessa objekt för att välja det tillhörande CRS.
- **Världens koordinatreferenssystem**: Detta är en lista över alla CRS som stöds av QGIS, inklusive geografiska, projicerade och anpassade koordinatreferenssystem. För att definiera ett CRS, välj det från listan genom att expandera lämplig nod och välja CRS. Det aktiva CRS är förvalt.
- **PROJ text**: Detta är den CRS-sträng som används av PROJ projektionsmotorn. Denna text är skrivskyddad och tillhandahålls för informationsändamål.

Om du vill ta bort en nyligen använd CRS från listan kan du göra det på flera sätt:

- Tryck på knappen Remove from recently used CRS på höger sida.
- Tryck på `Del` när du har valt den CRS som du vill ta bort.

- Högerklicka på en använd CRS och välj *Remove selected CRS from recently used CRS* för att ta bort den valda CRS:en eller *Clear all recently used CRS* för att ta bort alla använda CRS:er.

CRS-väljaren visar också en grov förhandsgranskning av det geografiska område där ett valt CRS är giltigt för användning. Många CRS är endast avsedda att användas inom små geografiska områden och du bör inte använda dem utanför det område de är avsedda för. Förhandsgranskningskartan visar ett ungefärligt användningsområde när ett CRS väljs från listan. Dessutom visar förhandsgranskningskartan en indikator på den aktuella omfattningen av huvudkartan.

6.5 Anpassat referenssystem för koordinater

Om QGIS inte tillhandahåller det koordinatreferenssystem som du behöver kan du definiera ett eget CRS. För att definiera ett CRS, välj *Custom CRS…* från menyn *Settings*. Anpassade CRS lagras i din QGIS-användardatabas. Förutom dina anpassade CRS innehåller denna databas även dina spatiala bokmärken och andra anpassade data.

För att kunna definiera ett anpassat CRS i QGIS krävs en god förståelse av PROJ projektionsbibliotek. Du kan börja med att läsa ”Cartographic Projection Procedures for the UNIX Environment - A User’s Manual” av Gerald I. Evenden, U.S. Geological Survey Open-File Report 90-284, 1990 (finns på https://pubs.usgs.gov/of/1990/of90-284/ofr90-284.pdf).

Denna manual beskriver användningen av `proj` och relaterade kommandoradsverktyg. De kartografiska parametrar som används med `proj` beskrivs i användarhandboken och är desamma som de som används av QGIS.

Dialogrutan *Custom Coordinate Reference System Definition* kräver endast två parametrar för att definiera ett CRS för användaren:

1. Ett beskrivande namn
2. De kartografiska parametrarna i PROJ- eller WKT-format

Så här skapar du en ny CRS:

1. Klicka på knappen Lägg till ny CRS
2. Ange ett beskrivande namn
3. Välj format: det kan vara *Proj String* eller *WKT*
4. Lägg till CRS *Parameters*.

> **Observera**
>
> **Företräcker att lagra CRS-definitionen i WKT-format**
>
> Även om både formaten `Proj String` och `WKT` stöds, rekommenderas det starkt att lagra projektionsdefinitioner i WKT-format. Om den tillgängliga definitionen är i proj-format väljer du därför det formatet, anger parametrarna och växlar sedan till WKT-format. QGIS konverterar definitionen till WKT-format som du senare kan spara.

5. Klicka på *Validate* för att testa om CRS-definitionen är en godtagbar projektionsdefinition.

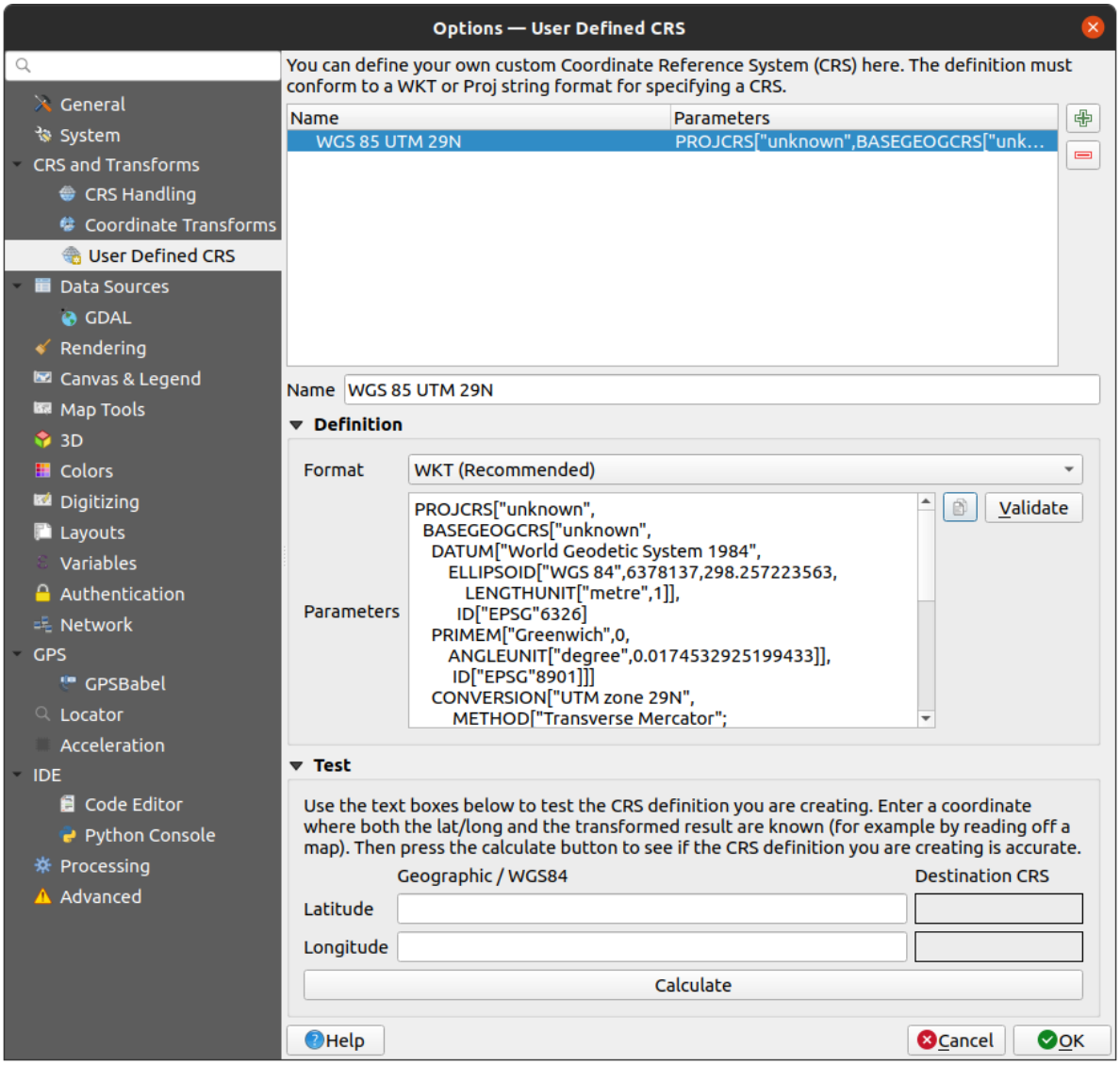


Fig. 6.4: Anpassad CRS-dialog

Du kan testa dina CRS-parametrar för att se om de ger rimliga resultat. Det gör du genom att ange kända värden för latitud och longitud enligt WGS 84 i fälten *North* respektive *East*. Klicka på *Calculate* och jämför resultatet med de kända värdena i ditt koordinatreferenssystem.

6.5.1 Integrera en NTv2-transformation i QGIS

För att integrera en NTv2-transformationsfil i QGIS behöver du ytterligare ett steg:

1. Placera NTv2-filen (.gsb) i CRS/Proj-mappen som QGIS använder (t.ex. `C:\OSGeo4W64\share\proj` för Windows-användare)
2. Lägg till **nadgrids** (`+nadgrids=namnpåfilen.gsb`) till Proj-definitionen i fältet *Parameters* i *Custom Coordinate Reference System Definition* (*Settings* ► *Custom Projections*…).

Fig. 6.5: Ställa in en NTv2-transformation

6.6 Datumtransformationer

I QGIS är CRS-transformering ”on-the-fly” aktiverad som standard, vilket innebär att när du använder lager med olika koordinatsystem så återprojicerar QGIS dem på ett transparent sätt till projektets CRS. För vissa CRS finns det ett antal möjliga transformationer tillgängliga för att återprojicera till projektets CRS!

Som standard kommer QGIS att försöka använda den mest exakta tillgängliga transformationen. I vissa fall är detta dock inte möjligt, t.ex. när det krävs ytterligare stödfiler för att använda en transformation. När en mer exakt omvandling finns tillgänglig, men inte kan användas för närvarande, kommer QGIS att visa ett informativt varningsmeddelande som informerar dig om den mer exakta omvandlingen och hur du aktiverar den på ditt system. Vanligtvis kräver detta nedladdning av ett externt paket med stödfiler för transformation och extrahering av dessa till mappen `proj` under mappen QGIS *user profile*.

Om så önskas kan QGIS också fråga dig när flera möjliga transformationer kan göras mellan två CRS, och låta dig göra ett välgrundat val av vilken transformation som är lämpligast att använda för dina data.

Denna anpassning görs i flikmenyn *Inställningar* ► *Alternativ* ► *Transformationer* under gruppen *Default datum transformations*:

- using *Ask for datum transformation if several are available*: när det finns mer än en lämplig datumtransformation för en CRS-kombination för källa/destination, öppnas automatiskt en dialogruta där användaren uppmanas att välja vilken av dessa datumtransformationer som ska användas för projektet. Om kryssrutan *Make default* är markerad när en transformation väljs från denna dialog, kommer valet att kommas ihåg och automatiskt tillämpas på alla nyskapade QGIS-projekt.
- eller definiera en lista över lämpliga datumtransformationer som ska användas som standard när ett lager laddas till ett projekt eller när ett lager omprojiceras.

  Använd knappen för att öppna dialogrutan *Select Datum Transformations*. Välj sedan:

  1. Välj lagrets *Source CRS* med hjälp av rullgardinsmenyn eller widgeten Select CRS.
  2. Tillhandahåll *Destination CRS* på samma sätt.
  3. En lista över tillgängliga transformationer från källa till destination visas i tabellen. Om du klickar på en rad visas detaljer om de inställningar som tillämpats och motsvarande noggrannhet och användningsområde för omvandlingen.

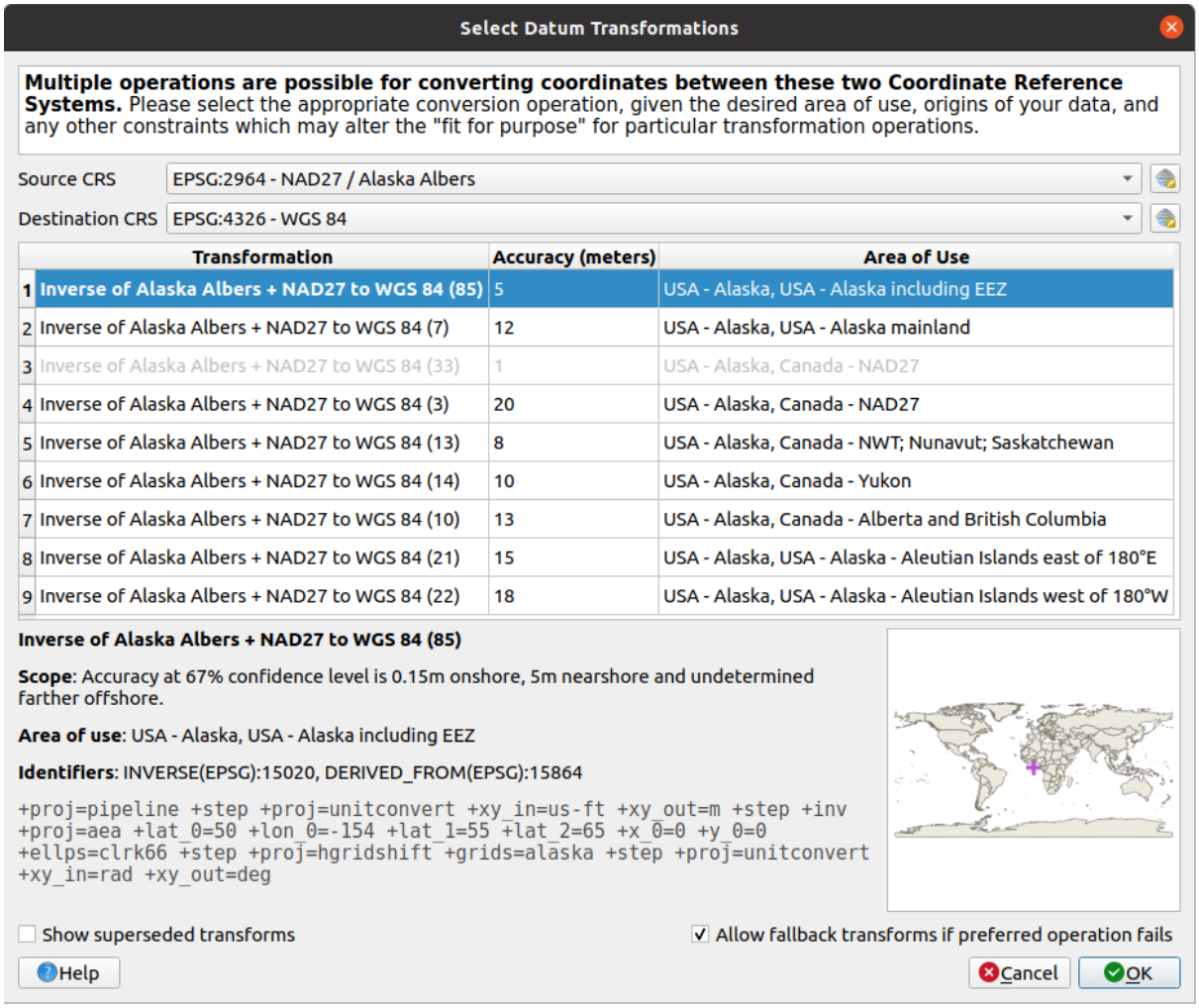


Fig. 6.6: Val av en föredragen standard datumtransformation

I vissa fall kan det hända att en transformation inte är tillgänglig för användning i ditt system. I så fall kommer transformationen fortfarande att visas (gråmarkerad) i listan men kan inte väljas förrän du har installerat det nödvändiga paketet med transformationsstöd. Vanligtvis finns det en knapp för att hämta och installera motsvarande grid, som sedan lagras under mappen `proj` i den aktiva katalogen *user profile*.

4. Hitta din önskade omvandling och välj den
5. Ange om du *Tillåt reservtransformationer om den föredragna operationen misslyckas*
6. Klicka på *OK*.

   En rad läggs till i tabellen under *Default Datum Transformations* med information om *Source CRS*, *Destination CRS*, *Operation* som används för transformationen och om *Allow fallback Transforms* är aktiverad.

Från och med nu använder QGIS automatiskt de valda datumtransformationerna för vidare transformation mellan dessa två CRS tills du tar bort den ( ) från listan eller ändrar posten ( ) i listan.

Datumtransformationer som anges i fliken *Inställningar* ► *Alternativ* ► *Transformationer* kommer att ärvas av alla nya QGIS-projekt som skapas i systemet. Dessutom kan ett visst projekt ha sin egen specifika uppsättning transformationer som specificeras via fliken *CRS* i dialogrutan *Projektegenskaper* (*Projekt* ► *Egenskaper…*). Dessa inställningar gäller endast för det aktuella projektet.

KAPITEL 7

Visualisering av kartor

Att skapa kartor är en av ”affärsidéerna” i QGIS. Kartvyer är ett sätt att visualisera spatiala lager som laddats i projektet, med de olika renderingsinställningar som tillämpas på dem. De kan vara av typen *2D* eller *3D* eller plottade längs en *profillinje*, visa olika skala eller omfattning, eller visa en annan uppsättning av de inlästa lagren tack vare *map themes*.

7.1 2D-kartvy

2D-kartvyn (även kallad **Map canvas**) är den centrala plats där kartor visas. QGIS öppnas som standard med en enda kartvy (kallad `huvudkarta`), som visar lager i 2D och är tätt knuten till panelen *Layers*. Det fönstret återspeglar den rendering (symbologi, märkning, synlighet …) som du har tillämpat på de laddade lagren.

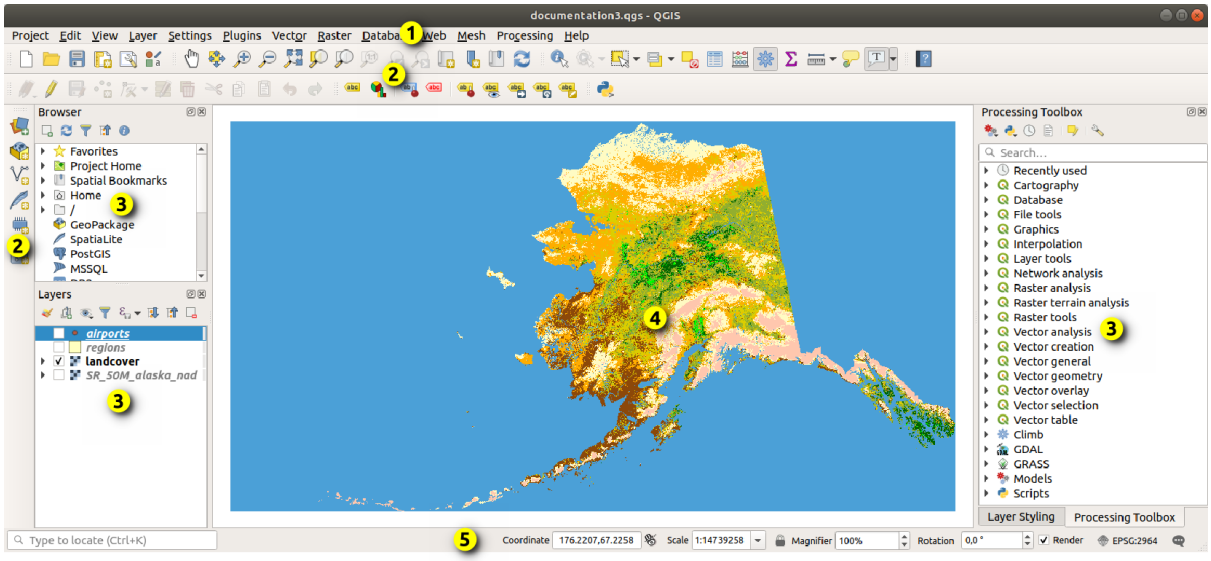


Fig. 7.1: QGIS grafiska gränssnitt

7.1.1 Utforska kartvyn

När du lägger till ett lager (se t.ex. *Öppningsdata*) letar QGIS automatiskt efter dess CRS. Om en annan CRS är inställd som standard för projektet (se *Referenssystem för projektkoordinater*) översätts lagrets omfattning ”i farten” till den CRS:en och kartvyn zoomas till den omfattningen om du börjar med ett tomt QGIS-projekt. Om det redan finns lager i projektet utförs ingen storleksändring av kartvyn, så endast funktioner som faller inom den aktuella kartvyns utsträckning kommer att vara synliga.

Klicka på kartvyn och du bör kunna interagera med den genom att panorera eller zooma till olika delar av kartan. Dedikerade verktyg finns i *Navigation Toolbar* och i *View*-menyn, med praktiska genvägar från tangentbordet eller musknapparna.

Table 7.1: Navigationsverktyg för kartor

Verktyg	Användning
Pan Map	• Ett vänsterklick: kartan centreras på den klickade punkten, i samma skala • Håll ned vänster musknapp och dra i kartbilden.
Zooma in	• Enbart vänsterklick: kartan centreras på den klickade punkten, medan skalan fördubblas • Dra en rektangel på kartbilden med vänster musknapp för att zooma in på ett område. • Håll ned `Alt` för att växla till verktyget Zooma ut.
Zooma ut	• Enbart vänsterklick: kartan centreras på den klickade punkten, medan skalan halveras • Dra en rektangel på kartbilden med vänster musknapp för att zooma ut från ett område. • Håll ned `Alt` för att växla till verktyget Zooma in.
Pan Map till urval	Panorera kartan till de valda objekten i alla valda lager i panelen *Layers*.
Zoom till val	Zooma till de valda objekten i alla valda lager i panelen *Layers*. *Finns även i lagrets kontextuella meny*
Zooma till lager`	Zooma till omfattningen av alla markerade lager i panelen *Layers*. *Finns även i lagrets kontextuella meny*
Zoom Full	Zooma till omfattningen av alla lager i projektet eller till *projektets fulla omfattning*.
Zoom Sista	Zooma kartan till den tidigare omfattningen i historiken.
Zoom Nästa	Zooma kartan till nästa utbredning i historien.
Zoom till ursprunglig upplösning	Zooma kartan till en nivå där en pixel i det aktiva rasterlagret täcker en skärmpixel. *Finns även i lagrets kontextuella meny*
Musens hjul	• Panorera över kartan: Håll mushjulet nedtryckt och dra det. • Zooma in: Rulla mushjulet för att zooma in eller zooma ut. Om `Ctrl`-tangenten hålls nedtryckt medan mushjulet rullar får du en finare zoom. • Tryck på bakåt- eller framåtknappen för att bläddra i zoomhistoriken för kartbilden.

continues on next page

Table 7.1 – fortsättning från föregående sida

Verktyg	Användning
Tangentbord	• Panorera kartan: Håll ned `Space`-tangenten och flytta musen. Tryck på piltangenterna för att panorera uppåt, nedåt, vänster och höger. • Zooma in: Tryck på `PgUp` eller `Ctrl++` • Zooma ut: Tryck på `PgDown` eller `Ctrl+-` • Zooma till område: När vissa kartverktyg är aktiva (Identifiera, Mäta…) kan du hålla ned `Shift` och dra en rektangel på kartan för att zooma till det området. Inte kompatibel med aktiva markerings- eller redigeringsverktyg.

Högerklicka på kartan och du bör kunna *Kopiera koordinater* för den klickade punkten i kartans CRS, i WGS84 eller i en anpassad CRS. Den kopierade informationen kan sedan klistras in i ett uttryck, ett skript, en textredigerare eller ett kalkylblad…

7.1.2 Styrning av kartrendering

Som standard återger QGIS alla synliga lager när kartbilden uppdateras. De händelser som utlöser en uppdatering av kartbilden inkluderar:

- ändra synligheten för ett lager
- ändra symbologi för ett synligt lager
- lägga till ett lager
- panorering eller zoomning
- ändra storlek på QGIS-fönstret

Med QGIS kan du styra renderingsprocessen på flera olika sätt.

- på *global nivå*
- per lager, t.ex. med hjälp av *skalaberoende rendering*
- eller med särskilda verktyg i det grafiska användargränssnittet.

För att stoppa kartritningen, tryck på `Esc`-tangenten. Detta stoppar uppdateringen av kartbilden och lämnar kartan delvis ritad. Det kan dock ta lite tid efter att du tryckt på `Esc` innan kartritningen stoppas.

Om du vill avbryta renderingen klickar du på kryssrutan *Rendera* i det nedre högra hörnet av statusfältet. När kryssrutan *Rendera* är avmarkerad, ritar QGIS inte om duken som svar på någon av de vanliga utlösarna som nämnts tidigare. Exempel på när du kanske vill avbryta rendering inkluderar:

- lägga till många lager och symbolisera dem innan de ritas
- lägga till ett eller flera stora lager och ställa in skalberoende innan du ritar
- lägga till ett eller flera stora lager och zooma till en specifik vy innan du ritar
- någon kombination av ovanstående

Om du markerar kryssrutan *Rendera* aktiveras rendering och en omedelbar uppdatering av kartbilden görs.

7.1.3 Tidsbaserad kontroll på kartbilden

QGIS kan hantera temporal kontroll på laddade lager, dvs. ändra renderingen av kartans canvas baserat på en tidsvariation. För att uppnå detta behöver du:

1. Lager som har dynamiska temporala egenskaper inställda. QGIS stöder temporal kontroll för olika dataleverantörer, med anpassade inställningar. Det handlar främst om att ställa in det tidsintervall inom vilket lagret ska visas:
   - *raster layers*: kontrollerar om lagret ska visas eller inte.
   - *WMTS-lager*: styr om data ska återges baserat på ett statiskt tidsintervall eller efter ett dynamiskt tidsintervall
   - *vektorlager*: funktioner filtreras baserat på tidsvärden som är associerade med deras attribut
   - *mesh layers*: visar dynamiskt de aktiva datasetgruppernas värden

   När dynamiska temporala alternativ är aktiverade för ett lager visas en ikon bredvid lagret i panelen *Lager* för att påminna dig om att lagret är temporalt kontrollerat. Klicka på ikonen för att uppdatera de temporala inställningarna.
2. Aktivera den temporala navigeringen av kartbilden med hjälp av *Temporal controller panel*. Panelen är aktiverad:
   - med hjälp av ikonen Temporal controller panel i verktygsfältet *Map Navigation*
   - eller från menyn *Visa* ► *Paneler* ► *Temporal controller panel*
   - eller från menyn *Visa* ► *Datafiltrering* ► *Temporal kontrollpanel*

Den temporala kontrollpanelen

Panelen *Temporal controller* har följande lägen:

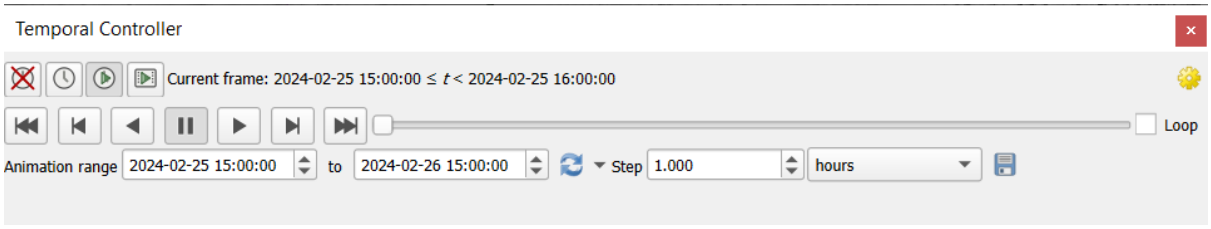


Fig. 7.2: Temporal Controller Panel i navigeringsläge

- Stäng av temporal navigering: alla temporala inställningar inaktiveras och synliga lager återges som vanligt
- :sup:`Fixed range temporal navigation`: ett tidsintervall anges och endast lager (eller funktioner) vars tidsintervall överlappar detta intervall visas på kartan.
- Animerad temporal navigering: ett tidsintervall anges, delas upp i steg och endast lager (eller funktioner) vars tidsintervall överlappar varje ram visas på kartan
- Animerad film: ett totalt antal bildrutor anges för animeringen, där kartinställningarna flyttas fram i varje steg och visar utvecklingen bildruta för bildruta utan någon tidsbaserad filtrering av data.
- Inställningar för allmän kontroll av animeringen
  - *Frames rate*: antal steg som visas per sekund

- *Kumulativt intervall*: alla animationsrutor kommer att ha samma startdatum och -tid men olika slutdatum och -tider. Detta är användbart om du vill ackumulera data i din temporala visualisering i stället för att visa ett ”rörligt tidsfönster” över dina data.

Animera en temporal navigering

En animation baseras på en varierande uppsättning synliga lager vid vissa tidpunkter inom ett tidsintervall. Så här skapar du en temporal animering:

1. Slå på Animerad temporal navigering, visar widgeten för animationsspelaren
2. Ange det *Tidsintervall* som ska beaktas. Med hjälp av -knappen kan detta definieras som:
   - *Set to full range* av alla tidsaktiverade lager
   - *Sätt till förinställt projektintervall* enligt definitionen i *projektegenskaper*
   - *Set to single layer's range* hämtat från ett tidsaktiverat lager
3. Fyll i tiden *Step* för att dela upp tidsintervallet. Olika enheter stöds, från `sekunder` till `hundratal`. Alternativet `källtidstämplar` finns också tillgängligt som steg: när det väljs får det den temporala navigeringen att stega mellan alla tillgängliga tidsintervall från lager i projektet. Det är användbart när ett projekt innehåller lager med icke sammanhängande tillgängliga tider, t.ex. en WMS-T-tjänst som tillhandahåller bilder som är tillgängliga vid oregelbundna datum. Med det här alternativet kan du bara stega mellan tidsintervall där nästa tillgängliga bild visas.
4. Klicka på knappen för att förhandsgranska animeringen. QGIS kommer att generera scener med hjälp av de lager som renderas vid de inställda tiderna. Lagrens visning beror på om de överlappar någon enskild tidsram.

   Animationen kan också förhandsgranskas genom att flytta tidsreglaget. Om du markerar kryssrutan *Loop* kommer animationen att köras upprepade gånger medan du klickar på för att stoppa en pågående animation. En komplett uppsättning videospelarknappar finns tillgänglig.

   Horisontell skrollning med mushjulet (där det stöds) med markören på kartbilden gör det också möjligt att navigera, eller ”skrubba”, det temporala navigeringsreglaget framåt och bakåt.
5. Klicka på knappen Export animation om du vill skapa en serie bilder som representerar scenen. De kan senare kombineras i ett videoredigeringsprogram:

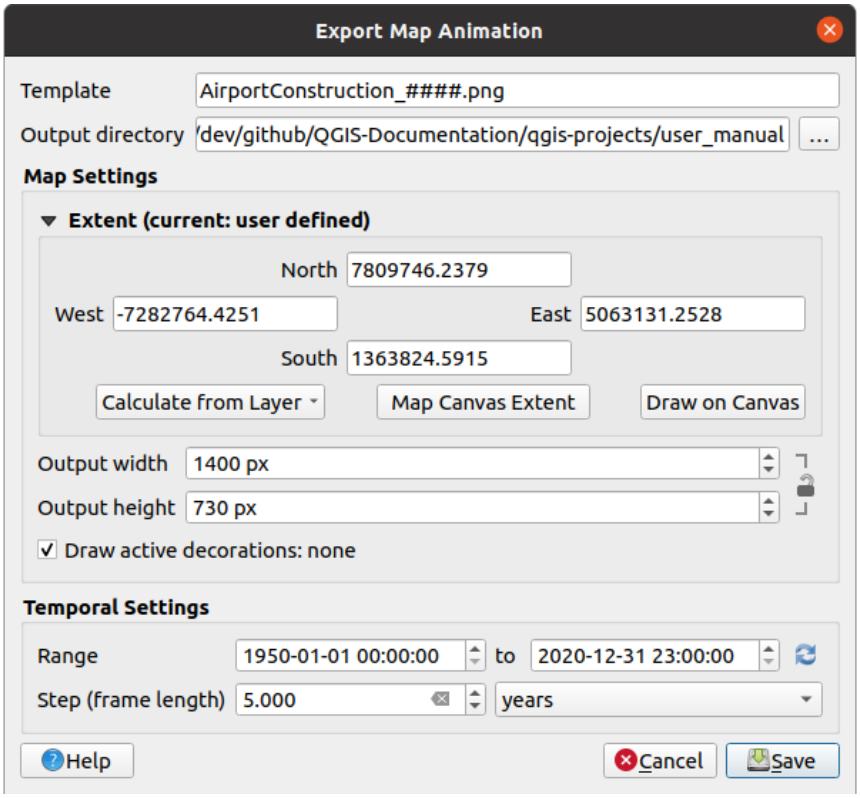


Fig. 7.3: Exportera animationsscener från Map Canvas till bilder

- Filnamnet *Template*: `####` ersätts med bildsekvensnummer
- Den *Utmatningskatalog*
- Under *Kartinställningar* kan du:
  - omdefiniera *spatial utsträckning* för att använda
  - kontrollera bildens *Resolution* (*Output width* och *Output height*)
  - *Draw active decorations*: om aktiva *dekorationer* ska sparas i utdata
- Under *Temporal settings* kan du omdefiniera:
  - tiden *Range* för animeringen
  - *Steg (ramlängd)* i den enhet som du väljer

7.1.4 Panelen för höjdkontroll

Elevation Controller låter dig hantera data för höjd z-område i 2D-kartor. Detta alternativ stöder för närvarande punktmolnslager och raster-DEM:er. Använd panelen för att visualisera och interagera med höjddata i kartbilden.

Aktivera panelen *Elevation Controller* i kartbilden genom att klicka på *Data Filtering* i menyn *Visa*. *Elevation Controller* visas som ett skjutreglage för intervall på vänster sida av kartbilden. Längst upp på skjutreglaget finns menyn Inställningar där du kan:

- Ställ in värdet *Fixed Range Size* för att låsa höjdintervallet till en viss storlek. Den här inställningen sparas i projektet och behålls under alla sessioner, vilket gör att höjdfiltret blir konsekvent. Använd alternativet *Invert Direction* för att vända höjdfilterreglaget, så att det rör sig från höga värden längst ned till låga värden längst upp.

- *Set Elevation Range* för att öppna projektets höjdflik, där du kan ställa in dedikerade nedre och övre gränser för höjdintervallet
- *Disable Elevation Filter* för att ta bort höjdfiltret från kartbilden

7.1.5 Lägga till bokmärken för extenter på kartan

Med Spatial Bookmarks kan du ”bokmärka” en geografisk plats och återvända till den senare. Som standard sparas bokmärkena i användarens profil (som *User Bookmarks*), vilket innebär att de är tillgängliga i alla projekt som användaren öppnar. De kan också sparas för ett enskilt projekt (med namnet *Project Bookmarks*) och lagras i projektfilen, vilket kan vara till hjälp om projektet ska delas med andra användare.

Skapa ett bokmärke

Så här skapar du ett bokmärke:

1. Zooma och panorera till det område som är av intresse.
2. Välj menyalternativet *View* ► *New Spatial Bookmark…*, tryck på `Ctrl+B` eller högerklicka på posten *Spatial Bookmarks* i panelen *Browser* och välj *New Spatial Bookmark*. Dialogrutan *Bookmark Editor* öppnas.

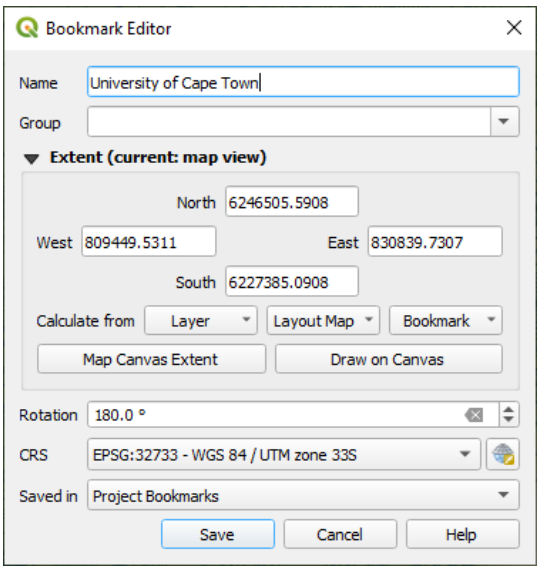


Fig. 7.4: Dialogrutan för redigering av bokmärken

3. Ange ett beskrivande namn för bokmärket
4. Ange eller välj ett gruppnamn där du vill lagra relaterade bokmärken
5. Välj omfattningen av det område som du vill spara med hjälp av widgeten *extent selector*
6. Ändra kartan *Rotation*
7. Ange vilken *CRS* som ska användas för omfattningen
8. Välj om bokmärket ska *Sparas i Användarens bokmärken* eller *Projektets bokmärken* (som standard är den här listrutan inställd på *Användarens bokmärken*)
9. Tryck på *Save* för att lägga till bokmärket i listan

Observera att du kan ha flera bokmärken med samma namn.

Arbeta med bokmärken

För att använda och hantera bokmärken kan du antingen använda panelen *Spatial Bookmarks* eller *Browser*.

Välj *View* ► *Show Spatial Bookmark Manager* eller tryck på `Ctrl+7` för att öppna panelen *Spatial Bookmarks Manager*. Välj *View* ► *Show Bookmarks* eller `Ctrl+Shift+B` för att visa posten *Spatial Bookmarks* i panelen *Browser*.

Du kan utföra följande uppgifter:

Table 7.2: Hantera bokmärkesåtgärder

Uppgift	Spatial bokmärkeshanterare	Bläddrare
Zoom till ett bokmärke	Dubbelklicka på det, eller markera bokmärket och tryck på knappen *Zooma till bokmärke*.	Dubbelklicka på det, dra och släpp det på kartbilden eller högerklicka på bokmärket och välj *Zoom to Bookmark*.
Ta bort ett bokmärke	Markera bokmärket och klicka på knappen *Ta bort bokmärke*. Bekräfta ditt val.	Högerklicka på bokmärket och välj *Delete Spatial Bookmark*. Bekräfta ditt val.
Exportera bokmärken till XML	Klicka på knappen *Importera/Exportera bokmärken* och välj *Exportera*. Alla bokmärken (användare eller projekt) sparas i en xml-fil.	Välj en eller flera mappar (användare eller projekt) eller undermappar (grupper), högerklicka sedan och välj *Exportera spatiala bokmärken…*. Den valda delmängden bokmärken sparas.
Importera bokmärken från XML	Klicka på knappen *Importera/Exportera bokmärken* och välj *Importera*. Alla bokmärken i XML-filen importeras som användarens bokmärken.	Högerklicka på posten *Spatiala bokmärken* eller en av dess mappar (användare eller projekt) eller undermappar (grupper) för att avgöra var bokmärkena ska importeras och välj sedan *Importera spatiala bokmärken*. Om detta utförs på posten *Spatiala bokmärken* läggs bokmärkena till i *User Bookmarks*.
Redigera bokmärke	Du kan ändra ett bokmärke genom att ändra värdena i tabellen. Du kan redigera namnet, gruppen, omfattningen och om det ska lagras i projektet eller inte.	Högerklicka på önskat bokmärke och välj *Edit Spatial Bookmark…*. Då öppnas *Bookmark Editor*, där du kan omdefiniera alla aspekter av bokmärket som om du skapade det för första gången. Du kan också dra och släppa bokmärket mellan mappar (användare och projekt) och undermappar (grupper).

Du kan hantera bokmärkesåtgärder genom att högerklicka på önskat bokmärke i *Spatial Bookmarks Manager*. Du kan också zooma till bokmärken genom att skriva in bokmärkets namn i *locator*.

7.1.6 Dekorera kartan

Dekorationer inkluderar rutnät, titeletikett, copyrightetikett, bild, norrpil, skalstreck och layoutextens. De används för att ”dekorera” kartan genom att lägga till kartografiska element.

Rutnät

Grid gör att du kan lägga till ett koordinatnät och koordinatanteckningar i kartbilden.

1. Välj menyalternativet *Visa* ► *Dekorationer* ► *Rutnät…* för att öppna dialogen.

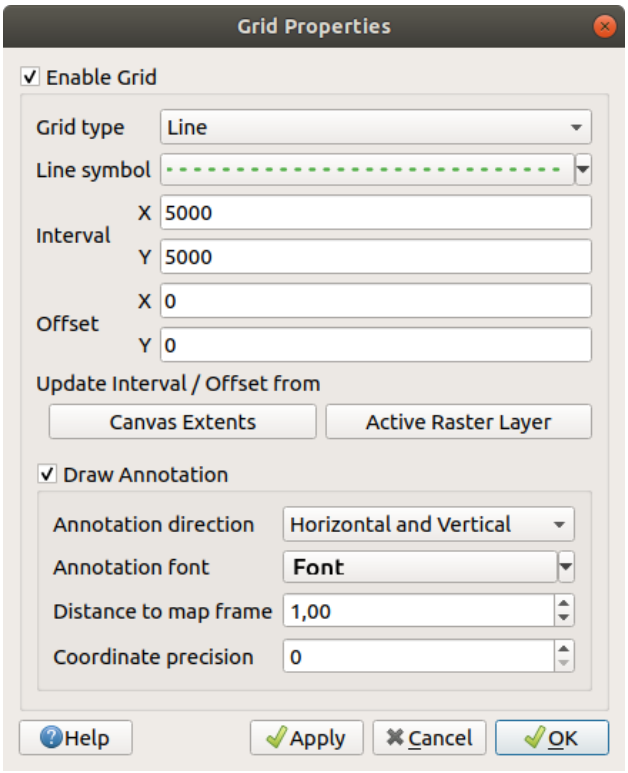


Fig. 7.5: Dialogrutan för rutnät

2. Tick *Aktivera rutnät* och ställ in rutnätsdefinitioner enligt de lager som laddats i kartbilden:
   - Typ av *Grid*: den kan vara *Line* eller *Marker*
   - Den associerade *linjesymbol* eller *markörsymbol* som används för att representera rutnätsmarkeringarna
   - *Intervall X* och *Intervall Y* mellan rutnätsmarkeringarna, i kartenheter
   - Ett *Offset X* och *Offset Y* avstånd för rutnätsmarkeringarna från det nedre vänstra hörnet av kartbilden, i kartenheter
   - Parametrarna för intervall och offset kan ställas in baserat på:
     - *Canvas Extents*: genererar ett rutnät med ett intervall som är ungefär 1/5 av dukens bredd
     - *Aktivt rasterlager*-upplösning
3. Tick *Draw annotations* för att visa koordinaterna för rutnätets markeringar och uppsättning:
   - *Annotation direction*, dvs. hur etiketterna ska placeras i förhållande till rutnätslinjen. Den kan vara:
     - *Horisontal* eller *Vertikal* för alla etiketter

  - *Horisontell och vertikal*, dvs. varje etikett är parallell med det rutnätsmärke som den hänvisar till
  - *Begränsningsriktning*, dvs. varje etikett följer dukens begränsning och är vinkelrät mot den rutnätsmarkering som den hänvisar till
- Teckensnittet *Annotation* (textformatering, buffert, skugga…) med hjälp av widgeten *fontselector*
- *Distance to map frame*, marginalen mellan anteckningar och kartans gränser. Praktiskt när man *exporterar kartbilden* t.ex. till ett bildformat eller PDF, och undviker att anteckningar hamnar på ”pappers”-gränserna.
- Den *Koordinatprecision*

4. Klicka på *Apply* för att kontrollera att det ser ut som förväntat eller på *OK* om du är nöjd.

Titelmärke

Title Label låter dig dekorera din karta med en **Title**.

Så här lägger du till en dekoration för titeletikett:

1. Välj menyalternativet *Visa* ► *Dekorationer* ► *Titellabel…* för att öppna dialogen.

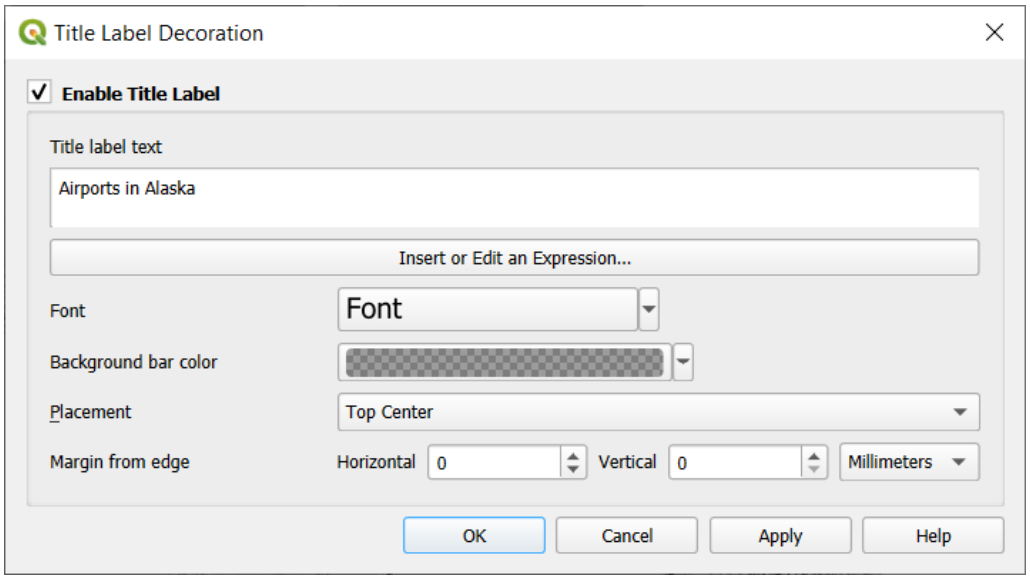


Fig. 7.6: Dialogrutan för dekorering av titlar

2. Kontrollera att *Enable Title Label* är markerad
3. Ange den titeltext som du vill placera på kartan. Du kan göra den dynamisk med hjälp av knappen *Insert or Edit an Expression…*.
4. Välj *Font* för etiketten med hjälp av *font selector widget* med full tillgång till QGIS *text formatting* alternativ. Du kan snabbt ställa in teckensnittets färg och opacitet genom att klicka på den svarta pilen till höger om kombinationsrutan för teckensnitt.
5. Välj *color* som ska gälla för titelns *Background bar color*.
6. Välj *Placement* för etiketten på duken: alternativen är *Top left*, *Top Center* (standard), *Top Right*, *Bottom left*, *Bottom Center* och *Bottom Right*.
7. Förfina placeringen av objektet genom att ange en horisontell och/eller vertikal *Marginal från kant*. Dessa värden kan vara i **Millimeter** eller **Pixel** eller anges som en **Procent** av kartans bredd eller höjd.
8. Klicka på *Apply* för att kontrollera att det ser ut som förväntat eller på *OK* om du är nöjd.

Copyright Etikett

Copyright Label kan användas för att dekorera din karta med en **Copyright**-etikett.

För att lägga till denna dekoration:

1. Välj menyalternativet *Visa* ► *Dekorationer* ► *Copyright Label…* för att öppna dialogen.

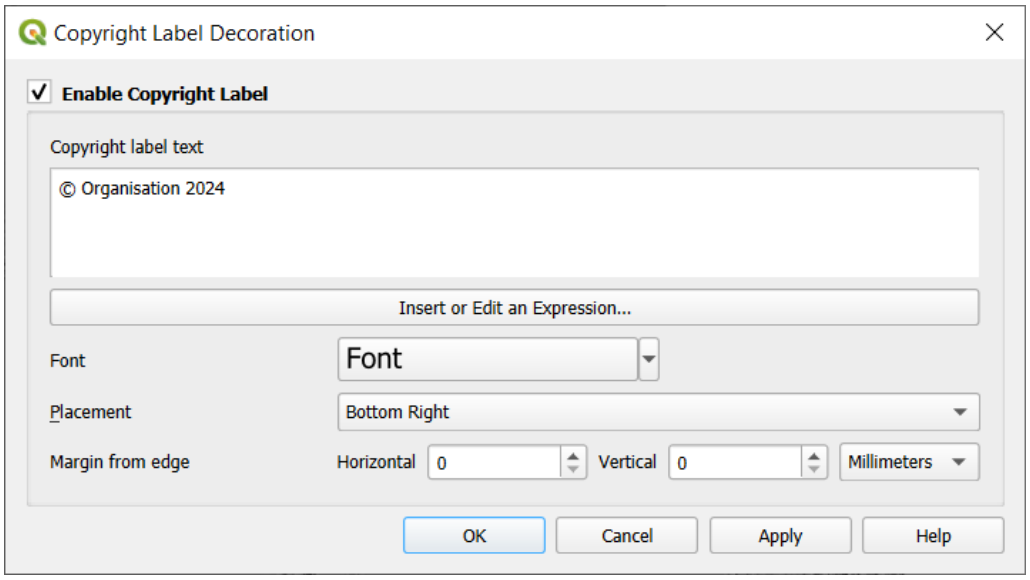


Fig. 7.7: Dialogrutan för dekorering av upphovsrätt

2. Kontrollera att *Enable Copyright Label* är markerad
3. Ange den copyright-text som du vill placera på kartan. Du kan göra den dynamisk med hjälp av knappen *Insert or Edit an Expression…*.
4. Välj *Font* för etiketten med hjälp av *font selector widget* med full tillgång till QGIS *text formatting* alternativ. Du kan snabbt ställa in teckensnittets färg och opacitet genom att klicka på den svarta pilen till höger om kombinationsrutan för teckensnitt.
5. Välj *Placement* för etiketten i duken: alternativen är *Top left*, *Top Center*, *Top Right*, *Bottom left*, *Bottom Center* och *Bottom Right* (standard för Copyright-dekoration)
6. Förfina placeringen av objektet genom att ange en horisontell och/eller vertikal *Marginal från kant*. Dessa värden kan vara i **Millimeter** eller **Pixel** eller anges som en **Procent** av kartans bredd eller höjd.
7. Klicka på *Apply* för att kontrollera att det ser ut som förväntat eller på *OK* om du är nöjd.

Dekoration av bilder

Image gör att du kan lägga till en bild (logotyp, förklaring, …) på kartbilden.

Lägga till en bild:

1. Välj menyalternativet *Visa* ► *Dekorationer* ► *Bild…* för att öppna dialogrutan.

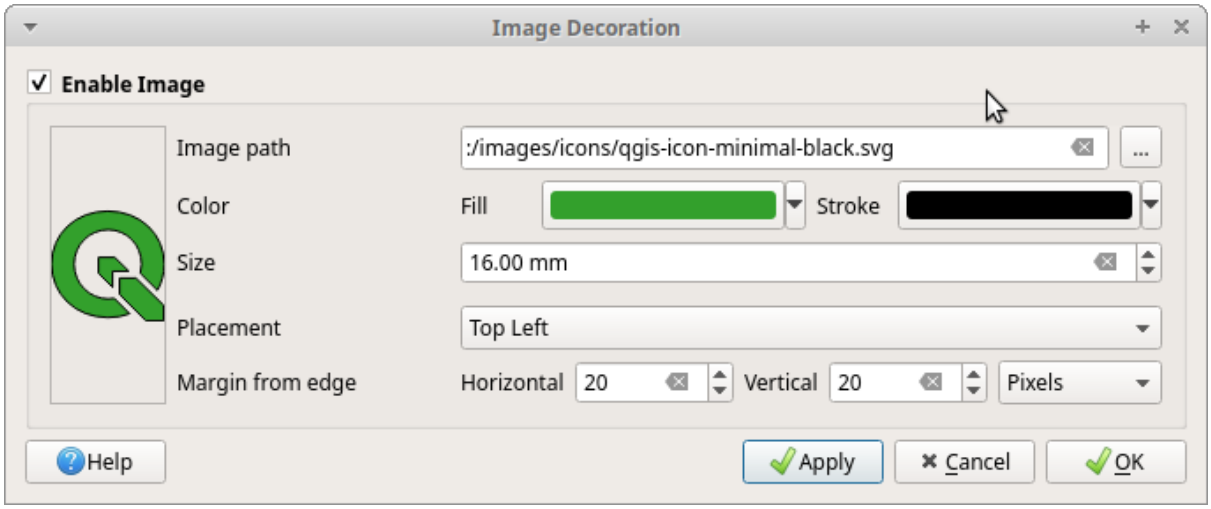


Fig. 7.8: Dialogrutan för bilddekoration

2. Kontrollera att *Enable Image* är markerad
3. Välj en bitmapp (t.ex. png eller jpg) eller SVG-bild med hjälp av … Browse-knappen
4. Om du har valt en parameter som aktiverar SVG kan du också ange en *Fill*- eller *Stroke*-färg (kontur). För bitmappsbilder är färginställningarna inaktiverade.
5. Ställ in en *Size* för bilden i mm. Bredden på den valda bilden används för att ändra storlek på den till angiven *Size*.
6. Välj var du vill placera bilden på kartbilden med hjälp av kombinationsrutan *Placement*. Standardpositionen är *Top Left*.
7. Ställ in *Horizontal* och *Vertical Margin from (Canvas) Edge*. Dessa värden kan anges i **Millimeter**, **Pixel** eller som en **Procent** av kartans bredd eller höjd.
8. Klicka på *Apply* för att kontrollera att det ser ut som förväntat och på *OK* om du är nöjd.

Nordpilen

North Arrow gör att du kan lägga till en norrpil på kartbilden.

Lägga till en norrpil:

1. Välj menyalternativet *Vy* ► *Dekorationer* ► *Nordpil…* för att öppna dialogrutan.

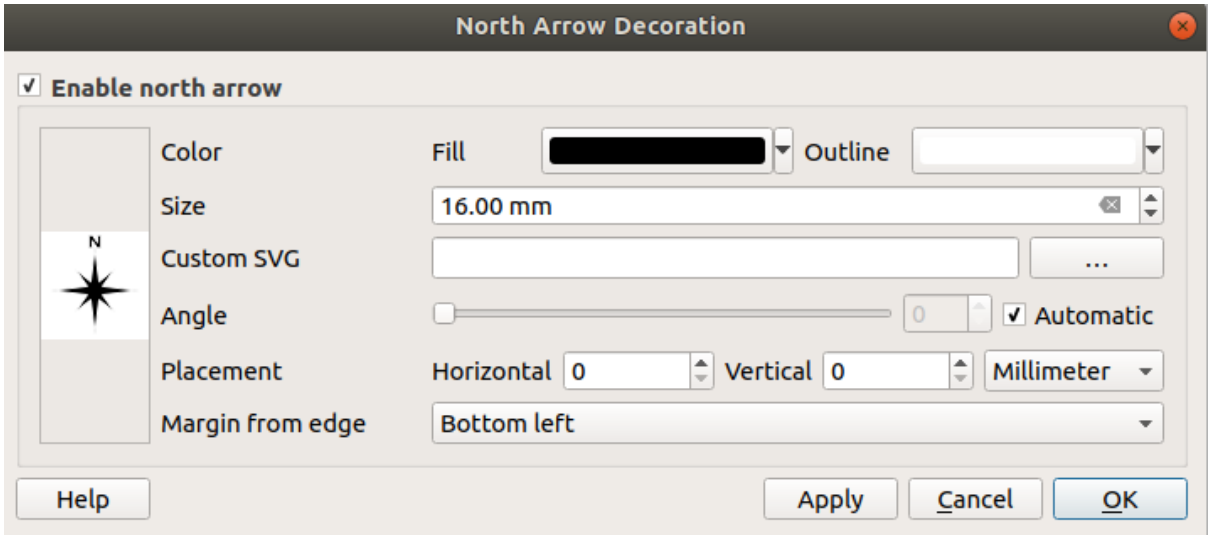


Fig. 7.9: Dialogrutan för norrpil

2. Kontrollera att *Enable north arrow* är markerad
3. Du kan ändra färg och storlek eller välja en anpassad SVG
4. Ändra vinkeln eller välj **Automatisk** för att låta QGIS bestämma riktningen
5. Du kan även välja placering i kombinationsrutan Placering
6. Förfina eventuellt pilens placering genom att ange en horisontell och/eller vertikal ”marginal från (Canvas) Edge”. Dessa värden kan vara i **Millimeter** eller **Pixel** eller anges som en **Procent** av kartans bredd eller höjd.
7. Klicka på *Apply* för att kontrollera att det ser ut som förväntat och på *OK* om du är nöjd.

Skala Bar

Scale Bar lägger till ett enkelt skalstreck i kartbilden. Du kan styra stilen och placeringen samt märkningen av stapeln. Skalstapeln respekterar det aktiva *projektets avståndsenhet* enligt definitionen i *Projektegenskaper* ► *Allmänt* ► *Enheter för avståndsmätning*.

För att lägga till en skalstreck:

1. Välj menyalternativet *View* ► *Decorations* ► *Scale Bar…* för att öppna dialogrutan

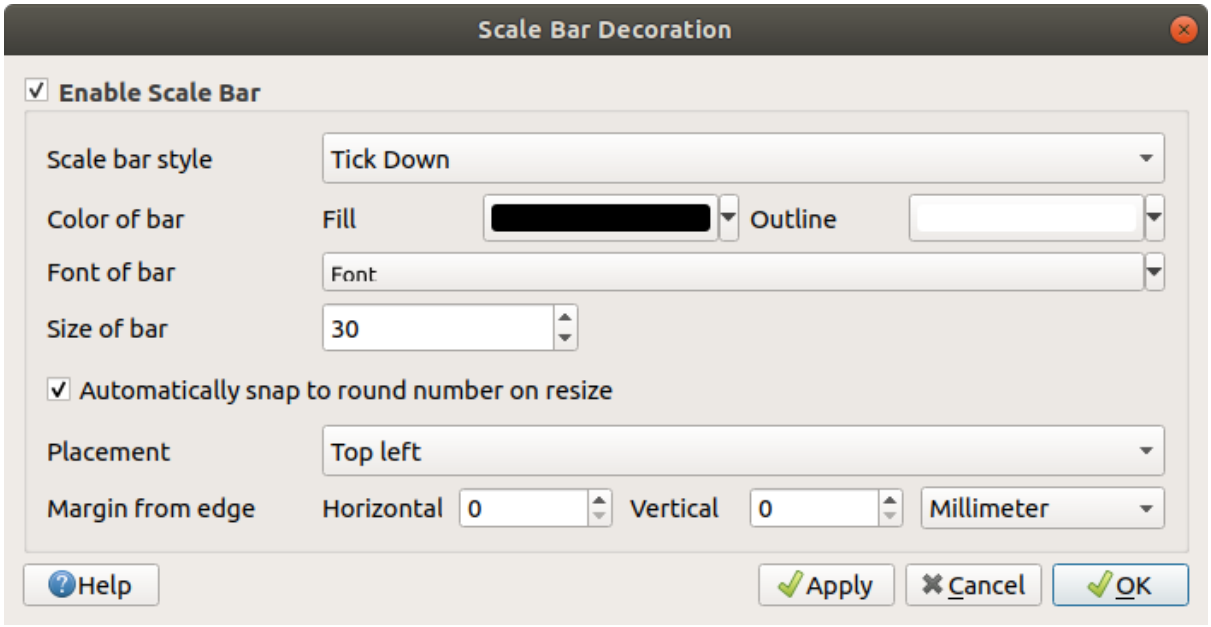


Fig. 7.10: Dialogrutan för skalfält

2. Kontrollera att *Enable scale bar* är markerad
3. Välj en stil från kombinationsrutan *Scale bar style* (välj sträng)
4. Välj *Color of bar* genom att välja en fyllningsfärg (standard: svart) och en konturfärg (standard: vit). Skalstapelns fyllning och kontur kan göras ogenomskinliga genom att klicka på nedåtpilen till höger om färginmatningen.
5. Välj teckensnitt för skalstrecket från kombinationsrutan *Teckensnitt för streck*
6. Ställ in *Storleken på baren* i den aktiva enheten
7. Markera valfritt *Automatiskt avrunda tal vid storleksändring* för att visa lättlästa värden
8. Välj placering från kombinationsrutan *Placement*
9. Du kan förfina placeringen av objektet genom att ange en horisontell och/eller vertikal *Marginal från (Canvas) Edge*. Dessa värden kan anges i **Millimeter** eller **Pixel** eller ställas in som en **Procent** av bredden eller höjden på kartbilden.
10. Klicka på *Apply* för att kontrollera att det ser ut som förväntat eller på *OK* om du är nöjd.

Layout Extents

Layout Extents lägger till utbredning av *kartobjekt* i utskriftslayout(er) på duken. När funktionen är aktiverad visas utdragen av alla kartobjekt i alla utskriftslayouter med en lätt prickad kant som är märkt med namnet på utskriftslayouten och kartobjektet. Du kan styra stilen och märkningen av de layoututbredningar som visas. Denna dekoration är användbar när du justerar positioneringen av kartelement, t.ex. etiketter, och behöver veta den faktiska synliga regionen för utskriftslayouter.

Fig. 7.11: Exempel på layoutextent som visas i ett QGIS-projekt med två utskriftslayouter. Utskriftslayouten med namnet ”Sights” innehåller två kartobjekt, medan den andra utskriftslayouten innehåller ett kartobjekt.

För att lägga till layoututsträckning(ar):

1. Välj *Vy* ► *Dekorationer* ► *Layoututbredning* för att öppna dialogrutan

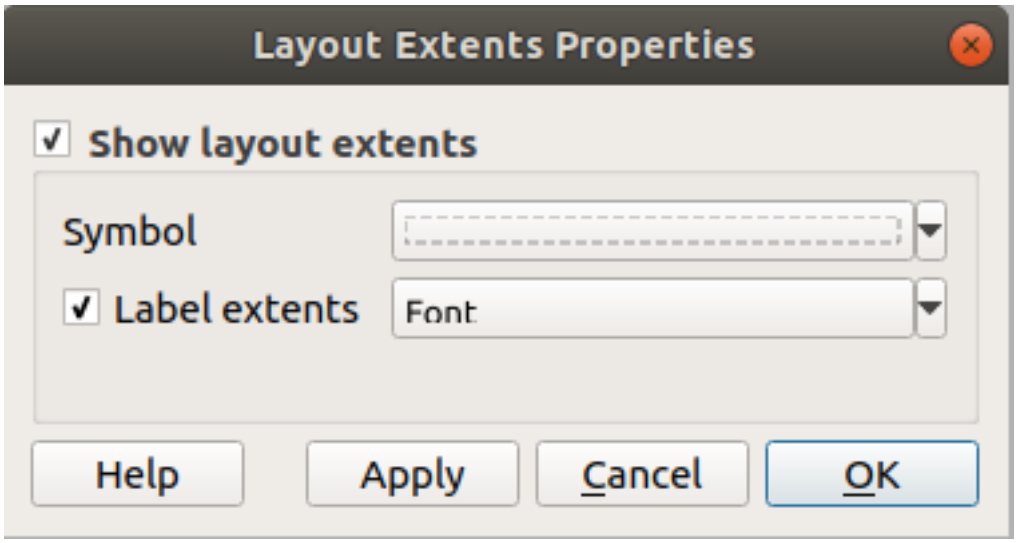


Fig. 7.12: Dialogrutan Layoututbredning

2. Se till att *Visa layoututdrag* är markerad.

3. Ändra eventuellt symbolen och märkningen av extenterna.
4. Klicka på *Apply* för att kontrollera att det ser ut som förväntat och på *OK* om du är nöjd.

Tips

Dekorationer Inställningar

När du sparar en QGIS-projektfil sparas alla ändringar som du har gjort i rutnät, norrpil, skalstreck, copyright och layoutextens i projektet och återställs nästa gång du laddar projektet.

7.1.7 Verktyg för anteckningar

Annotationer är en annan typ av element som läggs till i kartbilden för att ge ytterligare information som inte kan visas av de renderade lagren. Till skillnad från *labels* som förlitar sig på attributvärden som lagras i vektorlager, är annotationer oberoende detaljer som lagras i själva projektfilen.

Det finns två typer av annoteringar tillgängliga i QGIS:

- **Feature annotations**: de är faktiska georefererade funktioner av text-, markör-, linje- eller polygontyp som lagras i en speciell lagertyp som kallas ”annotation layer”. De är knutna till en viss geografisk plats, vilket innebär att om du flyttar kartan, ändrar skala eller projektion kommer dina anteckningar inte att hoppa runt på kartan. De kommer snarare att vara låsta till den plats där du har ritat dem.
- **Balloon annotations**: detta är individuella HTML- eller formannoteringar inuti en bubbla. De kan kopplas till valfritt lager för att synas och ritas ovanpå kartbilden. Storleken är beroende av kartans skala och dess position kan förankras.

Tips

Layout av kartan med anteckningar

Du kan skriva ut eller exportera anteckningar med din karta till olika format med hjälp av:

- verktyg för export av kartbilder som finns i menyn *Projekt*
- *print layout*, i vilket fall du måste markera *Draw map canvas items* i motsvarande kartobjektsegenskaper

Verktygsfältet *Annotations Toolbar* innehåller en uppsättning verktyg för att skapa och interagera med båda typerna av annoteringar.

Table 7.3: Åtgärder i verktygsfältet Anteckningar

Verktyg	Användning	Omfång
Nytt annoteringslager Egenskaper för huvudlager för an	Skapa ett nytt lager för att lagra anteckningar Kontrollinställningar för Main Annotation Layer	Anteckningar om funktioner
Modifiera anteckningar	Välj, flytta, ändra storlek på och modifiera symbologiegenskaper för anteckningar	
Skapa Polygon Annotation	Skapa en annotation som en polygonfunktion	
Skapa linjeannotering	Skapa en annotation som en polylinjefunktion	
Skapa markörannotering	Skapa en annotation som en punktfunktion	
Skapa textannotering vid punkt	Skapa en annotation som en textetikett	
Skapa textannotering längs linje	Skapa en annotering som en böjd text längs en linjefunktion	
Skapa textannotering i rektangel	Skapa en textformaterad anteckning som är bunden inom en rektangel	
Skapa bildannotering	Skapa en annotation som visar en `.SVG` eller rasterbild	
HTML Annotation	Välj och skapa en annotering med innehållet i en `HTML`-fil	Anteckningar om ballonger
FormAnnotation	Välj och skapa en annotering som visar attribut för ett vektorlager i en anpassad formulärfil	

Observera

Från och med QGIS 3.40 är det inte längre möjligt att skapa ballongannoteringar av SVG- eller texttyp. Vid inläsning av gamla projekt kommer alla dessa annoteringar automatiskt att konverteras till de nyare annoteringstyperna bild och rektangulär text.

Funktionskommentarer

Feature-annoteringar lagras i **annoteringslager**. Till skillnad från konventionella lager är ett annoteringslager endast tillgängligt i det aktuella projektet och kan innehålla funktioner av olika typer (text, markör, linje, polygon, bild). Lagret har inga attribut och ingen associerad symbologi, utan varje objekt kan symboliseras objekt för objekt genom *Layer Styling*-panelen.

Två typer av annoteringslager finns tillgängliga i QGIS:

- Ett vanligt *Annotation Layer*: du kan skapa ett med hjälp av verktyget New Annotation Layer. Det listas i panelen *Layers*, så att du kan styra dess funktioners synlighet, flytta det så att det visas ovanför eller under vissa lager i kartan, som vilket vanligt lager som helst. Dubbelklicka på lagret så får du tillgång till dess egenskaper.
- Den *Huvudlager för annoteringar*: Som standard är det här som anteckningar lagras när inget anteckningslager finns tillgängligt i projektet eller väljs vid skapandet. Det här lagret ritas alltid högst upp på kartan och du kommer inte att se det listat i panelen *Layers* tillsammans med de andra lagren i projektet, vilket innebär att dess funktioner alltid är synliga. Med hjälp av *Main Annotation Lageregenskaper* i verktygsfältet *Annotations* kan du öppna dialogrutan för dess egenskaper.

Egenskaper för lager

Egenskapsdialogen för ett annoteringslager innehåller följande flikar:

- *Information*: en skrivskyddad dialogruta som representerar en intressant plats för att snabbt hämta sammanfattad information och metadata om det aktuella lagret. Detta kan inkludera lagrets utsträckning, antal objekt per annoteringstyp och totalt antal, CRS-detaljer, …
- *Source*: definierar allmänna inställningar för anteckningslagret. Du kan göra följande:
  - Ange ett *Layernamn* som ska användas för att identifiera lagret i projektet (i *Layers Panel*, med uttryck, …)
  - Visa lagrets *Assigned Coordinate Reference System (CRS)*: du kan ändra lagrets CRS genom att välja ett nyligen använt i listrutan eller klicka på Select CRS-knappen (se *Väljare för referenssystem för koordinater*). Använd denna process endast om det CRS som tillämpas på lagret är fel eller om inget CRS har tillämpats.
- *Rendering*:
  - Du kan ställa in *Maximum (inklusive)* och *Minimum (exklusive)* skala, vilket definierar ett skalområde inom vilket funktioner ska vara synliga. Utanför detta intervall är de dolda. Knappen Set to current canvas scale hjälper dig att använda den aktuella kartans skala som gräns för synlighetsintervallet. Se *Väljare för synlighetsskala* för mer information.
  - *Länkat lager*: gör att du kan välja att ange ett länkat synlighetsskikt för annoteringsskiktet. Om detta anges ritas anteckningarna endast när det länkade lagret är synligt på kartan.
  - *Opacitet*: Med det här verktyget kan du göra det underliggande lagret i kartbilden synligt. Använd skjutreglaget för att anpassa synligheten för ditt vektorlager efter dina behov. Du kan också göra en exakt definition av den procentuella synligheten i menyn bredvid skjutreglaget.
  - *Blending mode* på nivån *Layer*: Med dessa verktyg kan du uppnå speciella renderingseffekter som du kanske tidigare bara kände till från grafikprogram. Pixlarna i de överliggande och underliggande lagren blandas med hjälp av de inställningar som beskrivs i *Blandningslägen*.
  - Applicera *paint effects* på alla lagerfunktioner med knappen *Draw Effects*.

  Vissa av dessa alternativ är tillgängliga från egenskaperna för feature annotation *Symbology*.

Interagera med funktioner

Feature-annoteringarna har särskilda verktyg för skapande beroende på vilken typ de är.

Table 7.4: Skapa anteckningar om funktioner

Famil / baser på	Verkty(för anteck	Procedur
Geom	Skapa Polygon Annotation Skapa linjeanmotering Skapa markörannotering	Detta är som att digitalisera en punkt, linje eller polygon vektorfunktion. Alla de vanliga genvägarna för att skapa funktioner gäller när du skapar annoteringsobjekt. En linje- eller polygonannotation ritas genom att vänsterklicka en gång för varje vertex, med ett sista högerklick för att slutföra formen. Snapping kan aktiveras medan du ritar, du kan använda *Advanced Digitizing Tools* för att exakt placera hörnpunkter, och till och med byta *ritverktyg* till exempelvis streaming-läget för helt friformade former.
Text	Skapa texta	Vänsterklicka på kartbilden.
	Skapa texta	Vänsterklicka för att lägga till hörnpunkter i geometrin för textbaslinjen och högerklicka för att avsluta. Som ovan finns även funktioner för digitaliseringsfunktioner tillgängliga.
	Skapa texta	Rita rektangelrutan för texten genom att vänsterklicka för att lägga till ett första hörn och vänsterklicka igen för att lägga till det motsatta.
Bild	Skapa bild	Rita upp bildens rektangel genom att vänsterklicka för att lägga till ett första hörn och vänsterklicka igen för att lägga till det motsatta. Välj sedan en bild eller SVG-fil från filutforskaren.

Till skillnad från vanliga lager behöver ett annoteringslager inte vara aktivt innan du väljer dess funktioner. Ta bara tag i verktyget Modify Annotations och du kommer att kunna interagera med alla annoterade funktioner:

- **Selection**: vänsterklicka på anteckningen. Som standard återges annotationer i den ordning de skapades, vilket innebär att nya annotationer placeras ovanpå äldre. Du kan behöva spela med egenskapen Z-index för funktioner för att välja dem som de ska placeras ovanför.
- **Flytta**: Vänsterklicka på ett markerat annoteringsobjekt för att börja flytta det. Om du högerklickar eller trycker på `Esc` avbryts förflyttningen, medan en andra vänsterklickning bekräftar förflyttningen. Förflyttningen kan också styras genom att trycka på markörknapparna:
  - `Shift+tangent` för stor rörelse
  - `Alt+tangent` för `1 px` rörelse
- **Geometrimodifiering**: flytta över en annotation och lila fyrkanter visas på noder i dess underliggande geometri. Vänsterklicka på fyrkanten, flytta och klicka igen. För linjära eller polygonala annotationer kan du dubbelklicka på ett segment för att lägga till en ny vertex.
- **Radera**: Om du trycker på `Del` eller `Backspace` medan en annotation är markerad raderas den annotationen
- *Ändra funktionssymbologi*

Funktionssymbologi

En markerad annotering visar sina *Symbology*-egenskaper i panelen *Layer styling*.

- För polygon-, polylinje- och marköranteckningar kan du ändra utseendet med hjälp av alla funktioner i *symbolegenskaperna*
- För textbaserade annoteringar finns ett område med textredigeringsverktyg som hjälper dig att konstruera strängen som ska visas. Den kan vara flerlinjig, använda HTML-formatering och förlita sig på QGIS uttrycksfunktioner. Du kan dessutom ändra utseendet med hjälp av alla funktioner i *textformategenskaperna*.

  Beroende på vilken typ av anteckning det gäller finns ytterligare alternativ tillgängliga.

  - För textannotering vid punkt är det också möjligt att:
    * ange texten *Alignment* (vänster, mitt eller höger om ankarpunkten)
    * konfigurerar om texten ska *Ignore map rotation* eller *Rotate with map*. I båda fallen kan en anpassad *Angle* ställas in för funktionens orientering.
  - För textanteckning längs en linje, konfigurera en *Offset from line* i valfri enhet
  - För textannotering i rektangel är det också möjligt att:
    * ange texten *Horizontal alignment* (vänster, mitt, höger eller justify) och *Vertical alignment* (topp, vertikal mitt eller botten) i rektangelrutan
    * konfigurera för rektangeln, avståndet *Margins* i de enheter du väljer, färgerna *Frame* och *Background*.
- Med bildkommentarer kan du visa en *Rasterbild* eller en *SVG-bild* med hjälp av *Väljare för fjärrstyrd eller inbäddad fil*. Bildens placering och storlek kan ställas in enligt följande:
  - *Skalaberoende storlek*: bilden skalas med kartan och blir större när man zoomar in och mindre när man zoomar ut. Storleken definieras av en enda referensstorlek i kartans enheter (t.ex. meter eller grader, beroende på lagrets CRS). Med *Lock aspect ratio* kan du hålla annoteringsstorleken proportionell mot den inbäddade bildens storlek eller sträcka ut bilden så att den fyller annoteringsområdet.
  - *Fixed size*: bildannoteringen behåller en konstant storlek i skärm- eller utdataenheter (t.ex. pixlar eller millimeter), oavsett kartans skala. Annotationens *Width* och *Height* kan fyllas i separat eller justeras proportionellt mot den inbäddade bildens storlek.
  - *Relative to map*: bildannoteringen kommer alltid att återges i samma position i förhållande till kartans canvasgränser, oavsett kartans skala. Annotationens *Width* och *Height* kan fyllas separat eller justeras proportionellt mot den inbäddade bildens storlek.

  Dessutom kan du aktivera visning av *Frame* och *Background* för rektangelboxen och konfigurera dem med *fill symbols*.
- För text vid punkt, text inom rektangel och bildannoteringar kan du *Visa upprop* när bildens eller textens ankarpunkt förskjuts från standardplaceringen. Så här skapar du en bildtext för en annotation:
  1. Välj anteckningsobjekt
  2. Flytta ankarpunkten, d.v.s. klicka på den centrala och gröna `X`-noden och klicka på den nya placeringen
  3. I anteckningsegenskaperna trycker du på knappen … för att konfigurera *callout properties*.

  Tips

  Om du lägger till en ballongtext till en text- eller bildtext får du den gamla ballongtextvisningen.
- Konfigurera en *Referensskala*: anger den kartskala som symbol- eller textstorlekar som använder pappersbaserade enheter (t.ex. millimeter eller punkter) relaterar till. Storlekarna kommer att skalas i enlighet med detta när kartan visas i en annan skala. Till exempel kommer en linje som är 2 mm bred i skala 1:2000 *Referensskala* att återges med 4 mm när kartan visas i skala 1:1000.
- Ange ett *Z-index*: en funktion med ett högre index placeras ovanpå funktioner med lägre index. En praktisk inställning för både visning och val av funktioner.
- Ändra några av *Layer rendering*-inställningarna

Anteckningar om ballonger

Du kan lägga till ballonganteckningar genom *Edit* ► *Add Annotation* ► menyn eller från *Annotations Toolbar*:

- HTML Annotation för att placera innehållet i en `html`-fil
- Form Annotation: användbart för att visa attribut för ett vektorlager i en anpassad `ui`-fil (se Fig. 7.13). Detta liknar *custom attribute forms*, men visas i ett anteckningsobjekt. Titta också på *den här videon <https://www.youtube.com/watch?v=0pDBuSbQ02o&feature=youtu.be&t=2m25s>* från Tim Sutton för mer information.

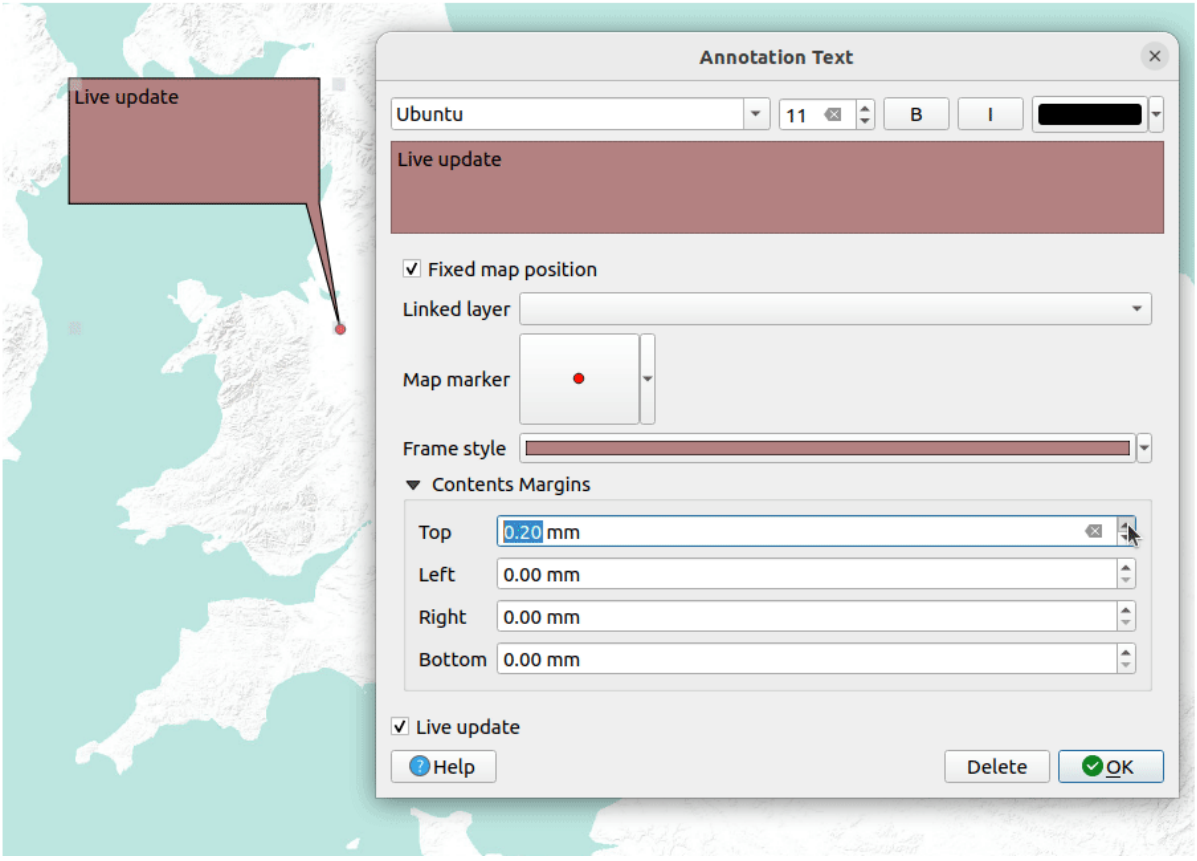


Fig. 7.13: Exempel på ballonganteckningar

För att lägga till en ballongannotation väljer du motsvarande verktyg och klickar på kartbilden. En tom ballong läggs till. Dubbelklicka på den så öppnas en dialogruta med olika alternativ. Denna dialog är nästan densamma för alla typer av anteckningar:

- Längst upp en filväljare som ska fyllas med sökvägen till en `html`- eller `ui`-fil beroende på vilken typ av annotering det gäller.
- *Fixed map position*: när den är avmarkerad baseras ballongens placering på skärmens position (istället för kartan), vilket innebär att den alltid visas oavsett kartans utbredning.
- *Linked layer*: associerar anteckningen med ett kartlager och gör den synlig endast när det lagret är synligt.
- *Kartmarkör*: använder *QGIS-symboler*, ställer in symbolen som ska visas vid ballongens ankarposition (visas endast när *Fixerad kartposition* är markerad).
- *Frame style*: anger ramens bakgrundsfärg, transparens, streckfärg eller bredd för ballongen som använder QGIS-symboler.
- *Contents margins*: anger inre marginaler för annoteringsramen.

- Med *Live update* kan du förhandsgranska dina ändringar i realtid.

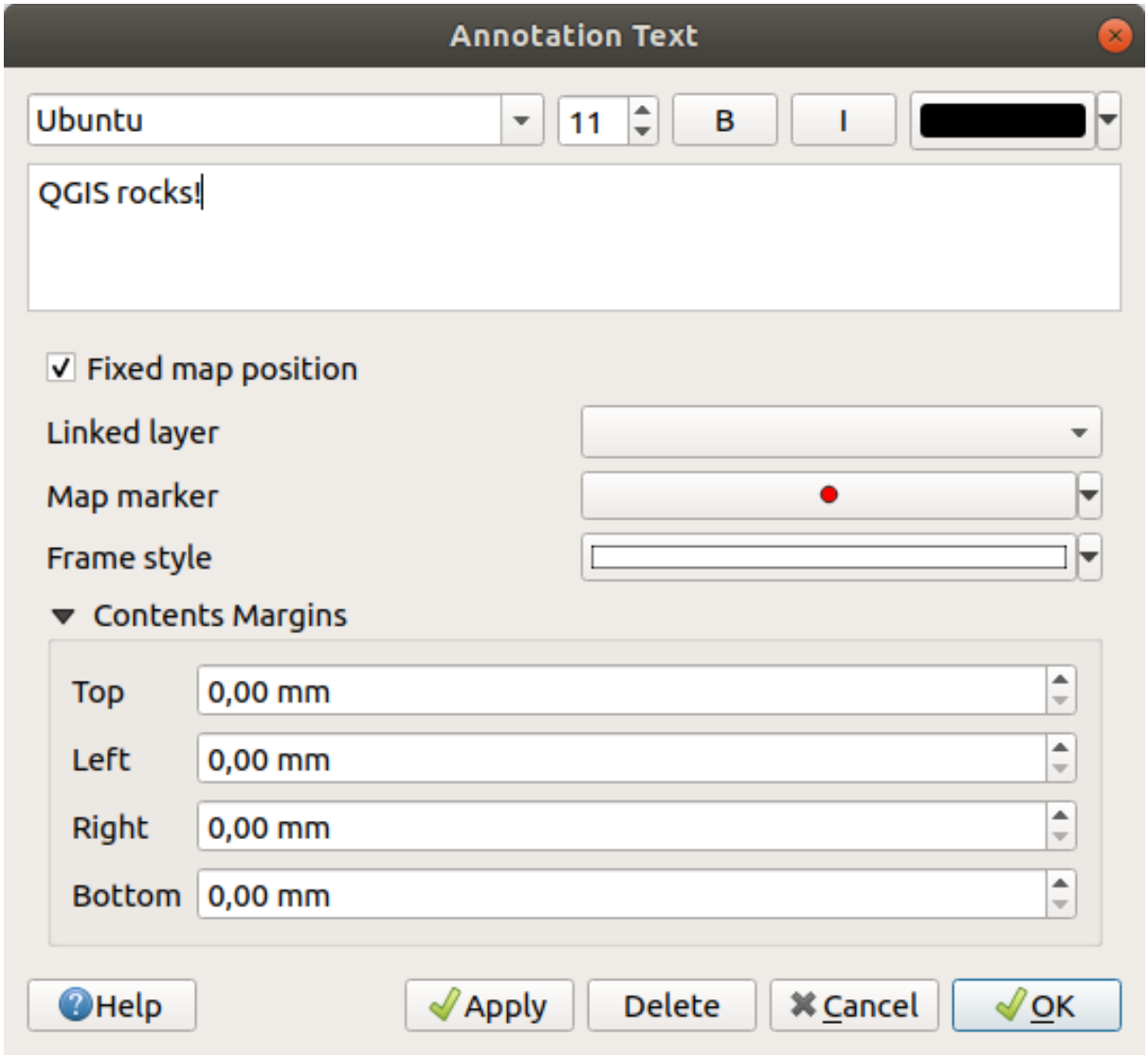


Fig. 7.14: En dialogruta med text om ballongens anteckning

När ett verktyg för ballonganteckningar är aktivt kan du också göra det:

- Välj en anteckning
- Ändra storlek på en annotering
- Flytta en annotation efter kartposition (genom att dra kartmarkören) eller genom att bara flytta ballongen.
- Ta bort en annotation: markera den och tryck antingen på knappen `Del` eller `Backspace`, eller dubbelklicka på den och tryck på knappen *Delete* i dialogrutan Egenskaper.
- Högerklicka och i den kontextuella menyn:
  - *Kopieringskoordinat* för anteckningens kartmarkör i olika CRS
  - *Edit* egenskaper för annotationen. Samma sak som att dubbelklicka på annotationen.
  - *Ta bort* anteckningen

7.1.8 Mätning

Allmän information

QGIS erbjuder fyra sätt att mäta geometrier:

- interaktiva mätverktyg
- mätning i fältet Fältkalkylator
- härledda mätningar i *Identifiering av funktioner*-verktyget
- verktyget för vektoranalys: *Vector* ► *Geometry Tools* ► *Export/Add Geometry Columns*

Mätning fungerar inom projicerade koordinatsystem (t.ex. UTM) och icke-projicerade data. De tre första mätverktygen fungerar lika bra i globala projektinställningar:

- Till skillnad från de flesta andra GIS är standardmätmetriken ellipsoid, med hjälp av den ellipsoid som definieras i *Projekt* ► *Egenskaper…* ► *Allmänt*. Detta gäller både när geografiska och projicerade koordinatsystem definieras för projektet.
- Om du vill beräkna den projicerade/planimetriska ytan eller avståndet med hjälp av cartesisk matematik måste mätellipsoiden ställas in på ”None/Planimetric” (*Projekt* ► *Egenskaper…* ► *General*). Men med ett geografiskt (dvs. icke-projicerat) CRS definierat för data och projekt kommer mätning av område och avstånd att vara ellipsoidisk.

Varken identifieringsverktyget eller fältkalkylatorn kommer dock att omvandla dina data till projektets CRS före mätningen. Om du vill uppnå detta måste du använda vektoranalysverktyget: *Vector* ► *Geometry Tools* ► *Add Geometry Attributes…*. Här är mätningen planimetrisk, om du inte väljer ellipsoidmätning.

Mät längd, ytor, lager och vinklar interaktivt

Klicka på ikonen i verktygsfältet Attribute för att påbörja mätningen. Nedåtpilen nära ikonen växlar mellan längd, yta, bäring eller vinkel. Standardenheten som används i dialogrutan är den som anges i *Projekt* ► *Egenskaper…* ► *Allmänt* menyn.

För *Mätlinje* och *Mätområde* kan mätningarna göras med *Cartesian* eller *Ellipsoidal* mått.

Observera

Konfigurera mätverktyget

När du mäter längd eller yta kan du klicka på knappen *Configuration* längst ned i widgeten för att öppna menyn :menuselection:` *Inställningar –> Alternativ –> Map Tools*, där du kan välja färg på gummibandet, precisionen i mätningarna och enhetens beteende. Du kan också välja önskade mät- eller vinkelenheter, men kom ihåg att dessa värden åsidosätts i det aktuella projektet av det val som görs i menyn *Projekt* ► *Egenskaper… -> General…* ► *General* och genom det val som görs i mätwidgeten.

Alla mätmoduler använder snappinginställningarna från digitaliseringsmodulen (se avsnitt *Inställning av snäpptolerans och sökradie*). Om du vill mäta exakt längs en linjefunktion eller runt en polygonfunktion måste du alltså först ställa in toleransen för snäppning av lagret. När du nu använder mätverktygen kommer varje musklick (inom toleransinställningen) att snäppas till det lagret.

Verktyget Mätlinje mäter avstånd mellan givna punkter. Verktyget låter dig sedan klicka på punkter på kartan. Varje segmentlängd, liksom den totala, visas i mätfönstret. I mätfönstret visas koordinater för alla dina punkter och avstånd. Tänk på att den första raden endast innehåller koordinater, eftersom den representerar din startpunkt. Nu är det möjligt att kopiera alla linjemätningar på en gång till urklipp med hjälp av knappen *Kopiera*. Om du klickar på

knappen *Configuration* kommer du till *Measure Tool Copy Settings* där du kan ställa in *copy options*. För att stoppa mätningen, klicka på höger musknapp.

Observera att du kan använda rullgardinsmenyn nära totalen för att ändra *mätningsenheter* interaktivt medan du arbetar med mätverktyget. Denna enhet behålls för widgeten tills ett nytt projekt skapas eller ett annat projekt öppnas.

Avsnittet *Info* i dialogrutan förklarar hur beräkningarna görs enligt de CRS-inställningar som finns tillgängliga.

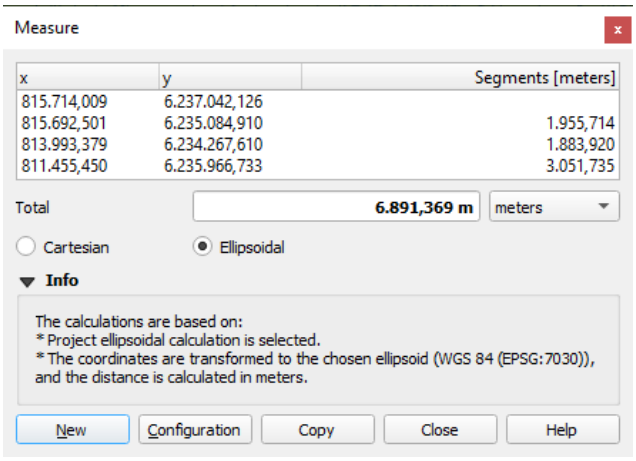


Fig. 7.15: Mät avstånd

Mät yta: Områden kan också mätas. I mätfönstret visas den ackumulerade områdesstorleken. Högerklicka för att stoppa ritningen. Infoavsnittet är också tillgängligt, liksom möjligheten att växla mellan olika *area units*.

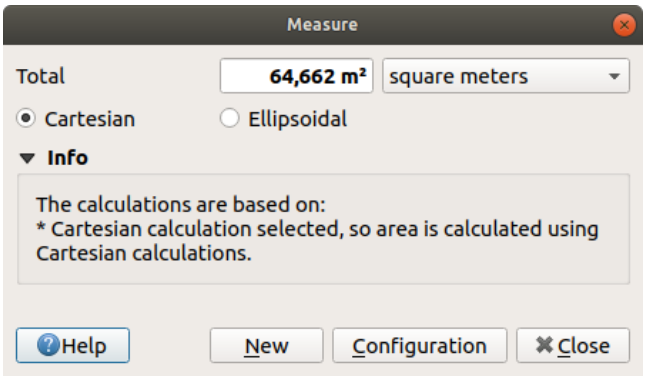


Fig. 7.16: Mätområde

Mätbäring: Du kan också mäta lager. Markören blir korsformad. Klicka för att rita den första punkten i lagret och flytta sedan markören för att rita den andra punkten. Mätningen visas i en popup-dialogruta.

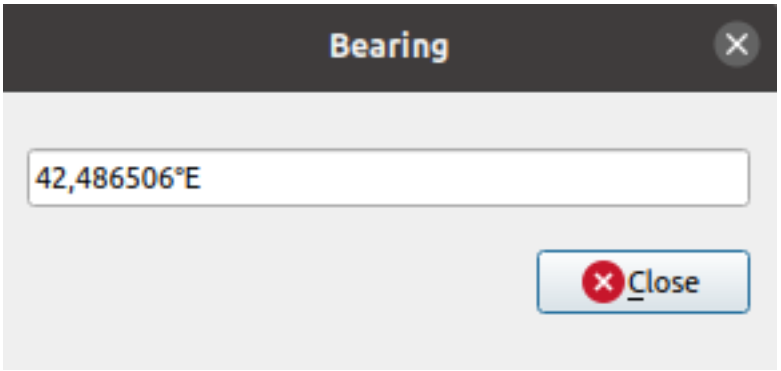


Fig. 7.17: Mätning av lager

Mät vinkel: Du kan också mäta vinklar. Markören blir korsformad. Klicka för att rita det första segmentet av den vinkel du vill mäta och flytta sedan markören för att rita önskad vinkel. Mätningen visas i en popup-dialogruta.

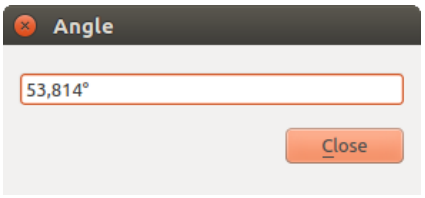


Fig. 7.18: Mät vinkel

7.1.9 Ställa in ytterligare kartvyer

Det är också möjligt att öppna ytterligare kartvyer vars innehåll kan avvika från *Layers*-panelens nuvarande tillstånd. För att lägga till en ny kartvy, gå till *View* ► *New Map View*. En ny flytande widget, som efterliknar renderingen av huvudkartvyn, läggs till i QGIS. Du kan lägga till så många kartvyer som du behöver. De kan hållas flytande, placeras sida vid sida eller staplas ovanpå varandra.

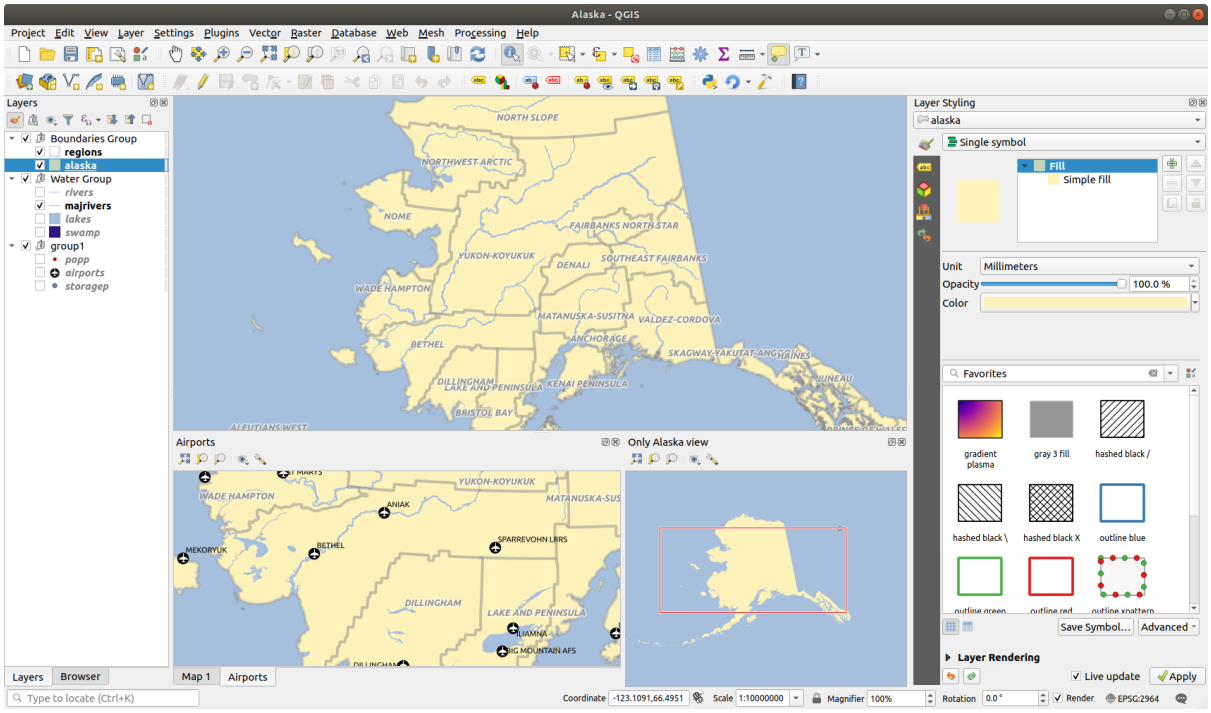


Fig. 7.19: Flera kartvyer med olika inställningar

Överst på en extra kartbild finns ett verktygsfält med följande funktioner:

- Zoom Full, Zoom to Selection och Zoom to Layer(s) för att navigera inom vyn
- Set View Theme för att välja det *karttema* som ska visas i kartvyn. Om inställningen är `(none)` kommer vyn att följa paneländringarna i *Layers*.
- View settings för att konfigurera kartvyn:
  - *Synkronisera vycentrum med huvudkartan*: synkroniserar centrum för kartvyerna utan att ändra skalan. Detta gör att du kan ha en översiktsstil eller förstorad karta som följer huvudkartans centrum.
  - *Synkronisera vy till urval*: samma som zoom till urval
  - *Skala*
  - *Rotation*
  - *förstoring*
  - *Synkronisera skalan* med huvudkartans skala. En *Skalfaktor* kan sedan tillämpas, vilket gör att du kan ha en vy som t.ex. alltid är 2x skalan på huvudkartan.
  - *Visa anteckningar*
  - *Visa markörens position*
  - *Visa huvuddukens utsträckning*
  - *Visa etiketter*: gör det möjligt att dölja etiketter oavsett om de är inställda i de visade lagrens egenskaper
  - *Ändra karta CRS…*
  - *Rename view…*

7.1.10 Exportera kartvyn

Kartor som du skapar kan layoutas och exporteras till olika format med hjälp av de avancerade funktionerna i *print layout eller report*. Det är också möjligt att direkt exportera den aktuella renderingen utan layout. Den här snabba ”skärmdumpen” av kartvyn har några praktiska funktioner.

För att exportera kartbilden med den aktuella renderingen:

1. Gå till *Projekt* ► *Importera/Exportera*
2. Beroende på ditt utmatningsformat väljer du antingen
   - *Exportera karta till bild…*
   - eller *Exportera karta till PDF…*

De två verktygen ger dig en gemensam uppsättning alternativ. I den dialogruta som öppnas:

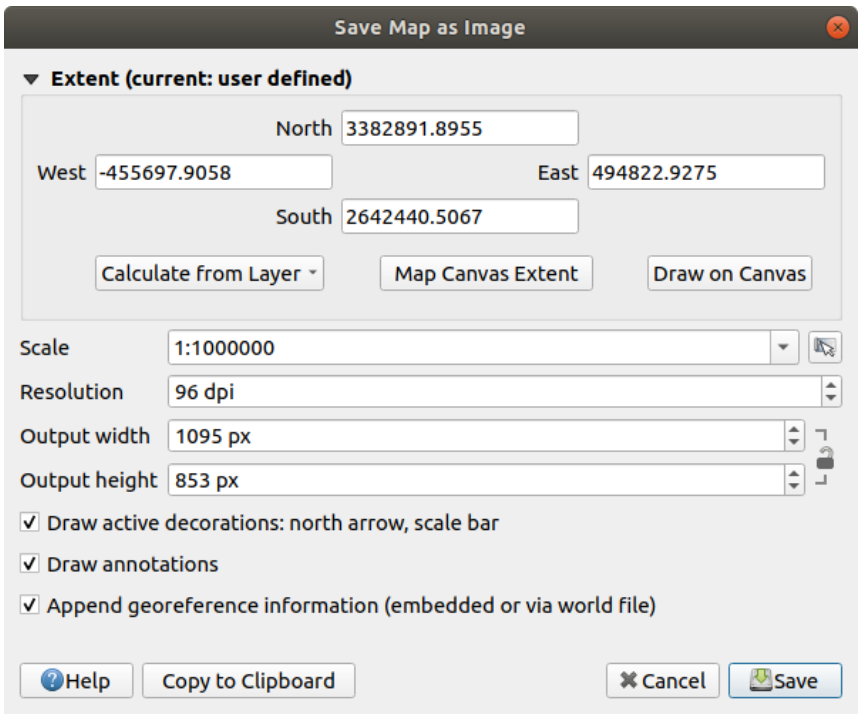


Fig. 7.20: Dialogrutan Spara karta som bild

1. Välj den *Extent* som ska exporteras: det kan vara den aktuella vyns utsträckning (standard), ett lagers utsträckning eller en anpassad utsträckning som ritas över kartbilden. Koordinaterna för det valda området visas och kan redigeras manuellt.
2. Ange *Skala* för kartan eller välj den från *fördefinierade skalor*: om du ändrar skalan ändras storleken på den utsträckning som ska exporteras (från mitten).
3. Ställ in *Resolution* för utdata
4. Styr *Output width* och *Output height* i pixlar för bilden: som standard baseras de på den aktuella upplösningen och omfattningen, men de kan anpassas och ändrar storleken på kartans omfattning (från mitten). Storleksförhållandet kan låsas, vilket kan vara särskilt praktiskt när man ritar utbredning på duken.
5. *Draw active decorations*: in use *dekorationer* (skalstreck, titel, rutnät, norrpil…) exporteras med kartan
6. *Draw annotations* för att exportera alla *annotation*

7. *Lägg till georeferensinformation (inbäddad eller via världsfil)*: beroende på utdataformat sparas en världsfil med samma namn (med tillägget `PNGW` för `PNG`-bilder, `JPGW` för `JPG`, …) i samma mapp som din bild. I formatet `PDF` bäddas informationen in i PDF-filen.
8. Vid export till PDF finns fler alternativ i dialogrutan *Save map as PDF…*:

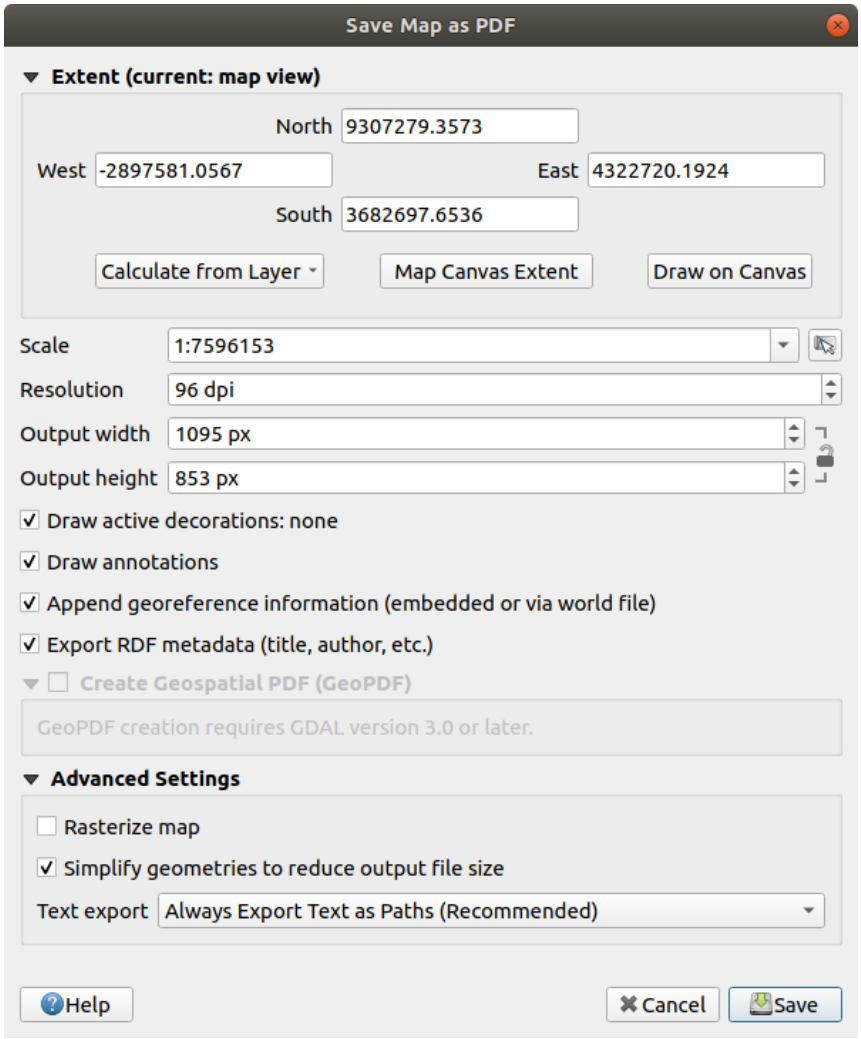


Fig. 7.21: Dialogrutan Spara karta som PDF

- *Exportera RDF-metadata* för dokumentet, t.ex. titel, författare, datum, beskrivning …
- *Skapa geospatial PDF*: Generera en georefererad PDF-fil. Det kan du göra:
  - Välj geospatial PDF *Format*
  - *Include vector feature information* in the Geospatial PDF file: kommer att inkludera all geometri- och attributinformation från funktioner som är synliga på kartan i den utmatade Geospatial PDF-filen.

> **Observera**
>
> En geospatial PDF-fil kan också användas som datakälla. Mer information om stöd för geospatiala PDF-filer i QGIS finns på https://north-road.com/2019/09/03/qgis-3-10-loves-geospatialpdf/.

- *Rasterisera karta*

- *Förenkla geometrier för att minska storleken på utdatafilen*: Geometrier kommer att förenklas vid export av kartan genom att ta bort hörn som inte är märkbart olika vid exportupplösningen (t.ex. om exportupplösningen är `300 dpi`, kommer hörn som är mindre än `1/600 tum` från varandra att tas bort). Detta kan minska exportfilens storlek och komplexitet (mycket stora filer kan inte laddas i andra program).
- Ange *Text export*: styr om textetiketter alltid eller helst ska exporteras som *text- eller konturobjekt*.

9. Klicka på *Save* för att välja plats, namn och format för filen.

   När du exporterar till bild är det också möjligt att *Kopiera till klippbord* det förväntade resultatet av ovanstående inställningar och klistra in kartan i ett annat program som LibreOffice, GIMP …

7.2 3D-kartvy

stöd för 3D-visualisering erbjuds genom 3D-kartvyn. Du kan skapa, hantera och öppna 3D-kartvyer via menyn *View* ► *3D Map Views* ►:

1. Genom att klicka på *New 3D Map View* kan du skapa en ny 3D-kartvy. En flytande och dockningsbar QGIS-panel visas (se *Dialogrutan 3D-kartvy*). Den har samma utbredning och vy som 2D-huvudkartan och innehåller en uppsättning navigeringsverktyg för att förvandla vyn till 3D.
2. Genom att klicka på *Manage 3D Map Views* kommer du till 3D Map Views Manager. Här får du möjlighet att öppna, duplicera, ta bort och byta namn på 3D-kartvyer.
3. Om du har skapat en eller flera 3D-kartvyer ser du dem listade i *3D Map Views*. Du kan slå på och av dem genom att klicka på . De sparas när du sparar projektet, även om de är avstängda.

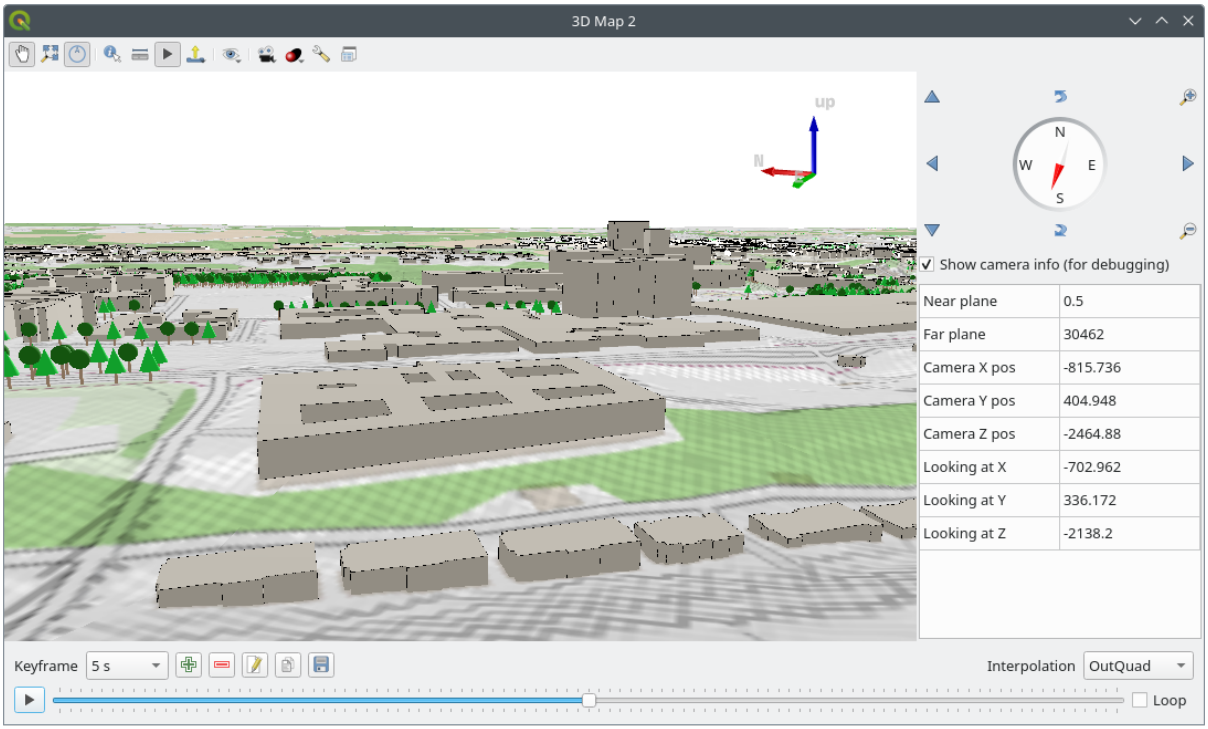


Fig. 7.22: Dialogrutan 3D-kartvy

Följande verktyg finns längst upp i panelen för 3D-kartvyn:

- Kamerakontroll: flyttar vyn, behåller samma vinkel och riktning för kameran

- Zoom Full: ändrar vyn till hela lagrets utsträckning, eller *referensutsträckningen* om den är inställd
- Toggle On-Screen Notification: visar/döljer navigeringswidgeten (som är tänkt att göra det lättare att styra kartvyn)
- Identify: returnerar information om den klickade punkten i terrängen eller den eller de klickade 3D-objekten – Mer information finns på *Identifiering av funktioner*
- Mätlinje: mäter det horisontella avståndet mellan punkter
- Animationer: visar/döljer widgeten *animationsspelare*
- Export-menyn gör det möjligt att exportera scenen till olika format:
  - Save as Image…: exporterar den aktuella vyn till ett bildfilsformat
  - Export 3D Scene: exporterar den aktuella vyn som en 3D-scen (`.obj`-fil), vilket möjliggör efterbearbetning i program som Blender… Terräng- och vektorobjekt exporteras som 3D-objekt. Exportinställningarna, som åsidosätter lagren *properties* eller kartvyn *configuration*, inkluderar:
    - *Scenens namn* och destination *Mapp*
    - *Terrängupplösning*
    - *Terrain texturupplösning*
    - *Modellskala*
    - *Mjuka kanter*
    - *Exportera normaler*
    - *Exportera texturer*
- Ställ in visningstema: Gör att du kan välja den uppsättning lager som ska visas i kartvyn från fördefinierade *kartteman*.
- Menyn Camera hjälper dig att kontrollera förhållandet mellan 2D- och 3D-vyerna:
  - Synkronisera vyerna (*2D-kartvy följer 3D-kamera* och/eller *3D-kamera följer 2D-kartvy*)
  - *Visa synligt kameraområde i 2D-kartvy*
  - *Set 3D scene on 2D map view*: gör det möjligt att klippa 3D-scenen och endast visa terräng och funktioner som skär en utsträckning som ritats på 2D-kartan. Fler alternativ finns på fliken *Generell konfiguration*.
- Effects lägger till visuella effekter till 3D-renderingen, t.ex. att visa *skuggor*, *ögonkupolbelysning* eller *ambient occlusion*.
- Knappen Options öppnar dialogrutan för att konfigurera 3D-kartvyn *settings*.
- Docka 3D-kartvy: växla från dockad widget till fönster på översta nivån

7.2.1 Konfiguration av scen

3D-kartvyn öppnas med några standardinställningar som du kan anpassa. Detta gör du genom att expandera menyn Options högst upp i 3D-panelen och trycka på knappen *Configure* för att öppna fönstret *3D configuration*.

I fönstret 3D Configuration finns olika alternativ för att finjustera 3D-scenen:

Allmänt

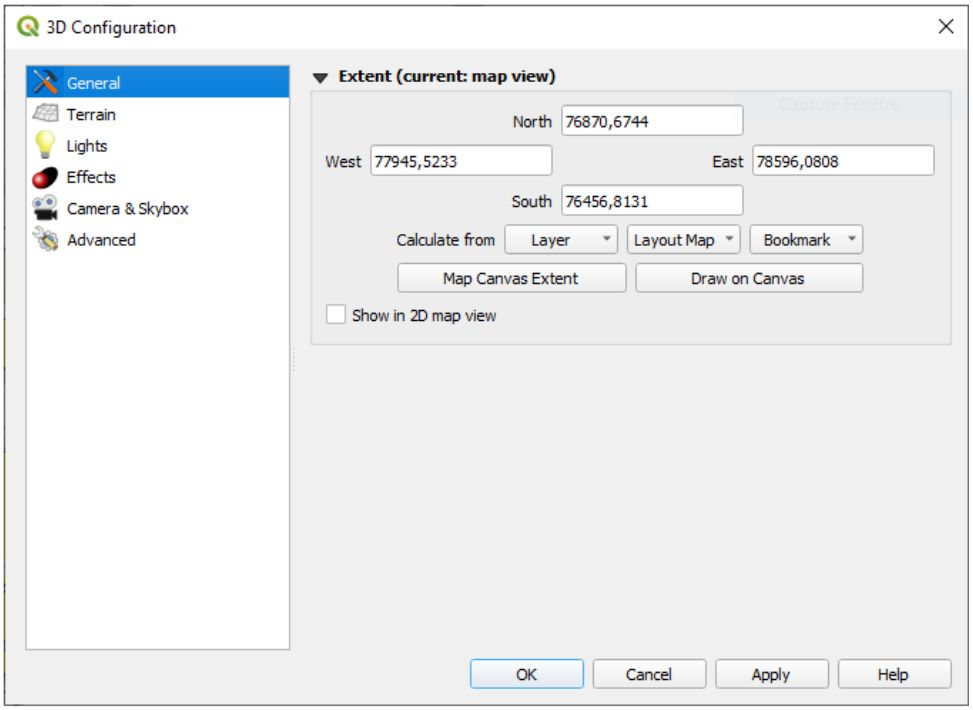


Fig. 7.23: Dialogrutan Allmän konfiguration för 3D-karta

Under fliken *General* kan du:

- Begränsa 3D-scenens innehåll till en 2D-kartutbredning med hjälp av *spatial extent selector*: terräng och funktioner klipps i den angivna utbredningen och endast de delar som faller inom utbredningen laddas i scenen. Höjdintervallet för terräng-, mesh- och punktmolnslagren tas också med i beräkningen så att kameran inte placeras under scenens innehåll, särskilt när terrängens vertikala skalinställning används för att överdriva höjdskillnaderna.

  Den inställda utbredningen används också som referensutbredning när du trycker på knappen Zoom full i 3D-kartvyn.

- Markera *Visa i 2D-kartvy* för att visa ett gummiband i huvudkartan som motsvarar den aktuella omfattningen av 3D-scenen.

Terräng

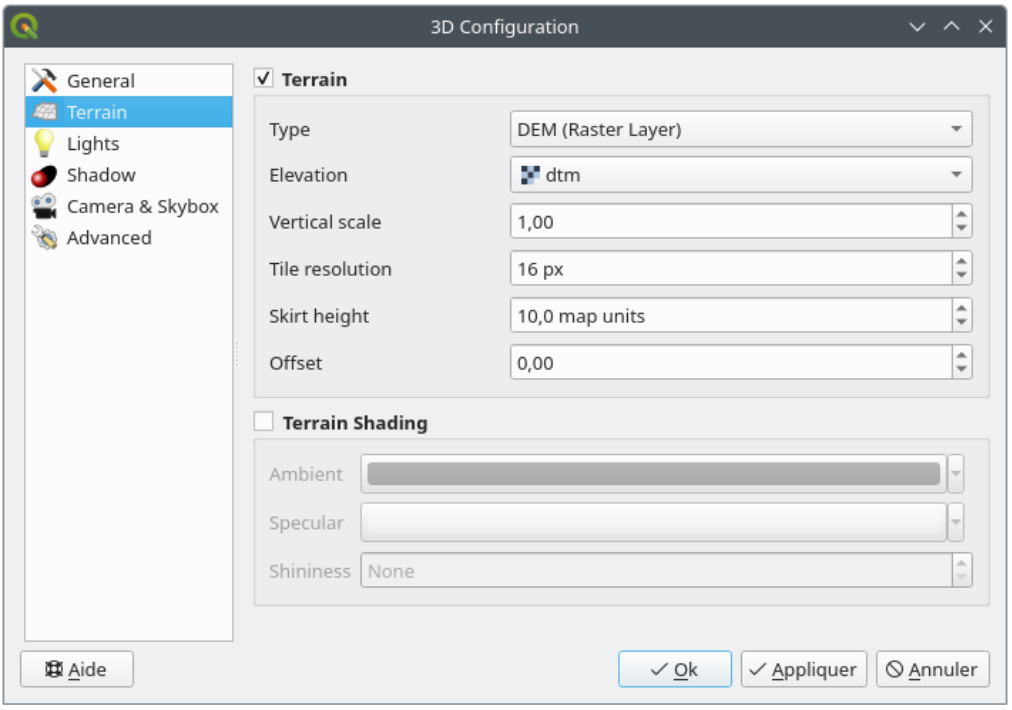


Fig. 7.24: Dialogrutan Konfiguration av 3D-kartans terräng

- *Terräng*: Innan vi går in på detaljerna är det värt att notera att terrängen i en 3D-vy representeras av en hierarki av terrängplattor och när kameran rör sig närmare terrängen ersätts befintliga plattor som inte har tillräckliga detaljer med mindre plattor med mer detaljer. Varje bricka har en meshgeometri som härrör från höjdrasterlagret och textur från 2D-kartlager.
  - Höjdterrängen *Type* kan vara:
    - a *Flat terräng*
    - en laddad *DEM (Raster Layer)*
    - en *Online*-tjänst, laddar höjdplattor producerade av Mapzen-verktyg – mer information på https://registry.opendata.aws/terrain-tiles/
    - en laddad *Mesh*-datauppsättning
    - ett *Quantized Mesh*-lager från projektet
  - *Elevation*: Raster- eller nätlager som ska användas för att generera terrängen. Rasterlagret måste innehålla ett band som representerar höjden över havet. För ett mesh-lager används hörnens Z-värden.
  - *Vertikal skala*: Skalfaktor för den vertikala axeln. Om skalan ökas kommer landformernas höjd att överdrivas.
  - *Tile resolution*: Hur många sampel från terrängrasterlagret som ska användas för varje kakelplatta. Ett värde på 16px innebär att geometrin för varje bricka kommer att bestå av 16x16 höjdprover. Högre siffror skapar mer detaljerade terrängplattor på bekostnad av ökad renderingskomplexitet.
  - *Kjolhöjd*: Ibland är det möjligt att se små sprickor mellan plattorna i terrängen. Om du höjer detta värde kommer vertikala väggar (”kjolar”) att läggas till runt terrängplattorna för att dölja sprickorna.
  - *Offset*: flyttar terrängen uppåt eller nedåt, t.ex. för att justera dess höjd i förhållande till marknivån för andra objekt i scenen.

    Detta kan vara användbart när det finns en diskrepans mellan höjden på terrängen och höjden på lager i din scen (t.ex. punktmoln som endast använder en relativ vertikal höjd). I sådana fall kan det förbättra

navigeringsupplevelsen att justera terränghöjden manuellt så att den sammanfaller med höjden på objekten i din scen.

- När ett mesh-lager används som terräng kan du konfigurera inställningarna för *Trianglar* (wireframe display, smooth triangles, level of detail) och *Renderingsinställningar för färg* (som en enhetlig färg eller *color ramp based*). Mer information finns i avsnittet *Mesh layer 3D properties*.
- *Terrängskuggning*: Gör det möjligt att välja hur terrängen ska återges:
  - Skuggning inaktiverad - terrängens färg bestäms endast utifrån kartans textur
  - Skuggning aktiverad - terrängens färg bestäms med hjälp av Phongs skuggningsmodell, som tar hänsyn till karttextur, terrängens normalvektor, scenljus och terrängmaterialets *Ambient* och *Specular* färger och *Shininess*

Lampor

På fliken *Lights* trycker du på menyn för att lägga till

- upp till åtta *Punktljus*: avger ljus i alla riktningar, som en sfär av ljus som fyller ett område. Objekt som är närmare ljuset blir ljusare och objekt som är längre bort blir mörkare. Ett punktljus har en bestämd position (*X*, *Y* och *Z*), en *Color*, en *Intensity* och en *Attenuation*
- upp till fyra *Directional lights*: efterliknar den belysning som du skulle få från en gigantisk blixtljuskälla mycket långt bort från dina objekt, alltid centrerad och som aldrig slocknar (t.ex. solen). Den avger parallella ljusstrålar i en enda riktning, men ljuset når ut i oändligheten. Ett riktat ljus kan roteras givet en *Azimuth*, ha en *Altitude*, en *Color* och en *Intensity*.

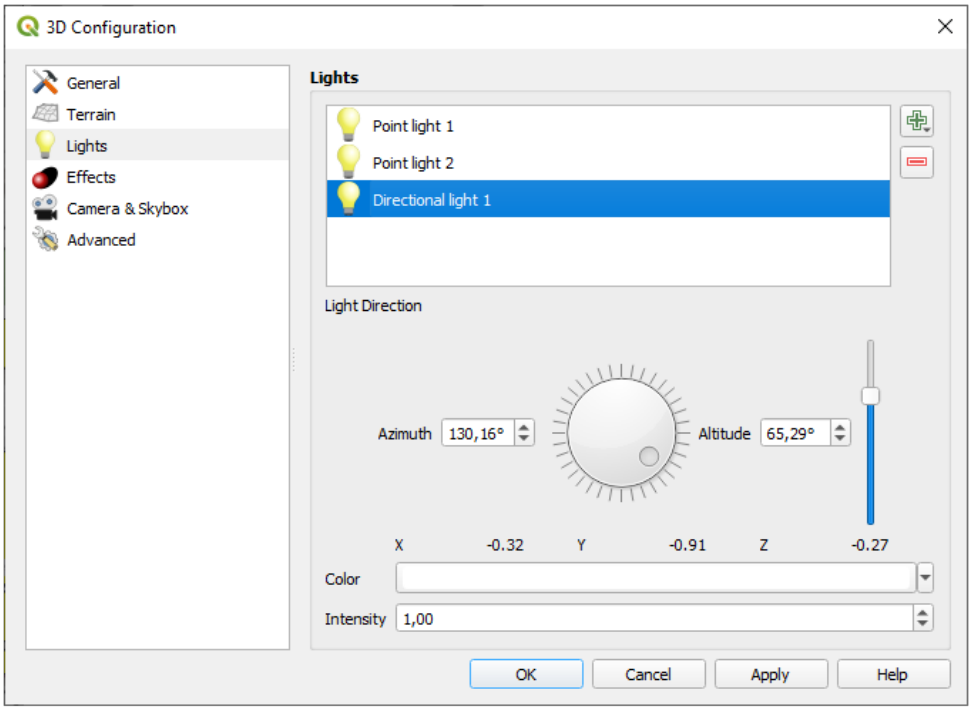


Fig. 7.25: Dialogrutan Konfiguration av lampor för 3D-karta

Effekter

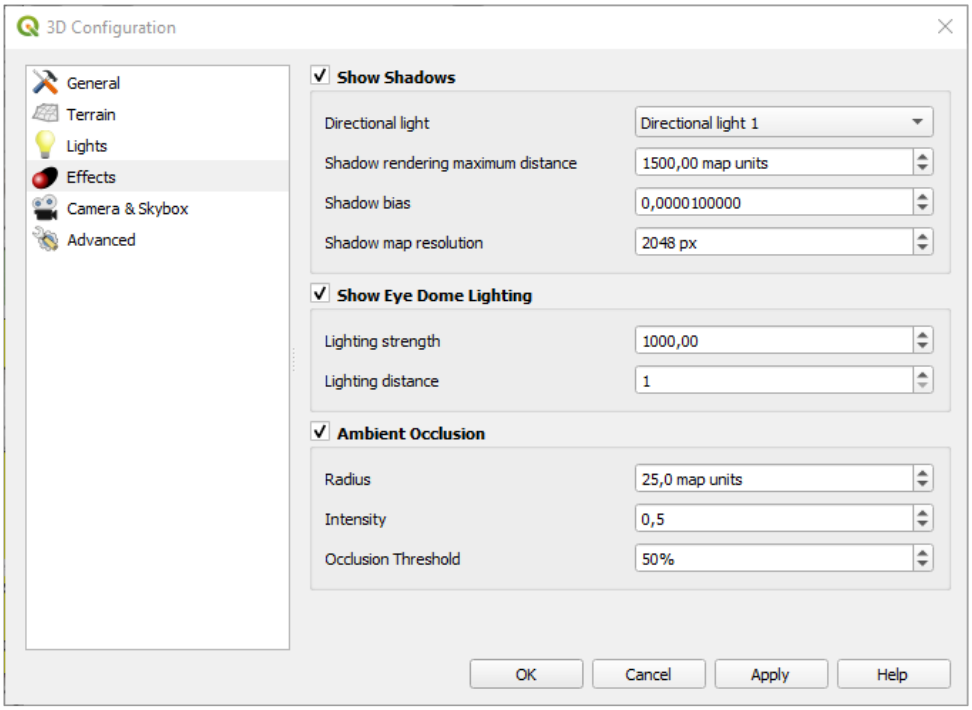


Fig. 7.26: Dialogrutan Konfiguration av 3D-kartans effekter

- Check *Show shadows* för att visa skuggor i din scen, given:
  - a *Riktningsljus*
  - a *Skuggåtergivning maximalt avstånd*: för att undvika att återge skuggan av alltför avlägsna objekt, särskilt när kameran tittar upp längs horisonten
  - a *Shadow bias*: för att undvika självskuggningseffekter som kan göra vissa områden mörkare än andra, på grund av skillnader mellan kartstorlekar. Ju lägre desto bättre
  - en *Shadow map resolution*: för att få skuggor att se skarpare ut. Det kan resultera i sämre prestanda om upplösningsparametern är för hög.
- *Visa Eye Dome Lighting* (EDL): en efterbehandlingseffekt som förbättrar djupseendet. Varje pixels djup (avstånd från kameran) jämförs med de närliggande pixlarnas djup och markeras i enlighet med djupskillnaden, vilket gör att kanterna framträder tydligare. Påverkar hela scenen och kan kombineras med *Screen Space Ambient Occlusion*. Följande parametrar kan kontrolleras:
  - *Ljusstyrka*: ökar kontrasten, vilket ger bättre djupseende
  - *Lighting distance*: representerar avståndet mellan de använda pixlarna och mittpixeln och gör att kanterna blir tjockare.
- Add screen-space *Ambient Occlusion* (SSAO): en efterbehandlingseffekt som också förbättrar djupseendet genom att applicera en mörkare skuggning på områden som är mindre utsatta för omgivande belysning. Påverkar hela scenen och kan kombineras med *Eye dome Lighting*. Följande parametrar kan kontrolleras:
  - *Radius*: hur långt vi ska nå för att beräkna omgivande ocklusion
  - *Intensitet*: hur stark effekten ska vara (högre värden gör saker mörkare)
  - *Ocklusionströskel*: hur många närliggande punkter som måste vara ockluderade för att effekten ska visas (lägre värden än 50% gör resultatet mörkare, men ger eventuellt ett större ocklusionsområde)

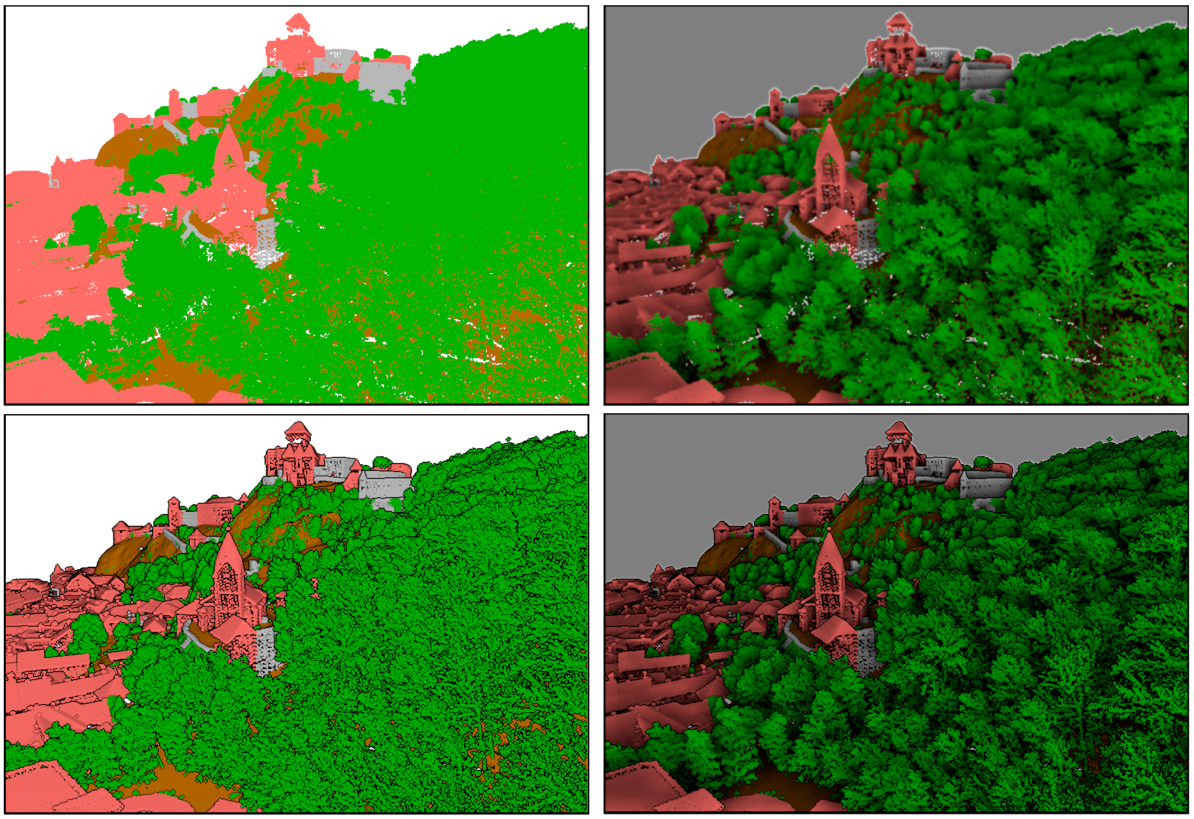


Fig. 7.27: Rendering av punktmoln i 3D-karta med hjälp av Eye Dome Lighting (EDL) och/eller Screen-Space Ambient Occlusion (SSAO)
Uppifrån, från vänster till höger: Ingen effekt – Endast SSAO – Endast EDL – SSAO och EDL

Kamera & Skybox

På den här fliken kan du styra olika parametrar som kamera, 3D-axel, navigationssynkronisering och skybox.

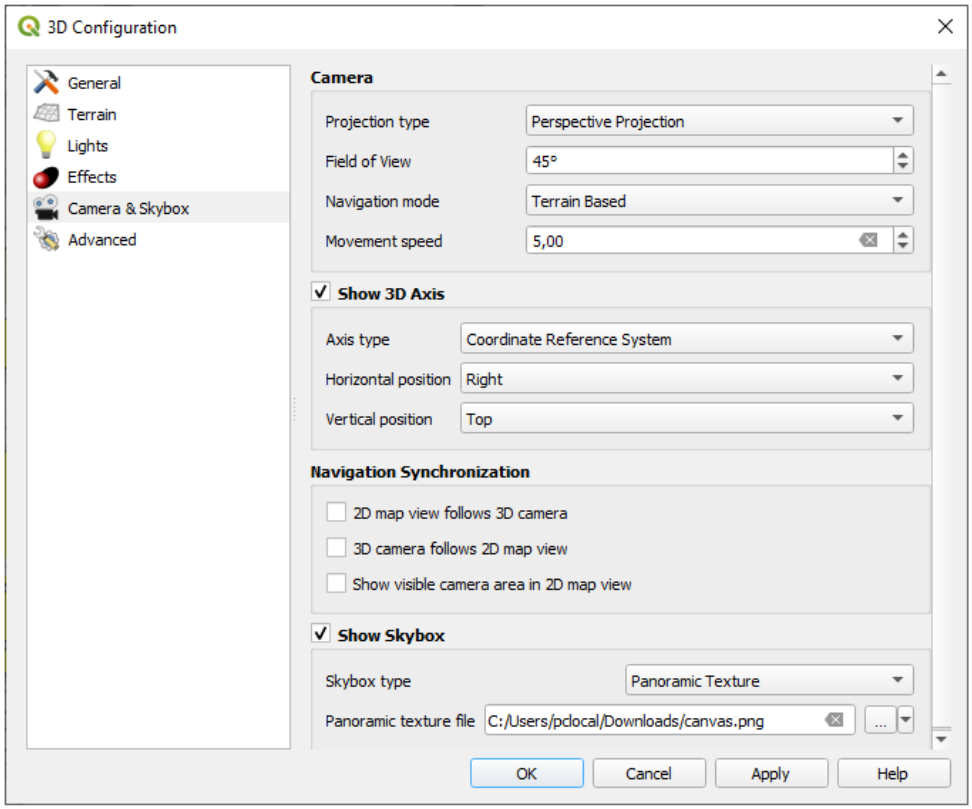


Fig. 7.28: Dialogrutan Konfiguration av kamera för 3D-karta

- Parametergruppen *Camera* åsidosätter vissa *standardinställningar för kameran* som görs i dialogrutan *Inställningar* ► *Alternativ* ► *3D*.
- Markera *Show 3D Axis* för att aktivera 3D-axelverktyget. Denna parametergrupp gör det möjligt att ställa in axeltyp och dess position.
  - Med typen *Coordinate Reference System* kommer en ortogonal axel att representeras.
  - Med typen *Kub* kommer en 3D-kub att representeras. Kubens ytor kan användas för att ändra kameravyn: klicka t.ex. på ytan *north* för att ställa in kameran så att den ser från norr.

Tips

Högerklicka på 3D-axeln för att snabbt ställa in dess position och typ samt kameravyn.

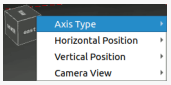


Fig. 7.29: Kontextmenyn för 3D-axeln

- Parametergruppen *Navigation Synchronization* lägger till alternativ för att synkronisera 2D-vyn med 3D-kamerans position eller 3D-kamerans position med 2D-vyn eller synkronisering i två riktningar. Det sista alternativet visar den utsträckning som är synlig från 3D-kameran över 2D-kartvyn.
- Markera eller avmarkera *Show skybox* för att aktivera rendering av skyboxar i scenen. Skybox-typen kan vara:
  - *Panoramic texture*, med en enda fil som ger sikt på 360
  - *Distinct faces*, med en texturfil för var och en av de sex sidorna av en låda som innehåller scenen

  Texturbildfiler för skyboxen kan vara filer på disken, fjärr-URL:er eller inbäddade i projektet (*mer detaljer*).

Avancerat

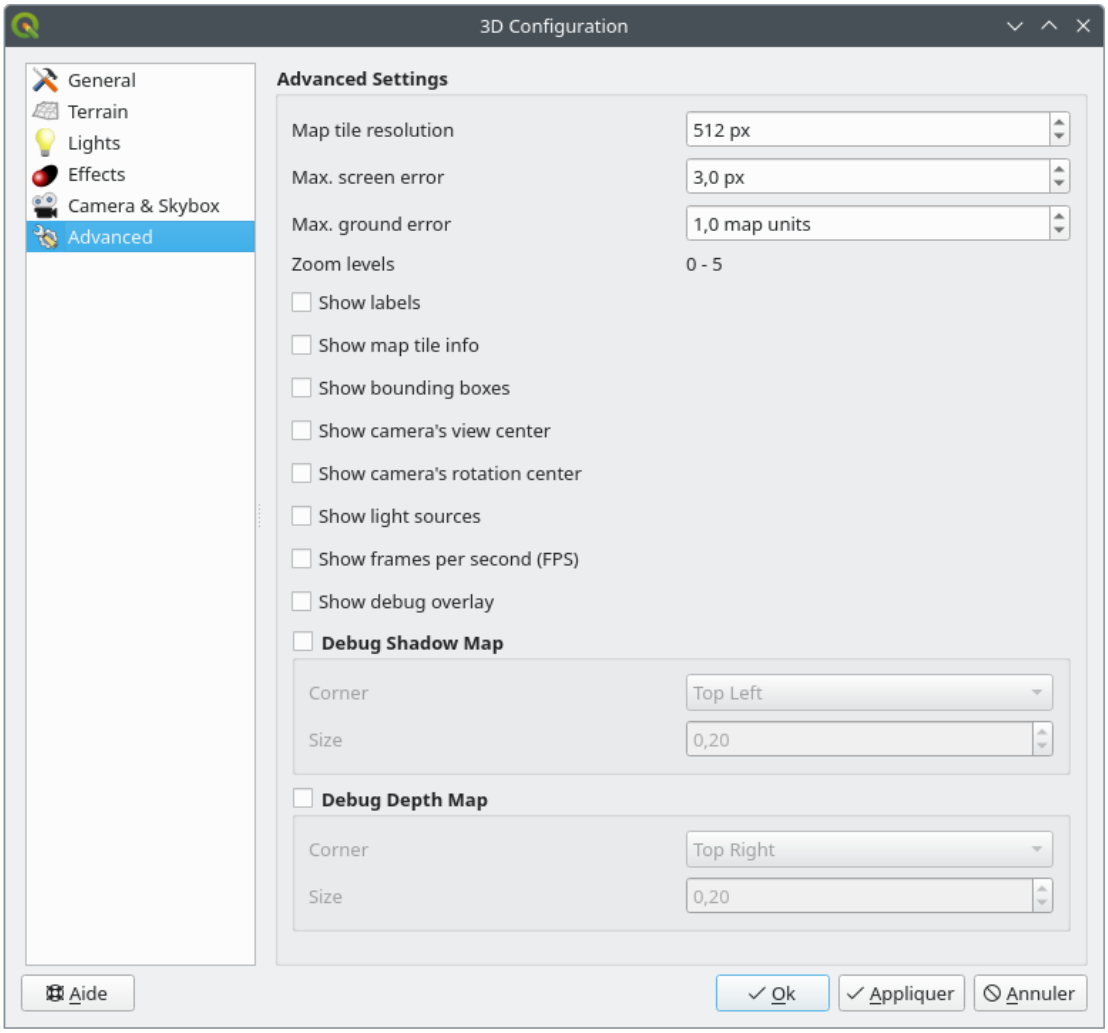


Fig. 7.30: Dialogrutan Avancerad konfiguration för 3D-karta

- *Kartplattornas upplösning*: Bredd och höjd på de 2D-kartbilder som används som texturer för terrängplattorna. 256px innebär att varje bricka renderas till en bild på 256x256 pixlar. Högre siffror skapar mer detaljerade terrängplattor på bekostnad av ökad renderingskomplexitet.
- *Max. skärmfel*: Bestämmer tröskeln för att byta ut terrängplattor mot mer detaljerade plattor (och vice versa) - dvs. hur snart 3D-vyn ska använda plattor av högre kvalitet. Lägre siffror innebär fler detaljer i scenen på bekostnad av ökad renderingskomplexitet.
- *Max. markfel*: Upplösningen på terrängplattorna vid vilken uppdelningen av plattorna i mer detaljerade plattor kommer att upphöra (att dela upp dem skulle ändå inte ge några extra detaljer). Detta värde begränsar djupet i hierarkin av plattor: lägre värden gör hierarkin djup, vilket ökar renderingskomplexiteten.
- *Zoom-nivåer*: Visar antalet zoomnivåer (beror på kartplattans upplösning och max. markfel).
- ☐ *Visa etiketter*: Aktiverar/inaktiverar kartetiketter
- ☐ *Visa info om kartplattor*: Inkludera gräns- och bricknummer för terrängbrickorna (användbart för felsökning av terrängproblem)
- ☐ *Visa begränsningsrutor*: Visa 3D-begränsningsrutor för terrängplattorna (användbart för felsökning av terrängproblem)

- *Visa kamerans bildcentrum*
- *Visa kamerans rotationscentrum*
- *Visa ljuskällor*: visar en sfär vid ljuskällans ursprung, vilket gör det lättare att flytta och placera ljuskällor i förhållande till sceninnehållet
- *Visa bilder per sekund (FPS)*
- *Visa felsökningsöverlägg*: visuellt överlägg som visar användbar felsökning- och profileringsinformation. Detta gör det möjligt att snabbt se ramgrafen och scengrafen.
- *Felsök skuggkarta*: återger scenen som en röd-svart bild från synvinkeln för det ljus som används för skuggor (för felsökning). Widgeten är inställd med en proportionell *Size* till 3D-kartvyn och dockad i en *Hörn*.
- *Felsök djupkarta*: återger scenens djupkarta som en bild där närmare pixlar är mörkare (för felsökning). Widgeten är inställd med en proportionell *Storlek* till 3D-kartvyn, och dockad i en *Hörn*.

7.2.2 Alternativ för navigering

För att utforska kartvyn i 3D:

- Luta terrängen (rotera den runt en horisontell axel som går genom mitten av fönstret)
  - Tryck på verktygen Tilt upp och Tilt ner
  - Tryck på `Shift` och använd upp-/nedknapparna
  - Dra musen framåt/bakåt med den mellersta musknappen nedtryckt
  - Tryck på `Shift` och dra musen framåt/bakåt med vänster musknapp nedtryckt
- Rotera terrängen (runt en vertikal axel som går genom mitten av fönstret)
  - Vrid kompassen i navigeringswidgeten till visningsriktningen
  - Tryck på `Shift` och använd vänster-/högerknapparna
  - Dra musen åt höger/vänster med den mittersta musknappen nedtryckt
  - Tryck på `Shift` och dra musen åt höger/vänster med vänster musknapp nedtryckt
- Ändra kamerans position (och vyns mittpunkt) genom att flytta runt den i ett horisontellt plan
  - Dra musen med vänster musknapp nedtryckt och Kamerakontroll-knappen aktiverad
  - Tryck på riktningspilarna på navigeringswidgeten
  - Använd knapparna upp/ner/vänster/höger för att flytta kameran framåt, bakåt, höger respektive vänster
- Ändra kamerans höjd: tryck på knapparna `Page Up`/`Page Down`
- Ändra kamerans orientering (kameran behålls i sin position men mittpunkten flyttas)
  - Tryck på `Ctrl` och använd piltangenterna för att vrida kameran uppåt, nedåt, åt vänster och höger
  - Tryck på `Ctrl` och dra musen med vänster musknapp nedtryckt
- Zooma in och ut
  - Tryck på motsvarande verktyg Zoom In och Zoom Out i navigeringswidgeten
  - Scrolla med mushjulet (håll `Ctrl` intryckt för finare zoomning)
  - Dra musen med höger musknapp intryckt för att zooma in (dra ner) och ut (dra upp)

För att återställa kameravyn klickar du på knappen Zoom Full högst upp på 3D-panelen.

7.2.3 Skapa en animation

En animation baseras på en uppsättning nyckelbilder - kamerapositioner vid särskilda tidpunkter. Så här skapar du en animering:

1. Slå på verktyget Animationer för att visa widgeten för animationsspelare
2. Klicka på knappen Add keyframe och ange en *Keyframe time* i sekunder. Kombinationsrutan *Keyframe* visar nu den inställda tiden.
3. Använd navigeringsverktygen för att flytta kameran till den position som ska associeras med den aktuella keyframe-tiden.
4. Upprepa de föregående stegen för att lägga till så många keyframes (med tid och position) som behövs.
5. Klicka på knappen för att förhandsgranska animeringen. QGIS genererar scener med hjälp av kamerans positioner/rotationer vid bestämda tidpunkter och interpolerar dem mellan dessa nyckelrutor. Det finns olika *Interpolation*-lägen för animeringar (t.ex. linjär, inQuad, outQuad, inCirc… – mer information finns på https://doc.qt.io/qt-5/qeasingcurve.html#EasingFunction-typedef).

   Animationen kan också förhandsgranskas genom att flytta tidsreglaget. Om du håller rutan *Loop* markerad kommer animationen att köras upprepade gånger medan du klickar på för att stoppa en pågående animering.

Klicka på Export animation frames för att generera en serie bilder som representerar scenen. Förutom filnamnet *Template* och *Output directory* kan du ställa in antalet *Frames per second*, *Output width* och *Output height*.

7.2.4 3D-vektorlager

Ett vektorlager med höjdvärden kan visas i 3D-kartvyn genom att markera *Enable 3D Renderer* i avsnittet *3D View* i vektorlagrets egenskaper. Det finns ett antal alternativ för att styra renderingen av 3D-vektorlagret.

7.3 Höjdprofil Vy

Panelen *Elevation Profile* är ett plottverktyg för sidovy, för att visualisera höjddata längs en linje. Den stöder vektor-, raster-, mesh- och punktmolnslager. Data kan vara av 2D- eller 3D-typ.

För att lägga till en höjdprofilvy, gå till *Visa* ► *Elevationsprofil* menyn. Du kan lägga till så många profilvyer du vill och de kan vara dockade, staplade på varandra eller flytande.

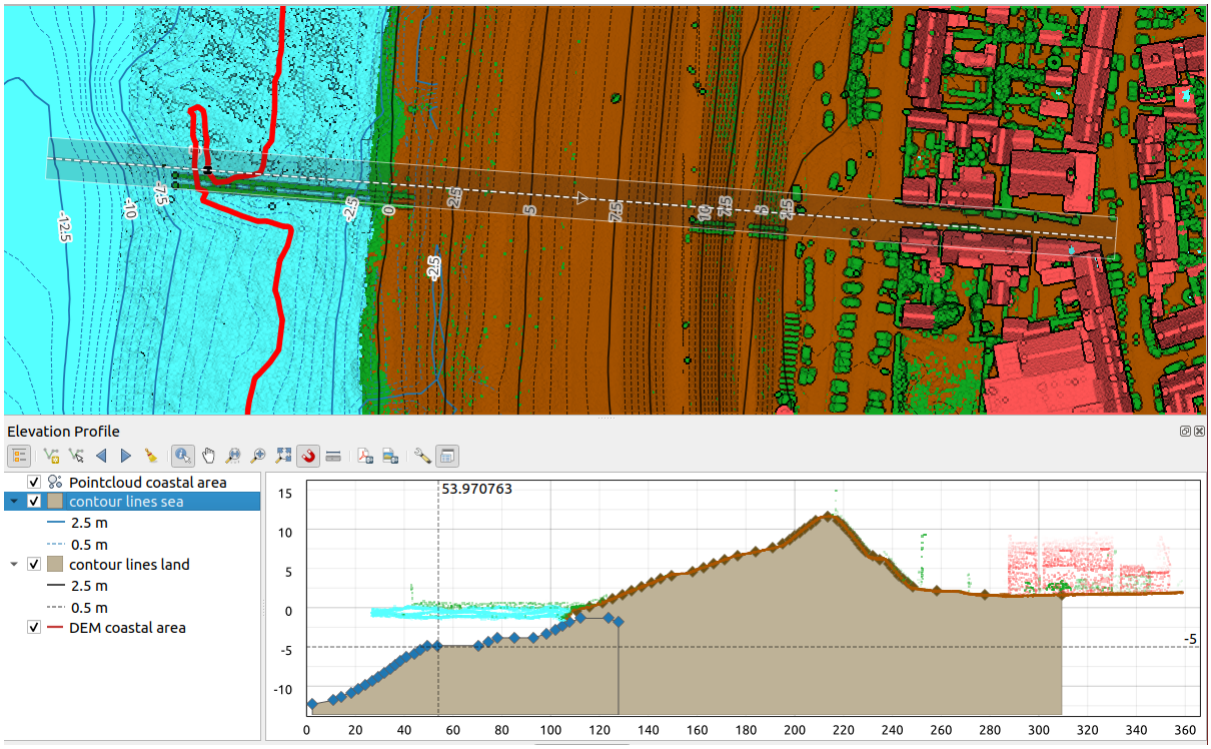


Fig. 7.31: Dialogruta för höjdprofil inbäddad under huvudkartan
Figuren visar en kustlinje med en stad bakom ett havsdike. En större tolerans som tillämpas på höjdprofillinjen returnerar fler punktmolnsfunktioner.

7.3.1 Gränssnittet

Överst på *Elevation Profile*-panelen finns ett verktygsfält med följande verktyg:

Table 7.5: Verktygsfält för vy av höjdprofil

Verktyg	Genväg	Beskrivning
Lägg till lager	håll in `Ctrl` medan du drar och släpper	Gör det möjligt att lägga till nya lager i en plot.
Visa lagerträdet		Visar eller döljer en lista över projektlager för att konfigurera rendering i profilvyn.
Fånga kurva		Ritar interaktivt en linje över kartbilden för att representera profilkurvan.
Fånga kurva från funktion		Välj en befintlig linjefunktion på kartbilden och skapa en profilkurva längs den linjen.
Putta åt vänster	`Ctrl+A]`	Gör det möjligt att sakta flytta upptagningslinjen över kartan till vänster (t.ex. för att hitta den optimala profillinjen baserat på höjden).
Putta åt höger	`Ctrl+A]`	Gör det möjligt att sakta flytta upptagningslinjen över kartan till höger (t.ex. för att hitta den optimala profillinjen baserat på höjden).
Töm		Tar bort profillinjen och alla diagram som visas i vyn *Elevation Profile* (höjdprofil)

continues on next page

Table 7.5 – fortsättning från föregående sida

Verktyg	Genväg	Beskrivning
Identifiera funktioner		Identifierar funktioner i plottduken med antingen ett enda klick eller en klick-och-drag-rektangel. Resultaten visas i standard *Identify Results* docka.
Panorera	`Space`	Klicka och dra för att panorera plottytan. Kan också göras med den mellersta musknappen.
Zoom X-axel		Zooma in/ut längs den horisontella axeln och behåll det vertikala förhållandet
zoom	`Ctrl+Sp`	Klicka eller klicka-och-drag en rektangel över diagrammet för att zooma in. Tryck på `Alt` och klicka för att istället zooma ut.
Zoom Full		Zoomar *Elevation Profile*-vyn till omfattningen av fångstlinjen
Aktivera snäppning		Gör det möjligt att snäppa till kanten eller toppunkten på plotelementen i profilvyn. Praktiskt för exakt hämtning av koordinater eller avståndsmätningar.
Mät avstånd		Mäter horisontella och vertikala avstånd
Exportera som PDF		Exporterar plottar till PDF (som högkvalitativa vektorobjekt)
Exportera som bild		Exporterar plottar till flera bildformat
Exportera resultat		Exporterar plottar som 3D-funktioner, 2D-profiler och avstånds-/höjdtabeller. (Resultaten kan sparas som DXF-filer, CSV-filer eller något av de skrivbara vektorformaten). Mer information finns på *Exportera från höjdprofilen*.
Options ►		Ger tillgång till konfigurationsinställningar för profilhöjningslinjen.
► *Lås skalor för avstånd/höjd*		Säkerställer att de horisontella och vertikala skalorna alltid är lika (så att t.ex. en 45° lutning visas som en 45° lutning i profilvyn).
► *Avståndsenheter*		Gör det möjligt att återge avstånd i profildiagrammet med andra enheter än kartans enheter.
► *Tolerans*		Ställer in hur långt från den faktiska profillinjen en funktion (vektorpunkt, linje eller polygon, punktmoln) kan ligga för att inkluderas i resultaten. Använder kartans enheter och ignoreras av andra lager- eller geometrityper.
► *Byt namn på profil…*		Gör det möjligt att byta namn på profilvyn.
Docka Elevation Profil View		Växla mellan dockad och flytande status för vyn

Längst ner till vänster kan en kopia av panelen *Layers* visas genom att trycka på knappen *Show Layer Tree*. Detta är dock en oberoende widget med sin egen uppsättning synliga lager i en anpassad stapelordning. Det gör att du kan kontrollera lagrens rendering och beteende inom plotcanvasen:

- Kryssa i rutan bredvid lagrets namn för att ange om det ska återges i plotcanvasen
- Dra och släpp lager upp eller ner för att ändra ordningen på lagren
- stilrendering av lager i profilvyn: dubbelklicka på ett lager eller högerklicka och välj *Properties…* för att öppna lagrets *Elevation* egenskapsflik för konfiguration. En sammanfattning av höjdinställningarna visas som verktygstips när du håller muspekaren över lagret.

Till höger om lagerträdet finns plot canvas som är den huvudsakliga platsen där du kan förhandsgranska höjdprofilen för de aktiverade lagren. Den är baserad på ett graderat rutnät där den horisontella axeln visar profillinjens längd och den vertikala axeln visar Z-höjden för de observerade objekten. Det är också möjligt att utföra en rad olika interaktioner, t.ex. zooma, panorera, mäta, identifiera funktioner, … med hjälp av verktygen längst upp.

7.3.2 Skapa en höjdprofil

För att skapa en profilvy kan du göra följande:

1. Gå till *View* ► *Elevation Profile* menyn. Panelen *Elevation profile* öppnas.
2. Skapa den profillinje längs vilken terrängen och objekten ska återges. Välj ett ritverktyg:
   - Capture Curve: klicka vänster över huvudkartan för att lägga till hörn och klicka höger för att slutföra en linje som kommer att användas som profillinje
   - eller Capture Curve From Feature: klicka på en linjefunktion på kartbilden för att välja den som profillinje. Om flera objekt finns vid den klickade punkten visas en popup-meny där du kan välja bland dem.

   Alla linjedigitaliseringsfunktioner som *snapping options*, *tracing*, *digitizing techniques* eller den avancerade *digitizing panel* är tillgängliga för användning.

   Plot canvas kan börja rendera vissa funktioner.
3. Nästa steg är att konfigurera höjdegenskaperna för de lager som du vill visualisera.
   1. Tryck på knappen Show Layer Tree för att visa listan över lager. Som standard laddas inte alla lager i projektet och refereras till i profilverktyget; t.ex. ignoreras rasterlager som inte är markerade som att de har höjddata i sina egenskaper. Om du klickar på knappen Add layers visas en filtrerad lista över möjliga lager som kan läggas till i plotten, men som för närvarande inte finns med i plotten. Om du använder dialogrutan med valda lager markeras de automatiskt som lager med höjddata och läggs omedelbart till i plotten.

      Tips

      Om du håller in `Ctrl`-tangenten kan du dra och släppa ytterligare lager från huvudlagerträdet till ett lager i höjdprofilverktyget.

   2. Växla synlighet för de lager du är intresserad av. Dessa är de enda som återges i profilvyn och valda lager kan skilja sig från huvudpanelen *Layers*.
   3. Dubbelklicka på ett lagernamn eller högerklicka och välj *Properties*. Fliken *Elevation* properties för lagret öppnas. Det är här du konfigurerar hur varje objekt eller terräng ska återges i profilvyn. Vilka egenskaper som är tillgängliga beror på lagertyp:
      - *Rasterhöjdsegenskaper*
      - *Vector Elevation Properties*
      - *Egenskaper för punktmolnens höjd*
      - *Mesh Elevation Properties*

      Profilvyn börjar rendera terräng eller funktioner i aktiva lager som korsar den givna profilkurvan, så snart de har konfigurerade höjdegenskaper.
4. Under Options rullgardinsmeny kan du ställa in värdet för *Tolerance*. Detta värde används för att skapa en platt buffert runt höjdprofillinjen, synlig i huvudkartan. Alla synliga objekt (punkt, linje, polygon, etc.) som överlappar denna buffert kommer att fångas upp i plottbilden.

Observera

Begränsningar med polygonextrudering

Geometriextrudering kan ställas in i egenskaperna *Elevation* för ett lager och återges i profilvyn. När toleransen är aktiverad är det dock inte trivialt att rendera extruderade polygoner, så för närvarande ignoreras polygonextrudering.

7.3.3 Interagera med profilen Vyer

När en höjdprofillinje skapas zoomas plottytan ut till sin fulla utsträckning. På X-axeln kan du se profilens längd och på Y-axeln höjdintervallet mellan lägsta och högsta höjd som registrerats, båda i kartenheter.

När du flyttar muspekaren i höjdprofilvyn ser du två korsande punktlinjer:

- den vertikala linjen visar höjdinformationen
- den horisontella linjen visar avståndet från början av höjdprofilen Line

När du flyttar muspekaren i höjdprofilen kan du också se en svart punkt röra sig längs höjdprofillinjen på huvudkartan. I mitten av profillinjen finns en pil som anger dess riktning.

Precis som för huvudkartan tillhandahåller QGIS medel för att navigera på plotkartan:

- Pan används för att flytta höjdprofilens utsträckning i vilken riktning du vill. Om du håller ned `Space` samtidigt som du flyttar musen flyttas också plottytans omfattning.
- Zoom X Axis används för att zooma in längs den horisontella axeln, samtidigt som förhållandet för den vertikala axeln (höjden) förblir detsamma. Vänsterklicka för att sträcka ut plotten längs X-axeln, med den klickade punkten i mitten av axeln, eller dra en rektangel för att sträcka ut plotten längs X-axeln till rektangelns bredd. Håll inne `Alt` medan du använder Zoom X Axis för att zooma ut längs den axeln.
- Zoom används för att zooma in på en viss punkt (med vänsterklick) eller till en viss utsträckning (genom att dra en rektangel över området). Håll in `Alt` medan du använder Zoom för att istället zooma ut. I kombination med `Ctrl` kan du zooma in eller ut på ett smidigare sätt.
- Zoom Full är den standardzoomnivå som används i början och visar profillinjens hela utsträckning, med alla återgivna funktioner. Använd den för att återställa zoomnivån.

Det är också möjligt att interagera med de element som visas i plotcanvasen:

- Tryck på knappen Enable snapping för att exakt fånga punkter, toppar eller kanter på objekten, för en exakt mätning eller koordinatrapport.
- Identify Features används för att identifiera funktioner i de synliga lagren i lagerträdet. Du kan dra en rektangel över flera funktioner i profilvyn för att söka efter dem alla. När formatet är kompatibelt (t.ex. vektor, punktmoln) kommer dessa funktioner att markeras i huvudkartan.
- Measurement Distances: klicka på eller välj två punkter i plottytan för att rapportera det horisontella *Distance*, *Elevation* och *Total distance* mellan dem, i kartenheter.
- Nudge Left och Nudge right används för att flytta höjdprofillinjens position i kartbilden till vänster eller höger. Kartbilden ritas om och visar funktioner och terräng som överlappar profillinjens buffert. Linjen flyttas i sidled med hjälp av värdet *Tolerance* i menyn Options.

Varning

Om du stänger en höjdprofilvy eller projektet tas vyn bort från projektet.

För mer information, ta en titt på QGIS elevation profile/cross section tool – a deep dive!, en presentation gjord av Nyall Dawson.

7.3.4 Exportera från höjdprofilen

I det övre verktygsfältet i dialogrutan *Elevation profile* finns en mängd olika verktyg för att exportera en höjdprofil, så att du kan återanvända plottarna i en annan programvara:

- Exportera som PDF: Exporterar plottar till PDF (som högkvalitativa vektorobjekt). Följande alternativ begärs för att generera utdatafilen:
  - *Sidstorlek*, välja en sida från en fördefinierad lista eller ange egna mått och sidans orientering
  - *Chart ranges* för att styra omfattningen av det diagram som ska exporteras:
    - på X-axeln, *Minimum distance* och *Maximum distance* från profilkurvans startpunkt
    - På Y-axeln visas *Minsta höjd* och *Maxsta höjd*
  - Med *Distance axis settings*: kan du konfigurera den horisontella graderingen av det rutnät som höjdprofilen återges över. Du kan anpassa *Label interval*, *Major grid line interval* och *Minor grid line interval*.
  - Inställningarna för *Elevation axis*, samma som ovan, för vertikal gradering
- Exportera som bild: Exporterar plottar till ett bildfilsformat. Specifika *Exportalternativ* såsom *Bildbredd* och *Bildhöjd* begärs, tillsammans med *Kartområden*, *Inställningar för avståndsaxel* och *Inställningar för höjdaxel* som tidigare nämnts.
- Exportera resultat: Exporterar profilresultaten som en uppsättning funktioner i ett vektorlager (`.DXF`, `.CSV`, `.SHP`, `.GPKG`, …). Beroende på utdataformat kan flera lager returneras när geometrityperna för exporterade funktioner skiljer sig åt.
  - *Exportera 3D-funktioner*: Exporterar profillinjer som 3D-funktioner, med höjdvärden som hämtas från höjdskivorna och lagras i den exporterade geometrins Z-värden.
  - *Exportera 2D-profil*: Exporterar profiler som 2D-profilinjer, med höjd lagrad i den exporterade geometrins Y-dimension och avstånd i X-dimension (som visas i höjdprofilwidgeten).
  - *Exportera tabell för avstånd/höjd*: Exporterar profiler som en tabell med samplade värden för avstånd och höjd.

KAPITEL 8

Allmänna verktyg

8.1 Hjälp med kontext

När du behöver hjälp med ett specifikt ämne kan du komma åt motsvarande sida i den aktuella användarhandboken via knappen *Help* som finns i de flesta dialogrutor — observera att tillägg från tredje part kan peka på särskilda webbsidor.

8.2 Paneler

Som standard erbjuder QGIS många paneler att arbeta med. Några av dessa paneler beskrivs nedan medan andra finns i olika delar av dokumentet. En fullständig lista över standardpaneler som tillhandahålls av QGIS finns tillgänglig via menyn *View* ► *Panels* ► och nämns i *Paneler*.

8.2.1 Panel för lager

Panelen *Layers* (även kallad `kartlegenden`) visar alla lager i projektet och hjälper dig att hantera deras synlighet och forma kartan. Du kan visa eller dölja panelen genom att trycka på `Ctrl+1`.

QGIS erbjuder en mängd olika sätt att lägga till lager i ett projekt:

- med hjälp av knappen *Add* på fliken Dedicated Data Provider i dialogrutan *Data Source Manager*
- från QGIS *Browser panel* eller *DB Manager*: dubbelklicka, dra och släpp filer och lager till QGIS eller använd snabbmenyn
- dra och släpp filer från operativsystemets filutforskare till QGIS

I alla dessa scenarier kan du öppna ett eller flera lager åt gången. Nya lager läggs till i panelen *Layers*:

1. om de släpps över *Layers*-panelen, på den exakta plats där de släpps
2. i övriga fall, på en plats som respekterar den globala *beteende som används när nya lager läggs till*-inställningen
3. och när det gäller flera lager sorteras de på ett sätt som ökar chansen att staplingen är logisk och att funktionerna är synliga så mycket som möjligt, enligt följande logik (uppifrån och ner):
    - vektorpunkt lager
    - vektorlinje lager

- vektor polygon lager
- lager av punktmoln
- nätlager
- rasterlager

Överst i panelen *Layers* finns ett verktygsfält där du kan göra följande:

- Öppna dockan för lagerstyling (F7): slå på och av panelen *Layer Styling*.
- Lägg till ny grupp: se *Interagera med grupper och lager*
- Manage Map Themes: kontrollera synligheten för lager och ordna dem i olika *map themes*.
- filterlager i förklaringsträdet:
  - *Filter Legend by Map Content*: endast de lager som är synliga och vars funktioner skär den aktuella kartbilden får sin stil återgiven i lagerpanelen. Annars används en generisk NULL-symbol för lagret. Baserat på skiktsymboliken är detta ett bekvämt sätt att identifiera vilken typ av funktioner från vilka lager som täcker ditt intresseområde.
  - *Show Private Layers*: en praktisk genväg för att visa och interagera med *privata lager* i panelen *Layers* utan att ändra projektinställningarna.
  - *Show Broken Layers Only*: endast lager med brutna datakällor visas.
- Filter Legend by Expression: tillämpa ett uttryck för att ta bort stilar från det valda lagerträdet som inte har någon funktion som uppfyller villkoret. Detta kan användas för att markera funktioner som ligger inom ett visst område/funktion i ett annat lager. I listrutan kan du redigera och rensa det uttryck som används för tillfället.
- Expand All eller Collapse All lager och grupper i lagerpanelen.
- Remove Layer/Group för närvarande valt.

Fig. 8.1: Verktygsfältet Layer i panelen Layers

Observera

Verktyg för att hantera lagerpanelen är också tillgängliga för kart- och legendobjekt i utskriftslayouter

Konfigurera kartteman

Rullgardinsmenyn Hantera kartteman ger tillgång till praktiska genvägar för att manipulera synligheten för lagren i panelen *Lager*:

- *Visa alla lager*
- *Dölj alla lager*
- *Visa valda lager*
- *Dölj valda lager*

- *Toggle Selected Layers*: ändrar synligheten för det första markerade lagret i panelen och tillämpar detta tillstånd på de andra markerade lagren. Kan även nås via genvägen `Space`.
- *Toggle Selected Layers Independently*: ändrar synlighetsstatus för varje markerat lager
- *Dölj bortvalda lager*

Utöver den enkla kontrollen av lagrens synlighet kan du med hjälp av menyn Manage Map Themes konfigurera **kartteman** i teckenförklaringen och växla från ett karttema till ett annat. Ett karttema är en **snapshot** av den aktuella kartlegenden som registrerar:

- de lager som anges som synliga i *Layers*-panelen
- **och** för varje synligt lager:
  - referensen till *style* som tillämpas på lagret
  - stilens synliga klasser, dvs. de lagermarkerade nodobjekten i *Layers-panelen*. Detta gäller för *symbologier* andra än rendering med en enda symbol
  - det kollapsade/expanderade tillståndet för lagernoden/lagernoderna och gruppen/grupperna som den är placerad i

För att skapa ett karttema:

1. Markera ett lager som du vill visa
2. Konfigurera lagrets egenskaper (symbologi, diagram, etiketter …) som vanligt
3. Expandera menyn *Style* ► längst ner och klicka på *Add…* för att lagra inställningarna som *en ny stil inbäddad i projektet*

   Observera

   Ett karttema kommer inte ihåg de aktuella detaljerna i egenskaperna: endast en referens till stilnamnet sparas, så när du gör ändringar i lagret medan den här stilen är aktiverad (t.ex. ändrar symbologin) uppdateras karttemat med ny information.

4. Upprepa de föregående stegen efter behov för de andra lagren
5. Om tillämpligt, expandera eller kollapsa grupper eller synliga lagernoder i *Layers*-panelen
6. Klicka på knappen Manage Map Themes högst upp på panelen och *Add Theme…*
7. Ange namnet på karttemat och klicka på *OK*

Det nya temat listas i den nedre delen av rullgardinsmenyn .

Du kan skapa så många kartteman som du behöver: när den aktuella kombinationen i kartlegenden (synliga lager, deras aktiva stil, kartlegendenoderna) inte matchar något befintligt karttema enligt definitionen ovan, klicka på *Add Theme…* för att skapa ett nytt karttema, eller använd *Replace Theme* ► för att uppdatera ett karttema. Du kan byta namn på det aktiva karttemat med *Rename Current Theme…* eller använda knappen *Remove Current Theme* för att ta bort det.

Kartteman är användbara för att snabbt växla mellan olika förkonfigurerade kombinationer: välj ett karttema i listan för att återställa dess kombination. Alla konfigurerade teman är också tillgängliga i utskriftslayouten, vilket gör att du kan skapa olika kartobjekt baserat på specifika teman och oberoende av den aktuella renderingen av huvudcanvasen (se *Kartobjektslager*).

Översikt över snabbmenyn i panelen Lager

Huvudkomponenten i panelen Layers längst ned i verktygsfältet är en ram som visar alla lager som har lagts till i projektet, eventuellt organiserade i grupper. Ett lager med en markerad ruta bredvid visas med ett innehåll som överlappar kartans omfattning, såvida inte en *skalbaserad synlighet* har angetts. Ett lager kan markeras och dras upp eller ned i teckenförklaringen för att ändra Z-ordningen. Z-ordning innebär att lager som är listade längst upp i förklaringen ritas över lager som är listade längre ned i förklaringen. Ett lager eller en grupp av lager kan också dras över flera QGIS-instanser.

Observera

Z-ordningsbeteendet kan åsidosättas med *Layer Order*-panelen.

Beroende på vilket objekt som har valts i panelen visas en särskild uppsättning alternativ som presenteras nedan när du högerklickar.

Table 8.1: Kontextuella menyer från *Layers*-panelobjekt

Alternativ	Grupp	Vektorlager	Rasterlager	Nätlager	Punktmolnslager	3D-lage
Zoom till lager/grupp	☑	☑	☑	☑	☑	☑
Zoom till urval		☑				
Visa i översikt		☑	☑	☑	☑	☑
Visa antal funktioner		☑				
Visa etikett		☑		☑		
Kopieringslager/grupp	☑	☑	☑	☑	☑	☑
Rename Layer/Group	☑	☑	☑	☑	☑	☑
Zoom till ursprunglig upplösning (100%)			☑			
Sträcka ut med nuvarande omfattning			☑			
Uppdatera SQL-lager…		☑				
Redigera virtuellt lager…		☑				
Lägg till grupp	☑					
Duplicate Layer		☑	☑	☑	☑	☑
Ta bort lager/grupp…	☑	☑	☑	☑	☑	☑
Flytta ut ur gruppen		☑	☑	☑	☑	
Flytta till toppen	☑	☑	☑	☑	☑	
Flytta till botten	☑	☑	☑	☑	☑	
Check och alla dess föräldrar		☑	☑	☑	☑	
Grupp vald		☑	☑	☑	☑	
Öppna attributtabell		☑				
Toggle Redigering		☑		☑		
Aktuella redigeringar ►		☑		☑		
Filter…		☑	☑		☑	
Ändra datakälla…		☑	☑	☑	☑	
Reparera datakälla…		☑	☑	☑	☑	

continues on next page

Table 8.1 – fortsättning från föregående sida

Alternativ	Grupp	Vektorlager	Rasterlager	Nätlager	Punktmolnslager	3D-lage
Åtgärder på urval ► (i redigeringsläge)		☑				
► *Duplicerat inslag*		☑				
► *Duplicera funktion och digitalisera*		☑				
Set Layer Scale Visibility…		☑	☑	☑	☑	☑
Zoom till synlig skala		☑	☑	☑	☑	
Lager CRS ►		☑	☑	☑	☑	☑
► *Ställ in projekt CRS från lager*		☑	☑	☑	☑	☑
► *Ställ in till…* (senaste CRS)				☑	☑	☑
► *Ställ in CRS-lagret…*		☑	☑	☑	☑	☑
Set Group CRS…	☑					
Set Group WMS Data…	☑					
☐ *Mutually Exclusive Group*	☑					
Check och alla dess barn (Ctrl-klick)	☑					
Avmarkera och alla dess barn (Ctrl-klick)	☑					
Göra permanent		☑				
Exportera ►	☑	☑	☑	☑	☑	☑
► *Spara som…*			☑			
► *Spara funktioner som…*		☑				
► *Spara valda funktioner som…*		☑				
► *Spara som lagerdefinitionsfil…*	☑	☑	☑	☑	☑	☑
► *Spara som QGIS Layer Style File…*		☑	☑	☑	☑	☑
Stilar ►		☑	☑	☑	☑	☑
► *Kopiera stil*		☑	☑	☑	☑	☑
► *Klistra in stil*	☑	☑	☑	☑	☑	
► *Lägg till…*		☑	☑	☑	☑	☑
► *Byt namn på nuvarande…*		☑	☑	☑	☑	☑
► *Redigera symbol…*		☑				
► *Kopiera symbol*		☑				
► *Klistra in symbol*		☑				
Lägg till lageranteckningar…		☑	☑	☑	☑	☑
Edit Layer Notes…		☑	☑	☑	☑	
Ta bort lageranteckningar		☑	☑	☑	☑	
Egenskaper…		☑	☑	☑	☑	☑

För GRASS-vektorlager är Toggle editing inte tillgängligt. Se avsnitt *Digitalisering och redigering av ett GRASS-vektorlager* för information om redigering av GRASS vektorlager.

Interagera med grupper och lager

Lagren i legendfönstret kan organiseras i grupper. Det finns olika sätt att göra detta på:

1. Tryck på ikonen för att lägga till en ny grupp. Skriv in ett namn på gruppen och tryck på `Enter`. Klicka nu på ett befintligt lager och dra det till gruppen.
2. Markera mer än ett lager och tryck sedan på ikonen . De markerade lagren infogas automatiskt i den nya gruppen.
3. Markera några lager, högerklicka i legendfönstret och välj *Group Selected*. De markerade lagren placeras automatiskt i en ny grupp.

Om du vill flytta ett lager ur en grupp drar du ut det eller högerklickar på det och väljer *Flytta ut ur gruppen*: lagret flyttas från gruppen och placeras ovanför den. Grupper kan också nästlas inuti andra grupper. Om ett lager placeras i en nästlad grupp kommer *Move Out of Group* att flytta lagret ut ur alla nästlade grupper.

Om du vill flytta en grupp eller ett lager till toppen av lagerpanelen kan du antingen dra det till toppen eller välja *Flytta till toppen*. Om du använder det här alternativet på ett lager som är inbäddat i en grupp flyttas lagret till toppen i den aktuella gruppen. Alternativet *Move to Bottom* följer samma logik för att flytta lager och grupper nedåt.

Kryssrutan för en grupp visar eller döljer de markerade lagren i gruppen med ett klick. Med `Ctrl` intryckt kommer kryssrutan också att slå på eller av alla lager i gruppen och dess undergrupper.

`Ctrl`-klick på ett markerat / avmarkerat lager avmarkerar / kontrollerar lagret och alla dess föräldrar.

Om du aktiverar alternativet **Mutually Exclusive Group** kan du göra så att en grupp bara har ett lager synligt samtidigt. När ett lager i gruppen görs synligt kommer de andra att växlas till att inte vara synliga.

Det är möjligt att välja mer än ett lager eller en grupp samtidigt genom att hålla ned `Ctrl`-tangenten medan du klickar på ytterligare lager. Du kan sedan flytta alla markerade lager till en ny grupp samtidigt.

Du kan också ta bort mer än ett lager eller en grupp samtidigt genom att markera flera objekt med `Ctrl` och sedan trycka på `Ctrl+D`: alla markerade lager eller grupper tas bort från lagerlistan.

Mer information om lager och grupper med hjälp av indikatorikonen

Under vissa omständigheter ändras formateringen eller ikonerna bredvid lagret eller gruppen i panelen *Layers* för att ge mer information om lagret/gruppen. Dessa element är:

- för att ange att lagret är i redigeringsläge och att du kan ändra data
- för att ange att lagret som redigeras har några osparade ändringar
- för att ange *ett filter* som tillämpas på lagret. Håll muspekaren över ikonen för att se filteruttrycket och dubbelklicka för att uppdatera frågan
- för att identifiera lager som är *required* i projektet, och därför inte kan flyttas
- för att identifiera en *inbäddad grupp eller lager* och sökvägen till deras ursprungliga projektfil
- för att identifiera ett lager vars datakälla inte var tillgänglig när projektfilen öppnades (se *Hantering av brutna filsökvägar*). Klicka på ikonen för att uppdatera källvägen eller välj *Repair Data Source…* i lagrets kontextuella meny.
- för att påminna dig om att lagret är ett *tillfälligt skrapskikt* och att dess innehåll kommer att kasseras när du stänger det här projektet. För att undvika dataförlust och göra lagret permanent klickar du på ikonen för att lagra lagret i något av GDAL-vektorformaten som stöds av QGIS.
- för att identifiera ett lager som används i *offline-redigeringsläge*.
- för att identifiera ett lager som inte har någon/okänd CRS

- för lager med koordinater som lagras i ett koordinatreferenssystem som har en inneboende låg noggrannhet (kräver att *motsvarande inställning* är aktiverad)
- för att identifiera ett temporalt lager som styrs av canvasanimering
- för att identifiera ett lager som har *notes* associerat
- Ett grått namn när kartans aktuella skala ligger utanför lagrets synlighetsskala (som anges i dess *Rendering*-egenskaper). Välj alternativet *Zoom to Visible Scale* i snabbmenyn för att zooma kartan till lagrets närmaste gräns för synlighetsskalan.

Styr rendering av lager genom gruppering

Grupper är ett sätt att strukturera lager i ett träd i projektet, men de kan också påverka hur deras komponentlager renderas, nämligen som ett enda platt objekt under kartrenderingar.

Alternativet för en sådan rendering är tillgängligt i panelen *Layer Styling* när en grupp väljs. Under fliken Symbology, markera *Render Layers as a Group* för att aktivera en uppsättning alternativ för att styra utseendet på de underordnade lagren som helhet, istället för enskilda lager:

- *Opacitet*: Funktioner från underordnade lager som skyms av andra underordnade lager förblir skymda, och opaciteten gäller endast för ”hela gruppen”.

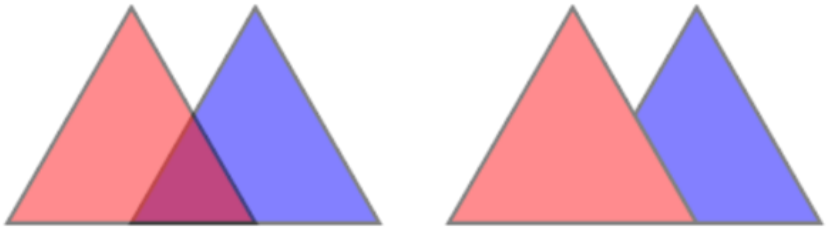


Fig. 8.2: Ställa in opacitet på lager kontra på en grupp
Bilden till vänster visar två lager som renderas med 50% opacity (underliggande funktioner är synliga, men halvmaskade av den 50% red funktionen ovanpå). Den andra bilden visar resultatet av att ställa in opaciteten på gruppen (delar av det blå underliggande barnlagret döljs helt av det röda lagret ovanpå och sedan renderas resultatet med 50% opacity).

- *Blandningslägen*: Precis som opacitet, att ställa in en *blend mode* (som multiply, overlay, …) för en hel grupp resulterar först i utplattning av funktioner i underordnade lager, med övre lager som döljer lägre. Renderingen erhålls sedan genom att blanda den platta gruppen och de lager som sitter under gruppen.
  - När barnlagren har tilldelats blandningslägen tillämpas det före utplattning, men omfattningen är begränsad till att endast påverka andra barnlager från den gruppen, och inte andra lager som sitter under hela gruppen.
  - Några fler *blending modes*-alternativ finns tillgängliga för underordnade lager i grupper, i deras *Symbology*-flik som utför ”clipping”-operationer på andra underordnade lager under renderingen. Du kan t.ex. klippa renderingen av innehållet i ett lager med innehållet i ett andra ”mask”-lager.
- *Layer effects*: tillämpar *effects* endast på den tillplattade renderingen av underordnade lager; Så t.ex. en droppskuggeffekt som tillämpas på gruppen skulle inte vara synlig för dolda underordnade lager.

När en grupp är inställd på *Render layers as a group*, visas endast gruppen i panellistan *Layer Order*. Gruppens barn kommer inte att synas i denna ordningslista, eftersom deras ordning bestäms av placeringen av gruppens lager.

Redigera stil för lager

Från panelen *Layers* har du genvägar för att snabbt och enkelt ändra lagerrenderingen.

Högerklicka på ett lager och välj *Styles* ► i listan för att:

- se de *stilar* som för närvarande är tillgängliga för lagret. Om du har definierat många stilar för lagret kan du växla från en stil till en annan och lagrets rendering kommer automatiskt att uppdateras på kartbilden.
- kopiera en del av eller hela den aktuella stilen och, i förekommande fall, klistra in en kopierad stil från ett annat lager
- *Rename current…* stil
- *Lägg till* en ny stil (som faktiskt är en kopia av den nuvarande)
- eller *Remove current* style (endast när flera stilar är tillgängliga).

> **Tips**
>
> **Snabbt dela ett lager stil**
>
> Kopiera stilen för ett lager från snabbmenyn och klistra in den i en grupp eller ett urval av lager: stilen tillämpas på alla lager som är av samma typ (vektor, raster, mesh, punktmoln, …) som det ursprungliga lagret och, för vektorlager, har samma geometrityp (punkt, linje eller polygon).

När du använder en symbologi baserad på klassificering av funktioner (t.ex. *categorized*, *graduated* eller *rule-based* för vektorlager, eller *classification* för punktmoln), kan du högerklicka på en klass i *Layers*-panelerna för att redigera synligheten för klasserna (och deras funktioner) och undvika att (av)markera dem en efter en:

- *Växla objekt*
- *Visa alla objekt*
- *Dölj alla objekt*

Med vektorlager ger den kontextuella menyn för en klassbladspost också tillgång till:

- *Välj funktioner*: väljer i lagret alla funktioner som matchar den klassen
- *Visa i attributtabell*: öppnar en attributtabell filtrerad till endast de funktioner som matchar den klassen
- uppdatera *symbolfärg* med hjälp av ett **färghjul**. För enkelhetens skull finns de senast använda färgerna också tillgängliga längst ner i färghjulet.
- *Edit Symbol…*: öppnar dialogrutan *Symbol Selector* för att ändra funktionssymbol (symbol, storlek, färg…).
- *Kopieringssymbol*
- *Pasta in symbol*

> **Tips**
>
> Genom att dubbelklicka på en klassbladspost öppnas också dialogrutan *Symbol Selector*.

8.2.2 Panel för lagerstyling

Panelen *Layer Styling* (även aktiverad med `Ctrl+3`) är en genväg till några av funktionerna i dialogrutan *Lageregenskaper*. Den ger ett snabbt och enkelt sätt att definiera rendering och beteende för ett lager och att visualisera dess effekter utan att behöva öppna dialogen för lageregenskaper.

Förutom att undvika den blockerande (eller ”modala”) dialogen för lageregenskaper, undviker lagerstylingpanelen också att skärmen fylls med dialogrutor och innehåller de flesta stilfunktioner (färgväljare, effektegenskaper, regelredigering, etikettbyte …): om du t.ex. klickar på färgknapparna i lagerstylingpanelen öppnas dialogrutan för färgväljare i själva lagerstylingpanelen i stället för som en separat dialogruta.

Välj ett objekt i en rullgardinslista med aktuella lager i lagerpanelen och:

- Beroende på det aktiva objektet, ställ in:
  - *Symbology* för grupper (se *Styr rendering av lager genom gruppering*)
  - *Symbology*, *Transparency*, och *Histogram* egenskaper för rasterlagret. Dessa alternativ är desamma som i *Dialog för egenskaper för raster*.
  - *Symbology*, *Labels*, *Mask* och *3D View* egenskaper för vektorlager. Dessa alternativ är desamma som i *Dialogrutan Egenskaper för vektor* och kan utökas med anpassade egenskaper som introduceras av tredjeparts tillägg.
  - *Symbology* och *3D View* egenskaper för mesh-lager. Dessa alternativ är desamma som i *Egenskaper för nätdataset*.
  - *Symbology*, *3D View* och *Elevation* egenskaper för punktmolnslagret. Dessa alternativ är desamma som i *Punktmoln Egenskaper*.
- Aktivera och konfigurera *global map shading* egenskaper
- Hantera den eller de associerade stilarna i *Style Manager* (mer information finns i *Hantera anpassade stilar*).
- Se historiken *History* över de ändringar som du har gjort i lagerstilen i det aktuella projektet: du kan därför avbryta eller återställa till valfritt tillstånd genom att markera det i listan och klicka på *Apply*.

En annan kraftfull funktion i den här panelen är kryssrutan *Live update*. Om du kryssar i den visas dina ändringar omedelbart på kartbilden: du behöver inte längre klicka på knappen *Apply*.

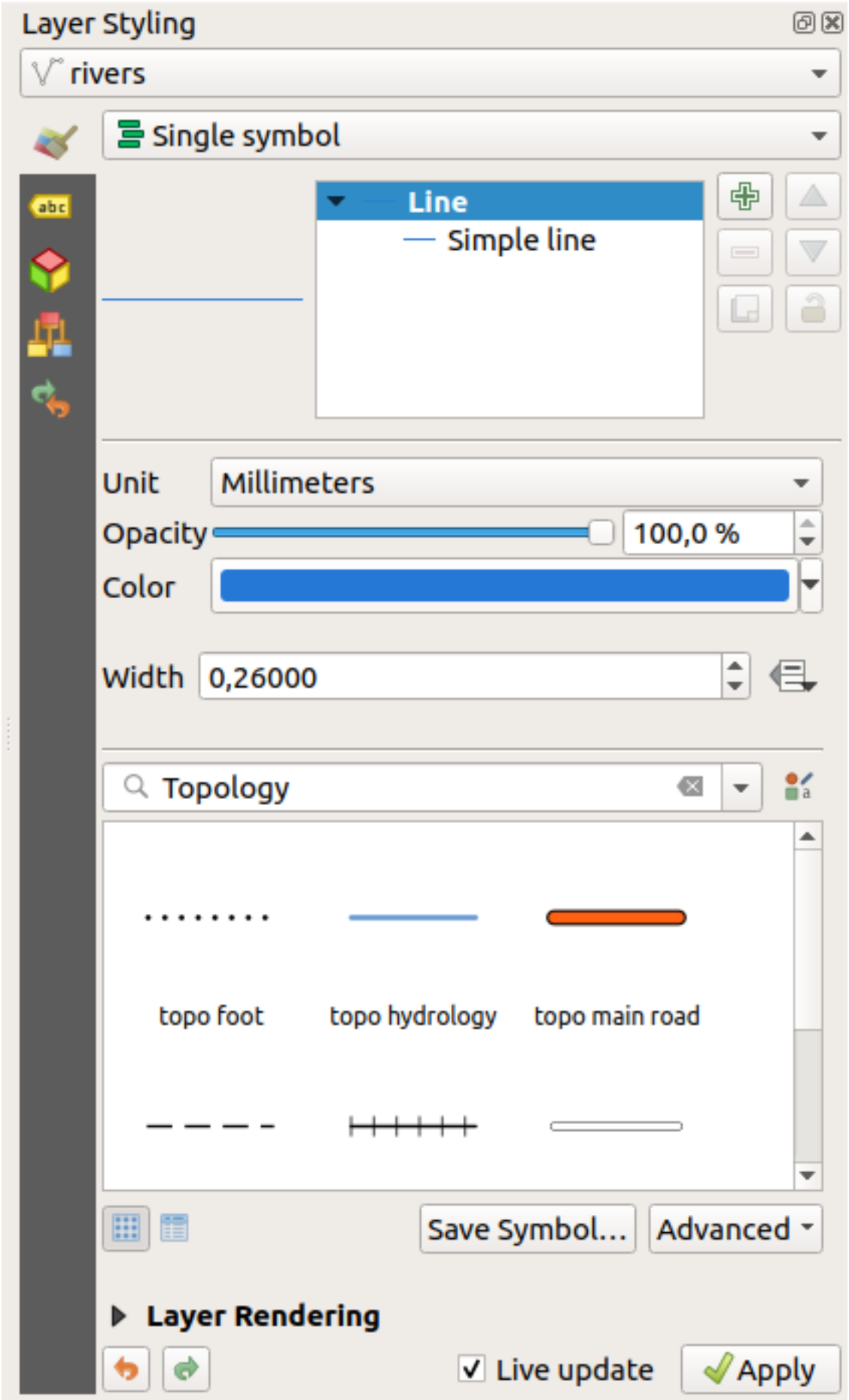


Fig. 8.3: Definiera ett lagers symbologi från panelen för lagerutformning

8.2.3 Panel för beställning av lager

Som standard ritas lager som visas på QGIS-kartvyn enligt deras ordning i panelen *Layers*: ju högre ett lager är i panelen, desto högre (och därmed mer synligt) kommer det att vara i kartvyn.

Du kan definiera en ritordning för lagren som är oberoende av ordningen i lagerpanelen med panelen *Lagerordning* aktiverad i menyn *Visa* ► *Paneler* ► eller med `Ctrl+9`. Markera *Control rendering order* under listan med lager och omorganisera lagren i panelen som du vill. Den här ordningen blir den som tillämpas på kartans canvas. I Fig. 8.4 kan du till exempel se att funktionerna `airports` visas över polygonen `alaska` trots dessa lagers respektive placering i panelen Lager.

Om du avmarkerar *Control rendering order* återgår du till standardbeteendet.

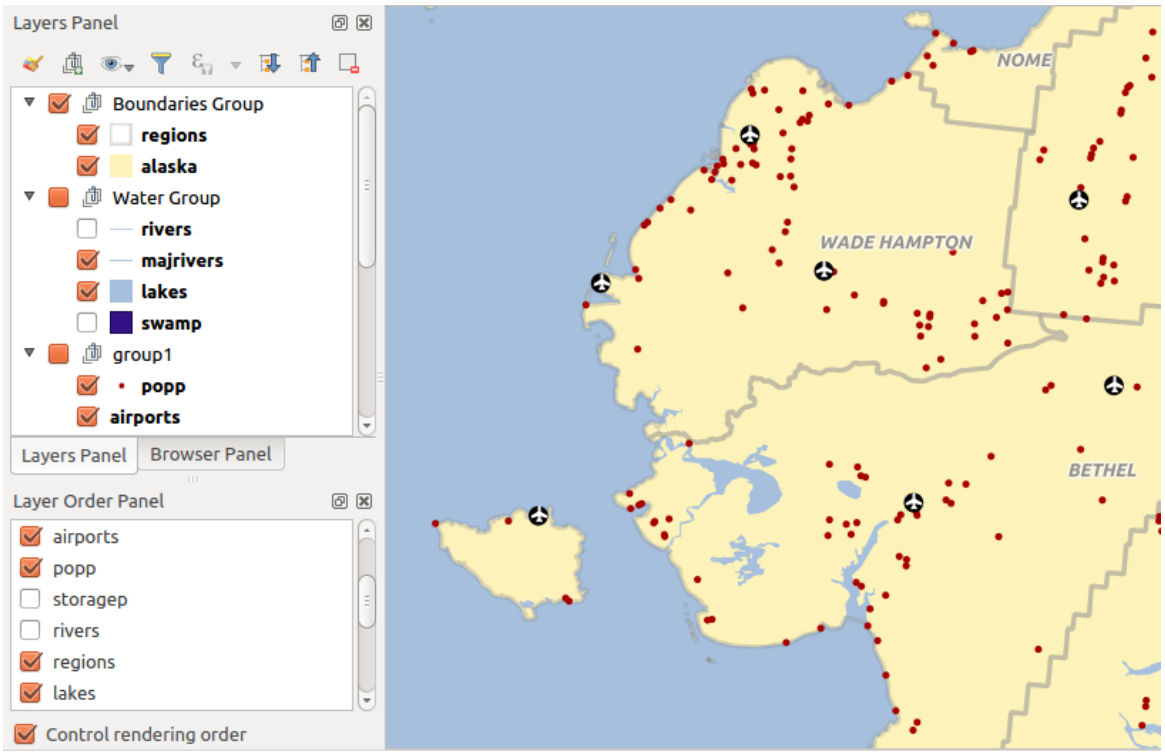


Fig. 8.4: Definiera en lagerordning som är oberoende av teckenförklaringen

8.2.4 Översiktspanel

Panelen *Overview* (`Ctrl+8`) visar en karta med en fullständig vy av några av lagren. Översiktskartan fylls med lager med hjälp av alternativet *Visa i översikt* i menyn *Layer* eller i den kontextuella menyn för lager. I vyn visar en röd rektangel den aktuella kartans omfattning, vilket hjälper dig att snabbt avgöra vilket område av hela kartan som du för närvarande tittar på. Om du klickar och drar i den röda rektangeln i översiktsramen uppdateras huvudkartans utbredning i enlighet med detta.

Observera att etiketter inte renderas till kartöversikten även om de lager som används i kartöversikten har ställts in för etikettering.

8.2.5 Panel för loggmeddelanden

När du laddar eller bearbetar vissa operationer kan du spåra och följa meddelanden som visas i olika flikar med hjälp av *Log Messages Panel*. Den kan aktiveras med hjälp av ikonen längst till höger i det nedre statusfältet.

8.2.6 Undo/Redo-panel

För varje lager som redigeras visar panelen *Undo/Redo* (`Ctrl+5`) listan över åtgärder som utförts, så att du snabbt kan ångra en uppsättning åtgärder genom att välja den åtgärd som anges ovan. Mer information finns i *Undo och Redo redigeringar*.

8.2.7 Panel för statistisk sammanfattning

Panelen *Statistics* (`Ctrl+6`) ger sammanfattad information om alla vektorlager. Med den här panelen kan du välja:

- det vektorlager som statistiken ska beräknas på: det kan väljas från den övre rullgardinsmenyn eller synkroniseras med det aktiva lagret i panelen *Layers* med kryssrutan *Follow selected layer* längst ner i rullgardinsmenyn för statistik
- fältet eller *uttryck* som ska användas: för varje lager kommer den sista posten ihåg och beräknas automatiskt när lagret väljs om.
- den statistik som ska returneras med hjälp av rullgardinsmenyn längst ned till höger i dialogrutan. Beroende på fältets (eller uttryckets) typ av värden finns följande statistik tillgänglig:

Table 8.2: Statistik tillgänglig för varje fälttyp

Statistik	Text	Heltal	Flytande	Datum
Antal	☑	☑	☑	☑
Räkna Distinkt Värde	☑			☑
Antal Saknat värde	☑	☑	☑	☑
Sum		☑	☑	
Genomsnitt		☑	☑	☑
Standardavvikelse		☑	☑	
Standardavvikelse på prov		☑	☑	
Minimalt värde	☑	☑	☑	☑
Maximalt värde	☑	☑	☑	☑
Intervall		☑	☑	☑
Minoritet	☑	☑	☑	
Majoritet	☑	☑	☑	
Sort (variety)		☑	☑	
Första kvartilen		☑	☑	
Tredje kvartilen		☑	☑	
Interkvartilintervall		☑	☑	
Minilängd	☑			
Maxlängden	☑			
Genomsnittlig längd	☑			

Den statistiska sammanfattningen kan vara:

- returneras för hela lagret eller *Bara utvalda funktioner*
- kopieras till urklipp och klistras in som en tabell i ett annat program
- omräknas med hjälp av knappen när den underliggande datakällan ändras (t.ex. nya eller borttagna funktioner/fält, attributändring)

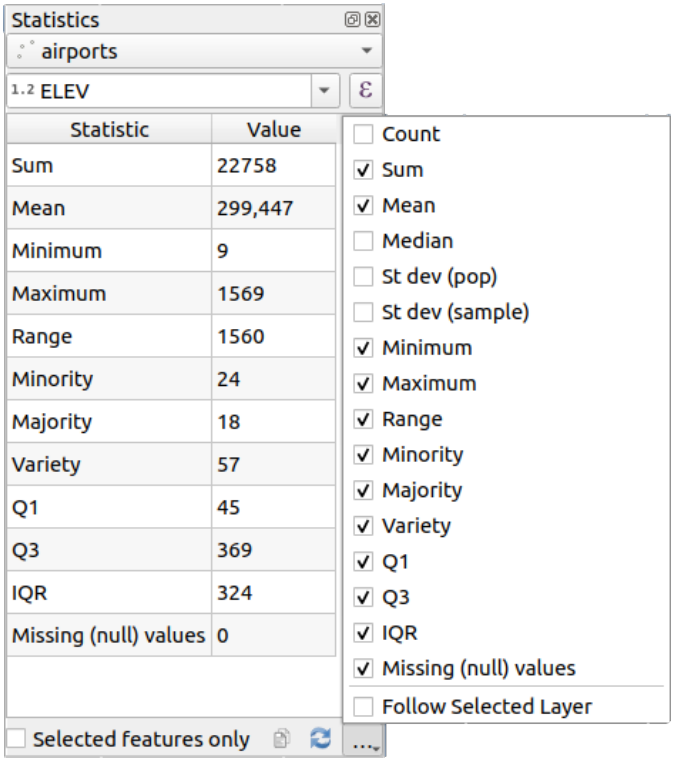


Fig. 8.5: Visa statistik för ett fält

8.2.8 Panel för felsöknings-/utvecklingsverktyg

Panelen *Debugging/Development Tools* (`F12`) ger en samlad plats för hantering och felsökning av åtgärder inom QGIS. Tillgängliga verktyg är organiserade under följande flikar:

- *Network Logger*: registrerar och visar nätverksförfrågningar och interaktioner
- *Query Logger*: loggar SQL-kommandon som utfärdas av dataleverantörer och hur länge de körs
- *Profiler*: ger laddningstider för alla åtgärder för att upptäcka orsaker till långsamhet

Observera

Tilläggsskapare kan utöka panelen med anpassade flikar för felsökning och utveckling av sina egna tillägg. Detta görs med hjälp av `registerDevToolWidgetFactory`-metoden.

Nätverkslogger

Fliken *Network Logger* hjälper dig att registrera och visa nätverksförfrågningar, tillsammans med en hel del användbara detaljer som status för förfrågan och svar, rubrik, fel, SSL-konfigurationsfel, timeouts, cachestatus etc.

Från det övre verktygsfältet kan du:

- *Record Log*: som startar eller stoppar loggningen.
- *Clear Log*: rensar logghistoriken.
- *Save Log…*: visar först en stor varning om att loggen är känslig och bör behandlas som konfidentiell och låter dig sedan spara loggen.
- Tryck på rullgardinsmenyn *Inställningar* för att välja om du vill *Visa lyckade förfrågningar*, *Visa timeouts* och *Visa svar som serveras från cacheminnet*.
- *Inaktivera cache*: inaktiverar cacheminnet så att varje förfrågan måste utföras.
- *Filtrera förfrågningar* baserat på delmängder av URL-strängar eller status för förfrågan

Genom att högerklicka på en förfrågan kan du:

- *Öppna URL* som öppnar URL:en i din standardwebbläsare.
- *Kopiera URL*
- *Copy As cURL* för att använda den i terminalen.
- *Copy as JSON*: kopierar trädvärdena som en json-sträng till urklipp, så att de enkelt kan klistras in i felrapporter eller för fjärrassistans.

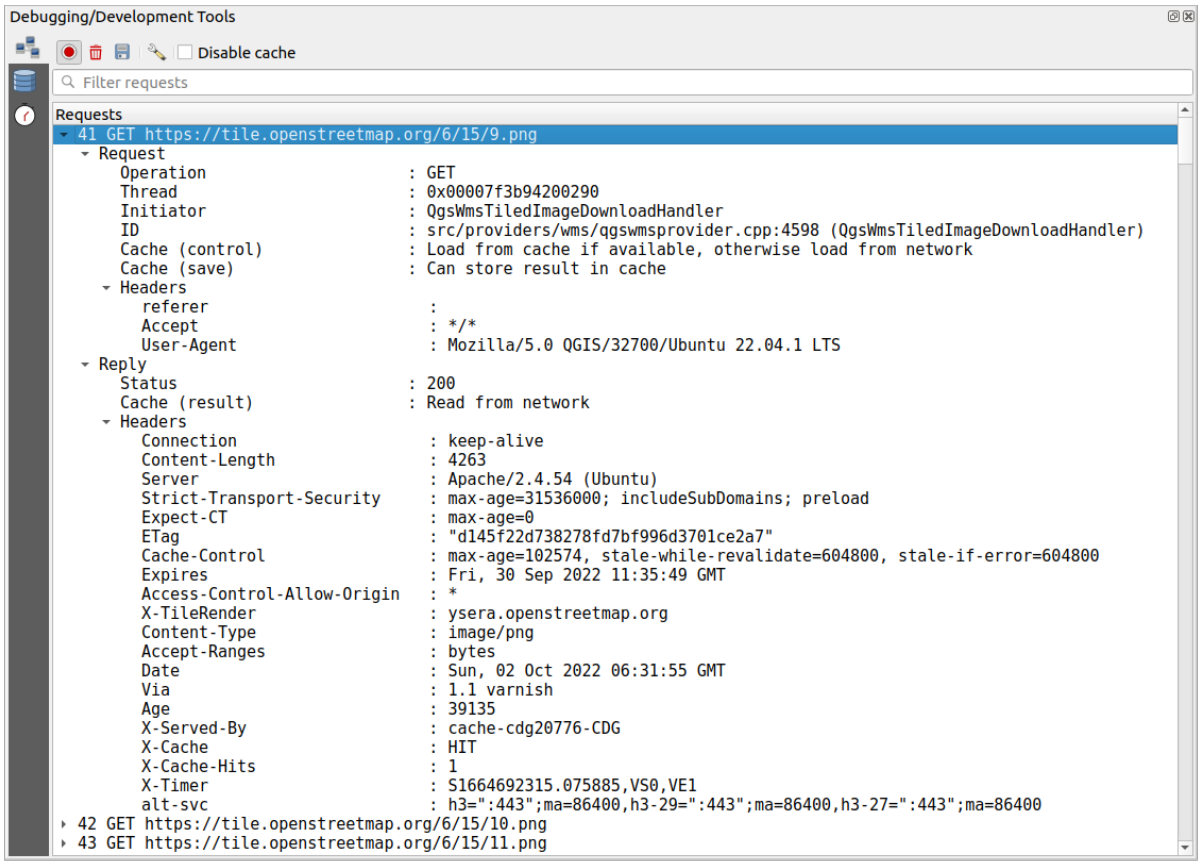


Fig. 8.6: Utdata från nätverkslogger för GET-begäran

Frågeloggare

Query Logger är platsen för att logga SQL-kommandon som skickas av dataleverantören och anslutnings-API:et till backend-databasen, tillsammans med deras exekveringstid som den mäts av QGIS (dvs. i den klient som skickade kommandona). Detta kan vara användbart när man undersöker prestanda för ett visst lager under felsökning eller utveckling av en QGIS-algoritm eller tillägg.

Från det övre verktygsfältet kan du:

- *Record Log*: som startar eller stoppar loggningen.
- *Clear Log*: rensar logghistoriken.
- *Save Log…*: visar först en stor varning om att loggen är känslig och bör behandlas som konfidentiell och låter dig sedan spara loggen.
- *Filtrera frågor* baserat på frågesträngens delmängder eller detaljer som leverantörstyp, starttid, initiativtagare, …

Högerklicka på en rapporterad fråga så kan du göra det:

- *Copy SQL*-kommando som anropas av QGIS på databasen
- *Copy as JSON*: kopierar trädvärdena som en json-sträng till urklipp, så att de enkelt kan klistras in i felrapporter eller för fjärrassistans.

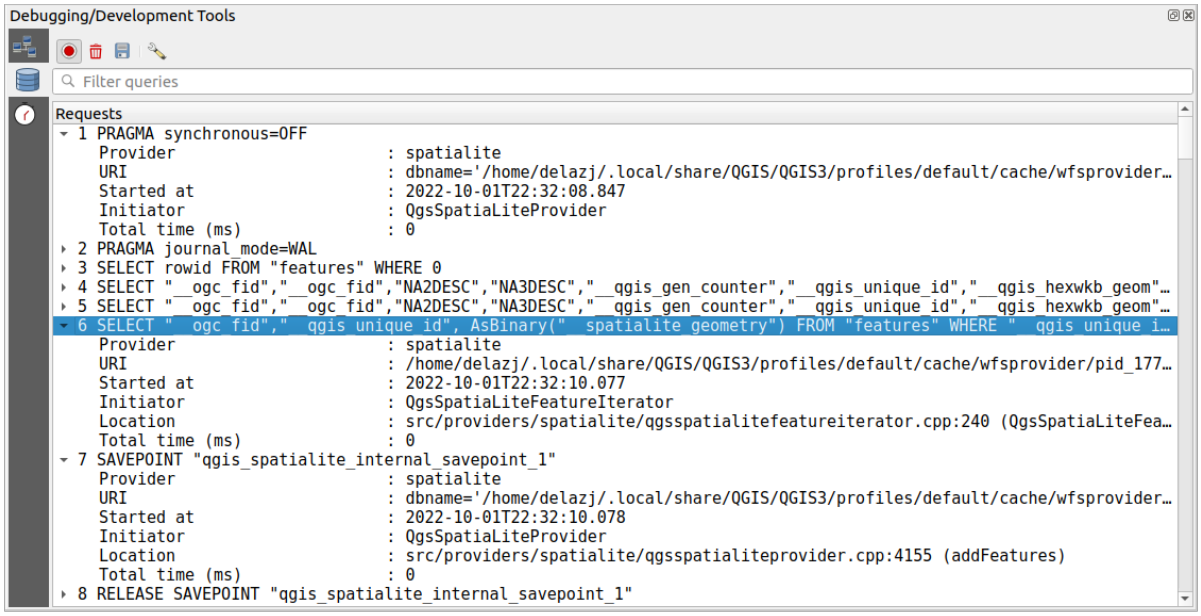


Fig. 8.7: Query Logger-utdata

Profilerare

Fliken *Profiler* gör det möjligt att få laddningstider för varje enskild operation som är involverad i åtgärder som begärs av användaren. Beroende på sammanhanget kan dessa åtgärder vara läsning av inställningar, skapande av meny, kartduk eller 3D-vyer, referenslösning för kartlager, laddning av bokmärken eller layout, … Detta hjälper till att identifiera orsaker till långsamma laddningstider.

Standardåtgärder som stöds kan väljas från rullgardinsmenyn *Category*:

- *Startup* hjälper dig att diagnostisera (och åtgärda) ibland långa uppstartstider för QGIS.
- *Project Load* ger dig möjlighet att få en uppdelning av de olika stegen i projektbelastningen, för att identifiera orsakerna till långsamma projektbelastningstider.
- Med verktyget *Map Render* kan du identifiera smärtpunkter i din kartrendering och spåra exakt vilka lager som orsakar långa omdragningar av kartan.

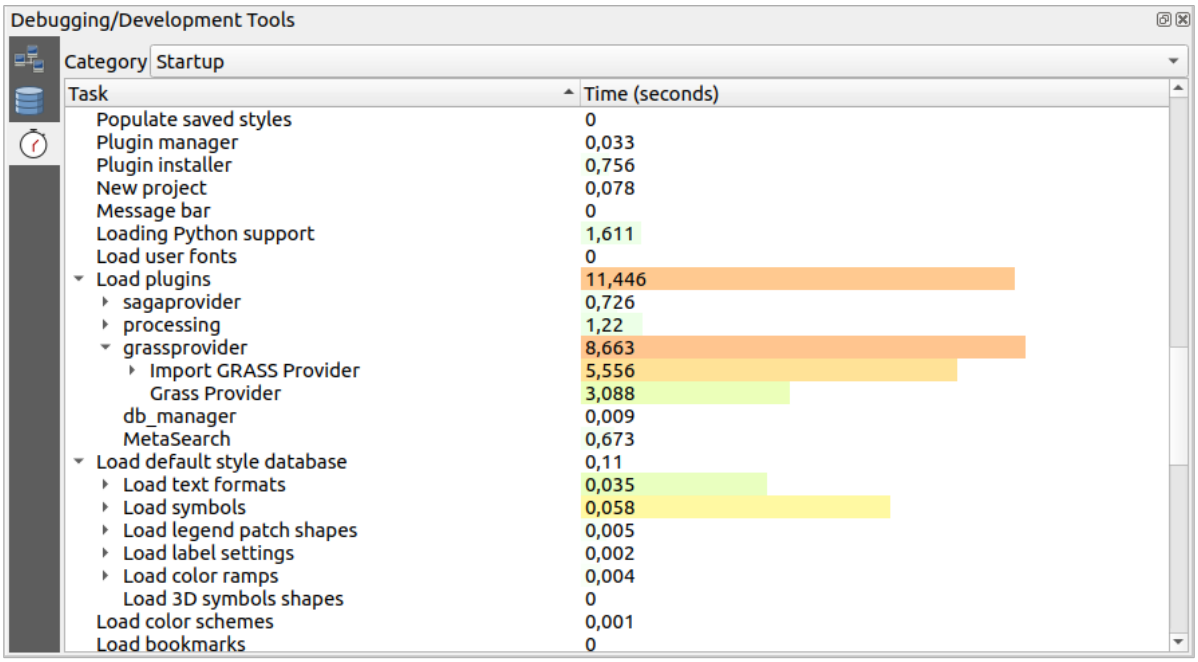


Fig. 8.8: Profiler för uppstart av QGIS

8.3 Inbäddning av lager från externa projekt

Ibland vill du behålla vissa lager i olika projekt, men med samma stil. Du kan antingen skapa en *standardstil* för dessa lager eller bädda in dem från ett annat projekt för att spara tid och ansträngning.

Att bädda in lager och grupper från ett befintligt projekt har vissa fördelar jämfört med styling:

- Alla typer av lager (vektor eller raster, lokalt eller online …) kan läggas till
- Genom att hämta grupper och lager kan du behålla samma trädstruktur för ”bakgrunds”-lagren i dina olika projekt
- Även om de inbäddade lagren är redigerbara kan du inte ändra deras egenskaper, t.ex. symbologi, etiketter, formulär, standardvärden och åtgärder, vilket säkerställer enhetlighet i olika projekt
- Ändra objekten i det ursprungliga projektet och ändringarna sprids till alla andra projekt

Om du vill bädda in innehåll från andra projektfiler i ditt projekt väljer du *Layer* ► *Embed Layers and Groups*:

1. Klicka på knappen … för att söka efter ett projekt: du kan se innehållet i projektet (se Fig. 8.9)
2. Håll ned `Ctrl` (eller `Cmd`) och klicka på de lager och grupper du vill hämta
3. Klicka på *OK*

De valda lagren och grupperna bäddas in i panelen *Layers* och visas på kartbilden. En ikon läggs till bredvid deras namn för igenkänning och om du håller muspekaren över visas ett verktygstips med den ursprungliga sökvägen till projektfilen.

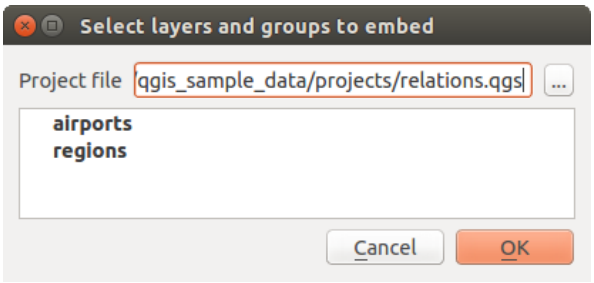


Fig. 8.9: Välj lager och grupper som ska bäddas in

Precis som alla andra lager kan ett inbäddat lager tas bort från projektet genom att högerklicka på lagret och klicka på Ta bort.

Tips

Ändra rendering av ett inbäddat lager

Det är inte möjligt att ändra renderingen av ett inbäddat lager om du inte gör ändringarna i den ursprungliga projektfilen. Om du högerklickar på ett lager och väljer *Duplicate* skapas dock ett lager som är fullt utrustat och inte beroende av originalprojektet. Du kan sedan ta bort det länkade lagret på ett säkert sätt.

8.4 Interagera med funktioner

8.4.1 Välja funktioner

QGIS tillhandahåller flera verktyg för att markera objekt på kartbilden. Markeringsverktygen finns tillgängliga i menyn *Edit* ► *Select* eller i verktygsfältet *Selection Toolbar*.

Observera

Markeringsverktygen arbetar med det aktiva lagret.

Manuell markering på kartbilden

Om du vill välja en eller flera funktioner med musen kan du använda något av följande verktyg:

- Välj funktioner efter område eller enstaka klick
- Välj funktioner efter polygon
- Välj funktioner med frihand
- Välj funktioner efter radie

Observera

Förutom Välj funktioner efter polygon, kan du med dessa manuella urvalsverktyg välja objekt på kartbilden med ett enda klick.

Observera

Använd verktyget Select Features by Polygon för att använda en befintlig polygonfunktion (från valfritt lager) för att välja överlappande funktioner i det aktiva lagret. Högerklicka i polygonen och välj den från snabbmenyn som visar en lista över alla polygoner som innehåller den klickade punkten. Alla överlappande funktioner från det aktiva lagret väljs.

Tips

Använd verktyget *Edit* ► *Select* ► *Reselect Features* för att göra om ditt senaste urval. Mycket användbart när du omsorgsfullt har gjort ett urval och sedan klickar någon annanstans av misstag och rensar ditt urval.

När du använder verktyget *Select Feature(s)* kan du hålla in `Shift` eller `Ctrl` (`Cmd` på macOS) för att växla om en funktion är markerad (dvs. antingen läggs till i det aktuella urvalet eller tas bort från det).

För de andra verktygen kan olika beteenden utföras genom att hålla ned:

- `Shift`: lägg till funktioner i det aktuella urvalet
- `Ctrl`/`Cmd`: subtrahera funktioner från det aktuella urvalet
- `Ctrl+Shift`/`Cmd+Shift`: skär med det aktuella urvalet, dvs. behåll endast överlappande funktioner från det aktuella urvalet
- `Alt`: välj funktioner som är helt inom urvalets form. Kombinerat med `Shift` eller `Ctrl`/`Cmd` kan du lägga till eller dra ifrån funktioner till/från det aktuella urvalet.

Automatisk markering

De andra urvalsverktygen, de flesta tillgängliga från *Attributtabellen*, utför ett urval baserat på en funktions attribut eller dess urvalstillstånd (observera att attributtabellen och kartbilden visar samma information, så om du väljer en funktion i attributtabellen kommer den också att väljas på kartbilden):

- Select By Expression… välj funktioner med hjälp av uttrycksdialog
- Välj funktioner efter värde… eller tryck på `F3`

- Deselect Features from Alla lager eller tryck på `Ctrl+Alt+A` för att avmarkera alla markerade funktioner i alla lager
- Välj bort funktioner från det aktuella aktiva lagret eller tryck på `Ctrl+Shift+A`
- Markera allt Features eller tryck på `Ctrl+A` för att välja alla funktioner i det aktuella lagret
- Invert Feature Selection för att invertera urvalet i det aktuella lagret
- Select by Location för att välja funktioner baserat på deras spatiala förhållande till andra funktioner (i samma eller ett annat lager - se *Välj efter plats*)
- Select within distance för att välja funktioner varhelst de befinner sig inom det angivna maximala avståndet från refererade funktioner - se *Välj inom avstånd*)

Om du t.ex. vill hitta regioner som är stadsdelar från `regions.shp` i QGIS exempeldata kan du göra det:

1. Använd ikonen Välj funktioner med hjälp av ett uttryck
2. Utöka *Fält och värden*-gruppen
3. Dubbelklicka på det fält som du vill fråga (”TYPE_2”)
4. Klicka på *All Unique* i panelen som dyker upp till höger
5. Dubbelklicka på ”Borough” i listan. I redigeringsfältet *Expression* skriver du följande fråga:

```
"TYPE_2"  =  'Borough'
```

6. Klicka på *Välj funktioner*

I dialogen för uttrycksbyggare kan du också använda *Funktionslista* ► *Recent (Selection)* för att göra ett urval som du har använt tidigare. Dialogrutan kommer ihåg de 20 senaste uttrycken som använts. Se *Uttryck* för mer information och exempel.

Tips

Spara ditt val i en ny fil

Användare kan spara valda funktioner i ett **Nytt temporärt Scratch-lager** eller ett **Nytt vektorlager** med hjälp av *Redigera* ► *Kopiera funktioner* och *Redigera* ► *Paste Features as* i önskat format.

Välj funktioner efter värde

Detta urvalsverktyg öppnar lagrets funktionsformulär så att användaren kan välja vilket värde som ska sökas i varje fält, om sökningen ska vara skiftlägeskänslig och vilken operation som ska användas. Verktyget har också autokomplettering, vilket automatiskt fyller i sökrutan med befintliga värden.

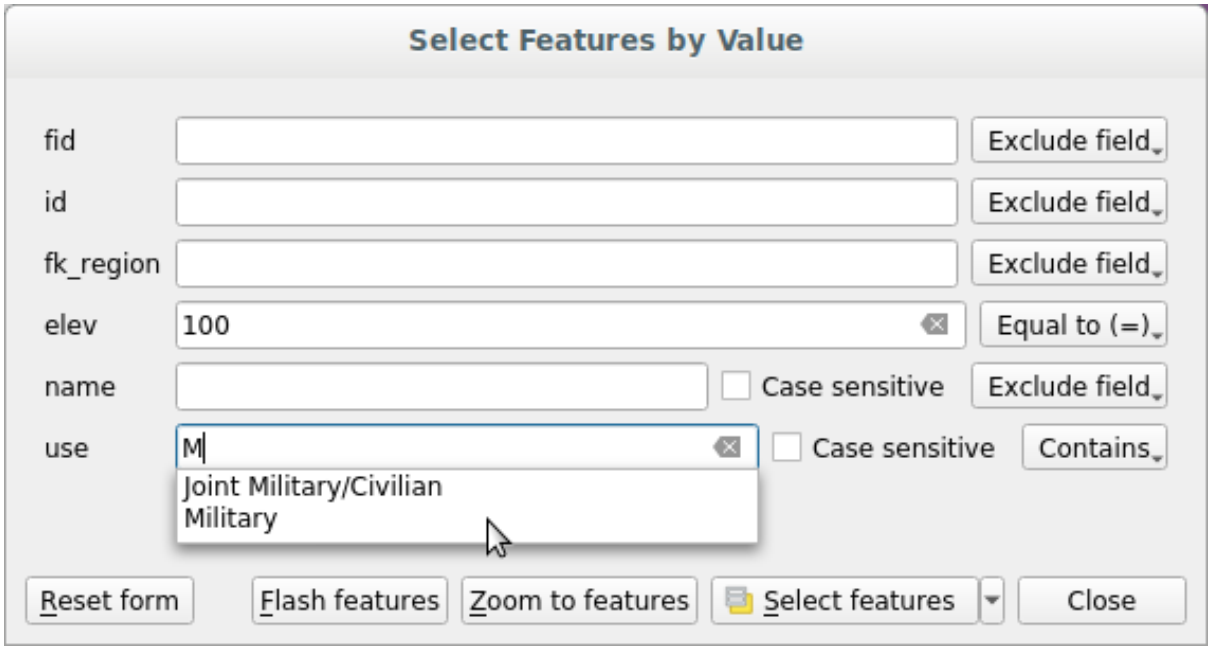


Fig. 8.10: Filtrera/välja funktioner med hjälp av formulärdialog

Vid sidan av varje fält finns en rullgardinslista med alternativ för att styra sökbeteendet:

Table 8.3: Frågeoperatorer per datatyp

Alternativ för fältsökning	Text	Numeric	Datum
Uteslut fält från sökningen	☑	☑	☑
Likvärdig med (=)	☑	☑	☑
Inte lika med (≠)	☑	☑	☑
Större än (>)		☑	☑
Mindre än (<)		☑	☑
Större än eller lika med (≥)		☑	☑
Mindre än eller lika med (≤)		☑	☑
Mellan (inklusive)		☑	☑
Inte mellan (inklusive)		☑	☑
Innehåller	☑		
Innehåller inte	☑		
Det saknas (null)	☑	☑	☑
Är inte saknad (inte null)	☑	☑	☑
Börjar med	☑		
Slutar med	☑		

För strängjämförelser är det också möjligt att använda alternativet ☑ *Case sensitive*.

När du har ställt in alla sökalternativ klickar du på *Select features* för att välja de matchande funktionerna. Alternativen i rullgardinsmenyn är:

- *Välj funktioner*
- *Lägg till i aktuellt urval*

- *Remove from current selection* (Ta bort från aktuellt urval)
- *Filtrera aktuellt urval*

Du kan också rensa alla sökalternativ med hjälp av knappen *Reset form*.

När villkoren är fastställda kan du också antingen:

- *Zoom till funktioner* på kartbilden utan att behöva göra ett förval
- *Flash features*, markerar de matchande funktionerna. Det här är ett praktiskt sätt att identifiera en funktion utan att markera den eller använda verktyget Identifiera. Observera att flashningen inte ändrar kartans utbredning och endast är synlig om objektet ligger inom gränserna för den aktuella kartans utbredning.

8.4.2 Identifiering av funktioner

Verktyget Identify Features gör att du kan interagera med kartbilden och få information om funktioner eller pixlar i ett popup-fönster. Det kan användas för att ställa frågor om de flesta av de lagertyper som stöds av QGIS (vektor, raster, mesh, punktmoln, wms, wfs, …). För att identifiera ett element, använd antingen:

- *Visa* ► *Identifiera funktioner*
- `Ctrl+Shift+I` (eller `Cmd+Shift+I`),
- Identifiera funktioner-knappen i verktygsfältet *Attribut*

Klicka sedan på en funktion eller pixel i det aktiva lagret. Det identifierade objektet markeras i kartbilden medan dialogrutan *Identify Results* öppnas med detaljerad information om det. Dialogrutan innehåller också en uppsättning knappar för avancerad konfiguration.

Dialogrutan Identifiera resultat

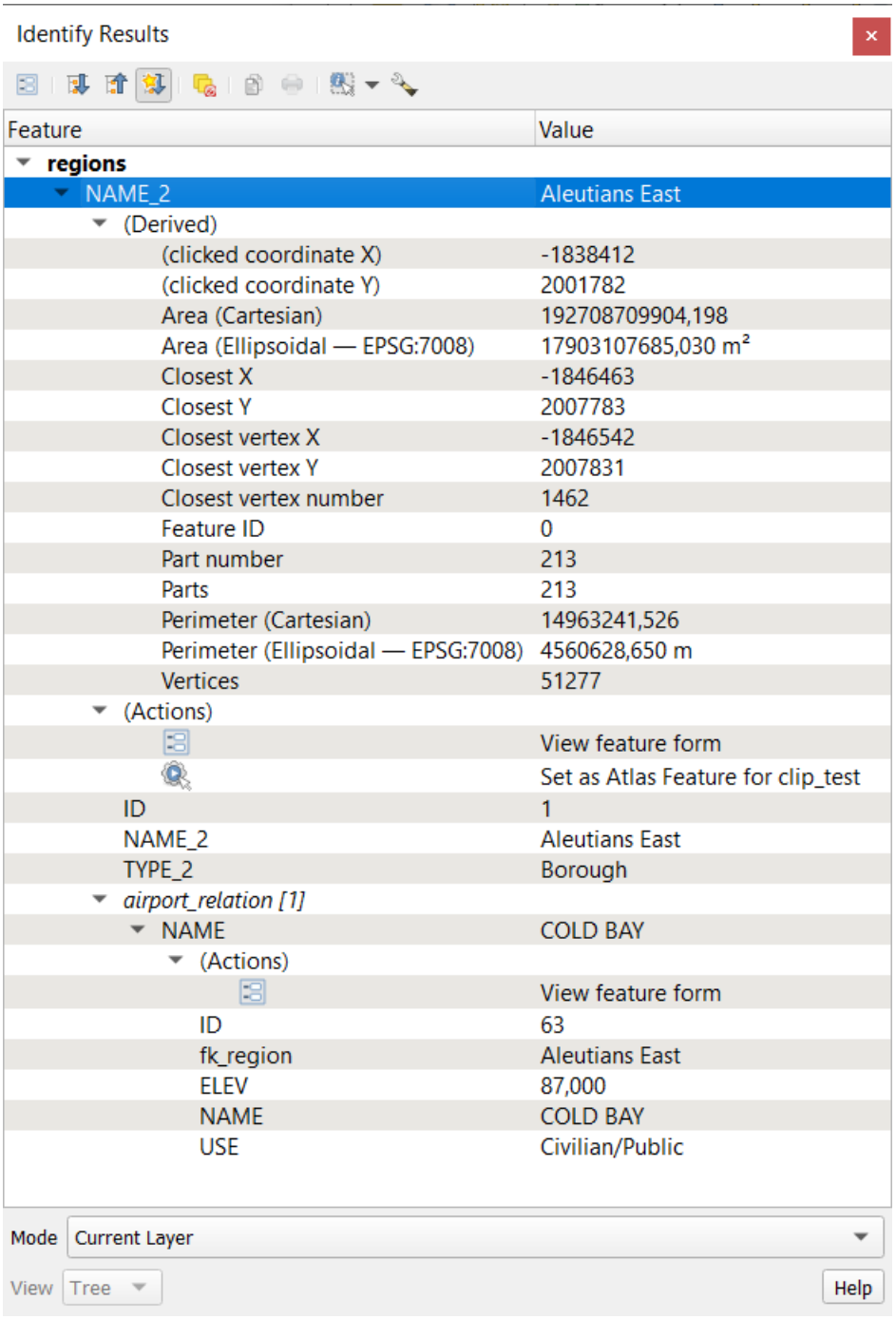


Fig. 8.11: Identifiera resultatdialog

Från botten till toppen:

- *View* styr den allmänna aspekten av dialogrutan och formateringen av resultaten; den kan ställas in som:
  - **Tree**: detta är standardvyn och visar resultaten i en trädstruktur
  - **Table**: endast tillgängligt för rasterbaserade lager, gör det möjligt att visa resultaten som en tabell vars kolumner är `Layer`, `FID`, `Attribute` och `Value`
  - eller **Graph**: endast tillgängligt för rasterbaserade lager

- Med *Mode* kan du välja de lager från vilka resultat kan returneras. Dessa lager bör vara synliga, så att data visas i kartbilden, och ha *identifierbara* från *Projektegenskaper* ► *Datakällor* ► *Lagerfunktioner*. Tillgängliga lägen är:
  - **Aktuellt lager**: endast det eller de lager som valts i *Layers*-panelen ger resultat. Om en grupp väljs, hämtas resultaten från dess bladlager.
  - **Top down, stop at first**: resultaten kommer från lagret med den översta funktionen eller pixeln under musen.
  - **Ovanifrån och ned**: resultaten kommer från de lager som har en funktion eller pixel under musen.
  - **Layer selection**: öppnar en kontextuell meny där användaren väljer det lager som ska användas för att identifiera funktioner. Om bara en enda funktion finns under musen visas resultaten automatiskt.
- I den övre delen av dialogrutan *Identify Results* visas en ram med *informationen* som returnerats av funktioner som en tabell, en graf eller ett träd, beroende på *vald vy*. I en trädvy har du en handfull verktyg ovanför resultaten:
  - Öppna formulär för den aktuella funktionen
  - Expandera träd
  - Kollapsa träd
  - Expand New Results by Default för att definiera om informationen för nästa identifierade funktion ska kollapsas eller expanderas
  - Rensa resultat
  - Kopiera den identifierade funktionen till urklipp, lämplig för att klistra in i ett kalkylblad.
  - Skriv ut valt HTML-svar: en textbaserad formatering av resultaten för att skriva ut på papper eller spara som en `.PDF`-fil
  - *interaktiva identifieringsverktyg*: en rullgardinsmeny med verktyg för att välja funktioner eller pixlar på kartbilden som ska identifieras
  - Under Identify Settings kan du aktivera om du vill:
    - *Öppna formuläret automatiskt för resultat från enstaka objekt*: Om rutan är markerad öppnas ett formulär med dess attribut varje gång en enskild funktion identifieras. Det här är ett praktiskt sätt att snabbt redigera attributen för en funktion.
    - *Dölj härledda attribut från resultat* för att bara visa fält som faktiskt definierats i lagret
    - *Dölj NULL-värden från resultat*
  - Hjälp för att komma åt den aktuella dokumentationen

Använda verktyget Identifiera funktioner

I standardvisningen (*View: Tree*) erbjuder panelen *Identify Results* flera verktyg för att interagera med de lager som ska undersökas. En smart kombination av dessa verktyg med *target layers selector* kan avsevärt förbättra identifieringsarbetet:

- Identifiera funktion(er) genom att klicka eller klicka och dra
  - ett enda klick eller klicka och dra: överlagrade funktioner i mållagren returneras
  - högerklicka: överlagrade funktioner från mållagren listas i snabbmenyn, grupperade efter lager. Du kan sedan välja att:
    - visa resultatet för en specifik funktion,

    * visa resultatet för alla funktioner i ett specifikt lager,
    * för vektorlager är det också möjligt att öppna dess attributtabell filtrerad till de returnerade funktionerna
    * eller visa alla returnerade funktioner.

- Identify Features on Mouse over: flytta muspekaren över kartbilden så markeras de objekt i mållagren som muspekaren håller över och visas i resultatpanelen.
- Identify Features by Polygon: returnerar objekt som överlappar en ritad eller markerad polygon.
  - Rita en polygon (vänsterklicka för att lägga till en punkt, högerklicka för att stänga polygonen) och alla överliggande funktioner från mållagren markeras och visas i resultatpanelen.
  - Högerklicka och du får en lista över alla synliga polygonfunktioner i projektet under klicket. Välj en post och QGIS kommer att returnera alla funktioner från mållagren som överlappar den valda polygonen.
- Identify Features by Freehand: returnerar objekt som överlappar en polygon ritad på fri hand. Rita en polygon (vänsterklicka för att starta, flytta pekaren för att forma området och högerklicka för att stänga polygonen). Alla överlappande objekt från mållagren markeras och returneras i resultatpanelen.
- Identify Features by Radius returnerar objekt som överlappar en ritad cirkel. Rita en cercle (vänsterklicka för att ange mittpunkten, flytta pekaren för att forma området eller ange radien i popup-textrutan och vänsterklicka eller tryck på `Enter` för att validera cirkeln). Alla överlagrade funktioner från mållagren markeras och returneras i resultatpanelen.

Tips

Filtrera de lager som ska sökas med verktyget Identify Features

Under *Projekt* ► *Egenskaper…* ► *Datakällor* ► *Lagerförmågor*, avmarkera kolumnen *Identifierbara* bredvid ett lager för att undvika att det efterfrågas när du använder verktyget Identifiera funktioner. Detta är ett praktiskt sätt att returnera funktioner från endast lager som är av intresse för dig.

Information om funktionen

När du identifierar data i kartbilden kommer dialogrutan *Identify Results* att visa information om de objekt som du klickat på (eller hållit muspekaren över, beroende på vilket verktyg som används). Standardvyn är en trädvy där det första objektet är namnet på lagret och dess underordnade objekt är de identifierade funktionerna. Varje funktion beskrivs av namnet på ett fält tillsammans med dess värde. Det här fältet är det som ställts in i *Lageregenskaper* ► *Display*. All övrig information om objektet följer.

Den funktionsinformation som visas av identifieringsverktyget beror på vilken typ av lager du har valt, om det är ett vektorlager (inklusive vektorplattor eller punktmolnsdata) eller ett rasterlager. Om ditt lager är ett rasterlager kommer den identifierade rasterpixeln att markeras om du klickar på en plats på kartbilden med verktyget identify. Dialogrutan Identify Results kan anpassas så att den visar anpassade fält, men som standard visas följande information:

- Funktionen *display name*;
- **Åtgärder**: Åtgärder kan läggas till i fönstren för identifiering av funktioner. Åtgärden körs genom att klicka på åtgärdsetiketten. Som standard läggs endast en åtgärd till, nämligen `View feature form` för redigering. Du kan definiera fler åtgärder i dialogen för lagrets egenskaper (se *Åtgärder Egenskaper*).
- **Härledd**: Denna information är beräknad eller härledd från annan information. Den inkluderar:
  - allmän information om objektets geometri:
    * beroende på geometrityp, de kartesiska måtten på längd, omkrets eller area i lagrets CRS-enheter. För 3D-linjevektorer är den kartesiska linjelängden tillgänglig.

        * beroende på geometrityp och om en ellipsoid har angetts i dialogen för projektegenskaper för *Measurements*, de ellipsoidiska värdena för längd, omkrets eller area med hjälp av de angivna enheterna
        * antalet geometriska delar i funktionen och numret på den del som klickades på
        * antalet toppar i objektet
    - koordinatinformation, med hjälp av projektegenskaperna *Coordinates display* inställningar:
        * koordinatvärdena `X` och `Y` för den klickade punkten
        * numret på det närmaste toppunktet till den punkt som klickades
        * koordinatvärdena `X` och `Y` för det närmaste toppunktet (och `Z`/`M` om tillämpligt)
        * om du klickar på ett krökt segment visas även radien för det avsnittet.
        * om både vektorlagret och projektet har vertikala datums inställda och de skiljer sig åt, kommer `Z`-värdet att visas för båda datums.
- **Dataattribut**: Detta är en lista över attributfält och värden för den funktion som har klickats på.
- information om den relaterade underordnade funktionen om du har definierat en *relation*:
    - namnet på relationen
    - posten i referensfältet, t.ex. namnet på den relaterade underordnade funktionen
    - **Actions**: listar åtgärder som definierats i dialogen för lagrets egenskaper (se *Åtgärder Egenskaper*) och standardåtgärden är `View feature form`.
    - **Dataattribut**: Detta är en lista över attributfält och värden för den relaterade underordnade funktionen.

Observera

Länkar i funktionens attribut är klickbara från *Identify Results*-panelen och öppnas i din standardwebbläsare.

Kontextuell meny för resultat

Andra funktioner finns i kontextmenyn för det identifierade objektet. Från snabbmenyn kan du t.ex:

- Se formuläret för funktionen
- Zooma till funktionen
- Kopiera funktion: Kopiera all geometri och alla attribut för objektet
- Växla val av funktion: Lägg till identifierad funktion i urvalet
- Kopiera attributvärde: Kopiera endast värdet för det attribut som du klickar på
- Kopiera funktionens attribut: Kopiera funktionens attribut
- Välj funktioner efter attributvärde: Välj alla funktioner i lagret som matchar det valda attributet
- Rensa resultat: Ta bort resultat i fönstret
- Ta bort höjdpunkter: Ta bort funktioner som är markerade på kartan
- Markera alla
- Markera lager
- Aktivera lager: Välj ett lager som ska aktiveras
- Egenskaper för lager: Öppna fönstret för lagrets egenskaper
- Visa alla
- Stäng alla

8.5 Spara och dela egenskaper för lager

8.5.1 Hantera anpassade stilar

När ett vektorlager läggs till i kartbilden använder QGIS som standard en slumpmässig symbol/färg för att återge dess funktioner. Du kan dock ställa in en standardsymbol i *Projekt* ► *Egenskaper…* ► *Default styles* som kommer att tillämpas på varje nytillagt lager enligt dess geometrityp.

För det mesta vill du dock hellre ha en anpassad och mer komplex stil som kan tillämpas automatiskt eller manuellt på lagren (med mindre ansträngning). Det kan du göra genom att använda menyn *Style* längst ned i dialogen Lageregenskaper. I den här menyn finns funktioner för att skapa, ladda och hantera stilar.

En stil lagrar olika information som anges i dialogen för lager- eller projektegenskaper för att rendera eller interagera med lagret. Detta inkluderar:

Table 8.4: Komponenter i lagrad stil och deras referenser

Kategori	Vector	Raster
Lagerkonfiguration	*Kapacitet för lager*	
	Egenskaper för display	*Egenskaper för display*
Symbologi	*Egenskaper för symbologi*	*Egenskaper för symbologi* [1]
3D Symbology	*egenskaper för 3D-vy*	N/A
Märkning	*Etiketter Egenskaper*	N/A
Fält	*Fältegenskaper* och *Begränsningar*	N/A
Attribut Formulär	*Attribut Formulär Egenskaper*	N/A
Åtgärder	*Åtgärder Egenskaper*	N/A
Karttips	*Egenskaper för display*	*Egenskaper för display*
Diagram	*Diagram Egenskaper*	N/A
Konfiguration av attributtabeller	*Arbeta med attributtabellen*	
Rendering	*Rendering av egenskaper*	*Rendering av egenskaper* [1]
Custom Properties (Anpassade egenskaper)	*Informationsegenskaper*	*Informationsegenskaper*
Geometrialternativ	*Egenskaper för digitalisering*	N/A
Relationer	*Inställning av relationer mellan flera lager*	N/A
Temporala egenskaper	*Temporala egenskaper*	*Temporala egenskaper*
Förklaringsinställningar	*Förklaringsegenskaper*	N/A
Höjningsegenskaper	*Elevation Egenskaper*	*Elevation Egenskaper*
Notes	*Anteckningar om lager*	

[1]: För rasterlager smälts symbologi- och renderingsobjekt samman, vilket innebär att om du sparar/laddar det ena så sparas/laddas även objekt från det andra.

Som standard har den stil som tillämpas på ett laddat lager namnet `default`. När du har fått den perfekta och lämpliga återgivningen för ditt lager kan du spara den genom att klicka på kombinationsrutan *Stil* och välja:

- ** Byt namn på nuvarande**: Den aktiva stilen byter namn och uppdateras med de aktuella alternativen

- **Lägg till**: En ny stil skapas med hjälp av de aktuella alternativen. Som standard sparas den i QGIS-projektfilen. Se nedan för att spara stilen i en annan fil eller en databas
- **Avlägsna**: Ta bort oönskad stil, om du har mer än en stil definierad för lagret.

Längst ner i rullgardinsmenyn Style kan du se de stilar som ställts in för lagret med den aktiva markerad.

Observera att varje gång du validerar dialogen för lageregenskaper uppdateras den aktiva stilen med de ändringar du har gjort.

Du kan skapa så många stilar du vill för ett lager, men bara en kan vara aktiv åt gången. I kombination med *Map Themes* erbjuder detta ett snabbt och kraftfullt sätt att hantera komplexa projekt utan att behöva duplicera något lager i kartlegenden.

Observera

Eftersom ändringarna lagras i den aktiva stilen när du ändrar lagrets egenskaper bör du alltid se till att du redigerar rätt stil så att du inte av misstag ändrar en stil som används i ett *mapptema*.

Tips

Hantera stilar från lagrets snabbmeny

Högerklicka på lagret i panelen *Layers* för att kopiera, klistra in, lägga till eller byta namn på lagerstilar. För både vektor- och rasterlager kan du välja vilka stilkategorier som ska kopieras/klistras in. Välj *All Style Categories* eller en av de tillgängliga kategorierna, beroende på lagertyp. Observera att för rasterlager kopieras kategorierna *Symbology* och *Rendering* alltid tillsammans.

8.5.2 Lagra stilar i en fil eller en databas

Stilar som skapas i kombinationsrutan *Style* sparas som standard i projektet och kan kopieras och klistras in från lager till lager i projektet, men det är också möjligt att spara dem utanför projektet så att de kan laddas i ett annat projekt.

Spara som textfil

Genom att klicka på *Stil* ► *Spara stil* kan du spara stilen som en:

- QGIS-lagerformatfil (`.qml`)
- SLD-fil (`.sld`), endast tillgänglig för vektorlager

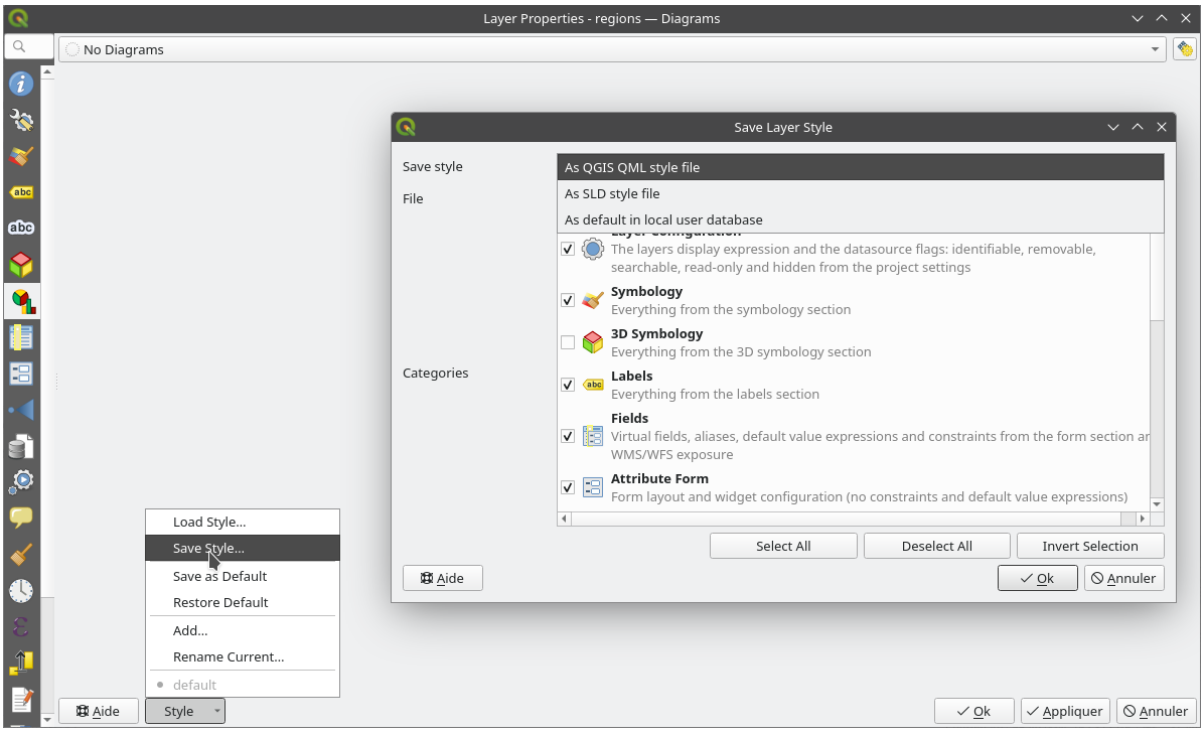


Fig. 8.12: Alternativ för kombinationsruta för vektorlager

Används på lager i filbaserat format (`.shp`, `.tab`…), *Save as Default* genererar en `.qml`-fil för lagret (med samma namn). SLD:er kan exporteras från alla typer av renderare – enstaka symboler, kategoriserade, graderade eller regelbaserade – men när en SLD importeras skapas antingen en enstaka symbol eller en regelbaserad renderare. Detta innebär att kategoriserade eller graderade stilar konverteras till regelbaserade. Om du vill bevara dessa renderare måste du använda QML-formatet. Å andra sidan kan det ibland vara mycket praktiskt att ha detta enkla sätt att konvertera stilar till regelbaserade.

Spara i databas

Vektorlagrets stilar kan också lagras i en databas om lagrets datakälla är en databasleverantör. Format som stöds är PostGIS, GeoPackage, SpatiaLite, MS SQL Server och Oracle. Skiktstilen sparas i en tabell (med namnet `layer_styles`) i databasen. Klicka på *Save Style…* ► *Save in database* och fyll sedan i dialogrutan för att definiera ett stilnamn, lägga till en beskrivning, en `.ui`-fil om tillämpligt och för att kontrollera om stilen ska vara standardstil.

Du kan spara flera stilar för en och samma tabell i databasen. Varje tabell kan dock bara ha en standardstil. Standardstilar kan sparas i lagerdatabasen eller i `qgis.db`, en lokal SQLite-databas i den aktiva *användarprofilen*-katalogen.

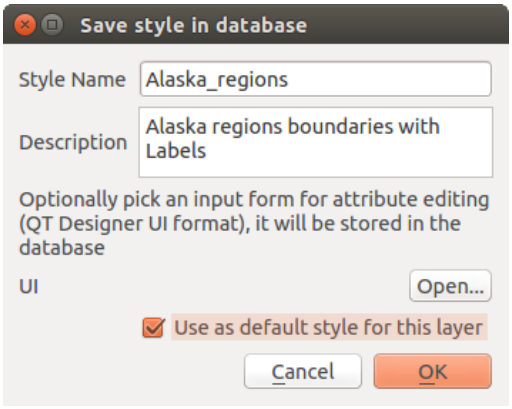


Fig. 8.13: Spara stil i databasdialog

Tips

Delning av stilfiler mellan databaser

Du kan bara spara din stil i en databas om lagret kommer från en sådan databas. Du kan inte blanda databaser (t.ex. lager i Oracle och stil i MS SQL Server). Använd i stället en vanlig textfil om du vill att stilen ska delas mellan olika databaser.

Observera

Du kan stöta på problem med att återställa tabellen `layer_styles` från en säkerhetskopia av PostgreSQL-databasen. Följ *Säkerhetskopiering av QGIS layer_style-tabell och databas* för att åtgärda det.

Ladda stil

När ett lager laddas i QGIS, om det redan finns en standardstil för lagret, laddar QGIS lagret med denna stil. Även *Style* ► *Restore Default* letar efter och laddar den filen och ersätter lagrets nuvarande stil.

Style ► *Load Style* hjälper dig att använda alla sparade stilar på ett lager. Medan textfilsstilar (`.sld` eller `.qml`) kan tillämpas på alla lager oavsett format, är det bara möjligt att ladda stilar som lagrats i en databas om lagret är från samma databas eller om stilen är lagrad i QGIS lokala databas.

I dialogrutan *Database Styles Manager* visas en lista över stilar som är relaterade till lagret i databasen och alla andra stilar som sparats i den, med namn och beskrivning.

Tips

Snabbt dela en lagerstil inom projektet

Du kan också dela lagerformatmallar inom ett projekt utan att importera en fil- eller databasformatmall: högerklicka på lagret i *Layers Panel* och, från kombinationsrutan *Styles* , kopiera formatmallen för ett lager och klistra in den i en grupp eller ett urval av lager: formatmallen tillämpas på alla lager som är av samma typ (vektor vs raster) som det ursprungliga lagret och, när det gäller vektorlager, har samma geometrityp (punkt, linje eller polygon).

8.5.3 Fil för definition av lager

Lagerdefinitioner kan sparas som en ”lagerdefinitionsfil” (`.qlr`) med hjälp av *Export* ► *Save As Layer Definition File…* i de aktiva lagrens kontextmeny. En lagerdefinitionsfil (`.qlr`) innehåller referenser till datakällan för lagren och deras stilar. `.qlr`-filer visas i panelen Browser och kan användas för att lägga till lagren (med den sparade stilen) i panelen Layers. Du kan också dra och släppa `.qlr`-filer från systemets filhanterare till kartbilden.

8.6 Dokumentera dina data

Förutom att visa och symbolisera data i lagren kan du med QGIS fylla i:

- **metadata**: information som hjälper människor att hitta och förstå datasetet, hur de kan få tillgång till och använda det … dessa är egenskaper hos datakällan och kan leva utanför QGIS-projektet.
- **noter**: instruktioner och kommentarer om lagret i det aktuella projektet

8.6.1 Metadata

I dialogrutan för lageregenskaper ger fliken *Metadata* dig alternativ för att skapa och redigera en metadatarapport om ditt lager.

Information för att fylla oro:

- data *Identification*: grundläggande attribution av datasetet (förälder, identifierare, titel, sammanfattning, språk…);
- de *Kategorier* som uppgifterna tillhör. Förutom **ISO**-kategorierna kan du lägga till egna kategorier;
- *Keywords* för att hämta data och tillhörande begrepp enligt en standardbaserad vokabulär;
- *Access* till datasetet (licenser, rättigheter, avgifter och begränsningar);
- *Extent* av datasetet, antingen spatial (CRS, kartutbredning, höjder) eller temporal;
- *Kontaktuppgifter* för ägaren/ägarna till datasetet;
- *Länkar* till tilläggsresurser och relaterad information;
- *Historia* för datasetet.

En sammanfattning av den ifyllda informationen finns på fliken *Validation* och hjälper dig att identifiera eventuella problem som är relaterade till formuläret. Du kan sedan antingen åtgärda dem eller ignorera dem.

Metadata sparas som standard i projektfilen, rullgardinsmenyn *Metadata* erbjuder alternativ för att ladda/spara metadata från `.qmd`-filen och för att ladda/spara metadata på ”Standard”-platsen.

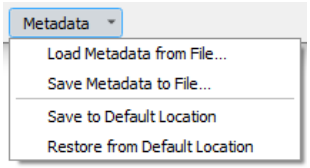


Fig. 8.14: Alternativ för laddning/sparning av metadata

Den ”standardplats” som används av *Save to Default Location* och *Restore from Default Location* ändras beroende på den underliggande datakällan och dess konfiguration:

- För PostgreSQL-datakällor om konfigurationsalternativet *Tillåt spara/ladda QGIS-skiktsmetadata i databasen* är markerat lagras metadata i en särskild tabell i databasen.

- För GeoPackage-datakällor *Save to Default Location* sparar alltid metadata i GeoPackages interna metadatatabeller.

  När metadata sparas i de interna tabellerna i PostgreSQL eller GeoPackage blir de tillgängliga för sökning och filtrering i bläddraren och i *layer metadata search panel*.

- För alla andra filbaserade datakällor sparar *Save to Default Location* metadata i en `.qmd`-fil tillsammans med filen.
- I alla andra fall sparar *Save to Default Location* metadata i en lokal `.sqlite`-databas.

8.6.2 Anteckningar om lager

Med Layer Notes kan du dokumentera lagret i det aktuella projektet. De kan användas för att lagra viktiga meddelanden för användare av projektet, t.ex. att göra-listor, instruktioner, varningar, …

Välj *Add layer notes…* från lagrets kontextuella meny i *Layers*-panelen och fyll i den öppna dialogrutan med nödvändiga texter.

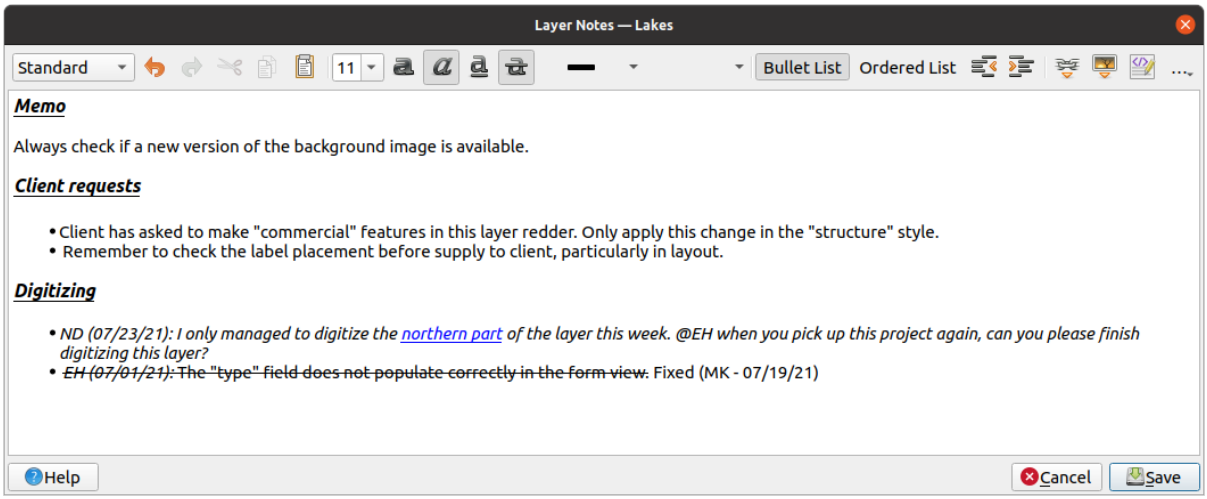


Fig. 8.15: Lägga till anteckningar i ett lager

Dialogrutan *Add layer notes* ger en html-baserad flerlinjig textruta med en komplett uppsättning verktyg för:

- textmanipulation: klippa, kopiera, klistra in, ångra, göra om
- teckenformatering, som tillämpas på hela eller delar av innehållet: teckenstorlek och färg, fet stil, kursiv stil, understrykning, genomstrykning, bakgrundsfärg, URL-markering
- styckeindelning: punktlistor och numrerade listor, indrag, fördefinierade rubriker
- filinmatning, även med dra-och-släpp
- redigering med HTML-kodning

Från rullgardinsmenyn … längst till höger i verktygsfältet kan du:

- *Avlägsna all formatering*
- *Avlägsna teckenformatering*
- *Rensa allt innehåll*

I *Lager*-panelen tilldelas ett lager med en anteckning ikonen som visar anteckningen när du håller muspekaren över lagret. Klicka på ikonen för att redigera anteckningen. Du kan också högerklicka på lagret och *Redigera lageranteckning…* eller *Ta bort lageranteckning*.

Observera

Anteckningar är en del av *lagerstil* och kan sparas i filen `.qml` eller `.qlr`. De kan också överföras från ett lager till ett annat genom att kopiera och klistra in lagerstilen.

8.7 Lagring av värden i variabler

I QGIS kan du använda variabler för att lagra användbara återkommande värden (t.ex. projektets titel eller användarens fullständiga namn) som kan användas i uttryck. Variabler kan definieras på applikationens globala nivå, projektnivå, lagernivå, bearbetningsmodelleringsnivå, layoutnivå och layoutobjektnivå. Precis som CSS-kaskadregler kan variabler skrivas över - t.ex. kommer en variabel på projektnivå att skriva över alla variabler på global nivå i programmet som har samma namn. Du kan använda dessa variabler för att skapa textsträngar eller andra anpassade uttryck med hjälp av tecknet `@` före variabelnamnet. I utskriftslayouten kan du t.ex. skapa en etikett med följande innehåll:

```
This map was made using QGIS [% @qgis_version %]. The project file for this
map is: [% @project_path %]
```

Etiketten återges så här:

```
This map was made using QGIS 3.4.4-Madeira. The project file for this map is:
/gis/qgis-user-conference-2019.qgs
```

Förutom *förinställda skrivskyddade variabler* kan du definiera dina egna anpassade variabler för någon av de nivåer som nämns ovan. Du kan hantera:

- **globala variabler** från menyn *Inställningar* ► *Alternativ*
- **projektvariabler** från dialogrutan *Projektegenskaper* (se *Projektets egenskaper*)
- **vektorskiktsvariabler** från dialogen *Lageregenskaper* (se *Dialogrutan Egenskaper för vektor*);
- **modelleringsvariabler** från dialogrutan *Model Designer* (se *Modellkonstruktören*);
- **layoutvariabler** från panelen *Layout* i utskriftslayouten (se *Layoutpanelen*);
- och **layoutobjektvariabler** från panelen *Item Properties* i utskriftslayouten (se *Layoutobjekt Gemensamma alternativ*).

För att skilja dem från redigerbara variabler visas namn och värden för skrivskyddade variabler i kursiv stil. Variabler på högre nivå som skrivs över av variabler på lägre nivå är däremot genomstrukna.

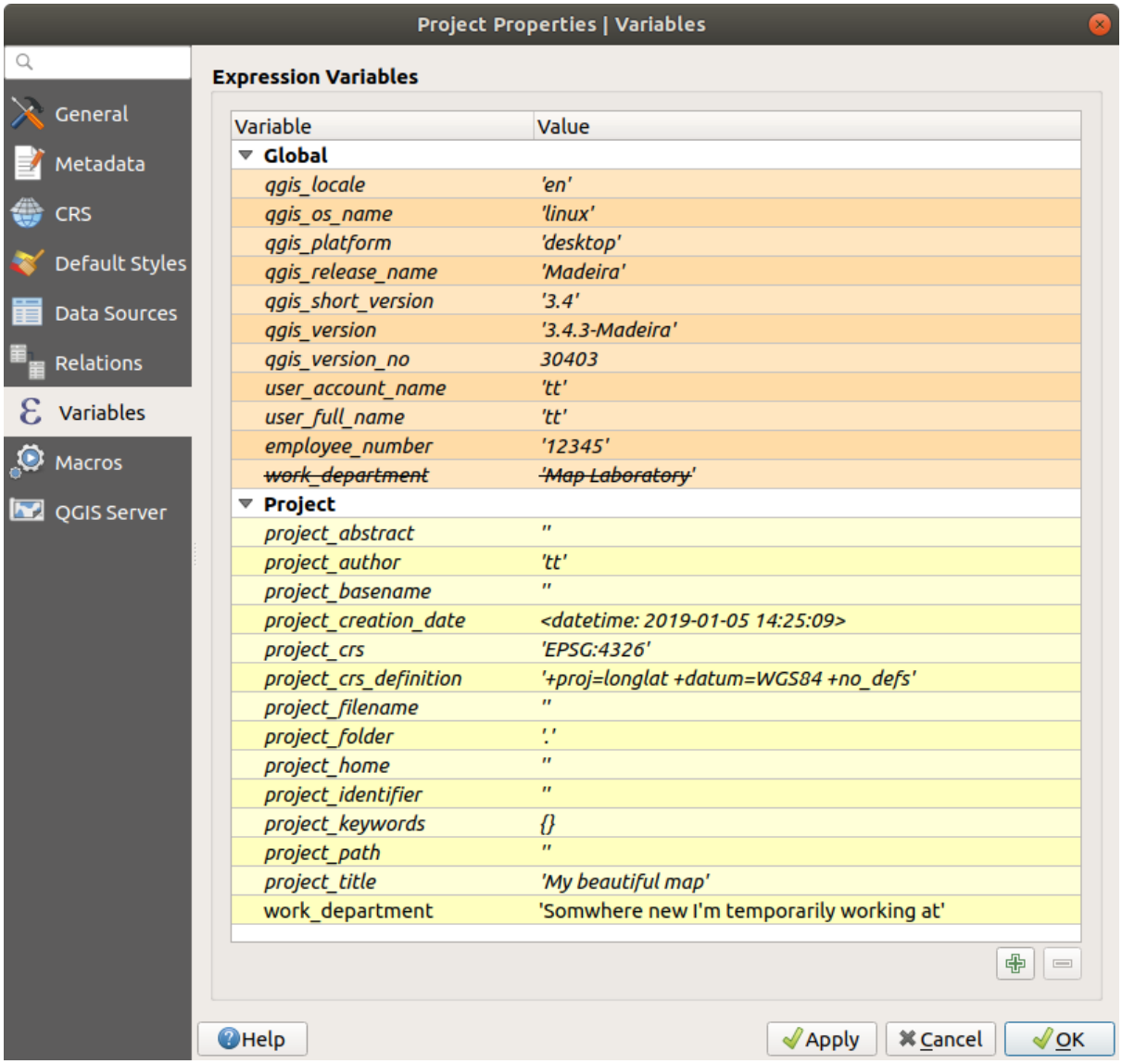


Fig. 8.16: Variabelredigerare på projektnivå

Observera

Du kan läsa mer om variabler och hitta några exempel i Nyall Dawsons blogginlägg Exploring variables in QGIS 2.12, part 1, part 2 och part 3.

8.8 Autentisering

QGIS har möjlighet att lagra/hämta autentiseringsuppgifter på ett säkert sätt. Användare kan på ett säkert sätt spara autentiseringsuppgifter i autentiseringskonfigurationer, som lagras i en portabel databas, kan tillämpas på server- eller databasanslutningar och refereras på ett säkert sätt av deras ID-tokens i projekt- eller inställningsfiler. För mer information se *System för autentisering*.

Ett huvudlösenord måste skapas när autentiseringssystemet och dess portabla databas initieras.

8.9 Vanliga widgetar

I QGIS finns det några alternativ som du ofta måste arbeta med. För enkelhetens skull tillhandahåller QGIS särskilda widgetar som presenteras nedan.

8.9.1 Färgväljare

Dialogrutan för färg

Dialogrutan *Välj färg* visas när du klickar på ikonen för att välja en färg. Funktionerna i denna dialog beror på tillståndet i kryssrutan för parametern *Use native color chooser dialogs* i *Inställningar* ► *Alternativ…* ► *Allmänt*. När kryssrutan är markerad används den inbyggda färgdialogen i det operativsystem som QGIS körs på. Annars används QGIS anpassade färgväljare.

Den anpassade färgväljardialogen har fyra olika flikar som gör att du kan välja färger med Color ramp, Color wheel, Color swatches eller Color picker. Med de två första flikarna kan du bläddra bland alla möjliga färgkombinationer och tillämpa ditt val på artikeln.

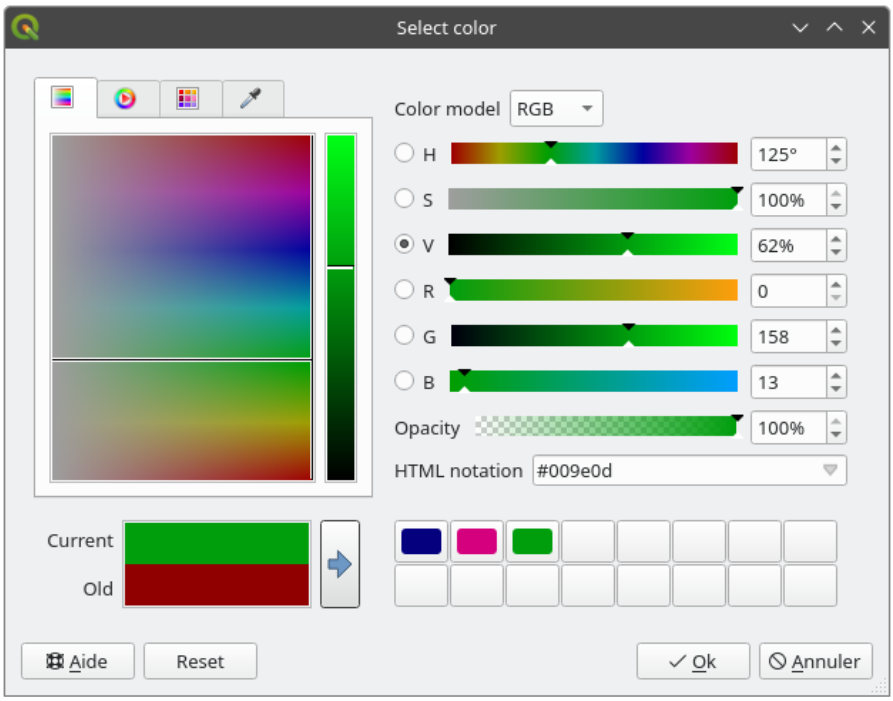


Fig. 8.17: Färgväljare rampflik

På fliken Färgprover kan du välja från en lista med färgpaletter (se *Färginställningar* för detaljer). Alla utom paletten *Recent colors* kan ändras med knapparna Add current color och Remove selected color längst ner i ramen.

Knappen … bredvid palettkombinationsrutan erbjuder också flera alternativ för att:

- kopiera, klistra in, importera eller exportera färger
- skapa, importera eller ta bort färgpaletter
- lägg till den anpassade paletten i färgväljarwidgeten med *Visa i färgknappar*-objektet (se Fig. 8.19)

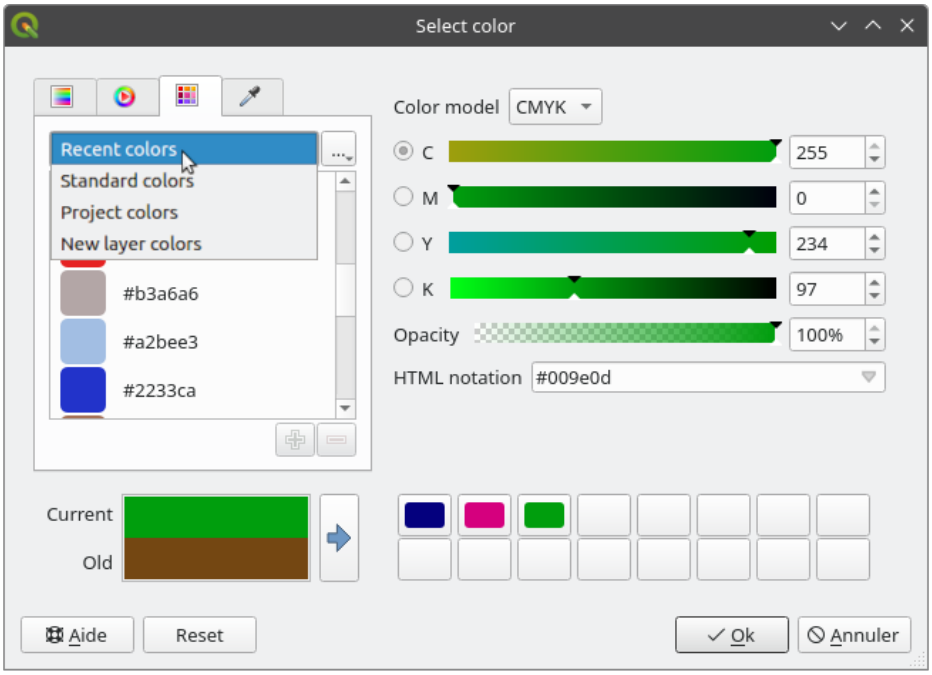


Fig. 8.18: Fliken färgväljare för färgprover

Ett annat alternativ är att använda Color picker som gör att du kan välja en färg under muspekaren var som helst i QGIS UI eller till och med från ett annat program: tryck på mellanslagstangenten medan fliken är aktiv, flytta musen över önskad färg och klicka på den eller tryck på mellanslagstangenten igen. Du kan också klicka på knappen *Sample Color* för att aktivera färgväljaren.

Oavsett vilken metod du använder beskrivs den valda färgen alltid genom alternativ för en *Färgmodell* som har *projektets färgmodell* som standard:

- *RGB*: färg ställs in med hjälp av reglage för `HSV` (Hue, Saturation, Value) eller `RGB` (Red, Green, Blue) värden.
- *CMYK* (Cyan, Magenta, Gul, Svart)

Att ändra en färg är så enkelt som att klicka på färghjulet eller rampen eller på något av skjutreglagen för färgparametrarna. Du kan justera sådana parametrar med spinboxen bredvid eller genom att rulla mushjulet över motsvarande skjutreglage. Du kan också skriva in färgen i HTML-notation. Slutligen finns det ett *Opacity*-reglage för att ställa in transparensnivån.

Dialogrutan ger också en visuell jämförelse mellan *Old*-färgen (som används på objektet) och *Current*-färgen (som väljs). Genom att dra och släppa eller trycka på knappen Add color to swatch kan vilken som helst av dessa färger sparas i en plats för enkel åtkomst.

Tips

Snabb färgändring

Dra och släpp en färgväljarwidget på en annan för att använda dess färg.

Genväg till rullgardinsmenyn för färg

Klicka på nedrullningspilen till höger om färgknappen för att visa en widget för snabbt färgval. Denna genväg ger tillgång till:

- en färghjul för att välja en färg från
- ett alpha-reglage för att ändra färgens opacitet
- de färgpaletter som tidigare ställts in på *Visa i färgknappar*
- kopiera den aktuella färgen och klistra in den i en annan widget
- välja en färg var som helst på din datorskärm
- välj en färg från färgväljardialogen
- dra och släpp färgen från en widget till en annan för snabb ändring

Tips

Dra mushjulet över en färgväljarwidget för att snabbt ändra opaciteten för den associerade färgen.

Observera

När färgwidgeten är inställd på en *projektfärg* genom de datadefinierade åsidosättningsegenskaperna är ovanstående funktioner för att ändra färgen inte tillgängliga. Du måste först *Unlink color* eller *Clear* definiera färgen.

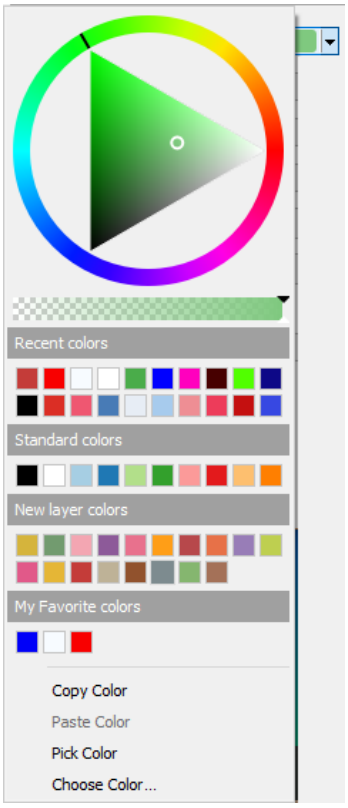


Fig. 8.19: Snabb färgväljarmeny

Kortkommando för färgrampens rullgardinsmeny

Färgramper är ett praktiskt sätt att använda en uppsättning färger på en eller flera funktioner. Hur de skapas beskrivs i avsnittet *Inställning av en färgramp*. När det gäller färgerna kan du trycka på färgrampsknappen för att öppna dialogrutan för motsvarande typ av färgramp och ändra dess egenskaper.

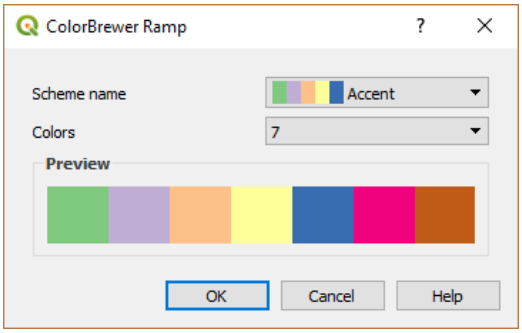


Fig. 8.20: Anpassning av en färgbryggeriramp

Rullgardinsmenyn till höger om knappen ger snabb åtkomst till en bredare uppsättning färgramper och alternativ:

- *Inverterad färgramp*
- *Clear Current Ramp* för att ta bort alla tilldelade färgramper till widgeten (endast tillgänglig i vissa sammanhang)
- *Slumpmässiga färger*: endast tillgänglig i vissa sammanhang (t.ex. när en färgramp används för en lagersymbologi), genom att markera denna post skapas och tillämpas en färgramp med slumpmässiga färger. Det gör det också möjligt för en *Shuffle random colors*-post att återskapa en ny slumpmässig färgramp om den nuvarande inte är tillfredsställande.
- en förhandsgranskning av färgramperna `gradient` eller `catalog: cpt-city` som är markerade som **Favoriter** i dialogrutan *Stilhanterare*
- *All Color Ramps* för att få tillgång till databasen för kompatibla färgramper
- *Create New Color Ramp…* av valfri typ som stöds och som kan användas i den aktuella widgeten (observera att den här färgrampen inte kommer att vara tillgänglig någon annanstans om du inte sparar den i biblioteket)
- *Edit Color Ramp…*, samma sak som att klicka på hela färgrampsknappen
- *Save Color Ramp…*, för att spara den aktuella färgrampen med dess anpassningar i stilbiblioteket

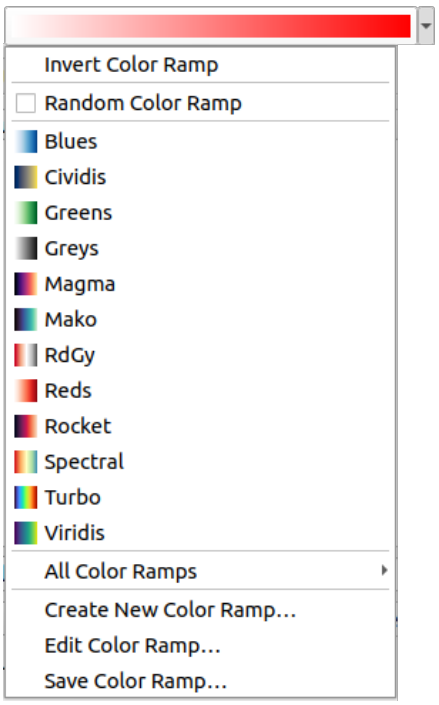


Fig. 8.21: Widget för snabbt val av färgramp

8.9.2 Widget för symboler

Väljarwidgeten *Symbol* är en praktisk genväg när du vill ställa in symbolegenskaper för en funktion. Om du klickar på rullgardinsmenyn visas följande symbolalternativ, tillsammans med funktionerna i *color drop-down widget*:

- *Configure Symbol…*: samma sak som att trycka på symbolväljarwidgeten. Öppnar en dialogruta för att ställa in *symbolparametrarna*.
- *Kopiera symbol* från det aktuella objektet
- *Pasta in symbol* till det aktuella objektet, vilket påskyndar konfigurationen
- *Clear Current Symbol* för att ta bort alla tilldelade symboler till widgeten (endast tillgänglig i vissa sammanhang)

Tips

Dra mushjulet över en widget med markör- eller linjesymboler för att snabbt ändra storleken på den tillhörande symbolen.

8.9.3 Väljare för fjärrstyrd eller inbäddad fil

Tillsammans med filväljarwidgeten kan … -knappen ibland visa en rullgardinsmeny. Detta är vanligtvis tillgängligt när du använder:

- en SVG-fil i en symbol eller en etikett
- en rasterbild för att anpassa symboler, etiketter, texturer eller dekorationer

Om du trycker på pilen kommer du till en meny för att:

- *Välj fil…*, för att ladda filen från filsystemet. Filen identifieras genom filsökvägen och QGIS måste lösa sökvägen för att kunna visa motsvarande bild

- *Från URL…*, för att ladda filen från en fjärr-URL. Precis som ovan laddas bilden endast när fjärrresursen har hämtats
- *Embed File…*, för att bädda in filen i det aktuella projektet, stildatabasen eller utskriftslayoutmallen. Filen återges då alltid som en del av objektet. Detta är ett bekvämt sätt att skapa självständiga projekt med anpassade symboler som enkelt kan delas mellan olika användare och installationer av QGIS.
- *Extract Embedded File…*, för att extrahera den inbäddade filen från widgeten och spara den på disken.

8.9.4 Väljare för synlighetsskala

Väljaren för synlighetsskala ger alternativ för att styra i vilken skala ett element ska göras synligt i kartbilden. Utanför det angivna skalintervallet visas inte elementen. Det kan tillämpas på t.ex. lager, etiketter eller diagram, från deras egenskapsflik *Rendering*.

1. Kryssa i rutan *Skalaberoende synlighet*
2. Fyll i rutan *Minimum (exklusiv)* med den mest utzoomade önskade skalan genom att skriva in värdet eller välja det från *fördefinierade skalor*
3. och/eller fylla *Maximum (inklusive)*-rutan med den mest inzoomade önskade skalan

   Knappen Set to current canvas scale bredvid skalrutorna ställer in den aktuella kartans skala som gräns för områdets synlighet. Tryck på pilen bredvid knappen för att få tillgång till skalor från layouternas kartor och återanvända dem för att fylla rutan.

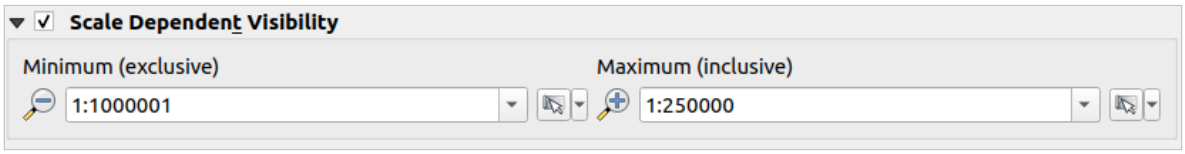


Fig. 8.22: Väljarwidget för synlighetsskala

8.9.5 Väljare för spatial utsträckning

Väljarwidgeten *Extent* är en praktisk genväg när du vill välja en spatial utsträckning som ska tilldelas ett lager eller begränsa de åtgärder som ska utföras. Beroende på sammanhanget erbjuder den val mellan:

- *Current Layer Extent*: t.ex. vid export av ett lager
- *Calculate from Layer* ►: använder omfattningen av ett lager som laddats i det aktuella projektet
- Använd nuvarande *Map Canvas Extent*
- *Draw on Canvas*: en rektangel vars koordinater sedan används
- *Calculate from Bookmark*: använder omfattningen av ett sparat *bokmärke*
- *Beräkna från layoutkarta*: använder omfattningen av en *layoutkarta*
- Ange eller redigera koordinaterna som `xmin, xmax, ymin, ymax`

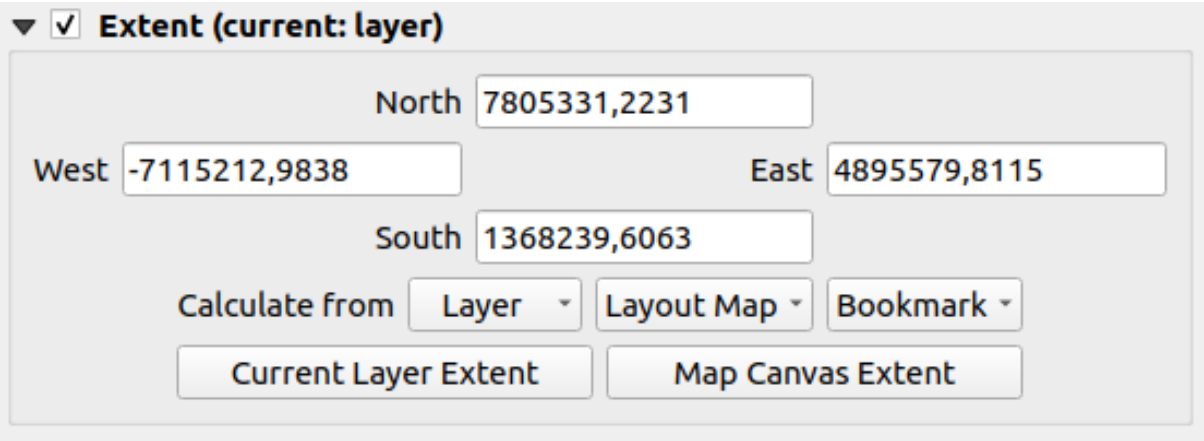


Fig. 8.23: Widget för extentväljare

8.9.6 Typsnittsväljare

Väljarwidgeten *Font* är en praktisk genväg när du vill ställa in teckensnittsegenskaper för textinformation (funktionsetiketter, dekorationsetiketter, text i kartlegenden, …). Om du klickar på rullgardinsmenyn visas några eller alla av följande alternativ:

Fig. 8.24: Rullgardinsmeny för val av teckensnitt

- *Clear Current Text Format* för att avaktivera alla tilldelade textformat för widgeten (endast tillgängligt i vissa sammanhang)
- *Fontstorlek* i den associerade enheten
- *Recent Fonts* ►-menyn med det aktiva teckensnittet markerat (högst upp)
- *Configure Format…*: samma som att trycka på widgeten för typsnittsväljare. Den öppnar dialogrutan *Textformat*, med avancerade formateringsalternativ som färg, opacitet, orientering, HTML-notation, buffert, bakgrund,

skugga, …

- *Kopiformat* av texten
- *Pasta in format* till texten, snabbare konfiguration
- *färgwidget* för snabb färginställning

Tips

Dra mushjulet över en widget för val av teckensnitt för att snabbt ändra teckenstorleken för den associerade texten.

8.9.7 Väljare för enhet

Storleksegenskaperna för objekten (etiketter, symboler, layoutelement, …) i QGIS är inte nödvändigtvis bundna till varken projektenheterna eller enheterna för ett visst lager. För en stor uppsättning egenskaper kan du med rullgardinsmenyn *Unit* justera deras värden enligt den återgivning du vill ha (baserat på skärmupplösning, pappersstorlek eller terräng). Tillgängliga enheter är:

- *Millimeter*
- *Poäng*
- *Pixlar*
- *Inches*
- *Percentage*: gör att du kan ställa in vissa egenskaper som en procentandel av en annan. Detta är t.ex. användbart för att skapa textformat där komponenterna (buffertstorlek, skuggradie …) snyggt skalas när textstorleken ändras, istället för att ha konstanta buffert-/skuggstorlekar. På så sätt behöver du inte justera dessa storlekar när textstorleken ändras.
- *Meter i skala*: Detta gör att du alltid kan ställa in storleken i meter, oavsett vad de underliggande kartanheterna är (t.ex. kan de vara i tum, fot, geografiska grader, …). Storleken i meter beräknas utifrån den aktuella inställningen för projektets ellipsoid och en projektion av avstånden i meter i mitten av den aktuella kartutbredningen. För kartor i ett projicerat koordinatsystem beräknas detta med hjälp av projicerade enheter. För kartor i ett geografiskt (latitud/longitud) baserat system approximeras storleken genom att beräkna meterstorlekar med hjälp av ellipsoidberäkningar för kartans vertikala skala.
- och *Kartenheter*: Storleken skalas enligt skalan i kartvyn. Eftersom detta kan leda till för stora eller för små värden kan du använda knappen bredvid posten för att begränsa storleken till ett intervall av värden baserat på:
  - *Minimum scale* och *Maximum scale*: Värdet skalas baserat på kartvyns skala tills du når någon av dessa skalgränser. Utanför skalområdet behålls värdet vid den närmaste skalgränsen.
  - och/eller *Minimum size* och *Maximum size* i `mm`: Värdet skalas baserat på kartvyns skala tills det når någon av dessa gränser; sedan behålls gränsstorleken.

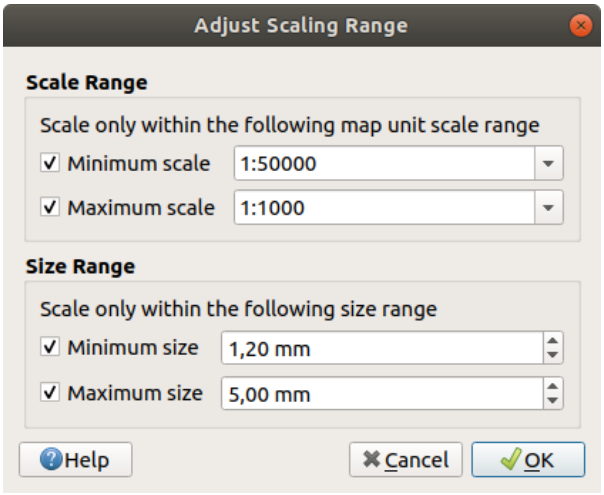


Fig. 8.25: Dialog för justering av skalningsintervall

8.9.8 Nummerformatering

Numeriska formaterare möjliggör formatering av numeriska värden för visning med hjälp av en mängd olika formateringstekniker (t.ex. vetenskaplig notation, valutavärden, procentvärden, anpassad formatering etc.) Detta kan t.ex. användas för att ställa in text i en layoutskala eller tabell, en färgrampslegende, en höjdprofilplott, …

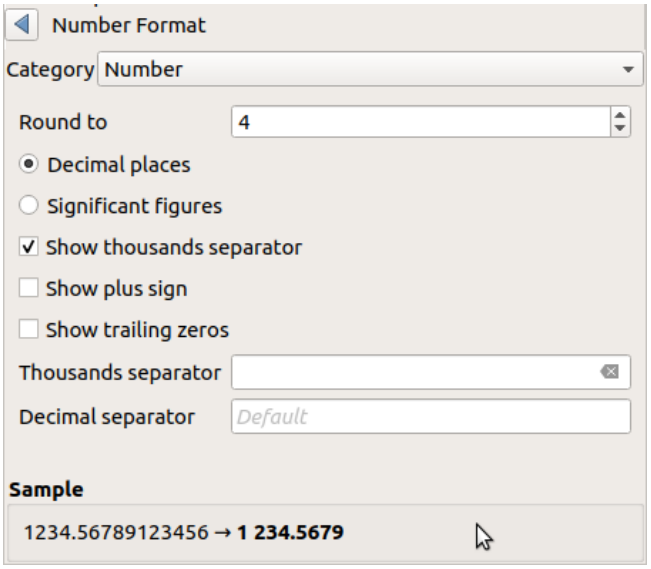


Fig. 8.26: Formatering av numeriskt värde

Olika kategorier av format stöds. För de flesta av dem kan du ställa in delar av eller alla följande numeriska alternativ:

- *Visa tusentalsavgränsare*
- *Visa plustecken*
- *Visa efterföljande nollor*

Men de kan också ha sina egna inställningar. Tillhandahållna kategorier är:

- *General*, standardkategorin: har ingen inställning och visar värden som anges i den överordnade widgetens egenskaper eller med hjälp av de globala inställningarna.

- *Custom expression*: gör att du kan skapa ett anpassat *QGIS-uttryck* för att formatera siffror. Uttrycket kan använda variabeln `@value` för att hämta det värde som ska formateras och sedan använda valfri standard QGIS-uttrycksfunktion för att formatera detta på önskat sätt.
- *Nummer*
  - Värdet kan vara *Round to* ett självdefinierat antal *Decimaler* eller deras *Signifikanta siffror*
  - anpassa *Tusentalsavskiljare* och *Decimalavskiljare*
- *Bearing* för en textrepresentation av en riktning/bäring med hjälp av:
  - *Format*: möjliga värdeintervall är `0 till 180°, med Ö/V-suffix`, `-180 till +180°` och `0 till 360°`
  - antal *Decimalställen*
- *Currency* för en textrepresentation av ett valutavärde.
  - *Prefix*
  - *Suffix*
  - antal *Decimalställen*
- *Fraction* för en vulgär fraktionell representation av ett decimalvärde (t.ex. *1/2* istället för *0.5*)
  - ☐ *Utnyttja unicode super/subscript* för att visa. Till exempel $^{1}/_{2}$ istället för 1/2
  - ☐ *Använda dedikerade Unicode-tecken*
  - anpassa *Tusentals-separatorn*
- *Percentage* - lägger till `%` till värdena, med inställning av:
  - antal *Decimalställen*
  - *Scaling* för att ange om de faktiska värdena redan representerar procent (då behålls de som de är) eller bråk (då konverteras de)
- *Vetenskaplig* notation i form av `2.56e+03`. Antalet *Decimalställen* kan ställas in.

En förhandsgranskning av inställningarna visas under avsnittet *Sample*.

8.9.9 Blandningslägen

QGIS erbjuder olika alternativ för speciella renderingseffekter med dessa verktyg som du kanske tidigare bara kände till från grafikprogram. Blandningslägen kan tillämpas på lager och funktioner, och även på utskriftslayoutobjekt:

- *Normal*: Detta är det vanliga blandningsläget, som använder alfakanalen i den översta pixeln för att blanda med pixeln under den. Färgerna blandas inte.
- *Lätta upp*: Detta väljer ut det maximala värdet för varje komponent från förgrunds- och bakgrundspixlarna. Var medveten om att resultatet tenderar att bli ojämnt och hårt.
- *Skärm*: Ljusa pixlar från källan målas över målet, medan mörka pixlar inte målas. Det här läget är mest användbart för att blanda texturen från ett objekt med ett annat objekt (t.ex. genom att använda en terrängskuggning för att texturera ett annat lager).
- *Dodge*: Ljusar upp och mättar underliggande pixlar baserat på ljusheten i den översta pixeln. Ljusare toppixlar gör att mättnaden och ljusstyrkan hos de underliggande pixlarna ökar. Detta fungerar bäst om de översta pixlarna inte är för ljusa. Annars blir effekten för extrem.
- *Addition*: Lägger till pixelvärden från ett objekt till det andra. Vid värden över det maximala värdet (i RGB) visas vitt. Detta läge är lämpligt för att framhäva funktioner.
- *Darken*: Behåller de lägsta värdena för varje komponent i förgrunds- och bakgrundspixlarna. Precis som med lighten tenderar resultatet att bli ojämnt och hårt.

- *Multiplicera*: Pixelvärdena för det övre objektet multipliceras med motsvarande värden för det nedre objektet. Resultatet blir mörkare.
- *Burn*: Mörkare färger i det översta objektet gör att de underliggande objekten blir mörkare. Burn kan användas för att justera och färgsätta underliggande lager.
- *Overlay*: Kombinerar multiplicerings- och skärmblandningslägen. Ljusa delar blir ljusare och mörka delar blir mörkare.
- *Mjukt ljus*: Mycket likt overlay, men istället för att använda multiply/screen används color burn/dodge. Detta är tänkt att efterlikna att skina ett mjukt ljus på en bild.
- *Hårt ljus*: Hard light är också mycket likt overlay-läget. Det är tänkt att efterlikna projicering av ett mycket intensivt ljus på en bild.
- *Difference*: Subtraherar den övre pixeln från den nedre, eller tvärtom, för att alltid få ett positivt värde. Att blanda med svart ger ingen förändring, eftersom skillnaden med alla färger är noll.
- *Subtract*: Subtraherar pixelvärdena för ett objekt från det andra. Vid negativa värden visas svart.

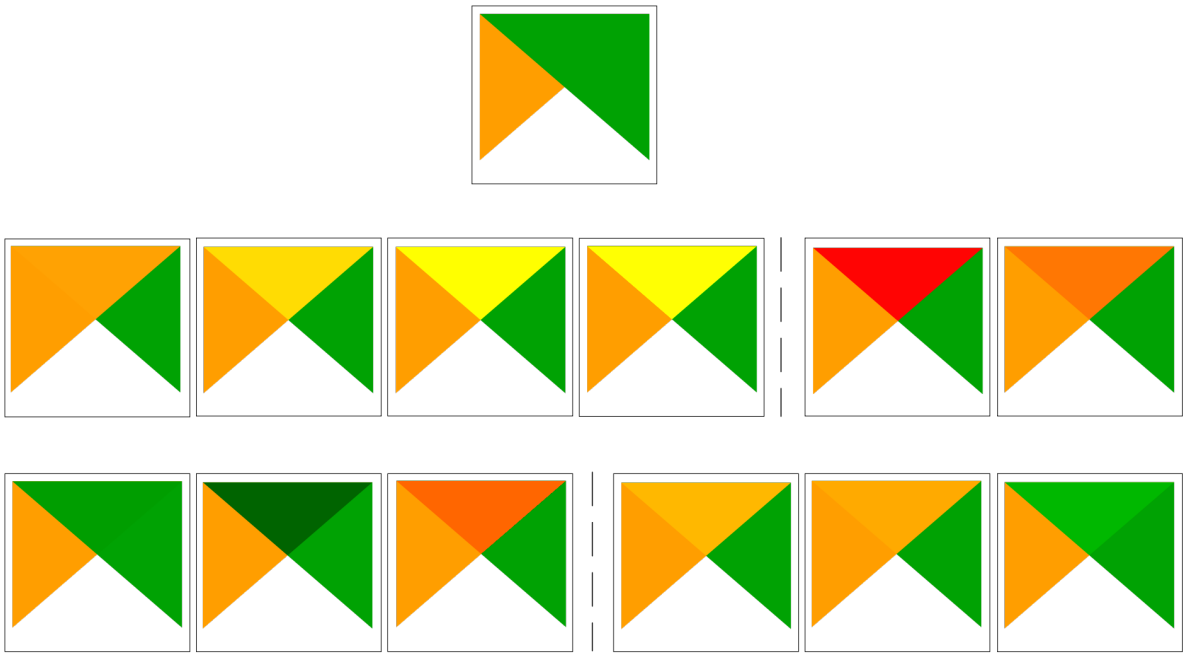


Fig. 8.27: Exempel på blandningslägen som tillämpas på en grön funktion över en orange
Från topp till botten, från vänster till höger: Normal – Ljusare, Screen, Dodge, Addition – Skillnad, Subtrahera – Mörkare, Multiplicera, Bränna – Överlagra, Mjukt ljus, Hårt ljus

När ett lager är en del av en grupp som *renderar lager som en grupp*, finns ytterligare blandningslägen tillgängliga för renderingen. De tillhandahåller metoder för att klippa renderingen av innehållet i ett lager med innehållet i ett andra ”mask”-lager.

- *Maskerad av nedan*: Utdata är den övre pixeln, där opaciteten reduceras med den nedre pixelns opacitet.
- *Mask nedanför*: Utdata är den nedre pixeln, där opaciteten minskas med den övre pixelns.
- *Inverse Masked By Below*: Utdata är den övre pixeln, där opaciteten reduceras med inversen av den nedre pixeln.
- *Inverse Mask Below*: Utdata är den nedre pixeln, där opaciteten reduceras med inversen av den övre pixeln.
- *Måla inuti nedanför*: Den övre pixeln blandas ovanpå den undre pixeln, med den övre pixelns opacitet minskad med den undre pixelns opacitet.
- *Måla under insidan*: Den nedre pixeln blandas ovanpå den övre pixeln, med den nedre pixelns opacitet minskad med den övre pixelns opacitet.

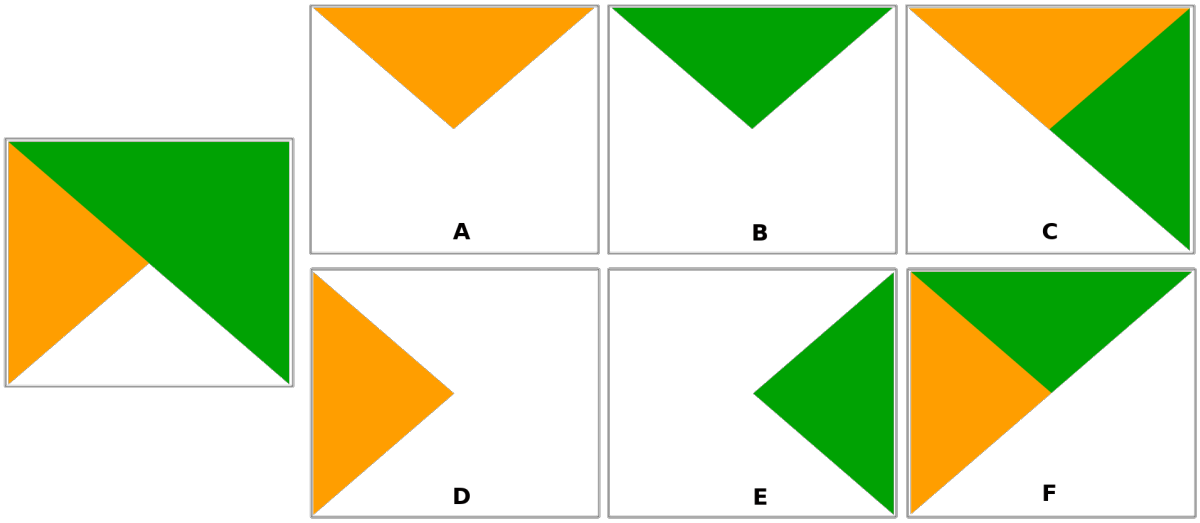


Fig. 8.28: Exempel på blend-klippläge som tillämpas på det översta gröna lagret i en grupp
A: Mask nedanför **B**: Maskerad av nedan **C**: Måla nedanför inuti **D**: Inverse Mask Below **E**: Omvänd maskerad av nedan **F**: Måla inuti nedanför

8.9.10 Inställning av datadefinierad åsidosättning

Bredvid många alternativ i dialogrutan för vektorlagrets egenskaper eller inställningar i utskriftslayouten finns ikonen Data defined override. Med hjälp av *expressions* baserat på lagerattribut eller objektinställningar, förbyggda eller anpassade funktioner och *variables* kan du med detta verktyg ställa in dynamiska värden för parametrar. När det är aktiverat tillämpas det värde som returneras av denna widget på parametern oavsett dess normala värde (kryssruta, textruta, skjutreglage …).

Den datadefinierade åsidosättningswidgeten

Om du klickar på ikonen Data defined override visas följande poster:

- *Beskrivning…* som anger om alternativet är aktiverat, vilken indata som förväntas, giltig indatatyp och aktuell definition. Om du håller muspekaren över widgeten visas också denna information.
- *Lagra data i projektet*: en knapp som gör att egenskapen kan lagras med hjälp av mekanismen *Egenskaper för hjälplagring*.
- *Field type*: en post för att välja bland lagrets fält som matchar den giltiga inmatningstypen.
- *Color*: när widgeten är länkad till en färgegenskap ger den här menyn tillgång till de färger som definieras som en del av det aktuella *projektets färger*-schema.
- *Variable*: en meny för att komma åt de tillgängliga användardefinierade *variablerna*
- *Edit…*-knappen för att skapa eller redigera det uttryck som ska tillämpas, med hjälp av dialogrutan *Expression String Builder*. För att hjälpa dig att fylla i uttrycket korrekt visas en påminnelse om det förväntade utdataformatet i dialogrutan.
- knapparna *Paste* och *Copy*.
- *Clear*-knappen för att ta bort inställningen.
- För numeriska egenskaper och färgegenskaper, *Assistant…* för att skala om hur funktionsdata tillämpas på egenskapen (mer information *under*)

Tips

Använd högerklick för att (av)aktivera dataöverstyrningen

När det datadefinierade åsidosättningsalternativet är korrekt inställt är ikonen gul eller . Om det är trasigt är ikonen röd eller .

Du kan aktivera eller inaktivera en konfigurerad Data-defined override-knapp genom att helt enkelt klicka på widgeten med höger musknapp.

Tips

Använd mittklick för att skapa eller redigera det uttryck som ska tillämpas

Du kan direkt öppna dialogrutan *Expression String Builder* för att skapa eller redigera det uttryck som ska tillämpas genom att klicka på widgeten Data-defined override med musens mittknapp.

Använda det datadefinierade assistentgränssnittet

När knappen Datadefinierad åsidosättning associeras med en storlek, en rotation, en opacitet eller en färgegenskap har den ett alternativ *Assistant…* som hjälper dig att ändra hur data tillämpas på parametern för varje funktion. Med hjälp av assistenten kan du:

- Definiera *Input*-data, dvs:
  - *Källa*: attributet som ska representeras, med hjälp av ett fält eller ett *uttryck*
  - det värdeintervall som ska representeras: du kan ange värdena manuellt eller använda knappen Fetch value range from layer för att fylla i dessa fält automatiskt med de minimi- och maximivärden som returneras av uttrycket *Source* som tillämpas på dina data
- *Apply transform curve*: som standard tillämpas utdatavärden (se nedan för inställning) på indatafunktioner enligt en linjär skala. Du kan åsidosätta denna logik: aktivera alternativet transform, klicka på grafiken för att lägga till brytpunkter och dra punkterna för att tillämpa en anpassad distribution.
- Definiera *Output*-värdena: alternativen varierar beroende på vilken parameter som ska definieras. Du kan göra en global inställning:
  - för en färginställning, den *färgramp* som ska tillämpas på värden och den enskilda färg som ska användas för NULL-värden
  - för de övriga, minimi- och maximivärdena som ska tillämpas på den valda egenskapen samt värdet för storlek/vinkel/opacitet för ignorerade eller NULL-källfunktioner
  - för storleksegenskaper, representationsmetoden *Scale* som kan vara **Flannery**, **Exponentiell**, **Surface**, **Radius** eller **Linear**
  - den *Exponent* som ska användas för skalning av data när *Scale method* är av exponentiell typ eller när opaciteten justeras

När det är kompatibelt med egenskapen visas en förhandsgranskning av liveuppdateringen till höger i dialogrutan för att hjälpa dig att kontrollera värdeskalningen.

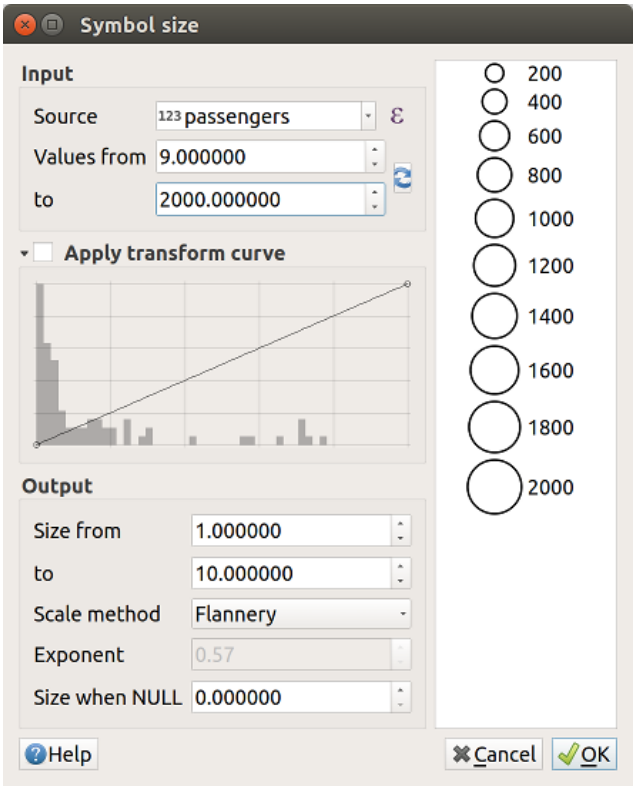


Fig. 8.29: Skalning av funktionsstorlek baserat på passagerarfältets värde

De värden som presenteras i assistenten för varierande storlek ovan kommer att ange den storlek som ”Data-defined override” ska användas med:

```
coalesce(scale_exp("passengers", 9, 2000, 1, 10, 0.57), 0)
```

KAPITEL 9

Lyft dig själv med uttryck

9.1 Uttryck

Baserat på lagerdata och förbyggda eller användardefinierade funktioner erbjuder **Expressions** ett kraftfullt sätt att manipulera attributvärde, geometri och variabler för att dynamiskt ändra geometristilen, etikettens innehåll eller position, värdet för diagram, höjden på ett layoutobjekt, välja vissa funktioner, skapa virtuella fält, …

Observera

En lista över standardfunktioner och variabler för att skriva uttryck finns på *Lista över funktioner*, med detaljerad information och exempel.

9.1.1 Strängbyggaren Expression

Huvuddialog för att bygga uttryck, *Expression string builder* är tillgänglig från många delar i QGIS och, kan särskilt nås när:

- genom att klicka på knappen ;
- *välja funktioner* med verktyget Välj efter uttryck…;
- *redigering av attribut* med t.ex. verktyget Field calculator;
- manipulera parametrar för symbologi, etikett eller layoutobjekt med verktyget Datadefinierad åsidosättning (se *Inställning av datadefinierad åsidosättning*);
- bygga ett *geometri-generator* symbollager;
- gör lite *geoprocessing*.

Dialogen Expression builder ger tillgång till:

- *Utryck* flik som, tack vare en lista med *fördefinierade funktioner*, hjälper till att skriva och kontrollera det uttryck som ska användas;

- *Fliken Funktionsredigerare* som hjälper till att utöka listan med funktioner genom att skapa egna.

Gränssnittet

Fliken *Expression* är det huvudsakliga gränssnittet för att skriva uttryck med funktioner, lagerfält och värden. Den innehåller följande widgetar:

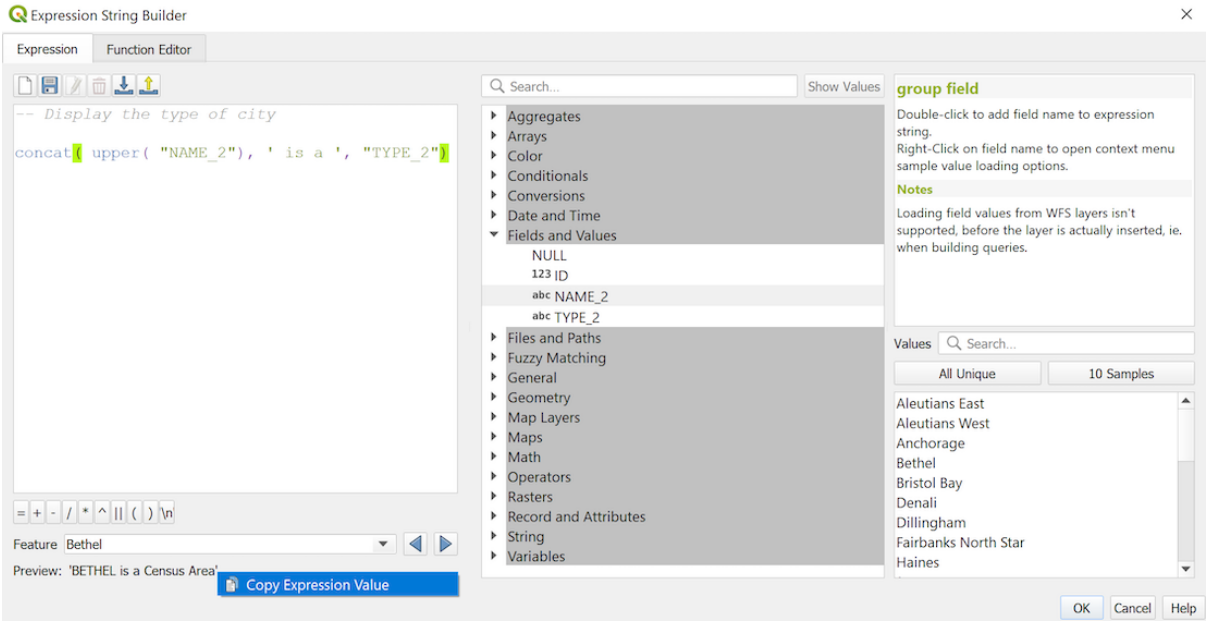


Fig. 9.1: Fliken Uttryck

- En uttrycksredigeringsyta för att skriva in eller klistra in uttryck. Autokomplettering finns tillgänglig för att snabba upp skrivandet av uttryck:
  - Motsvarande variabler, funktionsnamn och fältnamn till inmatningstexten visas nedan: använd pilarna `Upp` och `Ned` för att bläddra bland objekten och tryck på `Tab` för att infoga i uttrycket eller klicka helt enkelt på önskat objekt.
  - Funktionsparametrarna visas när du fyller i dem.

  QGIS kontrollerar också att uttrycket är korrekt och markerar alla fel som används:
  - *Understrykning*: för okända funktioner, felaktiga eller ogiltiga argument;
  - *Marker*: för alla andra fel (t.ex. saknad parentes, oväntat tecken) på en och samma plats.

Tips

Dokumentera ditt uttryck med kommentarer

När du använder komplexa uttryck är det bra att lägga till text antingen som en kommentar på flera rader eller som en kommentar i raden för att hjälpa dig att komma ihåg.

```
/*
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
)
```

- Utöver uttrycksredigeraren finns det en uppsättning verktyg som hjälper dig:
  - Rensa uttrycksredigeraren
  - Skapa och hantera *användaruttryck*
- Under uttrycksredaktören hittar du:
  - En uppsättning grundläggande operatorer som hjälper dig att bygga upp uttrycket
  - En indikation på det förväntade formatet för utdata när du datadefinierar egenskaper
  - En live *Output preview* av uttrycket (upp till 60 tecken), som utvärderas på den första funktionen i lagret som standard. Om du vill visa förhandsgranskningstext på mer än 60 tecken kan du hålla markören över texten för att visa ett popup-fönster med verktygstips som innehåller hela förhandsgranskningen av utdata. Om du vill kopiera förhandsgranskningstexten till urklipp högerklickar du på förhandsgranskningstexten och väljer *Kopiera uttrycksvärde*.

    Du kan bläddra bland och utvärdera andra funktioner i lagret med hjälp av kombinationsrutan *Feature* (värdena hämtas från lagrets egenskap *display name*).

    Om det finns ett fel anges det och du kan komma åt detaljerna via den angivna hyperlänken.
- En funktionsväljare visar listan över funktioner, variabler, fält … organiserade i grupper. En sökruta finns tillgänglig för att filtrera listan och snabbt hitta en viss funktion eller ett visst fält. Genom att dubbelklicka på ett objekt läggs det till i uttrycksredigeraren.
- En hjälppanel visar hjälp för varje valt objekt i funktionsväljaren.

> **Tips**
>
> Tryck på `Ctrl+Click` när du håller muspekaren över ett funktionsnamn i ett uttryck för att automatiskt visa dess hjälp i dialogrutan.

En widget med fältvärden som visas när ett fält väljs i funktionsväljaren hjälper till att hämta funktionsattribut:

- Leta efter ett visst fältvärde
- Visar listan med värdena *All Unique* eller *10 Samples*. Finns även tillgänglig via högerklick.

  När fältet är mappat med ett annat lager eller en uppsättning värden, dvs. om *field widget* är av typen *RelationReference*, *ValueRelation* eller *ValueMap*, är det möjligt att lista alla värden i det mappade fältet (från det refererade lagret, tabellen eller listan). Dessutom kan du filtrera listan till *Visa endast värden som används* i det aktuella fältet.

Om du dubbelklickar på ett fältvärde i widgeten läggs det till i uttrycksredigeraren.

> **Tips**
>
> Den högra panelen, som visar hjälp med funktioner eller fältvärden, kan kollapsas (osynliggöras) i dialogrutan. Tryck på knappen *Show Values* eller *Show Help* för att få tillbaka den.

Skriva ett uttryck

QGIS-uttryck används för att välja funktioner eller ställa in värden. Att skriva ett uttryck i QGIS följer vissa regler:

1. **Dialogen definierar sammanhanget**: om du är van vid SQL känner du säkert till frågor av typen *select features from layer where condition* eller *update layer set field = new_value where condition*. Ett QGIS-uttryck behöver också all denna information, men det verktyg som du använder för att öppna dialogen för att skapa ett uttryck tillhandahåller delar av dem. Till exempel kan du ange ett lager (`buildings`) med ett fält (`height`):

   - om du trycker på verktyget :sup:`Select by expression` betyder det att du vill ”välja egenskaper från byggnader”. **villkoret** är den enda information som du behöver ange i widgeten för uttryckstext, t.ex. skriv `"höjd" > 20` för att välja byggnader som är högre än 20.
   - med detta val gjort, tryck på knappen Field calculator och välj ”height” som *Update existing field*, du ger redan kommandot ”update buildings set height = ??? where height > 20”. Det enda som återstår att ange i det här fallet är det **nya värdet**, t.ex. ange bara `50` i textrutan för uttrycksredigeraren för att ställa in höjden på de tidigare valda byggnaderna.

2. **Var uppmärksam på citattecken**: enkla citattecken returnerar en bokstav, så en text som placeras mellan enkla citattecken (`'145'`) tolkas som en sträng. Dubbla citattecken ger dig värdet på den texten, så använd dem för fält (`"myfield"`). Fält kan också användas utan citationstecken (`myfield`). Inga citationstecken för siffror (`3.16`).

> **Observera**
>
> Funktioner tar normalt som argument en sträng för fältnamn. Gör:
>
> ```
> attribute( @atlas_feature, 'height' ) -- returns the value stored in the
> ↪"height" attribute of the current atlas feature
> ```
>
> Och inte:
>
> ```
> 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.
> ```

> **Tips**
>
> **Använd namngivna parametrar för att underlätta läsning av uttryck**
>
> Vissa funktioner kräver att många parametrar ställs in. Uttrycksmotorn stöder användning av namngivna parametrar. Detta innebär att istället för att skriva det kryptiska uttrycket `clamp( 1, 2, 9)`, kan du använda `clamp( min:=1, value:=2, max:=9)`. Detta gör det också möjligt att byta ut argument, t.ex. `clamp( value:=2, max:=9, min:=1)`. Att använda namngivna parametrar hjälper till att klargöra vad argumenten för en uttrycksfunktion hänvisar till, vilket är till hjälp när du försöker tolka ett uttryck senare!

Några användningsfall för uttryck

- Beräkna ett ”pop_density”-fält i fältkalkylatorn med hjälp av de befintliga fälten ”total_pop” och ”area_km2”:

```
"total_pop" / "area_km2"
```

- Märk eller kategorisera funktioner baserat på deras område:

```
CASE WHEN $area > 10 000 THEN 'Larger' ELSE 'Smaller' END
```

- Uppdatera fältet ”density_level” med kategorier enligt värdena för ”pop_density”:

```
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
```

- Använd en kategoriserad stil för alla funktioner beroende på om deras genomsnittliga huspris är lägre eller högre än 10000 € per kvadratmeter:

```
"price_m2" > 10000
```

- Med hjälp av verktyget ”Select By Expression…” väljer du alla funktioner som representerar områden med ”hög befolkningstäthet” och vars genomsnittliga huspris är högre än 10000 € per kvadratmeter:

```
"density_level" = 'High population density' and "price_m2" > 10000
```

  Det föregående uttrycket kan också användas för att definiera vilka funktioner som ska märkas ut eller visas på kartan.

- Välj funktioner som överlappar en naturlig zon från lagret ”lands”:

```
overlay_intersects( layer:='lands', filter:="zone_type"='Natural' )
```

- Räkna för varje funktion antalet byggnader som de innehåller:

```
array_length( overlay_contains( layer:='buildings', expression:=@id ) )
```

- Skapa en annan symbol (typ) för lagret med hjälp av geometrigeneratorn:

```
point_on_surface( @geometry )
```

- Givet en punktfunktion, generera en sluten linje (med `make_line`) runt dess geometri:

```
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 )␣
↪)
  )
)
```

- I en etikett för utskriftslayout visas namnet på de ”flygplatser” som finns inom layoutens ”Karta 1”-objekt:

```
with_variable( 'extent',
               map_get( item_variables( 'Map 1' ), 'map_extent' ),
               aggregate( 'airports', 'concatenate', "NAME",
                          intersects( @geometry, @extent ), ' ,'
```

(continues on next page)

(fortsättning från föregående sida)

```
                                    )
                )
```

Spara uttryck

Med hjälp av knappen Lägg till aktuellt uttryck i användaruttryck ovanför uttrycksredigerarens ram kan du spara viktiga uttryck som du vill ha snabb åtkomst till. Dessa finns tillgängliga i gruppen **Användaruttryck** i mittpanelen. De sparas under *användarprofil* (`<userprofile>/QGIS/QGIS3.ini`-filen) och finns tillgängliga i alla uttrycksdialoger i alla projekt i den aktuella användarprofilen.

En uppsättning verktyg som finns tillgängliga ovanför uttrycksredigeringsramen hjälper dig att hantera användaruttrycken:

- Lägg till det aktuella uttrycket i användaruttryck: lagrar uttrycket i användarprofilen. En etikett och en hjälptext kan läggas till för enkel identifiering.
- Redigera valt uttryck från användaruttryck, samt deras hjälp och etikett
- Ta bort valt uttryck från användaruttryck
- :sup:`Importera användaruttryck` från en `.json`-fil till den aktiva användarprofilmappen
- :sup:`Exportera användaruttryck` som en `.json`-fil; alla användaruttryck i användarprofilen `QGIS3.ini`-filen delas

9.1.2 Funktion Redaktör

Med fliken *Function Editor* kan du skriva dina egna funktioner på Python-språket. Detta är ett praktiskt och bekvämt sätt att tillgodose särskilda behov som inte täcks av de fördefinierade funktionerna.

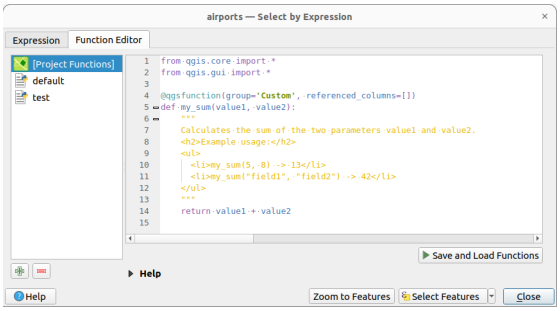


Fig. 9.2: Fliken Funktionsredigerare

För att skapa en ny funktion:

1. Tryck på knappen New File.
2. Välj var den nya funktionen ska lagras. Du kan välja att spara den antingen i en *Function file* eller i *Project functions*.
   - Om du väljer *Funktionsfil* måste du ange ett namn för Python-filen `.py`, som baseras på en QGIS-mallfil och lagras i mappen `/python/expressions` under den aktiva katalogen *användarprofil*, och därmed är tillgänglig för alla projekt som du kan öppna med den profilen.

     Anpassade Python-funktioner som lagras som funktionsfiler laddas automatiskt när du startar ett nytt projekt under användarprofilen. Tänk på att om du vill dela sådana Python-funktioner måste du dela filen `.py` i mappen `/python/expressions`.

- Om du väljer *Project functions* lagras den nya funktionen i projektfilen och är endast tillgänglig för det projektet.

  Anpassade Python-funktioner som lagras som projektfunktioner kan enkelt delas genom att dela projektfilen där de sparades. När ett projekt med projektfunktioner öppnas kan QGIS läsa in eller ignorera dem beroende på om *inbäddad Python-kod är aktiverad*.

Ett nytt objekt har lagts till i den vänstra panelen på fliken *Function Editor*.

3. I den högra panelen visas innehållet i filen: en pythonskriptmall.
4. Avkommentera kodblocket och börja redigera det enligt dina behov. Du kan förlita dig på autokomplettering av kod, kodformatering och skrivhjälpmedel, enligt definitionen i *Python-inställningar*.

   I widgeten *Help* hittar du allmän information om hur du ställer in dekoratorn @qgsfunction.

5. Tryck på knappen *Save and Load Functions*. Den funktion du skrev läggs till i funktionsträdet på fliken *Expression*, som standard under gruppen `Custom`.
6. Njut av din nya funktion.
7. Om funktionen behöver förbättras, aktivera fliken *Function Editor*, gör ändringarna och tryck igen på knappen *Save and Load Functions* för att göra dem tillgängliga i filen och därmed i alla uttrycksflikar.

Ta bort en anpassad funktion:

1. Aktivera fliken *Funktionsredigerare*
2. Välj funktion i listan
3. Tryck på Ta bort vald funktion. Funktionen tas bort från listan och, beroende på lagring, raderas motsvarande `.py`-fil från användarprofilmappen eller Python-funktionerna tas bort från projektfilen.

> **Observera**
>
> När en projektfil med projektfunktioner stängs av, avlastas motsvarande projektfunktioner och är inte längre tillgängliga i QGIS-sessionen.

Exempel

Här är ett kort exempel på hur du skapar din egen funktion `my_sum` som fungerar med två värden.

```
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
```

Dekoratorn `@qgsfunction` accepterar följande argument:

- Argumentet `group` anger i vilken grupp funktionen ska listas i dialogen Expression.
- `använder geometri`: Sätt detta till `True` om uttrycket kräver tillgång till funktionens geometri. Som standard `False`.

- `handlesnull`: Sätt detta till `True` om uttrycket har anpassad hantering för NULL-värden. Om `False` (standard), kommer resultatet alltid att vara NULL så snart någon parameter är NULL.
- `referenced_columns`: En matris med attributnamn som krävs för att köra funktionen. Standard är `[QgsFeatureRequest.ALL_ATTRIBUTES]`. Om du anger en delmängd av fält eller en tom lista blir körningen snabbare.
- `params_as_list`: Sätt detta till `True` för att skicka funktionsparametrarna som en lista. Om `False` (standard) kommer funktionen att ta emot parametrarna som enskilda argument.

Själva funktionen tar emot det antal och den typ av parametrar som du vill skicka till din funktion. Dessa bör ställas in före följande valfria argument:

- `feature`: den aktuella funktionen
- `parent`: objektet `QgsExpression`
- `context`: Om det finns ett argument som heter `context` på den sista positionen, kommer denna variabel att innehålla ett `QgsExpressionContext`-objekt, som ger tillgång till diverse ytterligare information som t.ex. uttrycksvariabler. T.ex. `context.variable( 'layer_id' )`

Om dessa argument finns i funktionssignaturen skickas de automatiskt till funktionen, utan att de behöver anges i uttrycket.

Funktionen i föregående exempel kan sedan användas i uttryck:

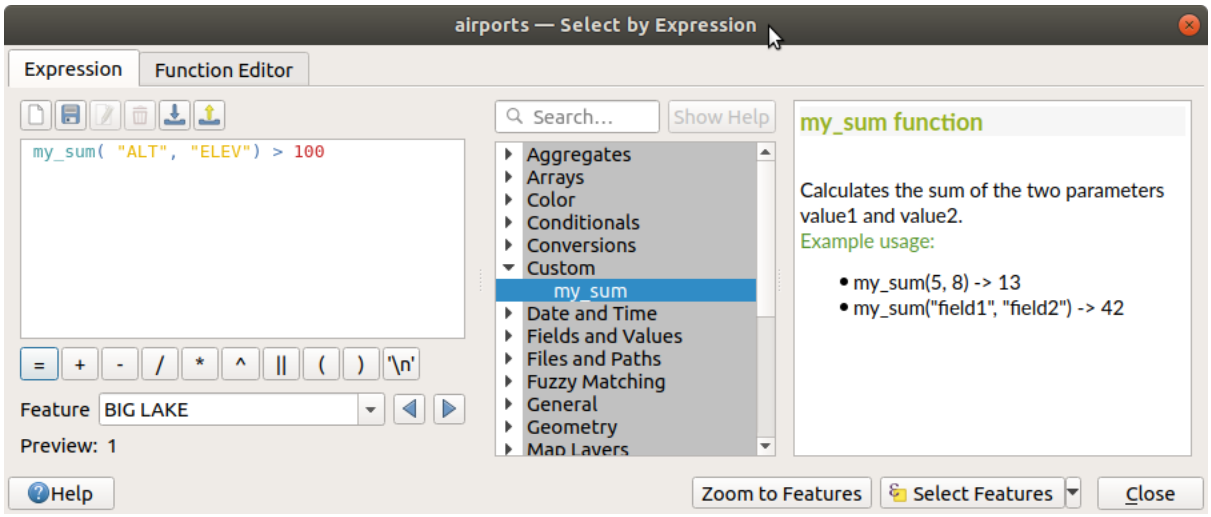


Fig. 9.3: Anpassad funktion tillagd på fliken Expression

Du hittar fler exempel och ytterligare argument för anpassade funktioner på `qgsfunction`. Mer information om hur du skapar Python-kod finns i PyQGIS-Developer-Cookbook.

9.2 Lista över funktioner

De funktioner, operatorer och variabler som finns tillgängliga i QGIS listas nedan, grupperade efter kategorier.

9.2.1 Aggregeringsfunktioner

Den här gruppen innehåller funktioner som aggregerar värden över lager och fält.

aggregate

Returnerar ett aggregerat värde som beräknats med hjälp av funktioner från ett annat lager.

Syntax	aggregate(layer, aggregate, expression, [filter], [concatenator=''], [order_by]) [] markerar valfria argument
Argument	• **lager** - en sträng som representerar antingen ett lagernamn eller ett lager-ID • **aggregera** - en sträng som motsvarar det aggregat som ska beräknas. Giltiga alternativ är: – count – count_distinct – count_missing – minimum eller min – maximalt eller max – sum – mean – median – stdev – stdevsample – range – minority – majority – q1: första kvartilen – q3: tredje kvartilen – iqr: intervall mellan kvartiler – min_length: minsta stränglängd – max_length: maximal stränglängd – konkatenera: sammanfoga strängar med en konkatenator – concatenate_unique: sammanfoga unika strängar med en konkatenator – collect: skapa en aggregerad multipartgeometri – array_agg: skapa en array med aggregerade värden • **expression** - underuttryck eller fältnamn som ska aggregeras • **filter** - valfritt filteruttryck för att begränsa de funktioner som används för att beräkna aggregatet. Fält och geometri kommer från objekten i det sammanfogade lagret. Källfunktionen kan nås med variabeln @parent. • **concatenator** - valfri sträng som ska användas för att sammanfoga värden för aggregaten ”concatenate” och ”concatenate_unique • **order_by** - valfritt filteruttryck för att ordna de funktioner som används för att beräkna aggregatet. Fält och geometri är från funktionerna i det sammanfogade lagret. Som standard kommer funktionerna att returneras i en ospecificerad ordning.
Exempel	• `aggregate(layer:='rail_stations',aggregate:='sum', expression:="passengers")` → summan av alla värden från fältet passengers i lagret rail_stations • `aggregate('rail_stations','sum', "passengers"/7)` → beräknar ett dagligt genomsnitt av ”passengers” genom att dela fältet ”passengers” med 7 och sedan summera värdena • `aggregate(layer:='rail_stations',aggregate:='sum', expression:="passengers",filter:="class">3)` → summerar alla värden från fältet ”passengers” från funktioner där attributet ”class” endast är större än 3 • `aggregate(layer:='rail_stations',aggregate:='concatenate', expression:="name", concatenator:=',')` → kommaseparerad lista över namnfältet för alla funktioner i lagret rail_stations • `aggregate(layer:='countries', aggregate:='max', expression:="code", filter:=intersects( @geometry, geometry(@parent) ) )` → Landskoden för ett korsande land i lagret ”countries • `aggregate(layer:='rail_stations',aggregate:='sum', expression:="passengers",filter:=contains( @atlas_geometry, @geometry ) )` → summan av alla värden från fältet passengers i rail_stations inom den aktuella atlasfunktionen • `aggregate(layer:='rail_stations', aggregate:='collect', expression:=centroid(@geometry), filter:="region_name" = attribute(@parent,'name') )` → aggregerar centroidgeometrier för rail_stations i samma region som den aktuella funktionen

array_agg

Returnerar en array med aggregerade värden från ett fält eller uttryck.

Syntax	array_agg(expression, [group_by], [filter], [order_by]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat • **order_by** - valfritt uttryck att använda för att ordna funktioner som används för att beräkna aggregat. Som standard returneras funktionerna i en ospecificerad ordning.
Exempel	• `array_agg("name",group_by:="state")` → lista över namnvärden, grupperade efter fältet state

collect

Returnerar multipartgeometrin för aggregerade geometrier från ett uttryck

Syntax	collect(expression, [group_by], [filter]) [] markerar valfria argument
Argument	• **expression** - geometriskt uttryck att aggregera • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat
Exempel	• `collect( @geometri )` → flerdelad geometri av aggregerade geometrier • `collect( centroid(@geometry), group_by:="region", filter:= "use" = 'civilian' )` → aggregerade centroider av de civila funktionerna baserat på deras regionvärde

concatenate

Returnerar alla aggregerade strängar från ett fält eller uttryck som förenas av en avgränsare.

Syntax	concatenate(expression, [group_by], [filter], [concatenator], [order_by]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat • **concatenator** - valfri sträng som ska användas för att sammanfoga värden. Tom som standard. • **order_by** - valfritt uttryck att använda för att ordna funktioner som används för att beräkna aggregat. Som standard returneras funktionerna i en ospecificerad ordning.
Exempel	• `concatenate("town_name",group_by:="state",concatenator:=', ')` → kommaseparerad lista över town_names, grupperade efter state-fält

concatenate_unique

Returnerar alla unika strängar från ett fält eller uttryck med en avgränsare.

Syntax	concatenate_unique(expression, [group_by], [filter], [concatenator], [order_by]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat • **concatenator** - valfri sträng som ska användas för att sammanfoga värden. Tom som standard. • **order_by** - valfritt uttryck att använda för att ordna funktioner som används för att beräkna aggregat. Som standard returneras funktionerna i en ospecificerad ordning.
Exempel	• `concatenate_unique("town_name",group_by:="state", concatenator:=',')` → kommaseparerad lista över unika town_names, grupperade efter state-fält

count

Returnerar antalet matchande funktioner.

Syntax	count(expression, [group_by], [filter]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat
Exempel	• `count("stations",group_by:="state")` → antal stationer, grupperade efter delstatsfält

count_distinct

Returnerar antalet distinkta värden.

Syntax	count_distinct(expression, [group_by], [filter]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat
Exempel	• `count_distinct("stations",group_by:="state")` → antal distinkta stationsvärden, grupperade efter fältet state

count_missing

Returnerar antalet saknade (NULL) värden.

Syntax	count_missing(expression, [group_by], [filter]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat
Exempel	• `count_missing("stations",group_by:="state")` → antal saknade (NULL) stationsvärden, grupperade efter state-fält

iqr

Returnerar det beräknade intervallet mellan kvartilerna från ett fält eller uttryck.

Syntax	iqr(expression, [group_by], [filter]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat
Exempel	• `iqr("population",group_by:="state")` → interkvartilintervall för befolkningsvärde, grupperat efter delstatsfält

majority

Returnerar den sammanlagda majoriteten av värden (det vanligast förekommande värdet) från ett fält eller uttryck.

Syntax	majority(expression, [group_by], [filter]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat
Exempel	• `majority("class",group_by:="state")` → vanligast förekommande klassvärde, grupperat efter state-fält

max_length

Returnerar den maximala längden på strängar från ett fält eller uttryck.

Syntax	max_length(expression, [group_by], [filter]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat
Exempel	• `max_length("town_name",group_by:="state")` → maximal längd på town_name, grupperat efter state-fält

maximum

Returnerar det aggregerade maxvärdet från ett fält eller uttryck.

Syntax	maximum(expression, [group_by], [filter]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat
Exempel	• `maximum("population",group_by:="state")` → maximalt befolkningsvärde, grupperat efter delstatsfält

mean

Returnerar det aggregerade medelvärdet från ett fält eller uttryck.

Syntax	mean(expression, [group_by], [filter]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat
Exempel	• `mean("population",group_by:="state")` → populationens medelvärde, grupperat efter delstatsfält

median

Returnerar det aggregerade medianvärdet från ett fält eller uttryck.

Syntax	median(expression, [group_by], [filter]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat
Exempel	• `median("population",group_by:="state")` → medianvärde för befolkningen, grupperat efter delstatsfält

min_length

Returnerar den minsta längden på strängar från ett fält eller uttryck.

Syntax	min_length(expression, [group_by], [filter]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat
Exempel	• `min_length("town_name",group_by:="state")` → minsta längd på town_name, grupperat efter state-fält

minimum

Returnerar det aggregerade minimivärdet från ett fält eller uttryck.

Syntax	minimum(expression, [group_by], [filter]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat
Exempel	• `minimum("population",group_by:="state")` → minsta befolkningsvärde, grupperat efter delstatsfält

minority

Returnerar den aggregerade minoriteten av värden (minst förekommande värde) från ett fält eller uttryck.

Syntax	minority(expression, [group_by], [filter]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat
Exempel	• `minority("class",group_by:="state")` → minst förekommande klassvärde, grupperat efter state-fält

q1

Returnerar den beräknade första kvartilen från ett fält eller uttryck.

Syntax	q1(expression, [group_by], [filter]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat
Exempel	• `q1("population",group_by:="state")` → första kvartilen av befolkningsvärdet, grupperat efter delstatsfält

q3

Returnerar den beräknade tredje kvartilen från ett fält eller uttryck.

Syntax	q3(expression, [group_by], [filter]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat
Exempel	• `q3("population",group_by:="state")` → tredje kvartilen av befolkningsvärdet, grupperat efter delstatsfält

range

Returnerar det aggregerade värdeintervallet (maximum - minimum) från ett fält eller uttryck.

Syntax	range(expression, [group_by], [filter]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat
Exempel	• `range("population",group_by:="state")` → intervall av befolkningsvärden, grupperade efter delstatsfält

relation_aggregate

Returnerar ett aggregerat värde som beräknas med hjälp av alla matchande underordnade funktioner från en lagerrelation.

Syntax	relation_aggregate(relation, aggregate, expression, [concatenator=''], [order_by]) [] markerar valfria argument
Argument	• **relation** - en sträng som representerar ett relations-ID • **aggregera** - en sträng som motsvarar det aggregat som ska beräknas. Giltiga alternativ är: – count – count_distinct – count_missing – minimum eller min – maximalt eller max – sum – mean – median – stdev – stdevsample – range – minority – majority – q1: första kvartilen – q3: tredje kvartilen – iqr: intervall mellan kvartiler – min_length: minsta stränglängd – max_length: maximal stränglängd – konkatenera: sammanfoga strängar med en konkatenator – concatenate_unique: sammanfoga unika strängar med en konkatenator – collect: skapa en aggregerad multipartgeometri – array_agg: skapa en array med aggregerade värden • **expression** - underuttryck eller fältnamn som ska aggregeras • **concatenator** - valfri sträng som ska användas för att sammanfoga värden för ''concatenate''-aggregatet • **order_by** - valfritt uttryck för att ordna de funktioner som används för att beräkna aggregatet. Fält och geometri kommer från funktionerna i det sammanfogade lagret. Som standard returneras funktionerna i en ospecificerad ordning.
Exempel	• `relation_aggregate(relation:='my_relation', aggregate:='mean',expression:="passengers")` → medelvärde för alla matchande underordnade funktioner med hjälp av relationen 'my_relation' • `relation_aggregate('my_relation','sum', "passengers"/7)` → summan av fältet passengers dividerat med 7 för alla matchande underordnade funktioner som använder relationen ''my_relation • `relation_aggregate('my_relation','concatenate',"towns", concatenator:=',')` → kommaseparerad lista över fältet towns för alla matchande underordnade funktioner som använder relationen 'my_relation' • `relation_aggregate('my_relation','array_agg', "id")` → array av fältet id från alla matchande underordnade funktioner som använder relationen 'my_relation'

Ytterligare läsning: *Inställning av relationer mellan flera lager*

stdev

Returnerar det aggregerade standardavvikelsevärdet från ett fält eller uttryck.

Syntax	stdev(expression, [group_by], [filter]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat
Exempel	• `stdev("population",group_by:="state")` → standardavvikelse för populationsvärde, grupperat efter delstatsfält

sum

Returnerar det aggregerade summerade värdet från ett fält eller uttryck.

Syntax	sum(expression, [group_by], [filter]) [] markerar valfria argument
Argument	• **expression** - underuttryck för fältet som ska aggregeras • **group_by** - valfritt uttryck att använda för att gruppera aggregerade beräkningar • **filter** - valfritt uttryck att använda för att filtrera egenskaper som används för att beräkna aggregat
Exempel	• `sum("population",group_by:="state")` → summerat befolkningsvärde, grupperat efter delstatsfält

9.2.2 Array-funktioner

Den här gruppen innehåller funktioner för att skapa och manipulera arrayer (även kända som listdatastrukturer). Ordningen på värdena i arrayen har betydelse, till skillnad från *'map' datastruktur*, där ordningen på nyckelvärdeparen är irrelevanta och värdena identifieras av sina nycklar.

array

Returnerar en array som innehåller alla värden som skickats som parameter.

Syntax	array(värde1, värde2, …)
Argument	• **value** - ett värde
Exempel	• `array(2,10)` → [2, 10] • `array(2,10)[0]` → 2

array_all

Returnerar TRUE om en array innehåller alla värden i en angiven array.

Syntax	array_all(array_a, array_b)
Argument	• **array_a** - en array • **array_b** - den array med värden som ska sökas
Exempel	• `array_all(array(1,2,3),array(2,3))` → TRUE • `array_all(array(1,2,3),array(1,2,4))` → FALSE

array_append

Returnerar en array med det angivna värdet tillagt i slutet.

Syntax	array_append(array, value)
Argument	• **array** - en array • **value** - det värde som ska läggas till
Exempel	• `array_append(array(1,2,3),4)` → [1, 2, 3, 4]

array_cat

Returnerar en array som innehåller alla de angivna arrayer som är sammanlänkade.

Syntax	array_cat(array1, array2, …)
Argument	• **array** - en array
Exempel	• `array_cat(array(1,2),array(2,3))` → [1, 2, 2, 3]

array_contains

Returnerar TRUE om en array innehåller det angivna värdet.

Syntax	array_contains(array, value)
Argument	• **array** - en array • **value** - det värde som ska sökas
Exempel	• `array_contains(array(1,2,3),2)` → TRUE

array_count

Räknar antalet förekomster av ett givet värde i en array.

Syntax	array_count(array, value)
Argument	• **array** - en array • **value** - det värde som ska räknas
Exempel	• `array_count(array('a', 'b', 'c', 'b'), 'b')` → 2

array_distinct

Returnerar en array som innehåller distinkta värden av den angivna arrayen.

Syntax	array_distinct(array)
Argument	• **array** - en array
Exempel	• `array_distinct(array(1,2,3,2,1))` → [1, 2, 3]

array_filter

Returnerar en array med endast de objekt för vilka uttrycket utvärderas till sant.

Syntax	array_filter(array, expression, [limit=0]) [] markerar valfria argument
Argument	• **array** - en array • **expression** - ett uttryck som ska utvärderas för varje element. Variabeln *@element* kommer att ersättas med det aktuella värdet. • **limit** - maximalt antal element som ska returneras. Använd 0 för att returnera alla värden.
Exempel	• `array_filter(array(1,2,3),@element < 3)` → [1, 2] • `array_filter(array(1,2,3),@element < 3, 1)` → [1]

array_find

Returnerar det lägsta indexet (0 för det första) för ett värde i en array. Returnerar -1 om värdet inte hittas.

Syntax	array_find(array, value)
Argument	• **array** - en array • **value** - det värde som ska sökas
Exempel	• `array_find(array('a', 'b', 'c'), 'b')` → 1 • `array_find(array('a', 'b', 'c', 'b'), 'b')` → 1

array_first

Returnerar det första värdet i en array.

Syntax	array_first(array)
Argument	• **array** - en array
Exempel	• `array_first(array('a','b','c'))` → 'a'

array_foreach

Returnerar en array med det angivna uttrycket utvärderat på varje objekt.

Syntax	array_foreach(array, expression)
Argument	• **array** - en array • **expression** - ett uttryck som ska utvärderas för varje objekt. Variabeln *@element* kommer att ersättas med det aktuella värdet och variabeln *@counter* med det aktuella indexet (börjar med 0).
Exempel	• `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

Returnerar det N:te värdet (0 för det första) eller det sista -N:te värdet (-1 för det sista) i en array.

Syntax	array_get(array, pos)
Argument	• **array** - en array • **pos** - det index som ska hämtas (0-baserat)
Exempel	• `array_get(array('a','b','c'),1)` → 'b' • `array_get(array('a','b','c'),-1)` → 'c'

Råd

Du kan också använda *indexoperatorn ([])* för att hämta ett värde från en array.

array_insert

Returnerar en array med det angivna värdet tillagt på den angivna positionen.

Syntax	array_insert(array, pos, value)
Argument	• **array** - en array • **pos** - positionen där tillägget ska göras (0-baserad) • **value** - det värde som ska läggas till
Exempel	• `array_insert(array(1,2,3),1,100)` → [1, 100, 2, 3]

array_intersect

Returnerar TRUE om minst ett element i array1 finns i array2.

Syntax	array_intersect(array1, array2)
Argument	• **array1** - en array • **array2** - en annan array
Exempel	• `array_intersect(array(1,2,3,4),array(4,0,2,5))` → TRUE

array_last

Returnerar det sista värdet i en array.

Syntax	array_last(array)
Argument	• **array** - en array
Exempel	• `array_last(array('a','b','c'))` → 'c'

array_length

Returnerar antalet element i en array.

Syntax	array_length(array)
Argument	• **array** - en array
Exempel	• `array_length(array(1,2,3))` → 3

array_majority

Returnerar de vanligaste värdena i en array.

Syntax	array_majority(array, [option='all']) [] markerar valfria argument
Argument	• **array** - en array • **option='all'** - en sträng som anger hur returvärdena ska hanteras. Giltiga alternativ är: – all: Standard, alla de vanligaste värdena returneras i en array. – any: Returnerar ett av de vanligaste värdena. – median: Returnerar medianen av de vanligaste värdena. Icke-aritmetiska värden ignoreras. – real_majority: Returnerar det värde som förekommer mer än hälften av arrayens storlek.
Exempel	• `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

Returnerar det maximala värdet i en array.

Syntax	array_max(array)
Argument	• **array** - en array
Exempel	• `array_max(array(0,42,4,2))` → 42

array_mean

Returnerar medelvärdet av aritmetiska värden i en array. Icke-numeriska värden i arrayen ignoreras.

Syntax	array_mean(array)
Argument	• **array** - en array
Exempel	• `array_mean(array(0,1,7,66.6,135.4))` → 42 • `array_mean(array(0,84,'a','b','c'))` → 42

array_median

Returnerar medianen för aritmetiska värden i en array. Icke-aritmetiska värden i arrayen ignoreras.

Syntax	array_median(array)
Argument	• **array** - en array
Exempel	• `array_median(array(0,1,42,42,43))` → 42 • `array_median(array(0,1,2,42,'a','b'))` → 1.5

array_min

Returnerar det lägsta värdet i en array.

Syntax	array_min(array)
Argument	• **array** - en array
Exempel	• `array_min(array(43,42,54))` → 42

array_minority

Returnerar de mindre vanliga värdena i en array.

Syntax	array_minority(array, [option='all']) [] markerar valfria argument
Argument	• **array** - en array • **option='all'** - en sträng som anger hur returvärdena ska hanteras. Giltiga alternativ är: – all: Standard, alla mindre vanliga värden returneras i en array. – any: Returnerar ett av de mindre vanliga värdena. – median: Returnerar medianen av de mindre vanliga värdena. Icke-aritmetiska värden ignoreras. – verklig_minoritet: Returnerar värden som förekommer mindre än halva storleken på arrayen.
Exempel	• `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

Returnerar en array med det angivna värdet tillagt i början.

Syntax	array_prepend(array, value)
Argument	• **array** - en array • **value** - det värde som ska läggas till
Exempel	• `array_prepend(array(1,2,3),0)` → [0, 1, 2, 3]

array_prioritize

Returnerar en array som är sorterad enligt den ordning som anges i en annan array. Värden som finns i den första arrayen men som saknas i den andra arrayen läggs till i slutet av resultatet.

Syntax	array_prioritize(array, array_prioritize)
Argument	• **array** - en array • **array_prioritize** - en array med värden ordnade efter prioritet
Exempel	• `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

Returnerar en array där alla poster med det angivna värdet har tagits bort.

Syntax	array_remove_all(array, value)
Argument	• **array** - en array • **value** - de värden som ska tas bort
Exempel	• `array_remove_all(array('a','b','c','b'),'b')` → ['a', 'c']

array_remove_at

Returnerar en array med elementet vid det angivna indexet borttaget. Stöder positivt (0 för det första elementet) och negativt (det sista -N:e värdet, -1 för det sista elementet) index.

Syntax	array_remove_at(array, pos)
Argument	• **array** - en array • **pos** - den position som ska tas bort (0-baserad)
Exempel	• `array_remove_at(array(1, 2, 3), 1)` → [1, 3] • `array_remove_at(array(1, 2, 3), -1)` → [1, 2]

array_replace

Returnerar en array med det angivna värdet, arrayen eller värdekartan ersatt.

Värde- och arrayvariant

Returnerar en array med det angivna värdet eller värdearrayen ersatt av ett annat värde eller en värdearray.

Syntax	array_replace(array, before, after)
Argument	• **array** - den matade arrayen • **before** - det värde eller den array av värden som ska ersättas • **after** - det värde eller den array av värden som ska användas som ersättning
Exempel	• `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', '-']

Kartvariant

Returnerar en array med de angivna map-nycklarna ersatta av sina parade värden.

Syntax	array_replace(array, map)
Argument	• **array** - den matade arrayen • **map** - kartan som innehåller nycklar och värden
Exempel	• `array_replace(array('APP', 'SHOULD', 'ROCK'),map('APP', 'QGIS','SHOULD','DOES'))` → ['QGIS', 'DOES', 'ROCK']

Ytterligare läsning: *replace*, *regexp_replace*

array_reverse

Returnerar den angivna arrayen med arrayvärden i omvänd ordning.

Syntax	array_reverse(array)
Argument	• **array** - en array
Exempel	• `array_reverse(array(2,4,0,10))` → [10, 0, 4, 2]

array_slice

Returnerar en del av arrayen. Skivan definieras av argumenten start_pos och end_pos.

Syntax	array_slice(array, start_pos, end_pos)
Argument	• **array** - en array • **start_pos** - index för startpositionen för skivan (0-baserad). Indexet för start_pos ingår i skivan. Om du använder ett negativt start_pos räknas indexet från slutet av listan (-1-baserat). • **end_pos** - index för skivans slutposition (0-baserat). Indexet för end_pos ingår i skivan. Om du använder ett negativt end_pos räknas indexet från slutet av listan (-1-baserat).
Exempel	• `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

Returnerar den angivna arrayen med dess element sorterade.

Syntax	array_sort(array, [ascending=true]) [] markerar valfria argument
Argument	• **array** - en array • **ascending** - sätt denna parameter till false för att sortera arrayen i fallande ordning
Exempel	• `array_sort(array(3,2,1))` → [1, 2, 3]

array_sum

Returnerar summan av aritmetiska värden i en array. Icke-numeriska värden i arrayen ignoreras.

Syntax	array_sum(array)
Argument	• **array** - en array
Exempel	• `array_sum(array(0,1,39.4,1.6,'a'))` → 42.0

array_to_string

Sammanfogar arrayelement till en sträng som separeras med en avgränsare och använder en valfri sträng för tomma värden.

Syntax	array_to_string(array, [delimiter=','], [empty_value='']) [] markerar valfria argument
Argument	• **array** - den matade arrayen • **delimiter** - den strängavgränsare som används för att separera sammanlänkade arrayelement • **empty_value** - den valfria strängen som ska användas som ersättning för tomma (noll-längd) matchningar
Exempel	• `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'

Ytterligare läsning: *string_to_array*

generate_series

Skapar en array som innehåller en sekvens av tal.

Syntax	generate_series(start, stop, [step=1]) [] markerar valfria argument
Argument	• **start** - första värdet i sekvensen • **stop** - värde som avslutar sekvensen när den har nåtts • **step** - värde som används som steg mellan värdena
Exempel	• `generera_serier(1,5)` → [1, 2, 3, 4, 5] • `generera_serier(5,1,-1)` → [5, 4, 3, 2, 1]

geometries_to_array

Delar upp en geometri i enklare geometrier i en array.

Syntax	geometries_to_array(geometry)
Argument	• **geometri** - inmatningsgeometrin
Exempel	• `geometries_to_array(geom_from_wkt('MultiPoint (1 2, 5 21)'))` → En array som innehåller 'Point (1 2)' och 'Point (5 21)' • `geometries_to_array(geom_from_wkt('GeometryCollection (Polygon ((5 8, 4 1, 3 2, 5 8)),LineString (3 2, 4 2))'))` → en array av en polygon och en linje geometrier • `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))'))` → en array av två polygongeometrier

regexp_matches

Returnerar en array med alla strängar som fångats av fångstgrupper, i den ordning som grupperna själva förekommer i det angivna reguljära uttrycket mot en sträng.

Syntax	regexp_matches(string, regex, [empty_value='']) [] markerar valfria argument
Argument	• **sträng** - strängen för att fånga grupper från mot det reguljära uttrycket • **regex** - det reguljära uttryck som används för att fånga grupper • **empty_value** - den valfria strängen som ska användas som ersättning för tomma (noll-längd) matchningar
Exempel	• `regexp_matches('QGIS=>rocks','(.*)=>(.*)')` → ['QGIS', 'rocks'] • `regexp_matches('key=>','(.*)=>(.*)','tomt värde')` → ['key', 'tomt värde']

Ytterligare läsning: *substr*, *regexp_substr*

string_to_array

Delar upp strängen i en array med hjälp av den medföljande avgränsaren och en valfri sträng för tomma värden.

Syntax	string_to_array(string, [delimiter=','], [empty_value='']) [] markerar valfria argument
Argument	• **sträng** - inmatningssträngen • **delimiter** - den strängavgränsare som används för att dela upp indatasträngen • **empty_value** - den valfria strängen som ska användas som ersättning för tomma (noll-längd) matchningar
Exempel	• `string_to_array('1,2,3',',')` → ['1', '2', '3'] • `string_to_array('1,,3',',','0')` → ['1', '0', '3']

Ytterligare läsning: *array_to_string*

9.2.3 Färgfunktioner

Denna grupp innehåller funktioner för att manipulera färger.

color_cmyk

Returnerar en strängrepresentation av en färg baserad på dess cyan-, magenta-, gul- och svartkomponenter

Syntax	color_cmyk(cyan, magenta, yellow, black)
Argument	• **cyan** - cyankomponent i färgen, som ett procentuellt heltalsvärde från 0 till 100 • **magenta** - magentakomponent i färgen, som ett procentuellt heltalsvärde från 0 till 100 • **yellow** - färgens gula komponent, som ett procentuellt heltalsvärde från 0 till 100 • **black** - färgens svarta komponent, som ett procentuellt heltalsvärde från 0 till 100
Exempel	• `färg_cmyk(100,50,0,10)` → '0,115,230'

color_cmyka

Returnerar en strängrepresentation av en färg baserad på dess cyan-, magenta-, gul-, svart- och alfa-komponenter (transparens)

Syntax	color_cmyka(cyan, magenta, yellow, black, alpha)
Argument	• **cyan** - cyankomponent i färgen, som ett procentuellt heltalsvärde från 0 till 100 • **magenta** - magentakomponent i färgen, som ett procentuellt heltalsvärde från 0 till 100 • **yellow** - färgens gula komponent, som ett procentuellt heltalsvärde från 0 till 100 • **black** - färgens svarta komponent, som ett procentuellt heltalsvärde från 0 till 100 • **alpha** - alfakomponent som ett heltalsvärde från 0 (helt transparent) till 255 (ogenomskinlig).
Exempel	• `färg_cmyka(100,50,0,10,200)` → '0,115,230,200'

color_cmykf

Returnerar ett färgobjekt baserat på dess cyan-, magenta-, gul-, svart- och alfakomponenter.

Syntax	color_cmykf(cyan, magenta, gul, svart, [alpha=1.0]) [] markerar valfria argument
Argument	• **cyan** - cyankomponent som ett flottörvärde från 0,0 till 1,0 • **magenta** - magentakomponent som ett flottörvärde från 0,0 till 1,0 • **yellow** - gul komponent som ett flottörvärde från 0,0 till 1,0 • **svart** - svart komponent som ett flottörvärde från 0,0 till 1,0 • **alpha** - alfakomponent som ett flottörvärde från 0,0 till 1,0
Exempel	• `färg_cmykf(1,0.9,0.81,0.62)` → CMYKA: 1.00,0.90,0.81,0.62,1.00

color_grayscale_average

Applicerar ett gråskalefilter på en färg och returnerar den. Returnerad typ är samma som färgargumentet, dvs. en färgsträngrepresentation eller ett färgobjekt.

Syntax	color_grayscale_average(color)
Argument	• **color** - en representation av en färgsträng eller ett färgobjekt
Exempel	• `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

Returnerar en strängrepresentation av en färg baserat på dess attribut för nyans, mättnad och ljushet.

Syntax	color_hsl(hue, saturation, lightness)
Argument	• **hue** - färgens nyans, som ett heltalsvärde från 0 till 360 • **saturation** - mättnadsprocent för färgen som ett heltal från 0 till 100 • **lightness** - färgens ljushetsprocent som ett heltalsvärde från 0 till 100
Exempel	• `färg_hsl(100,50,70)` → '166,217,140'

color_hsla

Returnerar en strängrepresentation av en färg baserat på dess attribut för nyans, mättnad, ljushet och alfa (transparens)

Syntax	color_hsla(hue, saturation, lightness, alpha)
Argument	• **hue** - färgens nyans, som ett heltalsvärde från 0 till 360 • **saturation** - mättnadsprocent för färgen som ett heltal från 0 till 100 • **lightness** - färgens ljushetsprocent som ett heltalsvärde från 0 till 100 • **alpha** - alfakomponent som ett heltalsvärde från 0 (helt transparent) till 255 (ogenomskinlig).
Exempel	• `färg_hsla(100,50,70,200)` → '166,217,140,200'

color_hslf

Returnerar ett färgobjekt baserat på dess attribut för nyans, mättnad och ljushet.

Syntax	color_hslf(nyans, mättnad, ljushet, [alpha=1.0]) [] markerar valfria argument
Argument	• **hue** - färgens nyans, som ett flottörvärde från 0,0 till 1,0 • **saturation** - färgens mättnad som ett flottörvärde från 0,0 till 1,0 • **lightness** - färgens ljushet som ett floatvärde från 0,0 till 1,0 • **alpha** - alfakomponent som ett flottörvärde från 0,0 till 1,0
Exempel	• `färg_hslf(0,3,0,52,0,7)` → HSLA: 0,30,0,52,0,70,1,00

color_hsv

Returnerar en strängrepresentation av en färg baserat på dess attribut för nyans, mättnad och värde.

Syntax	color_hsv(hue, saturation, value)
Argument	• **hue** - färgens nyans, som ett heltalsvärde från 0 till 360 • **saturation** - mättnadsprocent för färgen som ett heltal från 0 till 100 • **värde** - värdeprocent för färgen som ett heltal från 0 till 100
Exempel	• `färg_hsv(40,100,100)` → '255,170,0'

color_hsva

Returnerar en strängrepresentation av en färg baserat på dess attribut för nyans, mättnad, värde och alfa (transparens).

Syntax	color_hsva(hue, saturation, value, alpha)
Argument	• **hue** - färgens nyans, som ett heltalsvärde från 0 till 360 • **saturation** - mättnadsprocent för färgen som ett heltal från 0 till 100 • **värde** - värdeprocent för färgen som ett heltal från 0 till 100 • **alpha** - alfakomponent som ett heltalsvärde från 0 (helt transparent) till 255 (ogenomskinlig)
Exempel	• `färg_hsva(40,100,100,200)` → '255,170,0,200'

color_hsvf

Returnerar ett färgobjekt baserat på dess attribut för nyans, mättnad och värde.

Syntax	färg_hsvf(nyans, mättnad, värde, [alfa=1,0]) [] markerar valfria argument
Argument	• **hue** - färgens nyans, som ett flottörvärde från 0,0 till 1,0 • **saturation** - färgens mättnad som ett flottörvärde från 0,0 till 1,0 • **värde** - färgens värde i form av ett flottörvärde från 0,0 till 1,0 • **alpha** - alfakomponent som ett flottörvärde från 0,0 till 1,0
Exempel	• `färg_hsvf(0,4,1,0,6)` → HSVA: 0,40,1,00,0,60,1,00

color_mix

Returnerar en färg som blandar röd-, grön-, blå- och alfavärdena för två angivna färger baserat på ett givet förhållande. Returnerad typ är samma som färgargumenten, dvs. en färgsträngrepresentation eller ett färgobjekt.

Syntax	color_mix(color1, color2, ratio)
Argument	• **color1** - en färgsträng eller ett färgobjekt • **color2** - en färgsträng eller ett färgobjekt • **ratio** - ett förhållande
Exempel	• `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

Returnerar en sträng som representerar en färg som blandar röd-, grön-, blå- och alfavärdena för två angivna färger baserat på ett givet förhållande.

Syntax	color_mix_rgb(color1, color2, ratio)
Argument	• **färg1** - en färgsträng • **color2** - en färgsträng • **ratio** - ett förhållande
Exempel	• `color_mix_rgb('0,0,0','255,255,255',0.5)` → '127,127,127,255'

color_part

Returnerar en specifik komponent från en färgsträng eller ett färgobjekt, t.ex. den röda komponenten eller alfakomponenten.

Syntax	color_part(color, component)
Argument	• **color** - en färgsträng eller ett färgobjekt • **component** - en sträng som motsvarar den färgkomponent som ska returneras. Giltiga alternativ är: – röd: RGB röd komponent (0-255) – grön: RGB grön komponent (0-255) – blå: RGB blå komponent (0-255) – alfa: värde för alfa (transparens) (0-255) – nyans: HSV-nyans (0-360) – mättnad: HSV-mättnad (0-100) – värde: HSV-värde (0-100) – hsl_hue: HSL-nyans (0-360) – hsl_saturation: HSL-mättnad (0-100) – ljushet: HSL-ljushet (0-100) – cyan: CMYK-cyankomponent (0-100) – magenta: CMYK-komponent för magenta (0-100) – gul: CMYK gul komponent (0-100) – svart: CMYK svart komponent (0-100)
Exempel	• `färg_del('200,10,30','grön')` → 10 • `till_int(färg_del(färg_cmykf(0,1,0,2,0,3,0,9),'svart'))` → 90

color_rgb

Returnerar en strängrepresentation av en färg baserad på dess röda, gröna och blå komponenter.

Syntax	color_rgb(red, green, blue)
Argument	• **red** - röd komponent som ett heltalsvärde från 0 till 255 • **green** - grön komponent som ett heltalsvärde från 0 till 255 • **blue** - blå komponent som ett heltalsvärde från 0 till 255
Exempel	• `färg_rgb(255,127,0)` → '255,127,0'

color_rgba

Returnerar en strängrepresentation av en färg baserad på dess röda, gröna, blå och alfa (transparens) komponenter.

Syntax	color_rgba(red, green, blue, alpha)
Argument	• **red** - röd komponent som ett heltalsvärde från 0 till 255 • **green** - grön komponent som ett heltalsvärde från 0 till 255 • **blue** - blå komponent som ett heltalsvärde från 0 till 255 • **alpha** - alfakomponent som ett heltalsvärde från 0 (helt transparent) till 255 (ogenomskinlig).
Exempel	• `färg_rgba(255,127,0,200)` → '255,127,0,200'

color_rgbf

Returnerar ett färgobjekt baserat på dess röd-, grön-, blå- och alfakomponenter.

Syntax	color_rgbf(röd, grön, blå, [alpha=1,0]) [] markerar valfria argument
Argument	• **red** - röd komponent som ett flytvärde från 0,0 till 1,0 • **green** - grön komponent som ett flottörvärde från 0,0 till 1,0 • **blue** - blå komponent som ett flottörvärde från 0,0 till 1,0 • **alpha** - alfakomponent som ett flottörvärde från 0,0 till 1,0
Exempel	• `färg_rgbf(1,0,0,5,0)` → RGBA: 1,00,0,50,0,00,1,00

create_ramp

Returnerar en gradientramp från en karta med färgsträngar och steg.

Syntax	create_ramp(map, [discrete=false]) [] markerar valfria argument
Argument	• **map** - en karta över färgsträngar och steg • **discrete** - sätt denna parameter till true för att skapa en diskret färgramp
Exempel	• `ramp_color(create_ramp(map(0,'0,0,0',1,'255,0,0')),1)` → '255,0,0,255'

darker

Returnerar en mörkare (eller ljusare) färg. Returnerad typ är samma som för färgargument, dvs. en färgsträngrepresentation eller ett färgobjekt.

Syntax	darker(color, factor)
Argument	• **color** - en färgsträng eller ett färgobjekt • **factor** - ett heltal som motsvarar mörkläggningsfaktorn: – om faktorn är större än 100 ger funktionen en mörkare färg (om faktorn är 200 får man t.ex. en färg som är hälften så ljus); – om faktorn är mindre än 100 är returfärgen ljusare, men det rekommenderas att använda funktionen lighter() för detta ändamål; – om faktorn är 0 eller negativ är returvärdet ospecificerat.
Exempel	• `mörkare('200,10,30', 200)` → '100,5,15,255'

För vidare läsning: *lighter*

lighter

Returnerar en ljusare (eller mörkare) färg. Returnerad typ är samma som för färgargument, dvs. en färgsträngrepresentation eller ett färgobjekt.

Syntax	lighter(color, factor)
Argument	• **color** - en färgsträng eller ett färgobjekt • **factor** - ett heltal som motsvarar blixtfaktorn: – om faktorn är större än 100 ger funktionen en ljusare färg (om faktorn är 150 blir färgen t.ex. 50% ljusare); – om faktorn är mindre än 100 är returfärgen mörkare, men det rekommenderas att använda funktionen darker() för detta ändamål; – om faktorn är 0 eller negativ är returvärdet ospecificerat.
Exempel	• `lighter('200,10,30', 200)` → '255,158,168,255'

Ytterligare läsning: *darker*

project_color

Returnerar en färg från projektets färgschema.

Syntax	project_color(name)
Argument	• **namn** - ett namn på en färg
Exempel	• `project_color('Logofärg')` → '20,140,50'

Ytterligare läsning: *Ställa in projektfärger*

project_color_object

Returnerar en färg från projektets färgschema. I motsats till project_color som returnerar en färgsträng, returnerar project_color_object ett färgobjekt.

Syntax	project_color_object(name)
Argument	• **namn** - ett namn på en färg
Exempel	• `project_color_object('Logofärg')` → RGBA: 0,08,0,55,0,20,1,00

ramp_color

Returnerar en sträng som representerar en färg från en färgramp.

Sparad rampvariant

Returnerar en sträng som representerar en färg från en sparad ramp

Syntax	ramp_color(ramp_name, value)
Argument	• **ramp_name** - namnet på färgrampen i form av en sträng, t.ex. "Spectral • **värde** - den position på rampen som färgen ska väljas från som ett verkligt tal mellan 0 och 1
Exempel	• `ramp_color('Spectral',0.3)` → '253,190,115,255'

Observera

De tillgängliga färgramperna varierar mellan olika QGIS-installationer. Den här funktionen kanske inte ger förväntat resultat om du flyttar ditt QGIS-projekt mellan olika installationer.

Uttryck-skapad rampvariant

Returnerar en sträng som representerar en färg från en ramp som skapats med ett uttryck

Syntax	ramp_color(ramp, value)
Argument	• **ramp** - färgen ramp • **värde** - den position på rampen som färgen ska väljas från som ett verkligt tal mellan 0 och 1
Exempel	• `ramp_color(create_ramp(map(0,'0,0,0',1,'255,0,0')),1)` → '255,0,0,255'

Ytterligare läsning: *Inställning av en färgramp*, *Kortkommando för färgrampens rullgardinsmeny*

ramp_color_object

Returnerar ett färgobjekt från en färgramp. I motsats till ramp_color som returnerar en färgsträngrepresentation, returnerar ramp_color_object ett färgobjekt.

Sparad rampvariant

Returnerar ett färgobjekt från en sparad ramp

Syntax	ramp_color_object(ramp_name, value)
Argument	• **ramp_name** - namnet på färgrampen i form av en sträng, t.ex. "Spectral • **värde** - den position på rampen som färgen ska väljas från som ett verkligt tal mellan 0 och 1
Exempel	• `ramp_color_object('Spectral',0.3)` → RGBA: 0.99,0.75,0.45,1.00

Observera

De tillgängliga färgramperna varierar mellan olika QGIS-installationer. Den här funktionen kanske inte ger förväntat resultat om du flyttar ditt QGIS-projekt mellan olika installationer.

Uttryck-skapad rampvariant

Returnerar ett färgobjekt från en ramp som skapats med ett uttryck

Syntax	ramp_color_object(ramp, value)
Argument	• **ramp** - färgen ramp • **värde** - den position på rampen som färgen ska väljas från som ett verkligt tal mellan 0 och 1
Exempel	• `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

Ställer in en specifik färgkomponent för en färgsträng eller ett färgobjekt, t.ex. röd komponent eller alfakomponent.

Syntax	set_color_part(color, component, value)
Argument	• **color** - en färgsträng eller ett färgobjekt • **component** - en sträng som motsvarar den färgkomponent som ska ställas in. Giltiga alternativ är: – röd: RGB röd komponent (0-255) – grön: RGB grön komponent (0-255) – blå: RGB blå komponent (0-255) – alfa: värde för alfa (transparens) (0-255) – nyans: HSV-nyans (0-360) – mättnad: HSV-mättnad (0-100) – värde: HSV-värde (0-100) – hsl_hue: HSL-nyans (0-360) – hsl_saturation: HSL-mättnad (0-100) – ljushet: HSL-ljushet (0-100) – cyan: CMYK-cyankomponent (0-100) – magenta: CMYK-komponent för magenta (0-100) – gul: CMYK gul komponent (0-100) – svart: CMYK svart komponent (0-100) • **värde** - nytt värde för färgkomponenten, med hänsyn till de intervall som anges ovan
Exempel	• `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 Villkorliga funktioner

Den här gruppen innehåller funktioner för att hantera villkorliga kontroller i uttryck.

CASE

CASE används för att utvärdera en serie villkor och returnera ett resultat för det första villkoret som uppfylls. Villkoren utvärderas sekventiellt, och om ett villkor är sant stoppas utvärderingen och motsvarande resultat returneras. Om inget av villkoren är sant returneras värdet i ELSE-satsen. Om ingen ELSE-sats anges och inget av villkoren uppfylls, returneras dessutom NULL.

CASE

WHEN *villkor* THEN *resultat*

[…n]

[ELSE *resultat*]

END

[ ] markerar valfria komponenter

Argument	• **WHEN condition** - Ett villkorsuttryck som ska utvärderas • **THEN result** - Om *condition* utvärderas till True utvärderas och returneras *result*. • **ELSE result** - Om inget av ovanstående villkor utvärderas till True utvärderas och returneras *result*.
Exempel	• `CASE WHEN "name" IS NULL THEN "None" END` → Returnerar strängen ”None” om fältet ”name” är NULL • `CASE WHEN $area > 10000 THEN 'Stor fastighet' WHEN $area > 5000 THEN 'Medelstor fastighet' ELSE 'Liten fastighet' END` → Returnerar strängen ’Stor fastighet’ om området är större än 10000, ’Medelstor fastighet’ om området är mellan 5000 och 10000 och ’Liten fastighet’ för övriga

coalesce

Returnerar det första icke-NULL-värdet från uttrycksförteckningen.

Denna funktion kan ta ett valfritt antal argument.

Syntax	coalesce(uttryck1, uttryck2, …)
Argument	• **expression** - vilket giltigt uttryck eller värde som helst, oavsett typ.
Exempel	• `coalesce(NULL, 2)` → 2 • `coalesce(NULL, 2, 3)` → 2 • `coalesce(7, NULL, 3*2)` → 7 • `coalesce("fieldA", "fallbackField", 'ERROR')` → värdet av fieldA om det är icke-NULL annars värdet av ”fallbackField” eller strängen ’ERROR’ om båda är NULL

if

Testar ett villkor och returnerar ett annat resultat beroende på den villkorliga kontrollen.

Syntax	if(condition, result_when_true, result_when_false)
Argument	• **villkor** - det villkor som ska kontrolleras • **result_when_true** - det resultat som returneras när villkoret är sant eller ett annat värde som inte omvandlas till falskt. • **result_when_false** - det resultat som returneras när villkoret är falskt eller ett annat värde som konverteras till falskt, t.ex. 0 eller ”. NULL kommer också att konverteras till false.
Exempel	• `if( 1+1=2, 'Ja', 'Nej' )` → ’Ja’ • `if( 1+1=3, 'Ja', 'Nej' )` → ’Nej’ • `if( 5 > 3, 1, 0)` → 1 • `if( '', 'Det är sant (inte tomt)', 'Det är falskt (tomt)' )` → ’Det är falskt (tomt)’ • `if( ' ', 'Det är sant (inte tomt)', 'Det är falskt (tomt)' )` → ’Det är sant (inte tomt)’ • `if( 0, 'Ett', 'Noll' )` → ’Noll’ • `if( 10, 'Ett', 'Noll' )` → ’Ett’

nullif

Returnerar ett NULL-värde om värde1 är lika med värde2; annars returneras värde1. Detta kan användas för att villkorligt ersätta värden med NULL.

Syntax	nullif(värde1, värde2)
Argument	• **value1** - Det värde som antingen ska användas eller ersättas med NULL. • **value2** - Det kontrollvärde som kommer att utlösa NULL-ersättningen.
Exempel	• `nullif('(ingen)', '(ingen)')` → NULL • `nullif('text', '(ingen)')` → 'text' • `nullif("name", '')` → NULL, om name är en tom sträng (eller redan NULL), namnet i alla andra fall.

regexp_match

Returnerar den första matchande positionen som matchar ett reguljärt uttryck i en unikodsträng, eller 0 om delsträngen inte hittas.

Syntax	regexp_match(input_string, regex)
Argument	• **input_string** - den sträng som ska testas mot det reguljära uttrycket • **regex** - Det reguljära uttryck som ska testas mot. Backslash-tecken måste dubbeleskapsas (t.ex. "\\s" för att matcha ett tecken med vitt utrymme eller "\\b" för att matcha en ordgräns).
Exempel	• `regexp_match('QGIS ROCKS','\\sROCKS')` → 5 • `regexp_match('Budač','udač\\b')` → 2

try

Prövar ett uttryck och returnerar dess värde om det är felfritt. Om uttrycket ger ett fel returneras ett alternativt värde när det anges, annars returnerar funktionen NULL.

Syntax	try(expression, [alternative]) [] markerar valfria argument
Argument	• **expression** - det uttryck som ska köras • **alternative** - det resultat som kommer att returneras om uttrycket returnerar ett fel.
Exempel	• `try( to_int( '1' ), 0 )` → 1 • `try( to_int( 'a' ), 0 )` → 0 • `try( to_date( 'invalid_date' ) )` → NULL

9.2.5 Konverteringar Funktioner

Denna grupp innehåller funktioner för att konvertera en datatyp till en annan (t.ex. sträng från/till heltal, binär från/till sträng, sträng till datum, …).

from_base64

Avkodar en sträng i Base64-kodning till ett binärt värde.

Syntax	from_base64(string)
Argument	• **sträng** - den sträng som ska avkodas
Exempel	• `from_base64('UUdJUw==')` → 'QGIS'

hash

Skapar en hash från en sträng med en given metod. En byte (8 bitar) representeras av två hexadecimala "siffror", så "md4" (16 byte) ger en 16 * 2 = 32 tecken lång hexadisk sträng och "keccak_512" (64 byte) ger en 64 * 2 = 128 tecken lång hexadisk sträng.

Syntax	hash(string, method)
Argument	• **sträng** - strängen som ska hashas • **method** - Hashmetod bland följande: "md4", "md5", "sha1", "sha224", "sha384", "sha512", "sha3_224", "sha3_256", "sha3_384", "sha3_512", "keccak_224", "keccak_256", "keccak_384", "keccak_512
Exempel	• `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

Skapar en md5-hash från en sträng.

Syntax	md5(string)
Argument	• **sträng** - strängen som ska hashas
Exempel	• `md5('QGIS')` → '57470aaa9e22adaefac7f5f342f1c6da'

sha256

Skapar en sha256-hash från en sträng.

Syntax	sha256(string)
Argument	• **sträng** - strängen som ska hashas
Exempel	• `sha256('QGIS')` → 'eb045cba7a797aaa06ac58830846e40c8e8c780bc0676d3393605fae50c05309'

to_base64

Kodar ett binärt värde till en sträng med hjälp av Base64-kodning.

Syntax	to_base64(value)
Argument	• **värde** - det binära värdet som ska kodas
Exempel	• `to_base64('QGIS')` → 'UUdJUw=='

to_date

Konverterar en sträng till ett datumobjekt. En valfri formatsträng kan anges för att analysera strängen; se QDate::fromString eller dokumentationen av funktionen format_date för ytterligare dokumentation om formatet. Som standard används den aktuella QGIS-användarens locale.

Syntax	to_date(string, [format], [language]) [] markerar valfria argument
Argument	• **sträng** - sträng som representerar ett datumvärde • **format** - format som används för att konvertera strängen till ett datum • **language** - språk (gemener, två eller tre bokstäver, ISO 639 språkkod) som används för att konvertera strängen till ett datum. Som standard används den aktuella QGIS-användarens språk.
Exempel	• `to_date('2012-05-04')` → 2012-05-04 • `to_date('June 29, 2019','MMMM d, yyyy')` → 2019-06-29, om aktuell lokal använder namnet 'June' för den sjätte månaden, annars uppstår ett fel • `to_date('29 juni, 2019','d MMMM, åååå','fr')` → 2019-06-29

För vidare läsning: *format_date*

to_datetime

Konverterar en sträng till ett datetime-objekt. En valfri formatsträng kan anges för att analysera strängen; se QDate::fromString, QTime::fromString eller dokumentationen av funktionen format_date för ytterligare dokumentation om formatet. Som standard används den aktuella QGIS-användarens locale.

Syntax	to_datetime(string, [format], [language]) [] markerar valfria argument
Argument	• **sträng** - sträng som representerar ett datetime-värde • **format** - format som används för att konvertera strängen till en datetime • **language** - språk (gemener, två eller tre bokstäver, ISO 639 språkkod) som används för att konvertera strängen till en datetime. Som standard används den aktuella QGIS-användarens språk.
Exempel	• `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, om den aktuella lokala använder namnet 'June' för den sjätte månaden, annars uppstår ett fel • `to_datetime('29 juin, 2019 @ 12:34','d MMMM, yyyy @ HH:mm', 'fr')` → 2019-06-29T12:34

För vidare läsning: *format_date*

to_decimal

Konverterar en koordinat för grader, minuter och sekunder till dess decimala motsvarighet.

Syntax	to_decimal(value)
Argument	• **värde** - En sträng med grader, minuter och sekunder.
Exempel	• `till_decimal('6°21\'16.445')` → 6.3545680555

to_dm

Konverterar en koordinat till grad, minut.

Syntax	to_dm(coordinate, axis, precision, [formatting=]) [] markerar valfria argument
Argument	• **koordinat** - Ett värde för latitud eller longitud. • **axis** - Koordinatens axel. Antingen ”x” eller ”y”. • **precision** - Antal decimaler. • **formatting** - Anger formateringstypen. Acceptabla värden är NULL (standard), ’aligned’ eller ’suffix’.
Exempel	• `till_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

Konverterar en koordinat till grad, minut, sekund.

Syntax	to_dms(coordinate, axis, precision, [formatting=]) [] markerar valfria argument
Argument	• **koordinat** - Ett värde för latitud eller longitud. • **axis** - Koordinatens axel. Antingen ”x” eller ”y”. • **precision** - Antal decimaler. • **formatting** - Anger formateringstypen. Acceptabla värden är NULL (standard), ’aligned’ eller ’suffix’.
Exempel	• `till_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

Konverterar en sträng till ett heltal. Om ett värde inte kan konverteras till heltal är uttrycket ogiltigt (t.ex. ”123asd” är ogiltigt).

Syntax	to_int(string)
Argument	• **sträng** - sträng som ska konverteras till heltal
Exempel	• `to_int('123')` → 123

to_interval

Konverterar en sträng till en intervalltyp. Kan användas för att ta dagar, timmar, månad etc. från ett datum.

Syntax	to_interval(string)
Argument	• **sträng** - en sträng som representerar ett intervall. Tillåtna format är {n} dagar {n} timmar {n} månader.
Exempel	• `to_interval('1 dag 2 timmar')` → intervall: 1,08333 dagar • `to_interval( '0,5 timmar' )` → intervall: 30 minuter • `to_datetime('2012-05-05 12:00:00') - to_interval('1 dag 2 timmar')` → 2012-05-04T10:00:00

to_real

Konverterar en sträng till ett reellt tal. Om ett värde inte kan konverteras till ett reellt tal är uttrycket ogiltigt (t.ex. ”123.56asd” är ogiltigt). Tal avrundas efter att ändringar har sparats om precisionen är mindre än resultatet av konverteringen.

Syntax	to_real(string)
Argument	• **sträng** - sträng som ska konverteras till reellt tal
Exempel	• `till_verklig('123.45')` → 123.45

to_string

Konverterar ett tal till en sträng.

Syntax	to_string(number)
Argument	• **number** - Heltal eller verkligt värde. Det tal som ska konverteras till sträng.
Exempel	• `to_string(123)` → ’123’

Ytterligare läsning: *format_number*

to_time

Konverterar en sträng till ett tidsobjekt. En valfri formatsträng kan anges för att analysera strängen; se QTime::fromString för ytterligare dokumentation om formatet.

Syntax	to_time(string, [format], [language]) [] markerar valfria argument
Argument	• **sträng** - sträng som representerar ett tidsvärde • **format** - format som används för att konvertera strängen till en tid • **language** - språk (gemener, två eller tre bokstäver, ISO 639 språkkod) som används för att konvertera strängen till en tid
Exempel	• `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

För vidare läsning: *format_date*

9.2.6 Anpassade funktioner

Denna grupp innehåller funktioner som skapats av användaren. Se *Funktion Redaktör* för mer information.

9.2.7 Datum- och tidsfunktioner

Denna grupp innehåller funktioner för hantering av datum- och tidsdata. Den här gruppen delar flera funktioner med grupperna *Konverteringar Funktioner* (to_date, to_time, to_datetime, to_interval) och *Strängfunktioner* (format_date).

Observera

Lagring av datum, datatid och intervall i fält

Möjligheten att lagra *datum*-, *tid*- och *datumtid*-värden direkt i fält beror på datakällans leverantör (t.ex. accepterar Shapefile *datum*-format, men inte *datumtid*- eller *tid*-format). Följande är några förslag för att övervinna denna begränsning:

- *date*, *datetime* och *time* kan konverteras och lagras i fält av texttyp med hjälp av funktionen *format_date()*.
- *Intervall* kan lagras i fält av heltals- eller decimaltyp efter användning av en av funktionerna för datumutdragning (t.ex. *day()* för att få intervallet uttryckt i dagar)

age

Returnerar skillnaden mellan två datum eller datatider.

Skillnaden returneras som en `Interval` och måste användas med någon av följande funktioner för att få ut användbar information:

- `year`
- `month`
- `week`
- `day`
- `hour`
- `minute`
- `second`

Syntax	age(datetime1, datetime2)
Argument	• **datetime1** - en sträng, ett datum eller en datetime som representerar det senare datumet • **datetime2** - en sträng, ett datum eller en datetime som representerar det tidigare datumet
Exempel	• `day(age('2012-05-12','2012-05-02'))` → 10 • `timme(ålder('2012-05-12','2012-05-02'))` → 240

datetime_from_epoch

Returnerar en datatid vars datum och tid är antalet millisekunder, msecs, som har passerat sedan 1970-01-01T00:00:00.000, Coordinated Universal Time (Qt.UTC), och konverterat till Qt.LocalTime.

Syntax	datetime_from_epoch(int)
Argument	• **int** - antal (millisekunder)
Exempel	• `datatid_från_epoch(1483225200000)` → 2017-01-01T00:00:00

day

Extraherar dagen från ett datum, eller antalet dagar från ett intervall.

Datumvariant

Extraherar dagen från ett datum eller en datetime.

Syntax	day(date)
Argument	• **date** - ett datum- eller datetime-värde
Exempel	• `dag('2012-05-12')` → 12

Intervallvariant

Beräknar längden i dagar för ett intervall.

Syntax	day(interval)
Argument	• **interval** - intervallvärde för att returnera antal dagar från
Exempel	• `day(to_interval('3 dagar'))` → 3 • `dag(till_intervall('3 veckor 2 dagar'))` → 23 • `dag(ålder('2012-01-01','2010-01-01'))` → 730

day_of_week

Returnerar veckodagen för ett angivet datum eller en angiven datatid. Det returnerade värdet sträcker sig från 0 till 6, där 0 motsvarar en söndag och 6 en lördag.

Syntax	day_of_week(date)
Argument	• **date** - datum eller datetime-värde
Exempel	• `dag_av_vecka(till_datum('2015-09-21'))` → 1

epoch

Returnerar intervallet i millisekunder mellan unix-epoken och ett givet datumvärde.

Syntax	epoch(date)
Argument	• **date** - ett datum- eller datetime-värde
Exempel	• `epoch(to_date('2017-01-01'))` → 1483203600000

format_date

Formaterar en datatyp eller sträng till ett anpassat strängformat. Använder Qt datum/tid formatsträngar. Se QDateTime::toString.

Syntax	format_date(datetime, format, [language]) [] markerar valfria argument
Argument	• **datetime** - datum, tid eller datetime-värde • **format** - Strängmall som används för att formatera strängen.

Uttryck	Utmatning
d	dagen som ett nummer utan inledande nolla (1 till 31)
dd	dagen som ett nummer med en inledande nolla (01 till 31)
ddd	det förkortade lokala dagsnamnet (t.ex. ”mån” till ”sön”)
dddd	det långa lokaliserade dagsnamnet (t.ex. ”måndag” till ”söndag”)
M	månaden som ett nummer utan inledande nolla (1-12)
MM	månaden som ett nummer med en inledande nolla (01-12)
MMM	det förkortade lokala månadsnamnet (t.ex. ”Jan” till ”Dec”)
MMMM	det långa lokaliserade månadsnamnet (t.ex. ”januari” till ”december”)
yy	årtalet som tvåsiffrigt nummer (00-99)
yyyy	året som ett fyrsiffrigt tal

Dessa uttryck kan användas för tidsdelen av formatsträngen:

Uttryck	Utmatning
h	timmen utan inledande nolla (0 till 23 eller 1 till 12 om AM/PM-visning)
hh	timmen med en inledande nolla (00 till 23 eller 01 till 12 om AM/PM-visning)
H	timmen utan inledande nolla (0 till 23, även med AM/PM-visning)
HH	timmen med en inledande nolla (00 till 23, även med AM/PM-visning)
m	minuten utan inledande nolla (0 till 59)
mm	minuten med en inledande nolla (00 till 59)
s	den andra siffran utan inledande nolla (0 till 59)
ss	den andra siffran med en inledande nolla (00 till 59)
z	millisekunder utan nollor efter (0 till 999)
zzz	millisekunder med nollor efter (000 till 999)
AP eller A	tolkas som en AM/PM-tid. *AP* måste vara antingen ’AM’ eller ’PM’.
ap eller en	Tolkas som en AM/PM-tid. *ap* måste vara antingen ’am’ eller ’pm’.

- **language** - språk (gemener, två eller tre bokstäver, ISO 639 språkkod) som används för att formatera datumet till en anpassad sträng. Som standard används den aktuella QGIS-användarens språk.

Exempel

- `format_date('2012-05-15','dd.MM.åååå')` → ’15.05.2012’
- `format_date('2012-05-15','d MMMM åååå','fr')` → ’15 maj 2012’
- `format_date('2012-05-15','dddd')` → ”Tuesday”, om den aktuella locale är en engelsk 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

Extraherar timdelen från en datetime eller tid, eller antalet timmar från ett intervall.

Tidsvariant

Extraherar timdelen från en tid eller datetime.

Syntax	hour(datetime)
Argument	• **datetime** - ett tids- eller datetime-värde
Exempel	• `timme( till_datetime('2012-07-22 13:24:57') )` → 13

Intervallvariant

Beräknar längden i timmar för ett intervall.

Syntax	hour(interval)
Argument	• **interval** - intervallvärde för att returnera antal timmar från
Exempel	• `timme(till_intervall('3 timmar'))` → 3 • `hour(age('2012-07-22T13:00:00','2012-07-22T10:00:00'))` → 3 • `timme(ålder('2012-01-01','2010-01-01'))` → 17520

make_date

Skapar ett datumvärde från siffrorna år, månad och dag.

Syntax	make_date(year, month, day)
Argument	• **year** - Årtalets nummer. År 1 till 99 tolkas som de är. År 0 är ogiltigt. • **month** - Månadsnummer, där 1=Januari • **day** - Dagnummer, börjar med 1 för den första dagen i månaden
Exempel	• `make_date(2020,5,4)` → datumvärde 2020-05-04

make_datetime

Skapar ett datetime-värde från siffrorna år, månad, dag, timme, minut och sekund.

Syntax	make_datetime(year, month, day, hour, minute, second)
Argument	• **year** - Årtalets nummer. År 1 till 99 tolkas som de är. År 0 är ogiltigt. • **month** - Månadsnummer, där 1=Januari • **day** - Dagnummer, börjar med 1 för den första dagen i månaden • **hour** - Timnummer • **minut** - Minuter • **sekund** - Sekunder (fraktionerade värden inkluderar millisekunder)
Exempel	• `make_datetime(2020,5,4,13,45,30.5)` → datetime-värde 2020-05-04 13:45:30.500

make_interval

Skapar ett intervallvärde från värdena år, månad, veckor, dagar, timmar, minuter och sekunder.

Syntax	make_interval([years=0], [months=0], [weeks=0], [days=0], [hours=0], [minutes=0], [seconds=0]) [] markerar valfria argument
Argument	• **years** - Antal år (förutsätter en årslängd på 365,25 dagar). • **months** - Antal månader (förutsätter en månadslängd på 30 dagar) • **weeks** - Antal veckor • **days** - Antal dagar • **hours** - Antal timmar • **minuter** - Antal minuter • **sekunder** - Antal sekunder
Exempel	• `make_interval(hours:=3)` → intervall: 3 timmar • `make_interval(dagar:=2, timmar:=3)` → intervall: 2,125 dagar • `make_interval(minuter:=0,5, sekunder:=5)` → intervall: 35 sekunder

make_time

Skapar ett tidsvärde från tim-, minut- och sekundtal.

Syntax	make_time(hour, minute, second)
Argument	• **hour** - Timnummer • **minut** - Minuter • **sekund** - Sekunder (fraktionerade värden inkluderar millisekunder)
Exempel	• `make_time(13,45,30.5)` → tidsvärde 13:45:30.500

minute

Extraherar minutdelen från en datatid eller tid, eller antalet minuter från ett intervall.

Tidsvariant

Extraherar minutdelen från en tid eller datetime.

Syntax	minute(datetime)
Argument	• **datetime** - ett tids- eller datetime-värde
Exempel	• `minut( to_datetime('2012-07-22 13:24:57') )` → 24

Intervallvariant

Beräknar längden i minuter för ett intervall.

Syntax	minute(interval)
Argument	• **interval** - intervallvärde för att returnera antal minuter från
Exempel	• `minute(to_interval('3 minuter'))` → 3 • `minute(age('2012-07-22T00:20:00','2012-07-22T00:00:00'))` → 20 • `minute(age('2012-01-01','2010-01-01'))` → 1051200

month

Extraherar månadsdelen från ett datum, eller antalet månader från ett intervall.

Datumvariant

Extraherar månadsdelen från ett datum eller en datetime.

Syntax	month(date)
Argument	• **date** - ett datum- eller datetime-värde
Exempel	• `månad('2012-05-12')` → 05

Intervallvariant

Beräknar längden i månader för ett intervall.

Syntax	month(interval)
Argument	• **interval** - intervallvärde för att returnera antal månader från
Exempel	• `månad(till_intervall('3 månader'))` → 3 • `månad(ålder('2012-01-01','2010-01-01'))` → 4.03333

now

Returnerar aktuellt datum och aktuell tid. Funktionen är statisk och ger konsekventa resultat när den utvärderas. Den tid som returneras är den tid då uttrycket förbereds.

Syntax	now()
Exempel	• `now()` → 2012-07-22T13:24:57

sekund

Extraherar sekunddelen från en datetime eller tid, eller antalet sekunder från ett intervall.

Tidsvariant

Extraherar sekunddelen från en tid eller datetime.

Syntax	sekund(datetime)
Argument	• **datetime** - ett tids- eller datetime-värde
Exempel	• `sekund( to_datetime('2012-07-22 13:24:57') )` → 57

Intervallvariant

Beräknar längden i sekunder för ett intervall.

Syntax	second(interval)
Argument	• **interval** - intervallvärde för att returnera antal sekunder från
Exempel	• `sekund(till_intervall('3 minuter'))` → 180 • `second(age('2012-07-22T00:20:00','2012-07-22T00:00:00'))` → 1200 • `sekund(ålder('2012-01-01','2010-01-01'))` → 63072000

to_date

Konverterar en sträng till ett datumobjekt. En valfri formatsträng kan anges för att analysera strängen; se QDate::fromString eller dokumentationen av funktionen format_date för ytterligare dokumentation om formatet. Som standard används den aktuella QGIS-användarens locale.

Syntax	to_date(string, [format], [language]) [] markerar valfria argument
Argument	• **sträng** - sträng som representerar ett datumvärde • **format** - format som används för att konvertera strängen till ett datum • **language** - språk (gemener, två eller tre bokstäver, ISO 639 språkkod) som används för att konvertera strängen till ett datum. Som standard används den aktuella QGIS-användarens språk.
Exempel	• `to_date('2012-05-04')` → 2012-05-04 • `to_date('June 29, 2019','MMMM d, yyyy')` → 2019-06-29, om aktuell lokal använder namnet 'June' för den sjätte månaden, annars uppstår ett fel • `to_date('29 juni, 2019','d MMMM, åååå','fr')` → 2019-06-29

För vidare läsning: *format_date*

to_datetime

Konverterar en sträng till ett datetime-objekt. En valfri formatsträng kan anges för att analysera strängen; se QDate::fromString, QTime::fromString eller dokumentationen av funktionen format_date för ytterligare dokumentation om formatet. Som standard används den aktuella QGIS-användarens locale.

Syntax	to_datetime(string, [format], [language]) [] markerar valfria argument
Argument	• **sträng** - sträng som representerar ett datetime-värde • **format** - format som används för att konvertera strängen till en datetime • **language** - språk (gemener, två eller tre bokstäver, ISO 639 språkkod) som används för att konvertera strängen till en datetime. Som standard används den aktuella QGIS-användarens språk.
Exempel	• `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, om den aktuella lokala använder namnet 'June' för den sjätte månaden, annars uppstår ett fel • `to_datetime('29 juin, 2019 @ 12:34','d MMMM, yyyy @ HH:mm', 'fr')` → 2019-06-29T12:34

För vidare läsning: *format_date*

to_interval

Konverterar en sträng till en intervalltyp. Kan användas för att ta dagar, timmar, månad etc. från ett datum.

Syntax	to_interval(string)
Argument	• **sträng** - en sträng som representerar ett intervall. Tillåtna format är {n} dagar {n} timmar {n} månader.
Exempel	• `to_interval('1 dag 2 timmar')` → intervall: 1,08333 dagar • `to_interval( '0,5 timmar' )` → intervall: 30 minuter • `to_datetime('2012-05-05 12:00:00') - to_interval('1 dag 2 timmar')` → 2012-05-04T10:00:00

to_time

Konverterar en sträng till ett tidsobjekt. En valfri formatsträng kan anges för att analysera strängen; se QTime::fromString för ytterligare dokumentation om formatet.

Syntax	to_time(string, [format], [language]) [] markerar valfria argument
Argument	• **sträng** - sträng som representerar ett tidsvärde • **format** - format som används för att konvertera strängen till en tid • **language** - språk (gemener, två eller tre bokstäver, ISO 639 språkkod) som används för att konvertera strängen till en tid
Exempel	• `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

För vidare läsning: *format_date*

week

Extraherar veckonumret från ett datum, eller antalet veckor från ett intervall.

Datumvariant

Extraherar veckonumret från ett datum eller en datetime.

Syntax	week(date)
Argument	• **date** - ett datum- eller datetime-värde
Exempel	• `vecka('2012-05-12')` → 19

Intervallvariant

Beräknar längden i veckor för ett intervall.

Syntax	week(interval)
Argument	• **interval** - intervallvärde för att returnera antal månader från
Exempel	• `vecka(till_intervall('3 veckor'))` → 3 • `vecka(ålder('2012-01-01','2010-01-01'))` → 104.285

year

Extraherar årsdelen från ett datum, eller antalet år från ett intervall.

Datumvariant

Extraherar årsdelen från ett datum eller en datetime.

Syntax	year(date)
Argument	• **date** - ett datum- eller datetime-värde
Exempel	• `år('2012-05-12')` → 2012

Intervallvariant

Beräknar längden i år för ett intervall.

Syntax	year(interval)
Argument	• **interval** - intervallvärde för att returnera antal år från
Exempel	• `år(to_interval('3 år'))` → 3 • `år(ålder('2012-01-01','2010-01-01'))` → 1,9986

Några exempel:

Förutom dessa funktioner ger subtraktion av datum, datumtider eller tider med hjälp av operatorn − (minus) ett intervall.

Om du adderar eller subtraherar ett intervall till datum, datetime eller tid med operatörerna + (plus) och − (minus) får du tillbaka en datetime.

- Få antalet dagar tills QGIS 3.0 släpps:

```
to_date('2017-09-29') - to_date(now())
-- Returns <interval: 203 days>
```

- Samma sak med tiden:

```
to_datetime('2017-09-29 12:00:00') - now()
-- Returns <interval: 202.49 days>
```

- Hämta datatiden för 100 dagar från och med nu:

```
now() + to_interval('100 days')
-- Returns <datetime: 2017-06-18 01:00:00>
```

9.2.8 Fält och värden

Innehåller en lista över fält från det aktiva lagret och specialvärden. Fältlistan innehåller de som lagras i datasetet, *virtual* och *auxiliary* samt från *joins*.

Dubbelklicka på ett fältnamn för att lägga till det i ditt uttryck. Du kan också skriva fältnamnet (helst inom dubbla citattecken) eller dess *alias*.

För att hämta fältvärden som ska användas i ett uttryck väljer du lämpligt fält och i den widget som visas väljer du mellan *10 Samples* och *All Unique*. De begärda värdena visas sedan och du kan använda rutan *Search* högst upp i listan för att filtrera resultatet. Exempelvärden kan också nås genom att högerklicka på ett fält.

Om du vill lägga till ett värde i det uttryck du skriver, dubbelklickar du på det i listan. Om värdet är av strängtyp ska det ha ett enkelt citattecken, annars behövs inget citattecken.

NULL

Motsvarar ett NULL-värde.

Syntax	NULL
Exempel	• `NULL` → ett NULL-värde

Observera

Om du vill testa för NULL använder du ett *IS NULL*- eller *IS NOT NULL*-uttryck.

9.2.9 Funktioner för filer och sökvägar

Denna grupp innehåller funktioner som manipulerar fil- och sökvägsnamn.

base_file_name

Returnerar filens basnamn utan katalog- eller filsuffix.

Syntax	base_file_name(path)
Argument	• **path** - en filsökväg eller ett värde för ett kartlager. Om ett värde för ett kartlager anges kommer lagrets filkälla att användas.
Exempel	• `base_file_name('/home/qgis/data/country_boundaries.shp')` → 'country_boundaries'

exif

Hämtar exif-taggvärden från en bildfil.

Syntax	exif(path, [tag]) [] markerar valfria argument
Argument	• **path** - En sökväg till en bildfil eller ett värde för ett kartlager. Om ett värde för ett kartlager anges kommer filkällan för lagret att användas. • **tag** - Den tagg som ska returneras. Om den är tom returneras en karta med alla exif-taggvärden.
Exempel	• `exif('/my/photo.jpg','Exif.Image.Orientation')` → 0

file_exists

Returnerar TRUE om en filsökväg finns.

Syntax	file_exists(path)
Argument	• **path** - en filsökväg eller ett värde för ett kartlager. Om ett värde för ett kartlager anges kommer lagrets filkälla att användas.
Exempel	• `file_exists('/home/qgis/data/country_boundaries.shp')` → TRUE

file_name

Returnerar namnet på en fil (inklusive filtillägget), exklusive katalogen.

Syntax	file_name(path)
Argument	• **path** - en filsökväg eller ett värde för ett kartlager. Om ett värde för ett kartlager anges kommer lagrets filkälla att användas.
Exempel	• `file_name('/home/qgis/data/country_boundaries.shp')` → 'country_boundaries.shp'

file_path

Returnerar katalogkomponenten i en filsökväg. Detta inkluderar inte filnamnet.

Syntax	file_path(path)
Argument	• **path** - en filsökväg eller ett värde för ett kartlager. Om ett värde för ett kartlager anges kommer lagrets filkälla att användas.
Exempel	• `file_path('/home/qgis/data/country_boundaries.shp')` → '/home/qgis/data'

file_size

Returnerar storleken (i byte) på en fil.

Syntax	file_size(path)
Argument	• **path** - en filsökväg eller ett värde för ett kartlager. Om ett värde för ett kartlager anges kommer lagrets filkälla att användas.
Exempel	• `file_size('/home/qgis/data/country_boundaries.geojson')` → 5674

file_suffix

Returnerar filsuffixet (tillägget) från en filsökväg.

Syntax	file_suffix(path)
Argument	• **path** - en filsökväg eller ett värde för ett kartlager. Om ett värde för ett kartlager anges kommer lagrets filkälla att användas.
Exempel	• `file_suffix('/home/qgis/data/country_boundaries.shp')` → 'shp'

is_directory

Returnerar TRUE om en sökväg motsvarar en katalog.

Syntax	is_directory(path)
Argument	• **path** - en filsökväg eller ett värde för ett kartlager. Om ett värde för ett kartlager anges kommer lagrets filkälla att användas.
Exempel	• `is_directory('/home/qgis/data/country_boundaries.shp')` → FALSE • `is_directory('/home/qgis/data/')` → TRUE

is_file

Returnerar TRUE om en sökväg motsvarar en fil.

Syntax	is_file(path)
Argument	• **path** - en filsökväg eller ett värde för ett kartlager. Om ett värde för ett kartlager anges kommer lagrets filkälla att användas.
Exempel	• `is_file('/home/qgis/data/country_boundaries.shp')` → TRUE • `is_file('/home/qgis/data/')` → FALSE

9.2.10 Formulär Funktioner

Den här gruppen innehåller funktioner som uteslutande fungerar i attributformulärskontexten. Till exempel i inställningarna för *fältets widgets*.

current_parent_value

Denna funktion kan endast användas i ett inbäddat formulär och returnerar det aktuella, osparade värdet för ett fält i det överordnade formulär som för närvarande redigeras. Detta kommer att skilja sig från den överordnade funktionens faktiska attributvärden för funktioner som för närvarande redigeras eller som ännu inte har lagts till i ett överordnat lager. När den här funktionen används i ett filteruttryck för en widget med värderelation bör den förpackas i en ”coalesce()” som kan hämta den faktiska överordnade funktionen från lagret när formuläret inte används i ett inbäddat sammanhang.

Syntax	current_parent_value(field_name)
Argument	• **field_name** - ett fältnamn i det aktuella överordnade formuläret
Exempel	• `current_parent_value( 'FIELD_NAME' )` → Det aktuella värdet för fältet ’FIELD_NAME’ i det överordnade formuläret.

current_value

Returnerar det aktuella, osparade värdet för ett fält i formuläret eller tabellraden som för närvarande redigeras. Detta skiljer sig från objektets faktiska attributvärden för objekt som håller på att redigeras eller som ännu inte har lagts till i ett lager.

Syntax	current_value(field_name)
Argument	• **field_name** - ett fältnamn i det aktuella formuläret eller tabellraden
Exempel	• `current_value( 'FIELD_NAME' )` → Det aktuella värdet för fältet ’FIELD_NAME’.

9.2.11 Fuzzy matchningsfunktioner

Denna grupp innehåller funktioner för fuzzy-jämförelser mellan värden.

hamming_distance

Returnerar Hamming-avståndet mellan två strängar. Detta motsvarar antalet tecken vid motsvarande positioner i indatasträngarna där tecknen är olika. Inmatningssträngarna måste vara lika långa och jämförelsen är skiftlägeskänslig.

Syntax	hamming_distance(sträng1, sträng2)
Argument	• **sträng1** - en sträng • **sträng2** - en sträng
Exempel	• `hamming_distance('abc','xec')` → 2 • `hamming_distance('abc','ABc')` → 2 • `hamming_distance(övre('abc'),övre('ABC'))` → 0 • `hamming_distance('abc','abcd')` → NULL

levenshtein

Returnerar Levenshtein-editeringsavståndet mellan två strängar. Detta motsvarar det minsta antal teckenredigeringar (infogningar, borttagningar eller substitutioner) som krävs för att ändra en sträng till en annan.

Levenshtein-avståndet är ett mått på likheten mellan två strängar. Mindre avstånd innebär att strängarna är mer lika, och större avstånd innebär att strängarna är mer olika. Avståndet är skiftlägeskänsligt.

Syntax	levenshtein(string1, string2)
Argument	• **sträng1** - en sträng • **sträng2** - en sträng
Exempel	• `levenshtein('kattungar','vante')` → 2 • `levenshtein('Kitten','kitten')` → 1 • `levenshtein(övre('Kitten'),övre('kitten'))` → 0

longest_common_substring

Returnerar den längsta gemensamma delsträngen mellan två strängar. Denna delsträng är den längsta sträng som är en delsträng av de två indatasträngarna. Till exempel är den längsta gemensamma delsträngen av ”ABABC” och ”BABCA” ”BABC”. Substrängen är skiftlägeskänslig.

Syntax	longest_common_substring(string1, string2)
Argument	• **sträng1** - en sträng • **sträng2** - en sträng
Exempel	• `längsta_vanliga_substräng('ABABC','BABCA')` → 'BABC' • `längsta_vanliga_substräng('abcDeF','abcdef')` → 'abc' • `längsta_vanliga_substräng(övre('abcDeF'),övre('abcdex'))` → 'ABCDE'

soundex

Returnerar Soundex-representationen av en sträng. Soundex är en fonetisk matchningsalgoritm, så strängar med liknande ljud bör representeras av samma Soundex-kod.

Syntax	soundex(string)
Argument	• **sträng** - en sträng
Exempel	• `soundex('robert')` → 'R163' • `soundex('rupert')` → 'R163' • `soundex('rubin')` → 'R150'

9.2.12 Allmänna funktioner

Denna grupp innehåller allmänt sammansatta funktioner.

env

Hämtar en miljövariabel och returnerar dess innehåll som en sträng. Om variabeln inte hittas returneras NULL. Detta är praktiskt för att injicera systemspecifik konfiguration som enhetsbeteckningar eller sökvägsprefix. Definitionen av miljövariabler beror på operativsystemet, kontrollera med din systemadministratör eller dokumentationen för operativsystemet hur detta kan ställas in.

Syntax	env(name)
Argument	• **name** - Namnet på den miljövariabel som ska hämtas.
Exempel	• `env( 'LANG' )` → 'en_US.UTF-8' • `env( 'MY_OWN_PREFIX_VAR' )` → 'Z:' • `env( 'I_DO_NOT_EXIST' )` → NULL

eval

Utvärderar ett uttryck som skickas i form av en sträng. Användbart för att expandera dynamiska parametrar som skickas som kontextvariabler eller fält.

Syntax	eval(expression)
Argument	• **expression** - en uttrycksträng
Exempel	• ``eval("nice") `` → 'nice' • `eval(@expression_var)` → [vad resultatet av utvärderingen av @expression_var än kan bli…]

eval_template

Utvärderar en mall som skickas i form av en sträng. Användbar för att expandera dynamiska parametrar som skickas som kontextvariabler eller fält.

Syntax	eval_template(template)
Argument	• **mall** - en mallsträng
Exempel	• `eval_template('QGIS [% upper(\'rocks\') %]')` → QGIS ROCKS

is_layer_visible

Returnerar TRUE om ett angivet lager är synligt.

Syntax	is_layer_visible(layer)
Argument	• **lager** - en sträng som representerar antingen ett lagernamn eller ett lager-ID
Exempel	• `is_layer_visible('baseraster')` → TRUE

mime_type

Returnerar mimetypen för binärdata.

Syntax	mime_type(bytes)
Argument	• **bytes** - den binära datan
Exempel	• `mime_type('<html><body></body></html>')` → text/html • `mime_type(from_base64('R0lGODlhAQABAAAAACH5BAEKAAEALAAAAAABAAEAAAIAOw==` → image/gif

var

Returnerar det värde som lagrats i en angiven variabel.

Syntax	var(name)
Argument	• **namn** - ett variabelnamn
Exempel	• `var('qgis_version')` → '2.12'

Ytterligare läsning: Lista över standard *variabler*

with_variable

Denna funktion ställer in en variabel för varje uttryckskod som kommer att anges som 3:e argument. Detta är endast användbart för komplicerade uttryck, där samma beräknade värde måste användas på olika ställen.

Syntax	with_variable(name, value, expression)
Argument	• **namn** - namnet på den variabel som ska ställas in • **värde** - det värde som ska ställas in • **expression** - det uttryck som variabeln ska vara tillgänglig för
Exempel	• `med_variabel('min_summa', 1 + 2 + 3, @min_summa * 2 + @min_summa * 5)` → 42

9.2.13 Geometri Funktioner

Den här gruppen innehåller funktioner som arbetar med geometriska objekt (t.ex. buffert, transform, $area).

affine_transform

Returnerar geometrin efter en affin transformation. Beräkningarna görs i geometrins spatiala referenssystem. Operationerna utförs i ordningen skala, rotation, translation. Om det finns en Z- eller M-offset men koordinaten inte finns i geometrin kommer den att läggas till.

Syntax	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]) [] markerar valfria argument
Argument	• **geometri** - en geometri • **delta_x** - översättning av x-axeln • **delta_y** - förskjutning av y-axeln • **rotation_z** - rotation runt z-axeln i grader moturs • **scale_x** - Skalfaktor för x-axeln • **scale_y** - Skalfaktor för y-axeln • **delta_z** - översättning av z-axeln • **delta_m** - förskjutning av m-axeln • **scale_z** - Skalfaktor för z-axeln • **scale_m** - Skalfaktor för m-axeln
Exempel	• `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, 0, 1, 1, 5, 0))` → 'PointZ (3 1 5)'

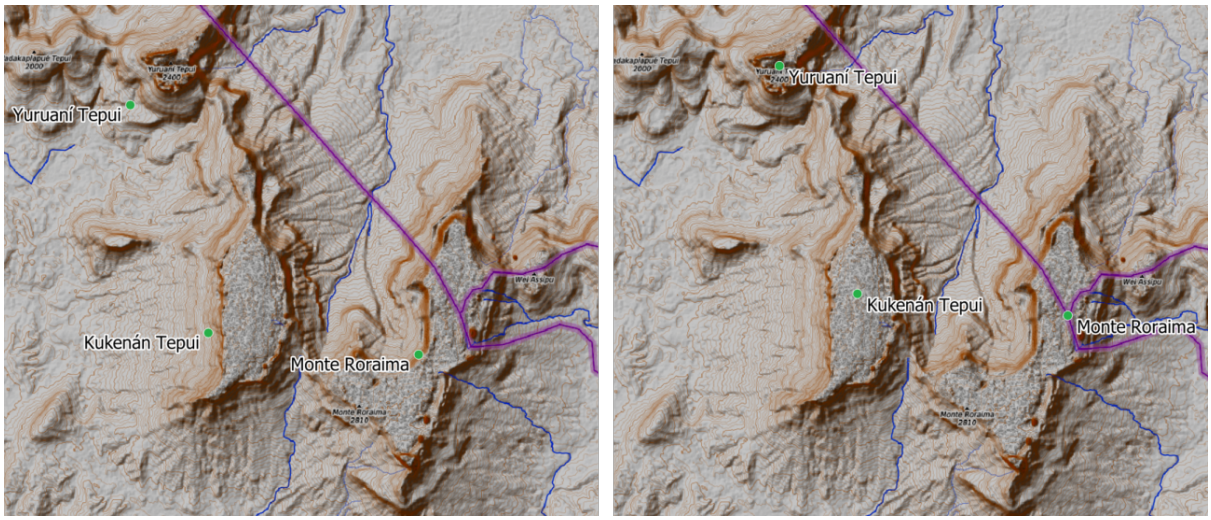


Fig. 9.4: Vektorpunktlager (gröna prickar) före (vänster) och efter (höger) en affin transformation (översättning).

angle_at_vertex

Returnerar bisektorvinkeln (medelvinkeln) till geometrin för ett angivet vertex på en linjestringgeometri. Vinklar anges i grader medurs från norr.

Syntax	angle_at_vertex(geometry, vertex)
Argument	• **geometri** - en geometri med linjestring • **vertex** - vertexindex, med början från 0; om värdet är negativt kommer det valda vertexindexet att vara dess totala antal minus det absoluta värdet
Exempel	• `vinkel_vid_vertex(geometri:=geom_from_wkt('LineString(0 0, 10 0, 10 10)'),vertex:=1)` → 45,0

apply_dash_pattern

Använder ett streckmönster på en geometri och returnerar en MultiLineString-geometri som är indatageometrin som streckas längs varje linje/ring med det angivna mönstret.

Syntax	apply_dash_pattern(geometry, pattern, [start_rule=no_rule], [end_rule=no_rule], [adjustment=both], [pattern_offset=0]) [] markerar valfria argument
Argument	• **geometry** - en geometri (accepterar (multi)linjestrings eller (multi)polygoner). • **pattern** - streckmönster, som en array av siffror som representerar längden på streck och mellanrum. Måste innehålla ett jämnt antal element. • **start_rule** - valfri regel för begränsning av mönstrets början. Giltiga värden är ”no_rule”, ”full_dash”, ”half_dash”, ”full_gap”, ”half_gap”. • **end_rule** - valfri regel för att begränsa slutet på mönstret. Giltiga värden är ”no_rule”, ”full_dash”, ”half_dash”, ”full_gap”, ”half_gap”. • **adjustment** - valfri regel för att ange vilken del av mönstret som ska justeras för att passa de önskade mönsterreglerna. Giltiga värden är ”both”, ”dash”, ”gap”. • **pattern_offset** - Valfritt avstånd som anger ett specifikt avstånd längs mönstret att börja på.
Exempel	• `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

Returnerar arean för den aktuella funktionen. Den area som beräknas med den här funktionen respekterar både ellipsoidinställningen och inställningarna för areaenhet i det aktuella projektet. Om t.ex. en ellipsoid har ställts in för projektet kommer den beräknade arean att vara ellipsoidisk, och om ingen ellipsoid har ställts in kommer den beräknade arean att vara planimetrisk.

Syntax	$area
Exempel	• `$area` → 42

area

Returnerar arean för ett geometriskt polygonobjekt. Beräkningar är alltid planimetriska i det spatiala referenssystemet (SRS) för denna geometri, och enheterna för den returnerade ytan kommer att matcha enheterna för SRS. Detta skiljer sig från de beräkningar som utförs av funktionen $area, som utför ellipsoidiska beräkningar baserat på projektets inställningar för ellipsoid och areaenheter.

Syntax	area(geometry)
Argument	• **geometry** - geometriobjekt för polygoner
Exempel	• `area(geom_from_wkt('POLYGON((0 0, 4 0, 4 2, 0 2, 0 0))'))` → 8,0

azimuth

Returnerar den nordbaserade azimuten som vinkeln i radianer mätt medurs från vertikalen på punkt_a till punkt_b.

Syntax	azimuth(point_a, point_b)
Argument	• **point_a** - punktgeometri • **point_b** - punktgeometri
Exempel	• `grader( azimut( make_point(25, 45), make_point(75, 100) ) )` → 42.273689 • `grader( azimut( make_point(75, 100), make_point(25,45) ) )` → 222.273689

bearing

Returnerar den nordbaserade bäringen som vinkeln i radianer mätt medurs på ellipsoiden från vertikalen på punkt_a till punkt_b.

Syntax	bearing(point_a, point_b, [source_crs], [ellipsoid]) [] markerar valfria argument
Argument	• **point_a** - punktgeometri • **point_b** - punktgeometri • **source_crs** - en valfri sträng som representerar punkternas CRS-källa. Som standard används det aktuella lagrets CRS. • **ellipsoid** - en valfri sträng som representerar akronymen eller auktoriteten:ID (t.ex. ”EPSG:7030”) för den ellipsoid som bäringen ska mätas på. Som standard används det aktuella projektets ellipsoidinställning.
Exempel	• `grader( bäring( make_point(16198544, -4534850), make_point(18736872, -1877769), 'EPSG:3857', 'EPSG:7030') )` → 49,980071 • `grader( bäring( make_point(18736872, -1877769), make_point(16198544, -4534850), 'EPSG:3857', 'WGS84') )` → 219.282386

boundary

Returnerar stängningen av geometrins kombinatoriska gräns (dvs. geometrins topologiska gräns). Till exempel kommer en polygongeometri att ha en gräns som består av linjestringarna för varje ring i polygonen. Vissa geometrityper har inte någon definierad gräns, t.ex. punkter eller geometrisamlingar, och returnerar NULL.

Syntax	boundary(geometry)
Argument	• **geometri** - en geometri
Exempel	• `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))’

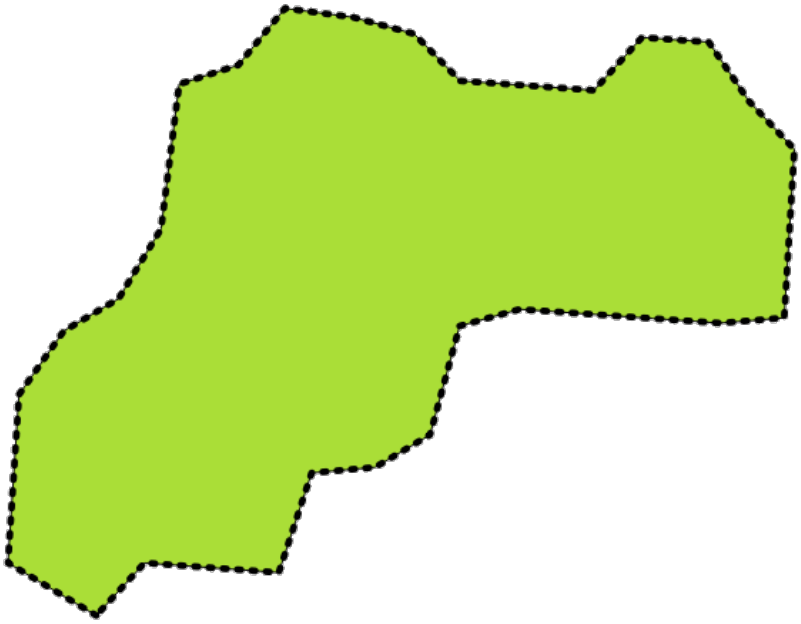


Fig. 9.5: Gräns (svart streckad linje) för källans polygonlager

Ytterligare läsning: *Begränsning*-algoritm

bounds

Returnerar en geometri som representerar begränsningsboxen för en indatageometri. Beräkningar görs i det spatiala referenssystemet för denna geometri.

Syntax	bounds(geometry)
Argument	• **geometri** - en geometri
Exempel	• `bounds(@geometry)` → avgränsningsbox för den aktuella funktionens geometri • `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))'

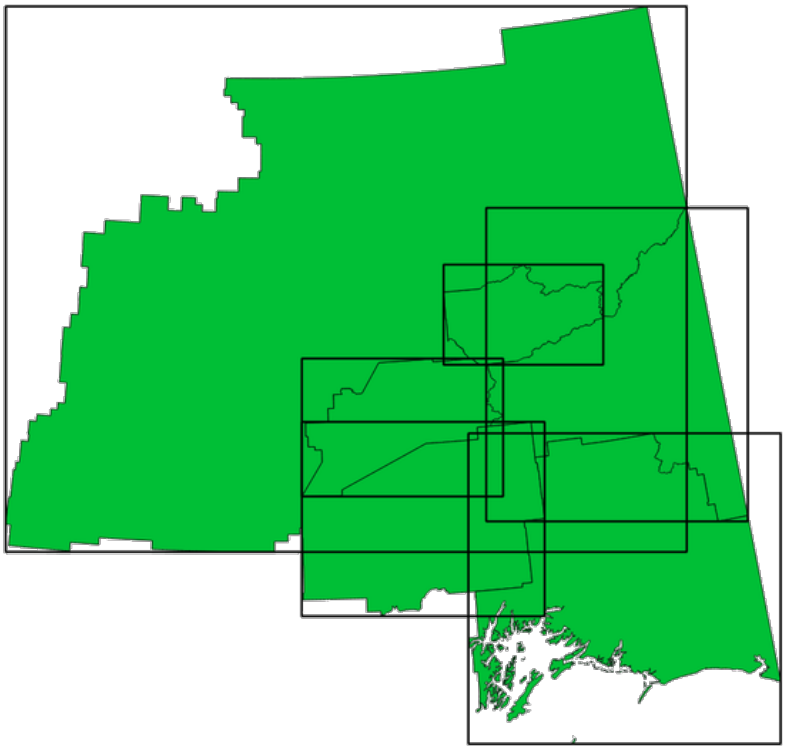


Fig. 9.6: Svarta linjer representerar de avgränsande boxarna för varje polygonfunktion

Ytterligare läsning: *Begränsande boxar* algoritm

bounds_height

Returnerar höjden på den avgränsande boxen för en geometri. Beräkningar görs i geometrins spatiala referenssystem.

Syntax	bounds_height(geometry)
Argument	• **geometri** - en geometri
Exempel	• `bounds_height(@geometry)` → höjden på begränsningsrutan för den aktuella funktionens geometri • `bounds_height(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))'))` → 1

bounds_width

Returnerar bredden på den avgränsande boxen för en geometri. Beräkningar görs i geometrins spatiala referenssystem.

Syntax	bounds_width(geometry)
Argument	• **geometri** - en geometri
Exempel	• `bounds_width(@geometry)` → bredden på begränsningsrutan för den aktuella funktionens geometri • `bounds_width(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))'))` → 2

buffer

Returnerar en geometri som representerar alla punkter vars avstånd från denna geometri är mindre än eller lika med distance. Beräkningar görs i det spatiala referenssystemet för denna geometri.

Syntax	buffert(geometri, avstånd, [segment=8], [cap='round'], [join='round'], [miter_limit=2]) [] markerar valfria argument
Argument	• **geometri** - en geometri • **distance** - buffertavstånd i lagerenheter • **segments** - antal segment som ska användas för att representera en kvartscirkel när en rund join-stil används. Ett större antal resulterar i en jämnare buffert med fler noder. • **cap** - typ av ändlock för buffert. Giltiga värden är "rund", "platt" eller "fyrkantig • **join** - fogningsstil för buffert. Giltiga värden är "round", "bevel" eller "miter". • **miter_limit** - gräns för miteravstånd, för användning när fogningsstilen är inställd på "miter
Exempel	• `buffer(@geometry, 10.5)` → polygon av den aktuella funktionens geometri buffrad med 10,5 enheter

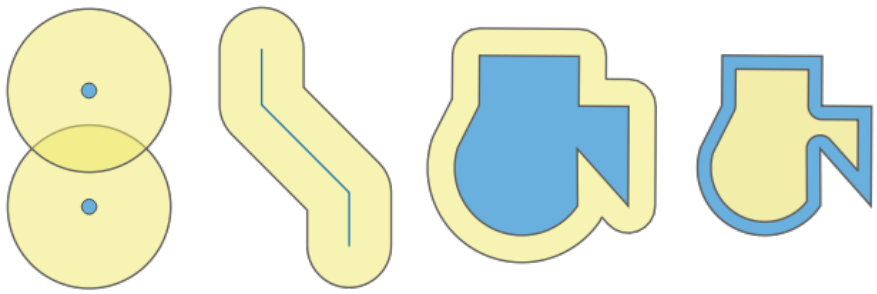


Fig. 9.7: Buffert (i gult) för punkter, linjer, polygoner med positiv buffert och polygoner med negativ buffert

Ytterligare läsning: *Buffra*-algoritm

buffer_by_m

Skapar en buffert längs en linjegeometri där buffertdiametern varierar beroende på m-värdena i linjens hörnpunkter.

Syntax	buffert_by_m(geometri, [segment=8]) [] markerar valfria argument
Argument	• **geometry** - geometri för inmatning. Måste vara en (multi)linjegeometri med m värden. • **segments** - antal segment för att approximera kvartscirkelkurvor i bufferten.
Exempel	• `buffer_by_m(geometry:=geom_from_wkt('LINESTRINGM(1 2 0,5, 4 2 0,2)'),segments:=8)` → En buffert med variabel bredd som börjar med en diameter på 0,5 och slutar med en diameter på 0,2 längs linjestringens geometri.

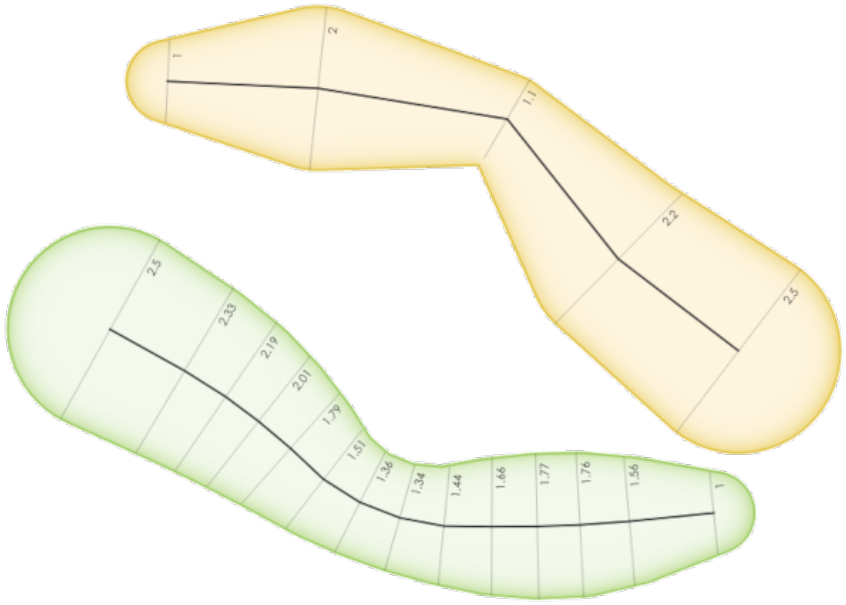


Fig. 9.8: Buffring av linjefunktioner med hjälp av m-värdet på topparna

Ytterligare läsning: *Buffert med variabel bredd (av M-värde)* algoritm

centroid

Returnerar den geometriska mittpunkten för en geometri.

Syntax	centroid(geometry)
Argument	• **geometri** - en geometri
Exempel	• `centroid(@geometri)` → en punktgeometri

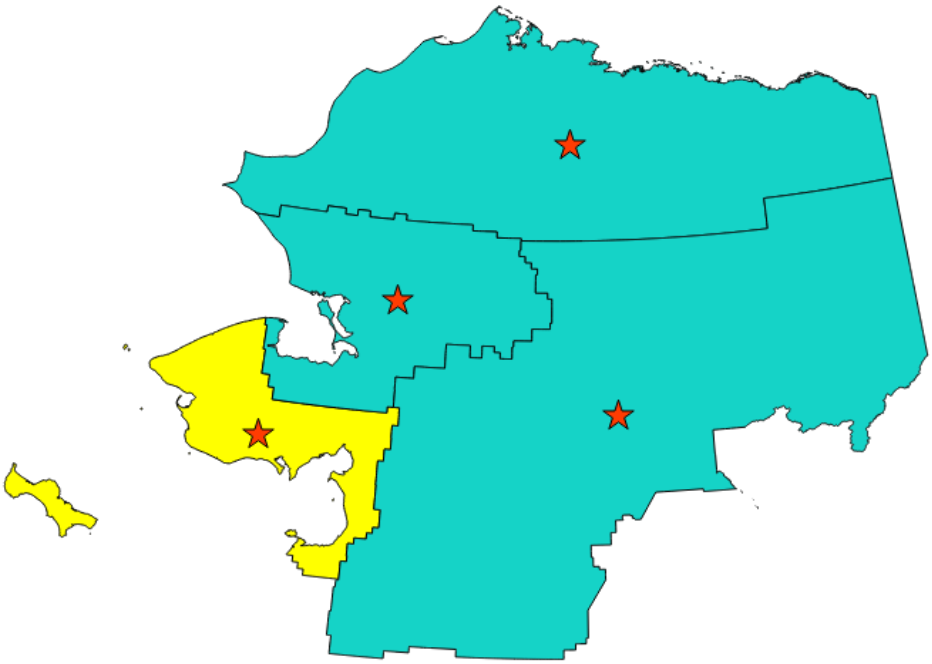


Fig. 9.9: De röda stjärnorna representerar centroiderna för funktionerna i inmatningslagret.

Ytterligare läsning: *Centroider* algoritm

close_line

Returnerar en sluten linjestring av den inmatade linjesträngen genom att lägga till den första punkten i slutet av linjen, om den inte redan är sluten. Om geometrin inte är en radsträng eller en sträng med flera rader blir resultatet NULL.

Syntax	close_line(geometry)
Argument	• **geometri** - en geometri för en linjesträng
Exempel	• `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

Returnerar den punkt på geometri1 som ligger närmast geometri2.

Syntax	closest_point(geometry1, geometry2)
Argument	• **geometri1** - geometri för att hitta närmaste punkt på • **geometry2** - geometri för att hitta närmaste punkt till
Exempel	• `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

Samlar en uppsättning geometrier till ett geometriobjekt med flera delar.

Lista över argument variant

Geometridelar anges som separata argument till funktionen.

Syntax	collect_geometries(geometri1, geometri2, …)
Argument	• **geometri** - en geometri
Exempel	• `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

Geometridelar specificeras som en array av geometridelar.

Syntax	collect_geometries(array)
Argument	• **array** - array av geometriobjekt
Exempel	• `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))'

Ytterligare läsning: *Samla in geometrier* algoritm

combine

Returnerar kombinationen av två geometrier.

Syntax	combine(geometry1, geometry2)
Argument	• **geometri1** - en geometri • **geometry2** - en geometri
Exempel	• `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

Returnerar en eventuellt konkav polygon som innehåller alla punkter i geometrin

Syntax	concave_hull(geometry, target_percent, [allow_holes=False]) [] markerar valfria argument
Argument	• **geometri** - en geometri • **target_percent** - den procentandel av området för den konvexa skålen som lösningen försöker närma sig. En target_percent på 1 ger samma resultat som det konvexa skrovet. En target_percent mellan 0 och 0,99 ger ett resultat som bör ha en mindre yta än den konvexa skalan. • **allow_holes** - valfritt argument som anger om hål ska tillåtas i utdatageometrin. Standardvärde är FALSE, inställt på TRUE för att tillåta hål i utdatageometrin.
Exempel	• `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))"

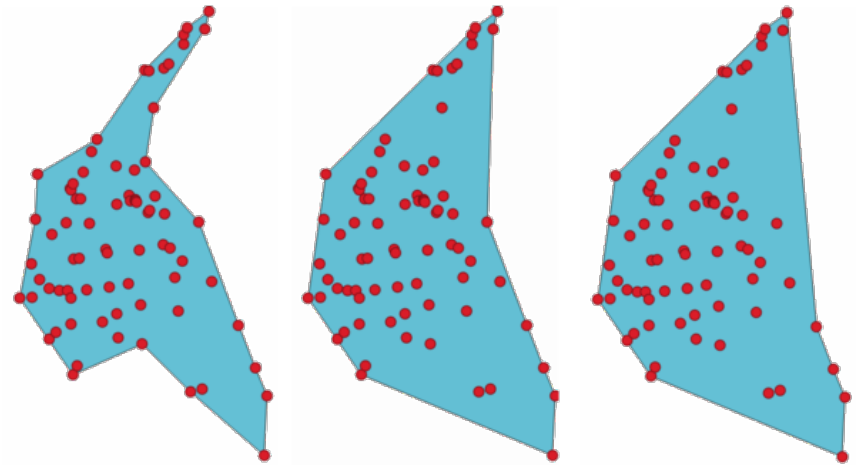


Fig. 9.10: Konkava skrov med ökande parameter för target_percent

Ytterligare läsning: *convex_hull*, *Konkavt skrov* algoritm

contains

Testar om en geometri innehåller en annan. Returnerar TRUE om och endast om inga punkter i geometri2 ligger i geometri1:s utsida och minst en punkt i geometri2:s insida ligger i geometri1:s insida.

Syntax	contains(geometry1, geometry2)
Argument	• **geometri1** - en geometri • **geometry2** - en geometri
Exempel	• `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

Ytterligare läsning: *overlay_contains*

convex_hull

Returnerar det konvexa skrovet för en geometri. Den representerar den minsta konvexa geometrin som omsluter alla geometrier i uppsättningen.

Syntax	convex_hull(geometry)
Argument	• **geometri** - en geometri
Exempel	• `geom_to_wkt( convex_hull( geom_from_wkt( 'LINESTRING(3 3, 4 4, 4 10)' ) ) )` → ”POLYGON((3 3, 4 10, 4 4, 3 3))

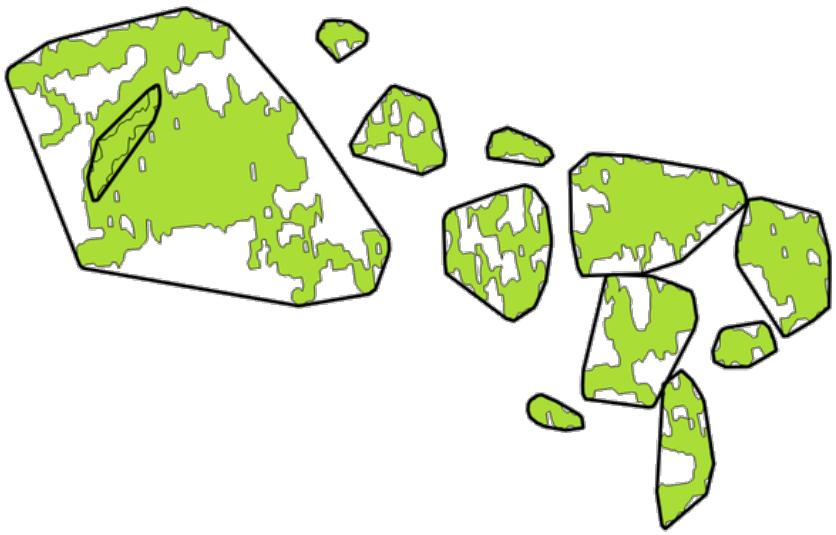


Fig. 9.11: Svarta linjer identifierar det konvexa skrovet för varje funktion

Ytterligare läsning: *concave_hull*, *Konvex skrovform* algoritm

crosses

Testar om en geometri korsar en annan. Returnerar TRUE om de angivna geometrierna har vissa, men inte alla, inre punkter gemensamt.

Syntax	crosses(geometry1, geometry2)
Argument	• **geometri1** - en geometri • **geometry2** - en geometri
Exempel	• `kors( geom_from_wkt( 'LINESTRING(3 5, 4 4, 5 3)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → SANT • `kors( geom_from_wkt( 'POINT(4 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → FALSE

Ytterligare läsning: *overlay_crosses*

densify_by_count

Tar en polygon- eller linjelagergeometri och genererar en ny där geometrierna har ett större antal hörn än den ursprungliga.

Syntax	densify_by_count(geometry, vertices)
Argument	• **geometry** - en geometri (accepterar (multi)linjestrings eller (multi)polygoner). • **vertices** - antal vertices som ska läggas till (per segment)
Exempel	• `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)

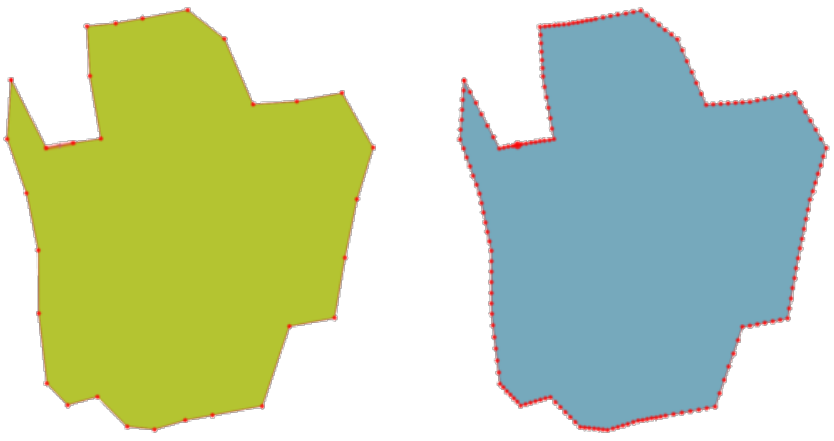


Fig. 9.12: Röda punkter visar topparna före och efter förtätningen

Ytterligare läsning: *Förtätning efter antal* algoritm

densify_by_distance

Tar en polygon- eller linjelagergeometri och genererar en ny där geometrierna förtätas genom att lägga till ytterligare hörn på kanter som har ett maximalt avstånd på det angivna intervallavståndet.

Syntax	densify_by_distance(geometry, distance)
Argument	• **geometry** - en geometri (accepterar (multi)linjestrings eller (multi)polygoner). • **distance** - maximalt intervallavstånd mellan topparna i utdatageometrin
Exempel	• `geom_to_wkt(densify_by_distance(geom_from_wkt('LINESTRING(1 1, 10 1)'), 4))` → LineString (1 1, 4 1, 7 1, 10 1)

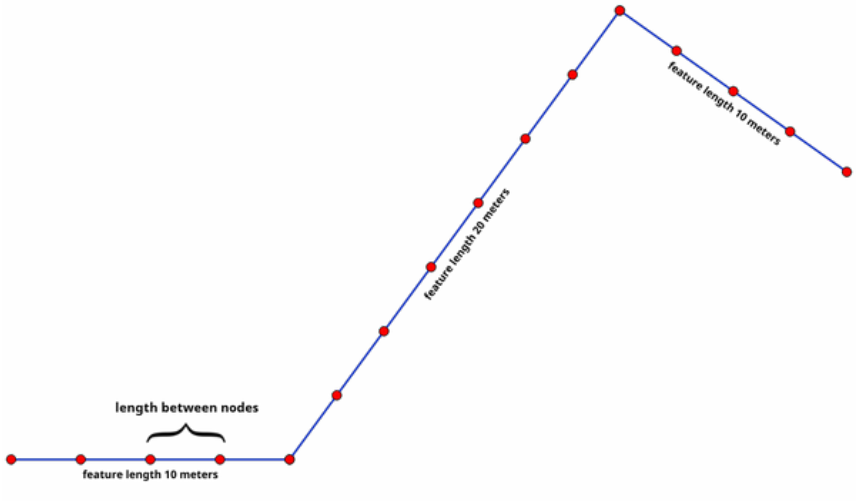


Fig. 9.13: Förtäta geometrin vid ett givet intervall

Ytterligare läsning: *Förtätning med intervall* algoritm

difference

Returnerar en geometri som representerar den del av geometri1 som inte korsar geometri2.

Syntax	difference(geometry1, geometry2)
Argument	• **geometri1** - en geometri • **geometry2** - en geometri
Exempel	• `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)

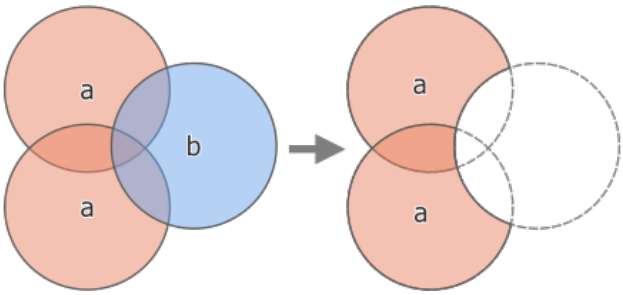


Fig. 9.14: Skillnadsoperation mellan ett inmatningslager med två funktioner ”a” och ett överlagringslager med en funktion ”b” (vänster) - resulterar i ett nytt lager med de modifierade funktionerna från ”a” (höger)

Ytterligare läsning: *Skillnad* algoritm

disjoint

Testar om geometrier inte korsar varandra spatialt. Returnerar TRUE om geometrierna inte delar något utrymme tillsammans.

Syntax	disjoint(geometry1, geometry2)
Argument	• **geometri1** - en geometri • **geometry2** - en geometri
Exempel	• `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

Ytterligare läsning: *overlay_disjoint*

distance

Returnerar det minsta avståndet (baserat på spatial referens) mellan två geometrier i projicerade enheter.

Syntax	distance(geometry1, geometry2)
Argument	• **geometri1** - en geometri • **geometry2** - en geometri
Exempel	• `avstånd( geom_from_wkt( 'POINT(4 4)' ), geom_from_wkt( 'POINT(4 8)' ) )` → 4

distance_to_vertex

Returnerar avståndet längs geometrin till en angiven vertex.

Syntax	distance_to_vertex(geometry, vertex)
Argument	• **geometri** - en geometri med linjestring • **vertex** - vertexindex, med början från 0; om värdet är negativt kommer det valda vertexindexet att vara dess totala antal minus det absoluta värdet
Exempel	• `avstånd_till_vertex(geometri:=geom_from_wkt('LineString(0 0, 10 0, 10 10)'),vertex:=1)` → 10.0

end_point

Returnerar den sista noden från en geometri.

Syntax	end_point(geometry)
Argument	• **geometry** - geometriobjekt
Exempel	• `geom_to_wkt(end_point(geom_from_wkt('LINESTRING(4 0, 4 2, 0 2)')))` → ”Punkt (0 2)

Fig. 9.15: Slutpunkt för en linjefunktion

Ytterligare läsning: *start_point*, *Extrahera specifika toppar* algoritm

exif_geotag

Skapar en punktgeometri från exif-geotaggarna i en bildfil.

Syntax	exif_geotag(path)
Argument	• **path** - En sökväg till en bildfil eller ett värde för ett kartlager. Om ett värde för ett kartlager anges kommer filkällan för lagret att användas.
Exempel	• `geom_to_wkt(exif_geotag('/my/photo.jpg'))` → ”Punkt (2 4)

extend

Förlänger början och slutet av en linjestringgeometri med en angiven mängd. Linjer förlängs med hjälp av bäringen för det första och sista segmentet i linjen. För en multilinestring förlängs alla delar. Avstånden är i det spatiala referenssystemet för denna geometri.

Syntax	extend(geometry, start_distance, end_distance)
Argument	• **geometri** - en (multi)linjestringsgeometri • **start_distance** - avstånd för att förlänga linjens start • **end_distance** - avstånd för att förlänga linjens slut.
Exempel	• `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))'

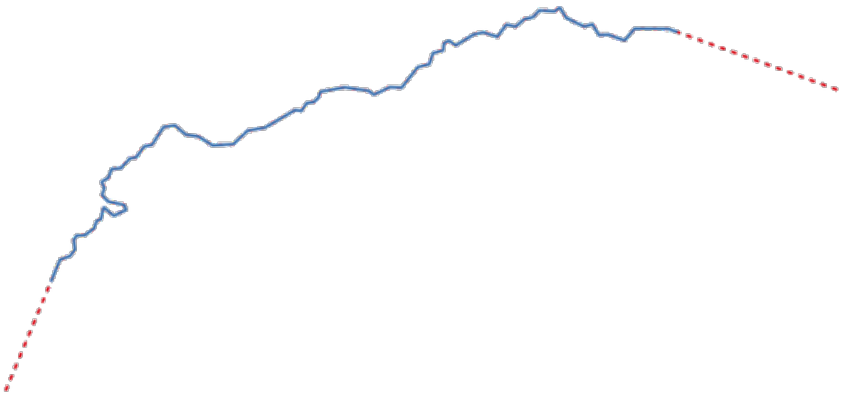


Fig. 9.16: De röda strecken representerar den initiala och slutliga förlängningen av det ursprungliga lagret

Ytterligare läsning: *Förlänga linjer* algoritm

exterior_ring

Returnerar en radsträng som representerar den yttre ringen i en polygongeometri. Om geometrin inte är en polygon kommer resultatet att vara NULL.

Syntax	exterior_ring(geometry)
Argument	• **geometri** - en polygongeometri
Exempel	• `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)'

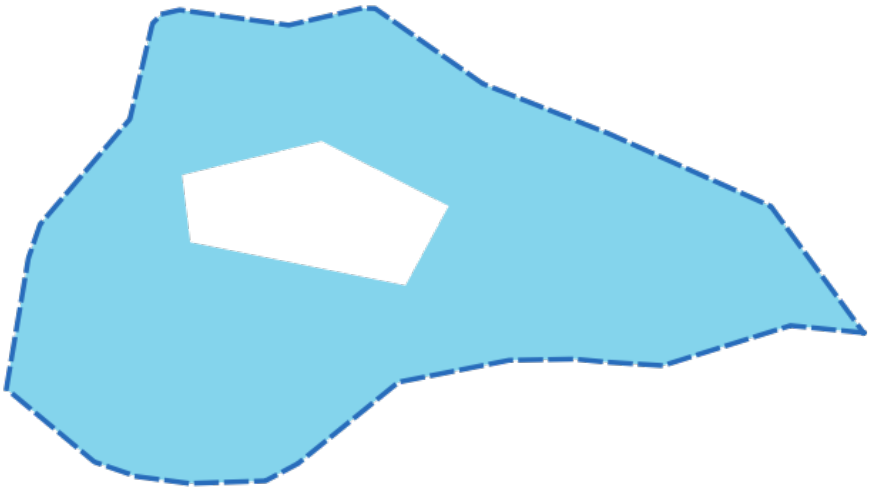


Fig. 9.17: Den streckade linjen representerar polygonens yttre ring

extrude

Returnerar en extruderad version av den inmatade (Multi-)Curve- eller (Multi-)Linestring-geometrin med en förlängning som anges av x och y.

Syntax	extrude(geometry, x, y)
Argument	• **geometri** - en kurva eller en linjestrings geometri • **x** - x-förlängning, numeriskt värde • **y** - y-förlängning, numeriskt värde
Exempel	• `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)))"

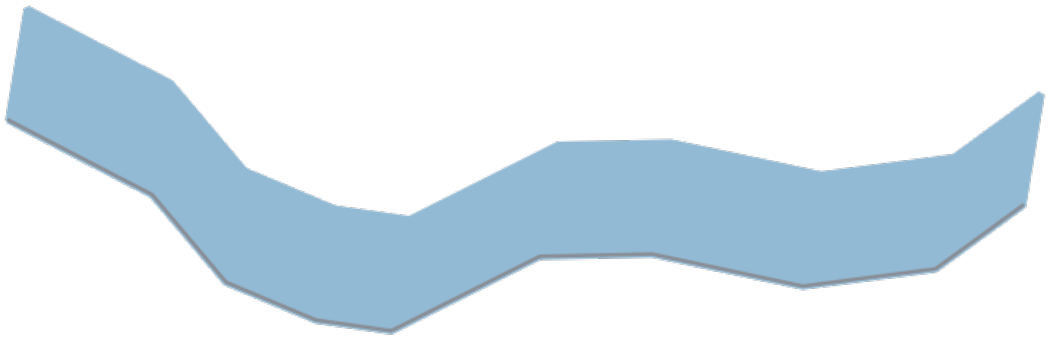


Fig. 9.18: Generera en polygon genom att extrudera en linje med förskjutning i x- och y-riktningarna

flip_coordinates

Returnerar en kopia av geometrin med x- och y-koordinaterna ombytta. Användbar för att reparera geometrier som har fått sina latitud- och longitudvärden omkastade.

Syntax	flip_coordinates(geometry)
Argument	• **geometri** - en geometri
Exempel	• `geom_to_wkt(flip_coordinates(make_point(1, 2)))` → 'Punkt (2 1)' • `geom_to_wkt(flip_coordinates(geom_from_wkt('LineString(0 2, 1 0, 1 6)')))` → 'LineString (2 0, 0 1, 6 1)'

Ytterligare läsning: *Byt X- och Y-koordinater* algoritm

force_polygon_ccw

Tvingar en geometri att följa konventionen att yttre ringar är motsols och inre ringar är medsols.

Syntax	force_polygon_ccw(geometry)
Argument	• **geometri** - en geometri. Alla icke-polygongeometrier returneras oförändrade.
Exempel	• `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))'

Ytterligare läsning: *force_polygon_cw*, *force_rhr*

force_polygon_cw

Tvingar en geometri att följa konventionen där yttre ringar är medurs och inre ringar är moturs.

Syntax	force_polygon_cw(geometry)
Argument	• **geometri** - en geometri. Alla icke-polygongeometrier returneras oförändrade.
Exempel	• `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))'

Ytterligare läsning: *force_polygon_ccw*, *force_rhr*

force_rhr

Tvingar en geometri att följa högerhandsregeln, enligt vilken det område som avgränsas av en polygon ligger till höger om gränsen. I synnerhet är den yttre ringen orienterad i medurs riktning och de inre ringarna i moturs riktning. På grund av inkonsekvensen i definitionen av Right-Hand-Rule i vissa sammanhang rekommenderas att man istället använder den explicita funktionen force_polygon_cw.

Syntax	force_rhr(geometry)
Argument	• **geometri** - en geometri. Alla icke-polygongeometrier returneras oförändrade.
Exempel	• `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))'

Ytterligare läsning: *Kraft högerhands-regel*-algoritm, *force_polygon_ccw*, *force_polygon_cw*

geom_from_gml

Returnerar en geometri från en GML-representation av geometri.

Syntax	geom_from_gml(gml)
Argument	• **gml** - GML-representation av en geometri som en sträng
Exempel	• `geom_from_gml('<gml:LineString srsName="EPSG:4326"><gml:coordinates>4,4 5,5 6,6</gml:coordinates></gml:LineString>')` → ett linjegeometriobjekt

geom_from_wkb

Returnerar en geometri som skapats från en WKB-representation (Well-Known Binary).

Syntax	geom_from_wkb(binary)
Argument	• **binary** - WKB-representation (Well-Known Binary) av en geometri (som en binär blob)
Exempel	• `geom_från_wkb( geom_till_wkb( make_point(4,5) ) )` → ett punktgeometriobjekt

geom_from_wkt

Returnerar en geometri som skapats från en WKT-representation (Well-Known Text).

Syntax	geom_from_wkt(text)
Argument	• **text** - WKT-representation (Well-Known Text) av en geometri
Exempel	• `geom_from_wkt( 'POINT(4 5)' )` → ett geometriobjekt

geom_to_wkb

Returnerar WKB-representationen (Well-Known Binary) av en geometri

Syntax	geom_to_wkb(geometry)
Argument	• **geometri** - en geometri
Exempel	• `geom_to_wkb( @geometry )` → binär blob som innehåller ett geometriobjekt

geom_to_wkt

Returnerar WKT-representationen (Well-Known Text) av geometrin utan SRID-metadata.

Syntax	geom_to_wkt(geometri, [precision=8]) [] markerar valfria argument
Argument	• **geometri** - en geometri • **precision** - numerisk precision
Exempel	• `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

Returnerar geometrin för den aktuella funktionen. Kan användas för bearbetning med andra funktioner. **VARNING: Denna funktion är föråldrad. Vi rekommenderar att du använder ersättningsvariabeln @geometry istället.**

Syntax	$geometry
Exempel	• `geom_to_wkt( $geometri )` → 'POINT(6 50)'

geometry

Returnerar en objekts geometri.

Syntax	geometry(feature)
Argument	• **feature** - ett funktionsobjekt
Exempel	• `geometry( @feature )` → geometrin för den aktuella funktionen. Föredrar att använda @geometry. • `geom_to_wkt( geometry( get_feature_by_id( 'streets', 1 ) ) )` → geometrin i WKT för objektet med id 1 i lagret ”streets”, t.ex. ”POINT(6 50) • `intersects( @geometry, geometry( get_feature( 'streets', 'name', 'Main St.' ) ) )` → TRUE om den aktuella funktionen spatialt skär den namngivna funktionen ”Main St.” i lagret ”streets

geometry_n

Returnerar en specifik geometri från en geometrisamling, eller NULL om indatageometrin inte är en samling. Returnerar även en del från en geometri med flera delar.

Syntax	geometry_n(geometry, index)
Argument	• **geometry** - geometrisamling • **index** - index för den geometri som ska returneras, där 1 är den första geometrin i samlingen
Exempel	• `geom_to_wkt(geometry_n(geom_from_wkt('GEOMETRYCOLLECTION(POINT(0 1), POINT(0 0), POINT(1 0), POINT(1 1))'),3))` → ’Punkt (1 0)’

geometry_type

Returnerar ett strängvärde som beskriver typen av geometri (punkt, linje eller polygon)

Syntax	geometry_type(geometry)
Argument	• **geometri** - en geometri
Exempel	• `geometry_type( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → ’Linje’ • `geometry_type( geom_from_wkt( 'MULTILINESTRING((2 5, 3 6, 4 8), (1 1, 0 0))') )` → ’Linje’ • `geometry_type( geom_from_wkt( 'POINT(2 5)') )` → ’Punkt’ • `geometry_type( geom_from_wkt( 'POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1))') )` → ’Polygon’

hausdorff_distance

Returnerar Hausdorff-avståndet mellan två geometrier. Detta är i princip ett mått på hur lika eller olika 2 geometrier är, där ett lägre avstånd indikerar mer likartade geometrier.

Funktionen kan utföras med ett valfritt densify-fraktionsargument. Om inget anges används en approximation av standard Hausdorff-avstånd. Denna approximation är exakt eller tillräckligt nära för en stor undergrupp av användbara fall. Exempel på dessa är:

- beräknar avståndet mellan Linestrings som är ungefär parallella med varandra och ungefär lika långa. Detta inträffar i matchande linjära nätverk.
- Testar likheten mellan geometrierna.

Om den approximativa standardmetoden som tillhandahålls av denna metod är otillräcklig, ange det valfria argumentet densify fraction. Om du anger detta argument utförs en segmentförtätning innan det diskreta Hausdorff-avståndet beräknas. Parametern anger den fraktion som varje segment ska förtätas med. Varje segment delas upp i ett antal lika långa undersegment, vars fraktion av den totala längden är närmast den angivna fraktionen. Om parametern för densify fraction minskas kommer det returnerade avståndet att närma sig det verkliga Hausdorff-avståndet för geometrierna.

Syntax	hausdorff_distance(geometry1, geometry2, [densify_fraction]) [] markerar valfria argument
Argument	• **geometri1** - en geometri • **geometry2** - en geometri • **densify_fraction** - mängd förtätningsfraktion
Exempel	• `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

Returnerar lutningen mätt från zenit (0) till nadir (180) på punkt_a till punkt_b.

Syntax	inclination(point_a, point_b)
Argument	• **point_a** - punktgeometri • **point_b** - punktgeometri
Exempel	• `inklination( make_point( 5, 10, 0 ), make_point( 5, 10, 5 ) )` → 0,0 • `inklination( make_point( 5, 10, 0 ), make_point( 5, 10, 0 ) )` → 90,0 • `inklination( make_point( 5, 10, 0 ), make_point( 50, 100, 0 ) )` → 90,0 • `inklination( make_point( 5, 10, 0 ), make_point( 5, 10, -5 ) )` → 180,0

interior_ring_n

Returnerar en specifik inre ring från en polygongeometri, eller NULL om geometrin inte är en polygon.

Syntax	interior_ring_n(geometry, index)
Argument	• **geometri** - polygongeometri • **index** - index för den interiör som ska returneras, där 1 är den första interiörringen
Exempel	• `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))'

intersection

Returnerar en geometri som representerar den delade delen av två geometrier.

Syntax	intersection(geometry1, geometry2)
Argument	• **geometri1** - en geometri • **geometry2** - en geometri
Exempel	• `geom_to_wkt( intersektion( 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( intersektion( 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)

Ytterligare läsning: *Korsningar* algoritm

intersects

Testar om en geometri skär en annan. Returnerar TRUE om geometrierna korsar varandra spatialt (delar någon del av utrymmet) och false om de inte gör det.

Syntax	intersects(geometry1, geometry2)
Argument	• **geometri1** - en geometri • **geometry2** - en geometri
Exempel	• `korsar( geom_from_wkt( 'POINT(4 4)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → SANT • `intersects( geom_from_wkt( 'POINT(4 5)' ), geom_from_wkt( 'POINT(5 5)' ) )` → FALSE

Ytterligare läsning: *overlay_intersects*

intersects_bbox

Testar om en geometris avgränsningsbox överlappar en annan geometris avgränsningsbox. Returnerar TRUE om geometrierna spatialt korsar den definierade begränsningsboxen och false om de inte gör det.

Syntax	intersects_bbox(geometry1, geometry2)
Argument	• **geometri1** - en geometri • **geometry2** - en geometri
Exempel	• `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

Returnerar TRUE om en linjestring är sluten (start- och slutpunkter sammanfaller), eller false om en linjestring inte är sluten. Om geometrin inte är en linjestring blir resultatet NULL.

Syntax	is_closed(geometry)
Argument	• **geometri** - en geometri för en linjesträng
Exempel	• `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

Returnerar TRUE om en geometri är tom (utan koordinater), false om geometrin inte är tom och NULL om det inte finns någon geometri. Se även is_empty_or_null.

Syntax	is_empty(geometry)
Argument	• **geometri** - en geometri
Exempel	• `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

Ytterligare läsning: *is_empty_or_null*

is_empty_or_null

Returnerar TRUE om en geometri är NULL eller tom (utan koordinater) eller false annars. Denna funktion är som uttrycket '@geometri ÄR NULL eller is_empty(@geometri)'

Syntax	is_empty_or_null(geometry)
Argument	• **geometri** - en geometri
Exempel	• `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

Ytterligare läsning: *is_empty*, *NULL*

is_multipart

Returnerar TRUE om geometrin är av typen Multi.

Syntax	is_multipart(geometry)
Argument	• **geometri** - en geometri
Exempel	• `is_multipart(geom_from_wkt('MULTIPOINT ((0 0),(1 1),(2 2))'))` → TRUE • `is_multipart(geom_from_wkt('POINT (0 0)'))` → FALSE

is_valid

Returnerar TRUE om en geometri är giltig; om den är välformad i 2D enligt OGC:s regler.

Syntax	is_valid(geometry)
Argument	• **geometri** - en geometri
Exempel	• `is_valid(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2, 0 0)'))` → TRUE • `is_valid(geom_from_wkt('LINESTRING(0 0)'))` → FALSE

Ytterligare läsning: *make_valid*, *Kontrollera giltighet* algoritm

$length

Returnerar längden på en linjestring. Om du behöver längden på en polygons kant använder du $perimeter istället. Den längd som beräknas med den här funktionen respekterar både ellipsoidinställningen och inställningarna för avståndsenheter i det aktuella projektet. Om till exempel en ellipsoid har ställts in för projektet kommer den beräknade längden att vara ellipsoidisk, och om ingen ellipsoid har ställts in kommer den beräknade längden att vara planimetrisk.

Syntax	$length
Exempel	• `$length` → 42.4711

length

Returnerar antalet tecken i en sträng eller längden på en geometrisk linjestring.

String variant

Returnerar antalet tecken i en sträng.

Syntax	length(string)
Argument	• **sträng** - sträng att räkna längden på
Exempel	• `längd('hello')` → 5

Geometrivariant

Beräknar längden på ett geometriskt linjeobjekt. Beräkningar är alltid planimetriska i det spatiala referenssystemet (SRS) för denna geometri, och enheterna för den returnerade längden kommer att matcha enheterna för SRS. Detta skiljer sig från de beräkningar som utförs av funktionen $length, som utför ellipsoidiska beräkningar baserat på projektets inställningar för ellipsoid och avståndsenheter.

Syntax	length(geometry)
Argument	• **geometry** - objekt för linjegeometri
Exempel	• `längd(geom_from_wkt('LINESTRING(0 0, 4 0)'))` → 4,0

Ytterligare läsning: *straight_distance_2d*

length3D

Beräknar 3D-längden för ett geometriskt linjeobjekt. Om geometrin inte är ett 3D-linjeobjekt returneras dess 2D-längd. Beräkningarna är alltid planimetriska i det spatiala referenssystemet (SRS) för denna geometri, och enheterna för den returnerade längden kommer att överensstämma med enheterna för SRS. Detta skiljer sig från de beräkningar som utförs av funktionen $length, som utför ellipsoidiska beräkningar baserat på projektets inställningar för ellipsoid och avståndsenheter.

Syntax	length3D(geometry)
Argument	• **geometry** - objekt för linjegeometri
Exempel	• `längd3D(geom_from_wkt('LINESTRINGZ(0 0 0, 3 0 4)'))` → 5,0

line_interpolate_angle

Returnerar vinkeln som är parallell med geometrin på ett angivet avstånd längs en linjestringgeometri. Vinklarna anges i grader medurs från norr.

Syntax	line_interpolate_angle(geometry, distance)
Argument	• **geometri** - en geometri med linjestring • **distance** - avstånd längs linjen för att interpolera vinkeln vid
Exempel	• `line_interpolate_angle(geometry:=geom_from_wkt('LineString(0 0, 10 0)'),distance:=5)` → 90,0

line_interpolate_point

Returnerar den punkt som interpolerats med ett angivet avstånd längs en linjestringsgeometri.

Syntax	line_interpolate_point(geometry, distance)
Argument	• **geometri** - en geometri med linjestring • **distance** - avstånd längs linjen som ska interpoleras
Exempel	• `geom_to_wkt(line_interpolate_point(geometry:=geom_from_wkt('LineString(0 0, 8 0)'), distance:=5))` → 'Punkt (5 0)' • `geom_to_wkt(line_interpolate_point(geometry:=geom_from_wkt('LineString(0 0, 1 1, 2 0)'), distance:=2,1))` → 'Punkt (1,48492424 0,51507576)' • `geom_to_wkt(line_interpolate_point(geometry:=geom_from_wkt('LineString(0 0, 1 0)'), distance:=2))` → NULL

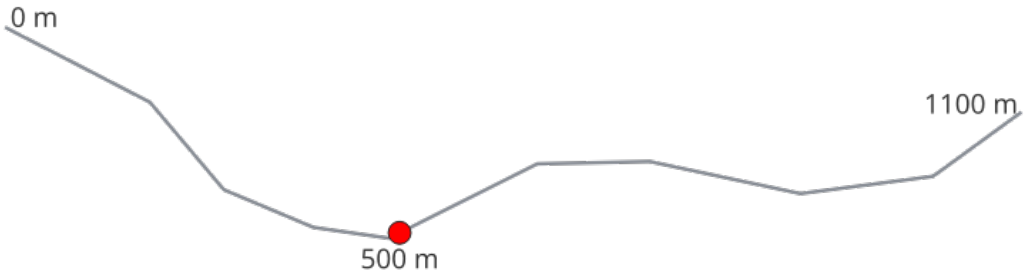


Fig. 9.19: Interpolerad punkt 500 meter från linjens början

Ytterligare läsning: *Interpolera punkt på linje*-algoritm

line_interpolate_point_by_m

Returnerar den punkt som interpoleras av ett matchande M-värde längs en linjestringsgeometri.

Syntax	line_interpolate_point_by_m(geometry, m, [use_3d_distance=false]) [] markerar valfria argument
Argument	• **geometri** - en geometri med linjestring • **m** - ett M-värde • **use_3d_distance** - styr om 2D- eller 3D-avstånd mellan hörnpunkter ska användas vid interpolering (detta alternativ beaktas endast för linjer med z-värden)
Exempel	• `geom_to_wkt(line_interpolate_point_by_m(geom_from_wkt('LineStringM(0 0 0, 10 10 10)'), m:=5))` → 'Punkt (5 5)'

line_locate_m

Returnerar avståndet längs en linjestring som motsvarar det första matchande interpolerade M-värdet.

Syntax	line_locate_m(geometry, m, [use_3d_distance=false]) [] markerar valfria argument
Argument	• **geometri** - en geometri med linjestring • **m** - ett M-värde • **use_3d_distance** - styr om 2D- eller 3D-avstånd mellan hörnpunkter ska användas vid interpolering (detta alternativ beaktas endast för linjer med z-värden)
Exempel	• `line_locate_m(geometry:=geom_from_wkt('LineStringM(0 0 0, 10 10 10)'),m:=5)` → 7,07106

line_locate_point

Returnerar avståndet längs en linjestring som motsvarar den närmaste positionen linjestringen kommer till en angiven punktgeometri.

Syntax	line_locate_point(geometry, point)
Argument	• **geometri** - en geometri med linjestring • **point** - punktgeometri för att lokalisera närmaste position på linjesträckan till
Exempel	• `line_locate_point(geometry:=geom_from_wkt('LineString(0 0, 10 0)'),point:=geom_from_wkt('Point(5 0)'))` → 5.0

line_merge

Returnerar en LineString- eller MultiLineString-geometri, där alla anslutna LineStrings från indatageometrin har sammanfogats till en enda linjestring. Denna funktion returnerar NULL om en geometri som inte är en LineString/MultiLineString skickas till den.

Syntax	line_merge(geometry)
Argument	• **geometri** - en LineString/MultiLineString-geometri
Exempel	• `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

Returnerar den del av en linjes (eller kurvas) geometri som ligger mellan de angivna start- och slutavstånden (mätt från linjens början). Z- och M-värden interpoleras linjärt från befintliga värden.

Syntax	line_substring(geometry, start_distance, end_distance)
Argument	• **geometri** - en geometri för en linjestring eller kurva • **start_distance** - avstånd till början av understrängen • **end_distance** - avstånd till slutet av delsträngen
Exempel	• `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)'

Fig. 9.20: Substringslinje med startavståndet inställt på 0 meter och slutavståndet på 250 meter.

Ytterligare läsning: *Linje delsträng* algoritm

m

Returnerar m-värdet (mått) för en punktgeometri.

Syntax	m(geometry)
Argument	• **geometri** - en punktgeometri
Exempel	• `m( geom_from_wkt( 'POINTM(2 5 4)' ) )` → 4

m_at

Hämtar en m-koordinat för geometrin, eller NULL om geometrin inte har något m-värde.

Syntax	m_at(geometry, vertex)
Argument	• **geometry** - geometriobjekt • **vertex** - index för geometrins vertex (index börjar på 0; negativa värden gäller från det sista indexet, med början på -1)
Exempel	• `m_at(geom_from_wkt('LineStringZM(0 0 0 0, 10 10 0 0 5, 10 10 0 0)'), 1)` → 5

m_max

Returnerar det maximala m-värdet (mått) för en geometri.

Syntax	m_max(geometry)
Argument	• **geometri** - en geometri som innehåller m värden
Exempel	• `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

Returnerar det minsta m-värdet (mått) för en geometri.

Syntax	m_min(geometry)
Argument	• **geometri** - en geometri som innehåller m värden
Exempel	• `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

Returnerar vinkeln på den långa axeln (medurs, i grader från norr) för den orienterade minimala begränsande rektangeln, som helt täcker geometrin.

Syntax	main_angle(geometry)
Argument	• **geometri** - en geometri
Exempel	• `main_angle(geom_from_wkt('Polygon ((321577 129614, 321581 129618, 321585 129615, 321581 129610, 321577 129614))'))` → 38,66

make_circle

Skapar en cirkulär polygon.

Syntax	make_circle(centrum, radie, [segment=36]) [] markerar valfria argument
Argument	• **center** - cirkelns mittpunkt • **radius** - cirkelns radie • **segments** - valfritt argument för polygonsegmentering. Som standard är detta värde 36
Exempel	• `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

Skapar en elliptisk polygon.

Syntax	make_ellipse(center, semi_major_axis, semi_minor_axis, azimuth, [segment=36]) [] markerar valfria argument
Argument	• **center** - ellipsens mittpunkt • **semi_major_axis** - ellipsens semi-major-axel • **semi_minor_axis** - ellipsens halvminoraxel • **azimut** - ellipsens orientering • **segments** - valfritt argument för polygonsegmentering. Som standard är detta värde 36
Exempel	• `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

Skapar en linjegeometri från en serie punktgeometrier.

Lista över argument variant

Linjepunkterna anges som separata argument till funktionen.

Syntax	make_line(punkt1, punkt2, …)
Argument	• **point** - en punktgeometri (eller en array av punkter)
Exempel	• `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

Linjens hörnpunkter anges som en array av punkter.

Syntax	make_line(array)
Argument	• **array** - array med punkter
Exempel	• `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

Skapar en punktgeometri från ett x- och y-värde (och valfritt z- och m-värde).

Syntax	make_point(x, y, [z], [m]) [] markerar valfria argument
Argument	• **x** - x-koordinat för punkt • **y** - punktens y-koordinat • **z** - valfri z-koordinat för punkten • **m** - valfritt m-värde för punkten
Exempel	• `geom_to_wkt(make_point(2,4))` → 'Punkt (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

Skapar en punktgeometri från en x-, y-koordinat och ett m-värde.

Syntax	make_point_m(x, y, m)
Argument	• **x** - x-koordinat för punkt • **y** - punktens y-koordinat • **m** - m-värde för punkt
Exempel	• `geom_to_wkt(make_point_m(2,4,6))` → 'PointM (2 4 6)'

make_polygon

Skapar en polygongeometri från en yttre ring och en valfri serie geometrier för inre ringar.

Syntax	make_polygon(outerRing, [innerRing1], [innerRing2], …) [] markerar valfria argument
Argument	• **outerRing** - geometri med sluten linje för polygonens yttre ring • **innerRing** - valfri geometri med sluten linje för innerringen
Exempel	• `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

Skapar en rektangel från 3 punkter.

Syntax	make_rectangle_3points(point1, point2, point3, [option=0]) [] markerar valfria argument
Argument	• **point1** - Första punkten. • **point2** - Andra punkten. • **point3** - Tredje punkten. • **option** - Ett valfritt argument för att konstruera rektangeln. Som standard är detta värde 0. Värdet kan vara 0 (avstånd) eller 1 (projicerad). Alternativ avstånd: Andra avståndet är lika med avståndet mellan 2:a och 3:e punkten. Alternativ projicerad: Andra avståndet är lika med avståndet för den vinkelräta projektionen av den 3:e punkten på segmentet eller dess förlängning.
Exempel	• `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

Skapar en regelbunden polygon.

Syntax	make_regular_polygon(centrum, radie, antal_sidor, [cirkel=0]) [] markerar valfria argument
Argument	• **center** - centrum av den regelbundna polygonen • **radius** - andra punkten. Den första om den reguljära polygonen är inskriven. Mittpunkten på den första sidan om den regelbundna polygonen är omskriven. • **number_sides** - Antal sidor/kanter på den regelbundna polygonen • **circle** - Valfritt argument för att konstruera den regelbundna polygonen. Som standard är detta värde 0. Värdet kan vara 0 (inskriven) eller 1 (omskriven)
Exempel	• `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

Skapar en kvadrat från en diagonal.

Syntax	make_square(point1, point2)
Argument	• **point1** - Första punkten på diagonalen • **point2** - diagonalens sista punkt
Exempel	• `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

Skapar en triangelpolygon.

Syntax	make_triangle(point1, point2, point3)
Argument	• **punkt1** - triangelns första punkt • **punkt2** - triangelns andra punkt • **punkt3** - triangelns tredje punkt
Exempel	• `geom_to_wkt(make_triangle(make_point(0,0), make_point(5, 5), make_point(0,10)))` → 'Triangel ((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

Returnerar en giltig geometri eller en tom geometri om geometrin inte kunde göras giltig.

Syntax	make_valid(geometry, [method=structure], [keep_collapsed=false]) [] markerar valfria argument
Argument	• **geometri** - en geometri • **method** - reparationsalgoritm. Kan vara antingen ”structure” eller ”linework”. Alternativet ”linework” kombinerar alla ringar till en uppsättning nodade linjer och extraherar sedan giltiga polygoner från detta linjesystem. Metoden ”structure” gör först alla ringar giltiga och slår sedan samman skal och subtraherar hål från skal för att generera ett giltigt resultat. Antar att hål och skal är korrekt kategoriserade. • **keep_collapsed** - om värdet är true behålls komponenter som har kollapsat till en lägre dimensionalitet. Till exempel en ring som kollapsar till en linje, eller en linje som kollapsar till en punkt.
Exempel	• `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)'))` → En tom geometri

Ytterligare läsning: *is_valid*, *Fixa geometrier* algoritm

minimal_circle

Returnerar den minimala omslutande cirkeln för en geometri. Den representerar den minsta cirkeln som omsluter alla geometrier i uppsättningen.

Syntax	minimal_cirkel(geometri, [segment=36]) [] markerar valfria argument
Argument	• **geometri** - en geometri • **segments** - valfritt argument för polygonsegmentering. Som standard är detta värde 36
Exempel	• `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))'

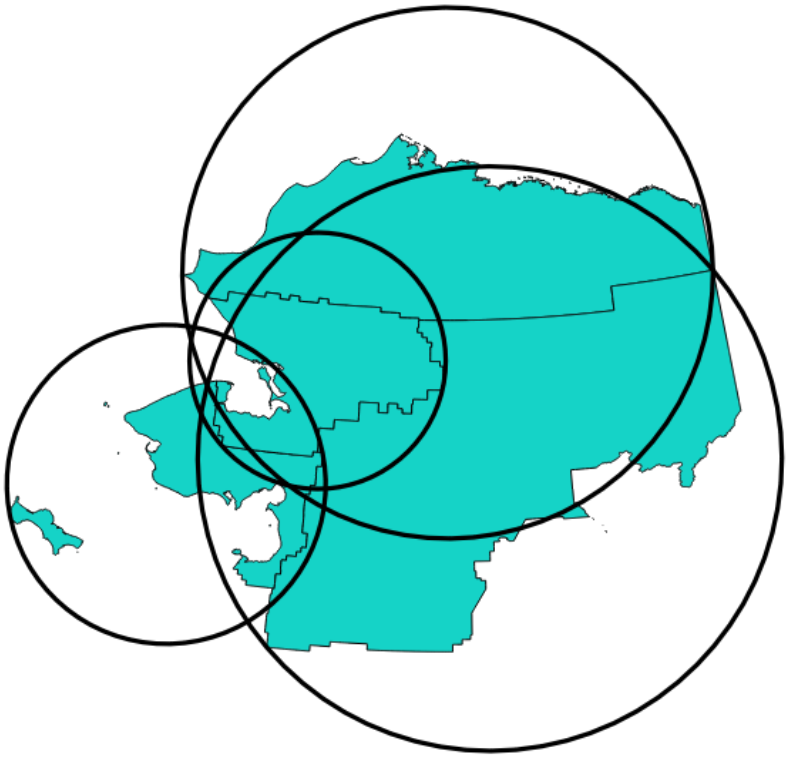


Fig. 9.21: Minimal omslutande cirkel för varje funktion

Ytterligare läsning: *Minsta omslutande cirklar* algoritm

nodes_to_points

Returnerar en multipunktsgeometri som består av varje nod i indatageometrin.

Syntax	nodes_to_points(geometry, [ignore_closing_nodes=false]) [] markerar valfria argument
Argument	• **geometry** - geometriobjekt • **ignore_closing_nodes** - valfritt argument som anger om duplicerade noder som stänger ringar av linjer eller polygoner ska inkluderas. Standardvärdet är false, inställt på true för att undvika att inkludera dessa duplicerade noder i utdatasamlingen.
Exempel	• `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))"

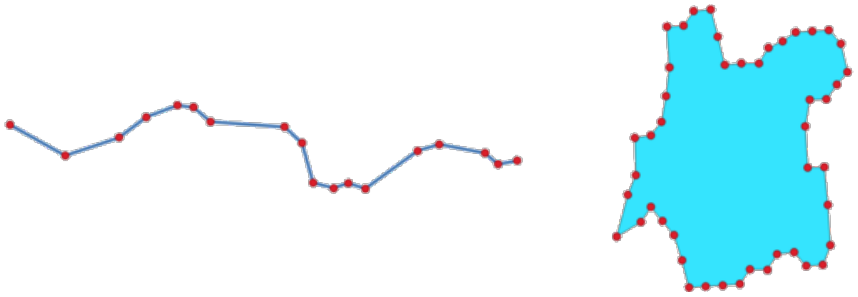


Fig. 9.22: Flerpunktsegenskaper extraherade från toppar

Ytterligare läsning: *Extrahera toppar* algoritm

num_geometries

Returnerar antalet geometrier i en geometrisamling, eller antalet delar i en geometri med flera delar. Funktionen returnerar NULL om indatageometrin inte är en samling.

Syntax	num_geometries(geometry)
Argument	• **geometry** - geometrisamling eller geometri med flera delar
Exempel	• `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

Returnerar antalet inre ringar i en polygon eller geometrisamling, eller NULL om indatageometrin inte är en polygon eller samling.

Syntax	num_interior_rings(geometry)
Argument	• **geometri** - inmatningsgeometri
Exempel	• `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

Returnerar antalet hörn i en geometri.

Syntax	num_points(geometry)
Argument	• **geometri** - en geometri
Exempel	• `num_points(@geometry)` → antal hörn i den aktuella funktionens geometri

num_rings

Returnerar antalet ringar (inklusive yttre ringar) i en polygon eller geometrisamling, eller NULL om indatageometrin inte är en polygon eller samling.

Syntax	num_rings(geometry)
Argument	• **geometri** - inmatningsgeometri
Exempel	• `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

Returnerar en geometri som bildas genom att en linjestringgeometri förskjuts åt sidan. Avstånden är i det spatiala referenssystemet för denna geometri.

Syntax	offset_curve(geometri, avstånd, [segment=8], [join=1], [miter_limit=2.0]) [] markerar valfria argument
Argument	• **geometri** - en (multi)linjestringsgeometri • **distance** - offsetavstånd. Positiva värden buffras till vänster om raderna, negativa värden till höger • **segments** - antal segment som ska användas för att representera en kvartscirkel när en rund anslutningsstil används. Ett större antal resulterar i en jämnare linje med fler noder. • **join** - fogningsstil för hörn, där 1 = rund, 2 = gering och 3 = fasning • **miter_limit** - begränsning av mitterförhållandet som används för mycket skarpa hörn (endast vid användning av mitterförband)
Exempel	• `offset_curve(@geometry, 10.5)` → linjen förskjuten åt vänster med 10,5 enheter • `offset_curve(@geometry, -10.5)` → linjen förskjuten åt höger med 10,5 enheter • `offset_curve(@geometry, 10.5, segments:=16, join:=1)` → linjen förskjuts åt vänster med 10,5 enheter, med fler segment för att få en jämnare kurva • `offset_curve(@geometry, 10.5, join:=3)` → linje förskjuten åt vänster med 10,5 enheter, med hjälp av en avfasad skarv

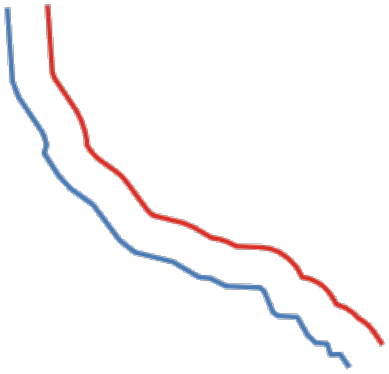


Fig. 9.23: I blått källagret, i rött det förskjutna lagret

Ytterligare läsning: *Offset-linjer* algoritm

order_parts

Ordnar delarna i en MultiGeometry efter ett givet kriterium

Syntax	order_parts(geometry, orderby, [ascending=true]) [] markerar valfria argument
Argument	• **geometri** - en geometri av flera typer • **orderby** - en uttrycksträng som definierar ordningskriterierna • **ascending** - boolean, True för stigande, False för fallande
Exempel	• `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

Returnerar en geometri som representerar den minimala orienterade begränsningsboxen för en indatageometri.

Syntax	oriented_bbox(geometry)
Argument	• **geometri** - en geometri
Exempel	• `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))

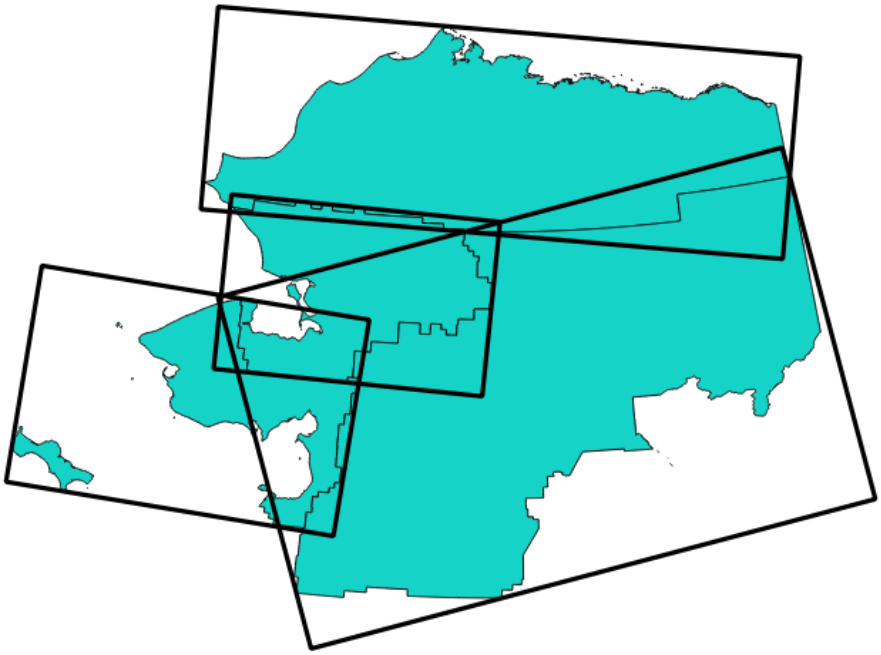


Fig. 9.24: Orienterad minsta avgränsande box

Ytterligare läsning: *Orienterad minsta avgränsande box*-algoritm

overlaps

Testar om en geometri överlappar en annan. Returnerar TRUE om geometrierna delar utrymme, har samma dimension, men inte är helt inneslutna av varandra.

Syntax	overlaps(geometry1, geometry2)
Argument	• **geometri1** - en geometri • **geometry2** - en geometri
Exempel	• `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

Returnerar om den aktuella funktionen spatialt innehåller minst en funktion från ett mållager, eller en array med uttrycksbaserade resultat för funktionerna i mållagret som ingår i den aktuella funktionen.

Läs mer om det underliggande GEOS ”Contains”-predikatet, som beskrivs i PostGIS ST_Contains-funktion.

Syntax	overlay_contains(layer, [expression], [filter], [limit], [cache=false]) [] markerar valfria argument
Argument	• **lager** - det lager vars överlagring kontrolleras • **expression** - ett valfritt uttryck som ska utvärderas på funktionerna från mållagret. Om det inte anges kommer funktionen bara att returnera ett booleanskt värde som anger om det finns minst en matchning. • **filter** - ett valfritt uttryck för att filtrera de målfunktioner som ska kontrolleras. Om det inte anges kommer alla funktioner att kontrolleras. • **limit** - ett valfritt heltal för att begränsa antalet matchande funktioner. Om inget anges kommer alla matchande funktioner att returneras. • **cache** - sätt detta till true för att bygga ett lokalt spatialt index (för det mesta är detta oönskat, såvida du inte arbetar med en särskilt långsam dataleverantör)
Exempel	• `overlay_contains('regions')` → TRUE om den aktuella funktionen spatialt innehåller en region • `overlay_contains('regions', filter:= population > 10000)` → TRUE om den aktuella funktionen spatialt innehåller en region med en population som är större än 10000 • `overlay_contains('regions', name)` → en array med namn för de regioner som ingår i den aktuella funktionen • `array_to_string(overlay_contains('regions', name))` → en sträng som en kommaseparerad lista med namn för de regioner som ingår i den aktuella funktionen • `array_sort(overlay_contains(layer:='regions', expression:="name", filter:= population > 10000))` → en ordnad array med namn, för de regioner som ingår i den aktuella funktionen och med en population större än 10000 • `overlay_contains(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → en array av geometrier (i WKT), för upp till två regioner som ingår i den aktuella funktionen

Ytterligare läsning: *contains*, *array manipulation*, *Välj efter plats* algoritm

overlay_crosses

Returnerar om den aktuella funktionen spatialt korsar minst en funktion från ett mållager, eller en uppsättning uttrycksbaserade resultat för de funktioner i mållagret som korsas av den aktuella funktionen.

Läs mer om det underliggande GEOS ”Crosses”-predikatet, som beskrivs i PostGIS ST_Crosses-funktionen.

Syntax	overlay_crosses(layer, [expression], [filter], [limit], [cache=false]) [] markerar valfria argument
Argument	• **lager** - det lager vars överlagring kontrolleras • **expression** - ett valfritt uttryck som ska utvärderas på funktionerna från mållagret. Om det inte anges kommer funktionen bara att returnera ett booleanskt värde som anger om det finns minst en matchning. • **filter** - ett valfritt uttryck för att filtrera de målfunktioner som ska kontrolleras. Om det inte anges kommer alla funktioner att kontrolleras. • **limit** - ett valfritt heltal för att begränsa antalet matchande funktioner. Om inget anges kommer alla matchande funktioner att returneras. • **cache** - sätt detta till true för att bygga ett lokalt spatialt index (för det mesta är detta oönskat, såvida du inte arbetar med en särskilt långsam dataleverantör)
Exempel	• `overlay_crosses('regions')` → TRUE om den aktuella funktionen spatialt korsar en region • `overlay_crosses('regions', filter:= population > 10000)` → TRUE om den aktuella funktionen spatialt korsar en region med en population som är större än 10000 • `overlay_crosses('regions', name)` → en array med namn för de regioner som korsas av den aktuella funktionen • `array_to_string(overlay_crosses('regions', name))` → en sträng som en kommaseparerad lista med namn för de regioner som korsas av den aktuella funktionen • `array_sort(overlay_crosses(layer:='regions', expression:="name", filter:= population > 10000))` → en ordnad array med namn, för de regioner som korsas av den aktuella funktionen och med en befolkning som är större än 10000 • `overlay_crosses(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → en array av geometrier (i WKT), för upp till två regioner som korsas av den aktuella funktionen

Ytterligare läsning: *crosses*, *array manipulation*, *Välj efter plats* algoritm

overlay_disjoint

Returnerar om den aktuella funktionen är spatialt disjunkt från alla funktioner i ett mållager, eller en array med uttrycksbaserade resultat för de funktioner i mållagret som är disjunkta från den aktuella funktionen.

Läs mer om det underliggande GEOS ”Disjoint”-predikatet, som beskrivs i PostGIS ST_Disjoint function.

Syntax	overlay_disjoint(layer, [expression], [filter], [limit], [cache=false]) [] markerar valfria argument
Argument	• **lager** - det lager vars överlagring kontrolleras • **expression** - ett valfritt uttryck som ska utvärderas på funktionerna från mållagret. Om det inte anges kommer funktionen bara att returnera ett booleanskt värde som anger om det finns minst en matchning. • **filter** - ett valfritt uttryck för att filtrera de målfunktioner som ska kontrolleras. Om det inte anges kommer alla funktioner att kontrolleras. • **limit** - ett valfritt heltal för att begränsa antalet matchande funktioner. Om inget anges kommer alla matchande funktioner att returneras. • **cache** - sätt detta till true för att bygga ett lokalt spatialt index (för det mesta är detta oönskat, såvida du inte arbetar med en särskilt långsam dataleverantör)
Exempel	• `overlay_disjoint('regions')` → TRUE om den aktuella funktionen är spatialt åtskild från alla regioner • `overlay_disjoint('regions', filter:= population > 10000)` → TRUE om den aktuella funktionen är spatialt åtskild från alla regioner med en population som är större än 10000 • `overlay_disjoint('regions', name)` → en array med namn för de regioner som är spatialt åtskilda från den aktuella funktionen • `array_to_string(overlay_disjoint('regions', name))` → en sträng som en kommaseparerad lista med namn för de regioner som är spatialt åtskilda från den aktuella funktionen • `array_sort(overlay_disjoint(layer:='regions', expression:="name", filter:= population > 10000))` → en ordnad array med namn, för de regioner som är spatialt åtskilda från den aktuella funktionen och har en befolkning som är större än 10000 • `overlay_disjoint(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → en array av geometrier (i WKT), för upp till två regioner som är spatialt åtskilda från den aktuella funktionen

Ytterligare läsning: *disjoint*, *array manipulation*, *Välj efter plats* algoritm

overlay_equals

Returnerar om den aktuella funktionen är spatialt lika med minst en funktion från ett mållager, eller en array med uttrycksbaserade resultat för funktionerna i mållagret som är spatialt lika med den aktuella funktionen.

Läs mer om det underliggande GEOS ”Equals”-predikatet, som beskrivs i PostGIS ST_Equals-funktionen.

Syntax	overlay_equals(layer, [expression], [filter], [limit], [cache=false]) [] markerar valfria argument
Argument	• **lager** - det lager vars överlagring kontrolleras • **expression** - ett valfritt uttryck som ska utvärderas på funktionerna från mållagret. Om det inte anges kommer funktionen bara att returnera ett booleanskt värde som anger om det finns minst en matchning. • **filter** - ett valfritt uttryck för att filtrera de målfunktioner som ska kontrolleras. Om det inte anges kommer alla funktioner att kontrolleras. • **limit** - ett valfritt heltal för att begränsa antalet matchande funktioner. Om inget anges kommer alla matchande funktioner att returneras. • **cache** - sätt detta till true för att bygga ett lokalt spatialt index (för det mesta är detta oönskat, såvida du inte arbetar med en särskilt långsam dataleverantör)
Exempel	• `overlay_equals('regions')` → TRUE om den aktuella funktionen är spatialt lika med en region • `overlay_equals('regions', filter:= population > 10000)` → TRUE om den aktuella funktionen är spatialt lika med en region med en befolkning större än 10000 • `overlay_equals('regions', name)` → en array med namn för de regioner som är spatialt lika med den aktuella funktionen • `array_to_string(overlay_equals('regions', name))` → en sträng som en kommaseparerad lista med namn, för de regioner som är spatialt lika med den aktuella funktionen • `array_sort(overlay_equals(layer:='regions', expression:="name", filter:= population > 10000))` → en ordnad array med namn, för de regioner som är spatialt lika med den aktuella funktionen och har en befolkning som är större än 10000 • `overlay_equals(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → en array av geometrier (i WKT), för upp till två regioner som är spatialt lika med den aktuella funktionen

Ytterligare läsning: *array manipulation*, *Välj efter plats* algoritm

overlay_intersects

Returnerar om den aktuella funktionen spatialt skär minst en funktion från ett mållager, eller en array med uttrycksbaserade resultat för de funktioner i mållagret som skärs av den aktuella funktionen.

Läs mer om det underliggande GEOS ”Intersects”-predikatet, som beskrivs i PostGIS ST_Intersects-funktion.

Syntax	overlay_intersects(layer, [expression], [filter], [limit], [cache=false], [min_overlap], [min_inscribed_circle_radius], [return_details], [sort_by_intersection_size]) [] markerar valfria argument
Argument	• **lager** - det lager vars överlagring kontrolleras • **expression** - ett valfritt uttryck som ska utvärderas på funktionerna från mållagret. Om det inte anges kommer funktionen bara att returnera ett booleanskt värde som anger om det finns minst en matchning. • **filter** - ett valfritt uttryck för att filtrera de målfunktioner som ska kontrolleras. Om det inte anges kommer alla funktioner att kontrolleras. • **limit** - ett valfritt heltal för att begränsa antalet matchande funktioner. Om inget anges kommer alla matchande funktioner att returneras. • **cache** - sätt detta till true för att bygga ett lokalt spatialt index (för det mesta är detta oönskat, såvida du inte arbetar med en särskilt långsam dataleverantör) • **min_overlap** - definierar ett valfritt uteslutningsfilter: – för polygoner, en minsta area i kvadratenheter för aktuell funktion för skärningspunkten. Om skärningspunkten resulterar i flera polygoner returneras skärningspunkten om minst en av polygonerna har en area som är större än eller lika med värdet – för linjer, en minimilängd i aktuella funktionsenheter. Om skärningen resulterar i flera linjer returneras skärningen om minst en av linjerna har en längd som är större än eller lika med värdet. • **min_inscribed_circle_radius** - definierar ett valfritt uteslutningsfilter (endast för polygoner): minsta radie i aktuella funktionsenheter för den maximalt inskrivna cirkeln i skärningspunkten. Om skärningspunkten resulterar i flera polygoner returneras skärningspunkten om minst en av polygonerna har en radie för den maximalt inskrivna cirkeln som är större än eller lika med värdet. Läs mer om det underliggande GEOS-predikatet, som beskrivs i PostGIS ST_MaximumInscribedCircle-funktionen. Detta argument kräver GEOS >= 3.9. • **return_details** - Sätt detta till true för att returnera en lista med kartor som innehåller (nyckelnamn inom citationstecken) funktionens ”id”, uttrycket ”result” och ”overlap”-värdet (för det största elementet vid multipart). Radien för den maximalt inskrivna cirkeln returneras också när mållagret är en polygon. Endast giltig när den används med parametern expression • **sort_by_intersection_size** - endast giltigt när det används med ett uttryck, sätt detta till ”des” för att returnera resultaten ordnade efter överlappningsvärdet i fallande ordning eller sätt detta till ”asc” för stigande ordning.
Exempel	• `overlay_intersects('regions')` → TRUE om den aktuella funktionen spatialt skär en region • `overlay_intersects('regions', filter:= population > 10000)` → TRUE om den aktuella funktionen spatialt skär en region med en befolkning större än 10000 • `overlay_intersects('regions', name)` → en array med namn för de regioner som genomkorsas av den aktuella funktionen • `array_to_string(overlay_intersects('regions', name))` → en sträng som en kommaseparerad lista med namn för de regioner som genomkorsas av den aktuella funktionen • `array_sort(overlay_intersects(layer:='regions', expression:="name", filter:= population > 10000))` → en ordnad array med namn, för de regioner som genomkorsas av den aktuella funktionen och med en befolkning som är större än 10000 • `overlay_intersects(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → en array av geometrier (i WKT), för upp till två regioner som genomskärs av den aktuella funktionen • `overlay_intersects(layer:='regions', min_overlap:=0.54)` → TRUE om den aktuella funktionen spatialt skär en region och skärningsområdet (för minst en av delarna när det gäller multipolygoner) är större än eller lika med 0,54 • `overlay_intersects(layer:='regions', min_inscribed_circle_radius:=0.54)` → TRUE om den aktuella funktionen spatialt skär en region och skärningsområdets maximala radie för den inskrivna cirkeln (för minst en av delarna om det gäller flera delar) är större än eller lika med 0,54

Ytterligare läsning: *intersects*, *array manipulation*, *Välj efter plats* algoritm

överlagring_närmaste

Returnerar om den aktuella funktionen har funktion(er) från ett mållager inom ett givet avstånd, eller en array med uttrycksbaserade resultat för funktionerna i mållagret inom ett avstånd från den aktuella funktionen.

Obs: Denna funktion kan vara långsam och förbruka mycket minne för stora lager.

Syntax	overlay_nearest(lager, [uttryck], [filter], [limit=1], [max_avstånd], [cache=false]) [] markerar valfria argument
Argument	• **lager** - mållagret • **expression** - ett valfritt uttryck som ska utvärderas på funktionerna från mållagret. Om det inte anges kommer funktionen bara att returnera ett booleanskt värde som anger om det finns minst en matchning. • **filter** - ett valfritt uttryck för att filtrera de målfunktioner som ska kontrolleras. Om det inte anges kommer alla funktioner i mållagret att användas. • **limit** - ett valfritt heltal för att begränsa antalet matchande funktioner. Om värdet inte anges returneras endast den närmaste funktionen. Om värdet sätts till -1 returneras alla matchande funktioner. • **max_distance** - ett valfritt avstånd för att begränsa sökningen efter matchande funktioner. Om det inte anges kommer alla funktioner i mållagret att användas. • **cache** - sätt detta till true för att bygga ett lokalt spatialt index (för det mesta är detta oönskat, såvida du inte arbetar med en särskilt långsam dataleverantör)
Exempel	• `overlay_nearest('airports')` → TRUE om lagret ”airports” har minst en funktion • `overlay_nearest('airports', max_distance:= 5000)` → TRUE om det finns en flygplats inom ett avstånd på 5000 kartenheter från den aktuella funktionen • `overlay_nearest('airports', name)` → namnet på den flygplats som ligger närmast den aktuella funktionen, som en array • `array_to_string(overlay_nearest('airports', name))` → namnet på den flygplats som ligger närmast den aktuella funktionen, som en sträng • `overlay_nearest(layer:='airports', expression:= name, max_distance:= 5000)` → namnet på den närmaste flygplatsen inom ett avstånd på 5000 kartenheter från den aktuella funktionen, som en array • `overlay_nearest(layer:='airports', expression:="name", filter:="Use"='Civilian', limit:=3)` → en array med namn, för upp till de tre närmaste civila flygplatserna ordnade efter avstånd • `overlay_nearest(layer:='airports', expression:="name", limit:= -1, max_distance:= 5000)` → en array med namn för alla flygplatser inom ett avstånd på 5000 kartenheter från den aktuella funktionen, ordnade efter avstånd

Ytterligare läsning: *array manipulation*, *Sammanfoga attribut efter närmaste* algoritm

överlägg_touches

Returnerar om den aktuella funktionen spatialt berör minst en funktion från ett mållager, eller en array med uttrycksbaserade resultat för de funktioner i mållagret som berörs av den aktuella funktionen.

Läs mer om det underliggande GEOS ”Touches” predikatet, som beskrivs i PostGIS ST_Touches funktion.

Syntax	overlay_touches(layer, [expression], [filter], [limit], [cache=false]) [] markerar valfria argument
Argument	• **lager** - det lager vars överlagring kontrolleras • **expression** - ett valfritt uttryck som ska utvärderas på funktionerna från mållagret. Om det inte anges kommer funktionen bara att returnera ett booleanskt värde som anger om det finns minst en matchning. • **filter** - ett valfritt uttryck för att filtrera de målfunktioner som ska kontrolleras. Om det inte anges kommer alla funktioner att kontrolleras. • **limit** - ett valfritt heltal för att begränsa antalet matchande funktioner. Om inget anges kommer alla matchande funktioner att returneras. • **cache** - sätt detta till true för att bygga ett lokalt spatialt index (för det mesta är detta oönskat, såvida du inte arbetar med en särskilt långsam dataleverantör)
Exempel	• `overlay_touches('regions')` → TRUE om den aktuella funktionen spatialt berör en region • `overlay_touches('regions', filter:= population > 10000)` → TRUE om den aktuella funktionen spatialt berör en region med en population som är större än 10000 • `overlay_touches('regions', name)` → en array med namn för de regioner som berörs av den aktuella funktionen • `string_to_array(overlay_touches('regions', name))` → en sträng som en kommaseparerad lista med namn för de regioner som berörs av den aktuella funktionen • `array_sort(overlay_touches(layer:='regions', expression:="name", filter:= population > 10000))` → en ordnad array med namn, för de regioner som berörs av den aktuella funktionen och med en befolkning som är större än 10000 • `overlay_touches(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → en array av geometrier (i WKT), för upp till två regioner som berörs av den aktuella funktionen

Ytterligare läsning: *touches*, *array manipulation*, *Välj efter plats* algoritm

overlay_within

Returnerar om den aktuella funktionen är spatialt inom minst en funktion från ett mållager, eller en array med uttrycksbaserade resultat för de funktioner i mållagret som innehåller den aktuella funktionen.

Läs mer om det underliggande GEOS ”Within”-predikatet, som beskrivs i PostGIS ST_Within-funktion.

Syntax	overlay_within(layer, [expression], [filter], [limit], [cache=false]) [] markerar valfria argument
Argument	• **lager** - det lager vars överlagring kontrolleras • **expression** - ett valfritt uttryck som ska utvärderas på funktionerna från mållagret. Om det inte anges kommer funktionen bara att returnera ett booleanskt värde som anger om det finns minst en matchning. • **filter** - ett valfritt uttryck för att filtrera de målfunktioner som ska kontrolleras. Om det inte anges kommer alla funktioner att kontrolleras. • **limit** - ett valfritt heltal för att begränsa antalet matchande funktioner. Om inget anges kommer alla matchande funktioner att returneras. • **cache** - sätt detta till true för att bygga ett lokalt spatialt index (för det mesta är detta oönskat, såvida du inte arbetar med en särskilt långsam dataleverantör)
Exempel	• `overlay_within('regions')` → TRUE om den aktuella funktionen är spatialt inom en region • `overlay_within('regions', filter:= population > 10000)` → TRUE om den aktuella funktionen ligger spatialt inom en region med en population som är större än 10000 • `overlay_within('regions', name)` → en array med namn för de regioner som innehåller den aktuella funktionen • `array_to_string(overlay_within('regions', name))` → en sträng som en kommaseparerad lista med namn för de regioner som innehåller den aktuella funktionen • `array_sort(overlay_within(layer:='regions', expression:="name", filter:= population > 10000))` → en ordnad array med namn, för de regioner som innehåller den aktuella funktionen och har en befolkning som är större än 10000 • `overlay_within(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → en array av geometrier (i WKT), för upp till två regioner som innehåller den aktuella funktionen

Ytterligare läsning: *within*, *array manipulation*, *Välj efter plats* algoritm

$perimeter

Returnerar omkretslängden för den aktuella funktionen. Den omkrets som beräknas med den här funktionen respekterar både ellipsoidinställningen och inställningarna för avståndsenhet i det aktuella projektet. Om t.ex. en ellipsoid har ställts in för projektet kommer den beräknade omkretsen att vara ellipsoidisk, och om ingen ellipsoid har ställts in kommer den beräknade omkretsen att vara planimetrisk.

Syntax	$perimeter
Exempel	• `$perimeter` → 42

perimeter

Returnerar omkretsen för ett geometriskt polygonobjekt. Beräkningarna är alltid planimetriska i det spatiala referenssystemet (SRS) för denna geometri, och enheterna för den returnerade omkretsen kommer att matcha enheterna för SRS. Detta skiljer sig från de beräkningar som utförs av funktionen $perimeter, som utför ellipsoidiska beräkningar baserat på projektets inställningar för ellipsoid och avståndsenheter.

Syntax	perimeter(geometry)
Argument	• **geometry** - geometriobjekt för polygoner
Exempel	• `perimeter(geom_from_wkt('POLYGON((0 0, 4 0, 4 2, 0 2, 0 0))'))` → 12,0

point_n

Returnerar en specifik nod från en geometri.

Syntax	point_n(geometry, index)
Argument	• **geometry** - geometriobjekt • **index** - index för den nod som ska returneras, där 1 är den första noden; om värdet är negativt kommer det valda vertexindexet att vara dess totala antal minus det absoluta värdet
Exempel	• `geom_to_wkt(point_n(geom_from_wkt('POLYGON((0 0, 4 0, 4 2, 0 2, 0 0))'),2))` → 'Punkt (4 0)'

Ytterligare läsning: *Extrahera specifika toppar* algoritm

point_on_surface

Returnerar en punkt som garanterat ligger på ytan av en geometri.

Syntax	point_on_surface(geometry)
Argument	• **geometri** - en geometri
Exempel	• `point_on_surface(@geometry)` → en punktgeometri

Ytterligare läsning: *Punkt på ytan* algoritm

pole_of_inaccessibility

Beräknar den ungefärliga otillgänglighetspolen för en yta, vilket är den mest avlägsna inre punkten från ytans gräns. Denna funktion använder ”polylabel”-algoritmen (Vladimir Agafonkin, 2016), som är en iterativ metod som garanterat hittar den verkliga otillgänglighetspolen inom en angiven tolerans. Mer exakta toleranser kräver fler iterationer och tar längre tid att beräkna.

Syntax	pole_of_inaccessibility(geometry, tolerance)
Argument	• **geometri** - en geometri • **tolerans** - maximalt avstånd mellan den returnerade punkten och den verkliga stolppositionen
Exempel	• `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)’

Fig. 9.25: Pol av otillgänglighet

Ytterligare läsning: *Pol av otillgänglighet* algoritm

project

Returnerar en punkt som projiceras från en startpunkt med hjälp av ett avstånd, en bäring (azimut) och en elevation i radianer.

Syntax	project(point, distance, azimuth, [elevation]) [] markerar valfria argument
Argument	• **punkt** - startpunkt • **avstånd** - avstånd till projektet • **azimuth** - azimut i radianer medurs, där 0 motsvarar norr • **elevation** - lutningsvinkel i radianer
Exempel	• `geom_to_wkt(project(make_point(1, 2), 3, radians(270)))` → ’Punkt(-2, 2)’

För vidare läsning: *Projektpunkter (kartesiska)* algoritm

relate

Testar DE-9IM (Dimensional Extended 9 Intersection Model) representation av förhållandet mellan två geometrier.

Relationship variant

Returnerar representationen av förhållandet mellan två geometrier enligt DE-9IM (Dimensional Extended 9 Intersection Model).

Syntax	relate(geometry, geometry)
Argument	• **geometri** - en geometri • **geometri** - en geometri
Exempel	• `relate( geom_from_wkt( 'LINESTRING(40 40,120 120)' ), geom_from_wkt( 'LINESTRING(40 40,60 120)' ) )` → 'FF1F00102'

Mönstermatchningsvariant

Testar om DE-9IM-förhållandet mellan två geometrier stämmer överens med ett angivet mönster.

Syntax	relate(geometry, geometry, pattern)
Argument	• **geometri** - en geometri • **geometri** - en geometri • **mönster** - DE-9IM-mönster som ska matchas
Exempel	• `relate( geom_from_wkt( 'LINESTRING(40 40,120 120)' ), geom_from_wkt( 'LINESTRING(40 40,60 120)' ), '**1F001**' )` → TRUE

reverse

Omvänder riktningen på en linjestring genom att vända ordningen på dess hörn.

Syntax	reverse(geometry)
Argument	• **geometri** - en geometri
Exempel	• `geom_to_wkt(reverse(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)')))` → 'LINESTRING(2 2, 1 1, 0 0)'

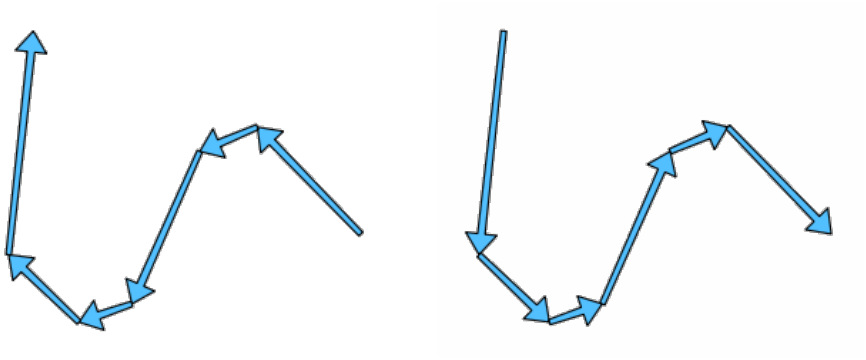


Fig. 9.26: Vändning av linjens riktning

Ytterligare läsning: *Omvänd riktning på linjen* algoritm

rotate

Returnerar en roterad version av en geometri. Beräkningar görs i geometrins spatiala referenssystem.

Syntax	rotate(geometry, rotation, [center=NULL], [per_part=false]) [] markerar valfria argument
Argument	• **geometri** - en geometri • **rotation** - medsols rotation i grader • **center** - rotationens mittpunkt. Om den inte anges används mitten av geometrins begränsningsbox. • **per_part** - tillämpa rotation per del. Om true, kommer rotation att tillämpas runt mitten av varje dels begränsningsbox när indatageometrin är flerdelad och en explicit rotationscentrumpunkt inte har angetts.
Exempel	• `rotate(@geometry, 45, make_point(4, 5))` → geometrin roterad 45 grader medurs runt punkten (4, 5) • `rotate(@geometry, 45)` → geometrin roterad 45 grader medurs runt mitten av dess begränsande box

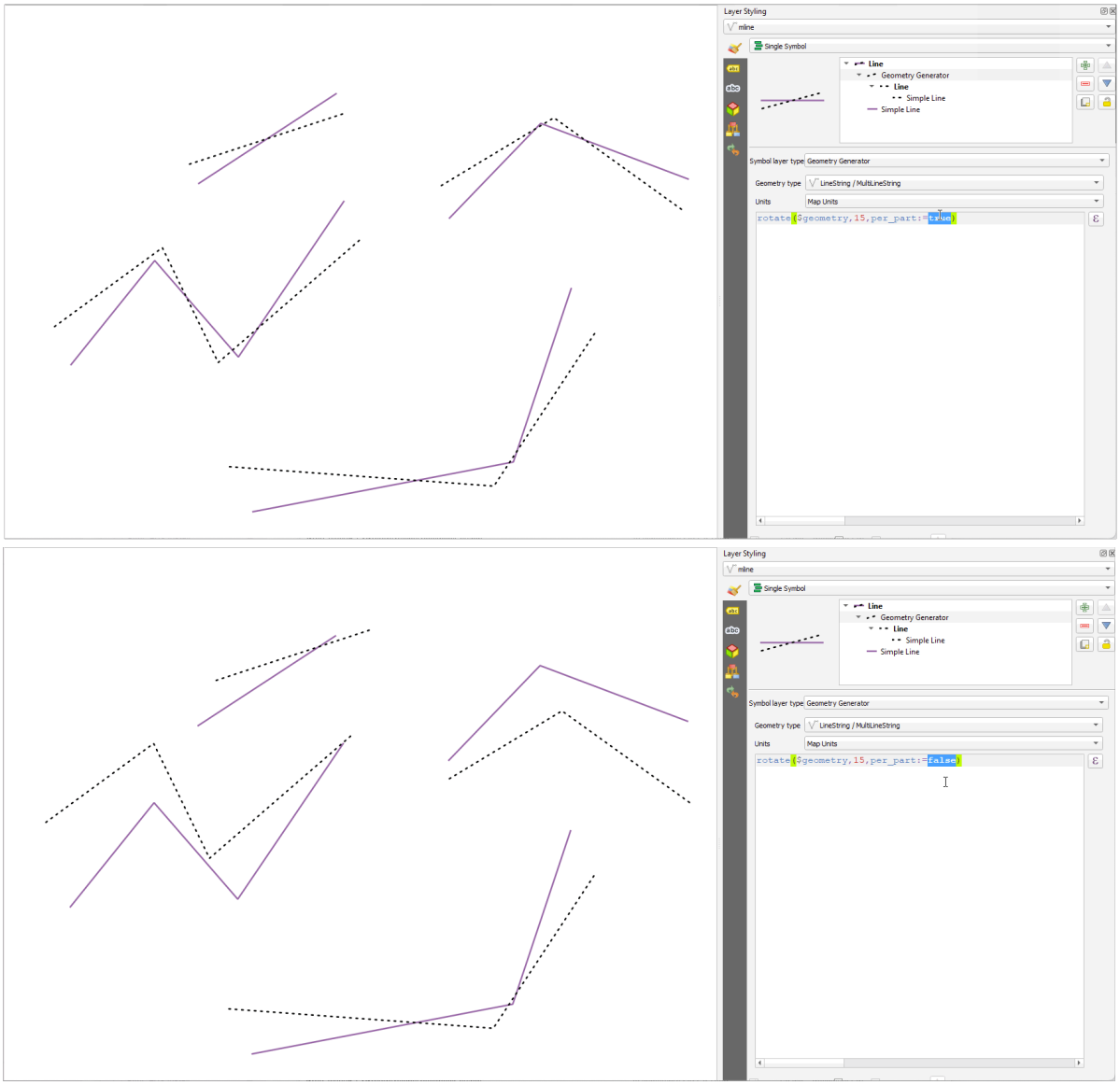


Fig. 9.27: Roterande funktioner

roundness

Beräknar hur nära en polygonform är en cirkel. Funktionen returnerar SANT när polygonformen är en perfekt cirkel och 0 när den är helt platt.

Syntax	roundness(geometry)
Argument	• **geometry** - en polygon
Exempel	• `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

Ytterligare läsning: *Rundhet*-algoritm

scale

Returnerar en skalad version av en geometri. Beräkningar görs i geometrins spatiala referenssystem.

Syntax	scale(geometry, x_scale, y_scale, [center]) [] markerar valfria argument
Argument	• **geometri** - en geometri • **x_scale** - Skalningsfaktor för x-axeln • **y_scale** - Skalningsfaktor för y-axeln • **center** - skalningens mittpunkt. Om den inte anges används mitten av geometrins begränsningsbox.
Exempel	• `scale(@geometry, 2, 0,5, make_point(4, 5))` → geometrin skalas två gånger horisontellt och halveras vertikalt, runt punkten (4, 5) • `scale(@geometri, 2, 0,5)` → geometri två gånger horisontellt och halverat vertikalt, runt mitten av dess begränsningsbox

segments_to_lines

Returnerar en geometri med flera linjer som består av en linje för varje segment i indatageometrin.

Syntax	segments_to_lines(geometry)
Argument	• **geometry** - geometriobjekt
Exempel	• `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))"

Ytterligare läsning: *Explodera linjer* algoritm

shared_paths

Returnerar en samling som innehåller banor som delas av de två inmatade geometrierna. De som går i samma riktning finns i det första elementet i samlingen, de som går i motsatt riktning finns i det andra elementet. Själva banorna anges i den första geometrins riktning.

Syntax	shared_paths(geometry1, geometry2)
Argument	• **geometry1** - en LineString/MultiLineString-geometri • **geometry2** - en LineString/MultiLineString-geometri
Exempel	• `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

Returnerar den kortaste linjen som förbinder geometri1 med geometri2. Den resulterande linjen kommer att börja vid geometri1 och sluta vid geometri2.

Syntax	shortest_line(geometry1, geometry2)
Argument	• **geometri1** - geometri för att hitta kortaste vägen från • **geometry2** - geometri för att hitta kortaste vägen till
Exempel	• `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

Förenklar en geometri genom att ta bort noder med hjälp av ett avståndsbaserat tröskelvärde (t.ex. Douglas Peucker-algoritmen). Algoritmen bevarar stora avvikelser i geometrier och minskar antalet toppar i nästan raka segment.

Syntax	simplify(geometry, tolerance)
Argument	• **geometri** - en geometri • **tolerans** - maximal avvikelse från raka segment för punkter som ska tas bort
Exempel	• `geom_to_wkt(simplify(geometry:=geom_from_wkt('LineString(0 0, 5 0.1, 10 0)'),tolerans:=5))` → 'LineString(0 0, 10 0)'

Fig. 9.28: Från vänster till höger, källager och ökande förenklingstoleranser

Ytterligare läsning: *Förenkla* algoritm

simplify_vw

Förenklar en geometri genom att ta bort noder med hjälp av ett ytbaserat tröskelvärde (t.ex. Visvalingam-Whyatt-algoritmen). Algoritmen tar bort hörn som skapar små områden i geometrier, t.ex. smala spikar eller nästan raka segment.

Syntax	simplify_vw(geometry, tolerance)
Argument	• **geometri** - en geometri • **tolerans** - ett mått på den maximala yta som skapas av en nod för att noden ska kunna tas bort
Exempel	• `geom_to_wkt(simplify_vw(geometry:=geom_from_wkt('LineString(0 0, 5 0, 5.01 10, 5.02 0, 10 0)'),tolerans:=5))` → 'LineString(0 0, 10 0)'

Ytterligare läsning: *Förenkla* algoritm

single_sided_buffer

Returnerar en geometri som bildats genom att buffra ut bara en sida av en linjestringgeometri. Avstånden är i det spatiala referenssystemet för denna geometri.

Syntax	single_sided_buffer(geometri, avstånd, [segment=8], [join=1], [miter_limit=2.0]) [] markerar valfria argument
Argument	• **geometri** - en (multi)linjestringsgeometri • **distance** - buffertavstånd. Positiva värden kommer att buffras till vänster om linjerna, negativa värden till höger • **segments** - antal segment som ska användas för att representera en kvartscirkel när en rund join-stil används. Ett större antal resulterar i en jämnare buffert med fler noder. • **join** - fogningsstil för hörn, där 1 = rund, 2 = gering och 3 = fasning • **miter_limit** - begränsning av mitterförhållandet som används för mycket skarpa hörn (endast vid användning av mitterförband)
Exempel	• `single_sided_buffer(@geometry, 10.5)` → linje buffrad till vänster med 10,5 enheter • `single_sided_buffer(@geometry, -10.5)` → linje buffrad åt höger med 10,5 enheter • `single_sided_buffer(@geometry, 10.5, segments:=16, join:=1)` → linjen buffras åt vänster med 10,5 enheter, med fler segment för att få en jämnare buffert • `single_sided_buffer(@geometry, 10.5, join:=3)` → linje buffrad till vänster med 10,5 enheter, med en avfasad skarv

Fig. 9.29: Vänster- och högersidig buffert på samma vektorlinjelager

Ytterligare läsning: *Enkelsidig buffert*-algoritm

sinuosity

Returnerar en kurvas sinuositet, vilket är förhållandet mellan kurvans längd och det raka (2D) avståndet mellan dess ändpunkter.

Syntax	sinuosity(geometry)
Argument	• **geometry** - Inmatningskurva (cirkulärsträng, linjesträng)
Exempel	• `rund(sinuositet(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

Utjämnar en geometri genom att lägga till extra noder som avrundar geometrins hörn. Om indatageometrier innehåller Z- eller M-värden kommer dessa också att jämnas ut och utdatageometrin kommer att behålla samma dimensionalitet som indatageometrin.

Syntax	smooth(geometri, [iterationer=1], [offset=0,25], [min_längd=-1], [max_vinkel=180]) [] markerar valfria argument
Argument	• **geometri** - en geometri • **iterations** - antal iterationer för utjämning som ska tillämpas. Större antal resulterar i jämnare men mer komplexa geometrier. • **offset** - värde mellan 0 och 0,5 som styr hur nära den utjämnade geometrin följer originalgeometrin. Mindre värden resulterar i en snävare utjämning, större värden resulterar i en lösare utjämning. • **min_length** - minsta längd på segmenten för att tillämpa utjämning. Denna parameter kan användas för att undvika att placera alltför många extra noder i kortare segment av geometrin. • **max_angle** - maximal vinkel vid noden för att utjämning ska tillämpas (0-180). Genom att sänka den maximala vinkeln kan avsiktligt skarpa hörn i geometrin bevaras. Ett värde på 80 grader innebär t.ex. att räta vinklar i geometrin bibehålls.
Exempel	• `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)'

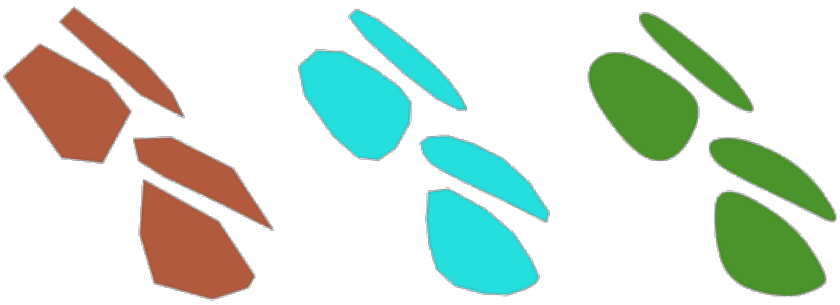


Fig. 9.30: Ökat antal iterationer ger jämnare geometrier

Ytterligare läsning: *Smidig* algoritm

square_wave

Konstruerar kvadratiska/rektangulära vågor längs en geometris gräns.

Syntax	square_wave(geometry, wavelength, amplitude, [strict=False]) [] markerar valfria argument
Argument	• **geometri** - en geometri • **våglängd** - våglängd för fyrkantig vågform • **amplitud** - amplitud för fyrkantig vågform • **strict** - Som standard behandlas våglängdsargumentet som en ”maximal våglängd”, där den faktiska våglängden kommer att justeras dynamiskt så att ett exakt antal kvadratvågor skapas längs geometrins gränser. Om argumentet strict är satt till true kommer våglängden att användas exakt och ett ofullständigt mönster kan komma att användas för den slutliga vågformen.
Exempel	• `square_wave(geom_from_wkt('LineString(0 0, 10 0)'), 3, 1)` → Kvadratvågor med våglängd 3 och amplitud 1 längs linjestringen

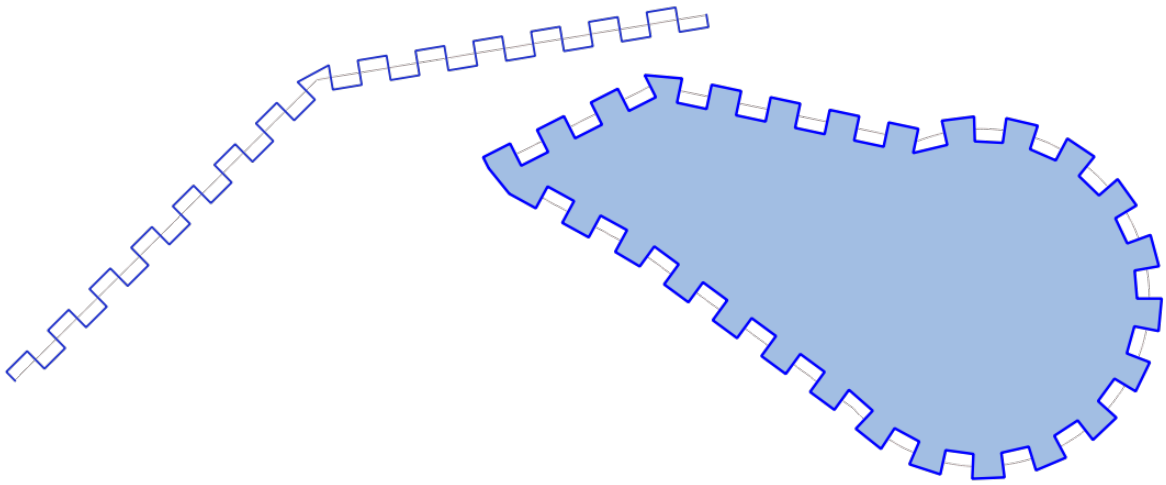


Fig. 9.31: Symbolisering av funktioner med fyrkantsvågor

square_wave_randomized

Konstruerar slumpmässiga kvadratiska/rektangulära vågor längs gränsen för en geometri.

Syntax	square_wave_randomized(geometry, min_våglängd, max_våglängd, min_amplitud, max_amplitud, [seed=0]) [] markerar valfria argument
Argument	• **geometri** - en geometri • **min_våglängd** - vågornas minsta våglängd • **max_våglängd** - vågornas maximala våglängd • **min_amplitude** - vågornas minsta amplitud • **max_amplitude** - vågornas maximala amplitud • **seed** - anger ett slumpmässigt frö för generering av vågor. Om seed är 0 kommer en helt slumpmässig uppsättning vågor att genereras.
Exempel	• `square_wave_randomized(geom_from_wkt('LineString(0 0, 10 0)'), 2, 3, 0.1, 0.2)` → Slumpmässigt dimensionerade fyrkantvågor med våglängder mellan 2 och 3 och amplituder mellan 0.1 och 0.2 längs linjestringen

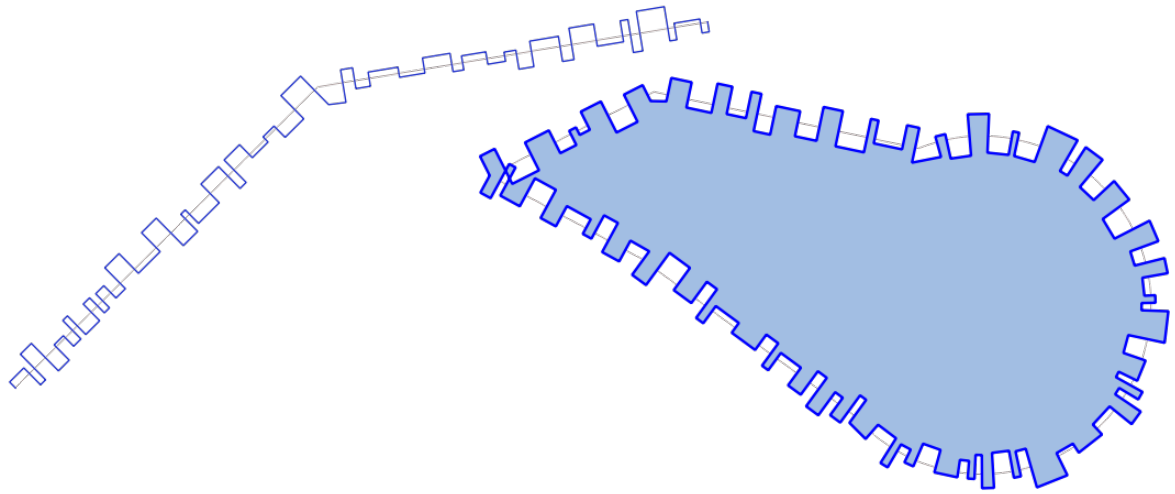


Fig. 9.32: Symbolisering av funktioner med kvadratiska slumpmässiga vågor

start_point

Returnerar den första noden från en geometri.

Syntax	start_point(geometry)
Argument	• **geometry** - geometriobjekt
Exempel	• `geom_to_wkt(start_point(geom_from_wkt('LINESTRING(4 0, 4 2, 0 2)')))` → 'Punkt (4 0)'

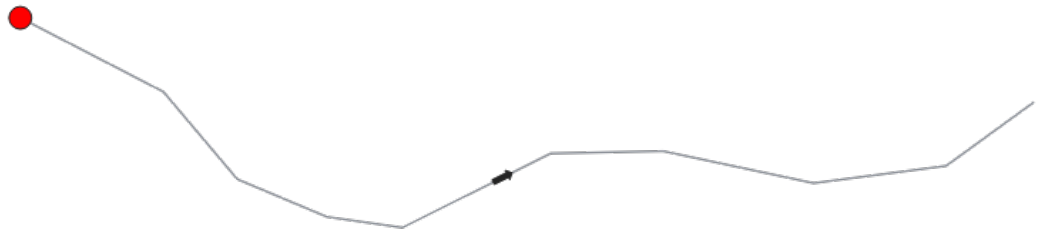


Fig. 9.33: Startpunkt för en linjefunktion

Ytterligare läsning: *end_point*, *Extrahera specifika toppar* algoritm

straight_distance_2d

Returnerar det direkta/euklidiska avståndet mellan det första och sista toppunktet i en geometri. Geometrin måste vara en kurva (circularstring, linestring).

Syntax	straight_distance_2d(geometry)
Argument	• **geometry** - Geometrin.
Exempel	• `rätt_avstånd_2d(geom_from_wkt('LINESTRING(1 0, 1 1)'))` → 1 • `rund(rak_avstånd_2d(geom_från_wkt('LINESTRING(1 4, 3 5, 5 0)')), 3)` → 5,657

Ytterligare läsning: *length*

sym_difference

Returnerar en geometri som representerar de delar av två geometrier som inte skär varandra.

Syntax	sym_difference(geometry1, geometry2)
Argument	• **geometri1** - en geometri • **geometry2** - en geometri
Exempel	• `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)

För vidare läsning: *Symmetrisk skillnad* algoritm

tapered_buffer

Skapar en buffert längs en linjegeometri där buffertdiametern varierar jämnt över linjens längd.

Syntax	tapered_buffer(geometry, start_width, end_width, [segments=8]) [] markerar valfria argument
Argument	• **geometry** - geometri för inmatning. Måste vara en (multi)linjegeometri. • **start_width** - buffertens bredd vid början av raden, • **end_width** - buffertens bredd vid slutet av raden. • **segments** - antal segment för att approximera kvartscirkelkurvor i bufferten.
Exempel	• `tapered_buffer(geometry:=geom_from_wkt('LINESTRING(1 2, 4 2)'),start_width:=1,end_width:=2,segments:=8)` → En avsmalnande buffert som börjar med en diameter på 1 och slutar med en diameter på 2 längs linestring-geometrin.

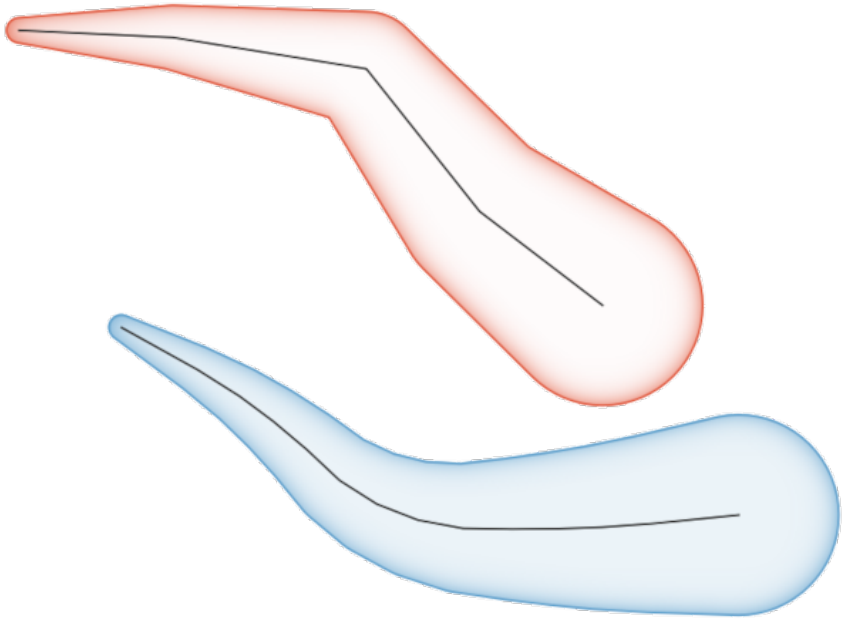


Fig. 9.34: Avsmalnande buffert på linjefunktioner

Ytterligare läsning: *Avsmalnande buffertar*-algoritm

touches

Testar om en geometri berör en annan. Returnerar TRUE om geometrierna har minst en gemensam punkt, men deras interiörer inte korsar varandra.

Syntax	touches(geometry1, geometry2)
Argument	• **geometri1** - en geometri • **geometry2** - en geometri
Exempel	• `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

Ytterligare läsning: *överlägg_touches*

transform

Returnerar den geometri som transformerats från ett CRS-källsystem till ett CRS-målsystem.

Syntax	transform(geometry, source_auth_id, dest_auth_id)
Argument	• **geometri** - en geometri • **source_auth_id** - CRS-ID för autentisering av källan • **dest_auth_id** - CRS-ID för autentisering av destinationen
Exempel	• `geom_to_wkt( transform( make_point(488995.53240249, 7104473.38600835), "EPSG:2154", "EPSG:4326" ) )` → ”POINT(0 51)

Ytterligare läsning: *Återprojicera lagret* algoritm

translate

Returnerar en översatt version av en geometri. Beräkningar görs i geometrins spatiala referenssystem.

Syntax	translate(geometry, dx, dy)
Argument	• **geometri** - en geometri • **dx** - delta x • **dy** - delta y
Exempel	• `translate(@geometry, 5, 10)` → en geometri av samma typ som den ursprungliga

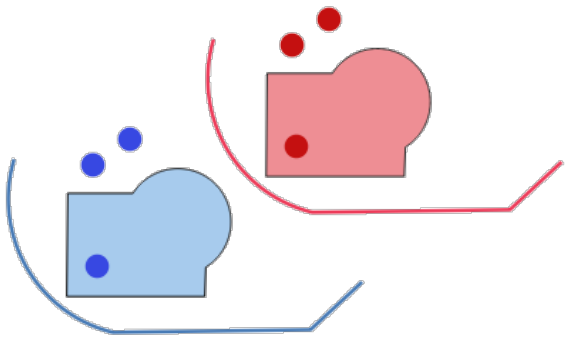


Fig. 9.35: Översättning av funktioner

För vidare läsning: *Översätt* algoritm

triangular_wave

Konstruerar triangulära vågor längs gränsen för en geometri.

Syntax	triangular_wave(geometry, wavelength, amplitude, [strict=False]) [] markerar valfria argument
Argument	• **geometri** - en geometri • **våglängd** - våglängd för triangulär vågform • **amplitud** - amplitud för triangulär vågform • **strict** - Som standard behandlas våglängdsargumentet som en ”maximal våglängd”, där den faktiska våglängden kommer att justeras dynamiskt så att ett exakt antal triangulära vågor skapas längs geometrins gränser. Om argumentet strict är satt till true kommer våglängden att användas exakt och ett ofullständigt mönster kan komma att användas för den slutliga vågformen.
Exempel	• `triangular_wave(geom_from_wkt('LineString(0 0, 10 0)'), 3, 1)` → Triangulära vågor med våglängd 3 och amplitud 1 längs linjestringen

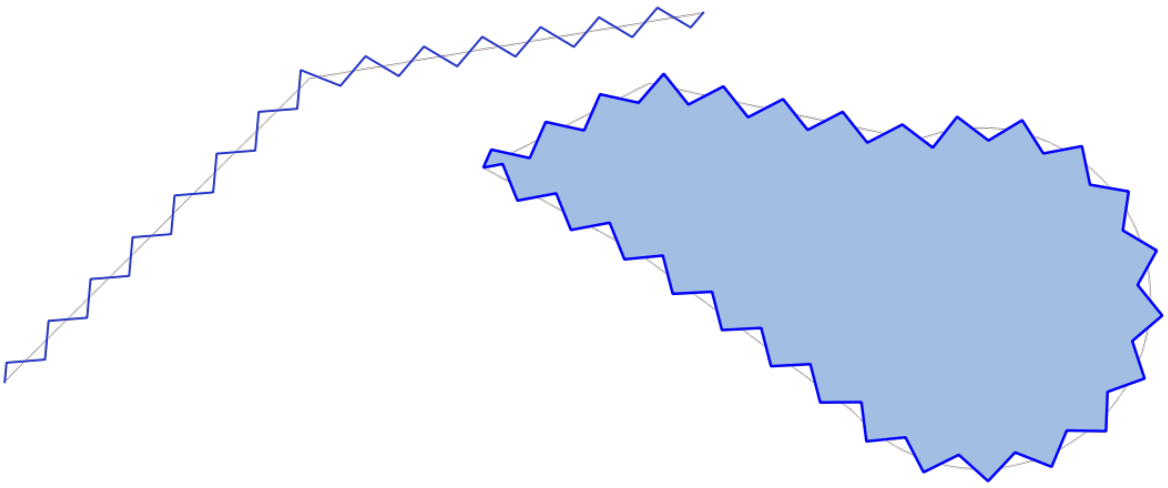


Fig. 9.36: Symbolisering av funktioner med triangulära vågor

triangular_wave_randomized

Konstruerar slumpmässiga triangulära vågor längs geometrins gränser.

Syntax	triangular_wave_randomized(geometry, min_wavelength, max_wavelength, min_amplitude, max_amplitude, [seed=0]) [] markerar valfria argument
Argument	• **geometri** - en geometri • **min_våglängd** - vågornas minsta våglängd • **max_våglängd** - vågornas maximala våglängd • **min_amplitude** - vågornas minsta amplitud • **max_amplitude** - vågornas maximala amplitud • **seed** - anger ett slumpmässigt frö för generering av vågor. Om seed är 0 kommer en helt slumpmässig uppsättning vågor att genereras.
Exempel	• `triangular_wave_randomized(geom_from_wkt('LineString(0 0, 10 0)'), 2, 3, 0.1, 0.2)` → Triangulära vågor av slumpmässig storlek med våglängder mellan 2 och 3 och amplituder mellan 0,1 och 0,2 längs linjestringen

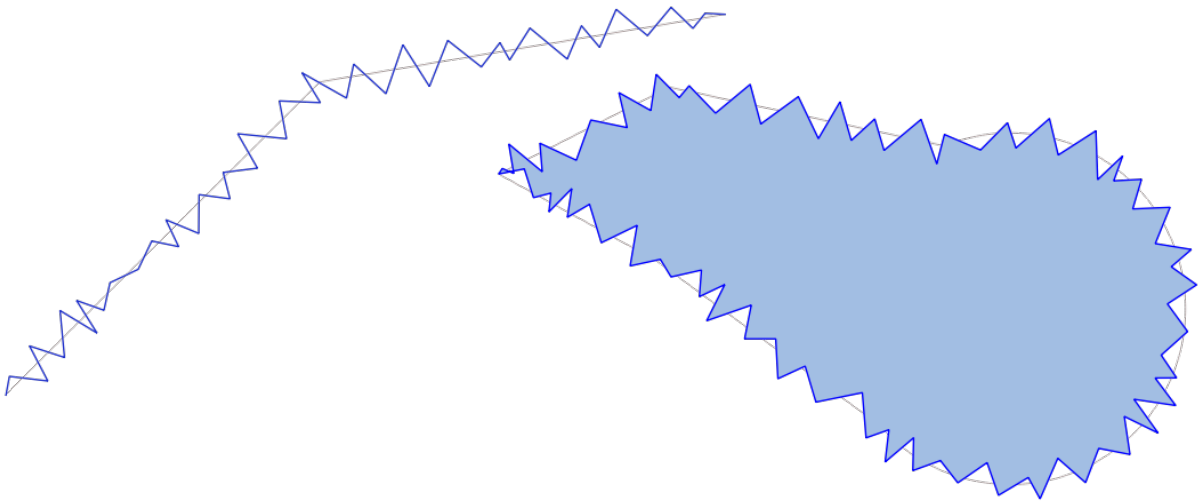


Fig. 9.37: Symbolisering av funktioner med triangulära slumpmässiga vågor

union

Returnerar en geometri som representerar punktuppsättningens förening av geometrierna.

Syntax	union(geometry1, geometry2)
Argument	• **geometri1** - en geometri • **geometry2** - en geometri
Exempel	• `geom_to_wkt( union( make_point(4, 4), make_point(5, 5) ) )` → ”MULTIPOINT(4 4, 5 5)

wave

Konstruerar rundade (sinusliknande) vågor längs en geometris gräns.

Syntax	wave(geometry, wavelength, amplitude, [strict=False]) [] markerar valfria argument
Argument	• **geometri** - en geometri • **våglängd** - våglängd för sinusliknande vågform • **amplitud** - amplitud för sinusliknande vågform • **strict** - Som standard behandlas våglängdsargumentet som en ”maximal våglängd”, där den faktiska våglängden kommer att justeras dynamiskt så att ett exakt antal vågor skapas längs geometrins gränser. Om argumentet strict är satt till true kommer våglängden att användas exakt och ett ofullständigt mönster kan komma att användas för den slutliga vågformen.
Exempel	• `wave(geom_from_wkt('LineString(0 0, 10 0)'), 3, 1)` → Sinusliknande vågor med våglängd 3 och amplitud 1 längs linjestringen

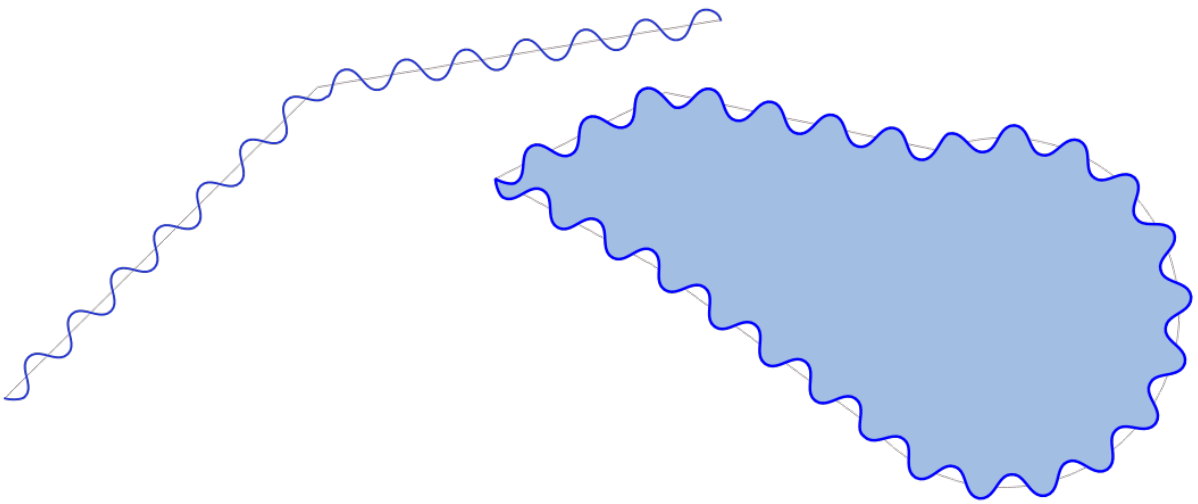


Fig. 9.38: Symbolisering av funktioner med vågor

wave_randomized

Konstruerar slumpmässiga krökta (sinusliknande) vågor längs en geometris gräns.

Syntax	wave_randomized(geometry, min_wavelength, max_wavelength, min_amplitude, max_amplitude, [seed=0]) [] markerar valfria argument
Argument	• **geometri** - en geometri • **min_våglängd** - vågornas minsta våglängd • **max_våglängd** - vågornas maximala våglängd • **min_amplitude** - vågornas minsta amplitud • **max_amplitude** - vågornas maximala amplitud • **seed** - anger ett slumpmässigt frö för generering av vågor. Om seed är 0 kommer en helt slumpmässig uppsättning vågor att genereras.
Exempel	• `wave_randomized(geom_from_wkt('LineString(0 0, 10 0)'), 2, 3, 0,1, 0,2)` → Slumpmässigt valda krökta vågor med våglängder mellan 2 och 3 och amplituder mellan 0,1 och 0,2 längs linjestringen

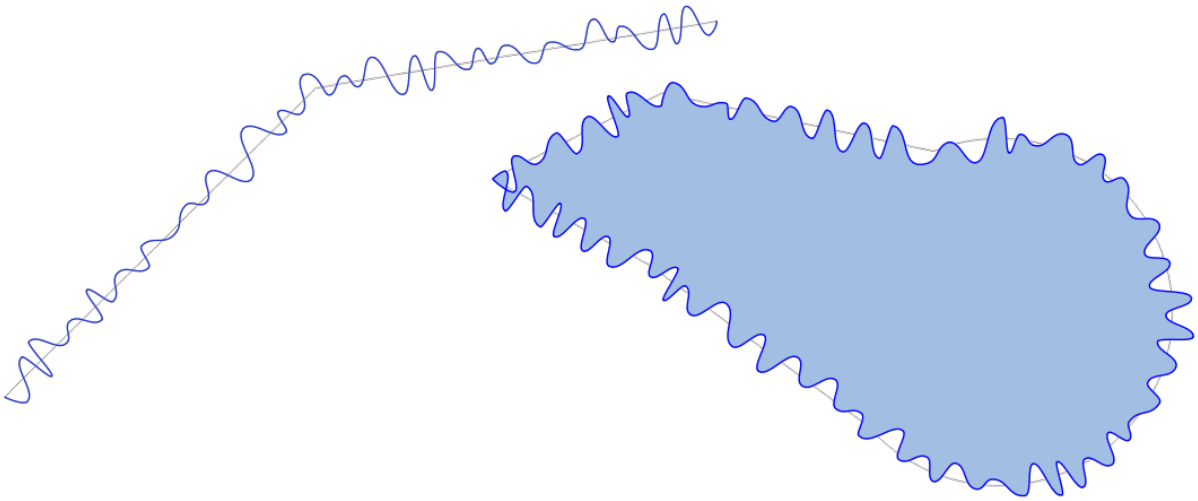


Fig. 9.39: Symbolisering av funktioner med slumpmässiga vågor

wedge_buffer

Returnerar en kilformad buffert som utgår från en punktgeometri.

Syntax	wedge_buffer(center, azimuth, width, outer_radius, [inner_radius=0.0]) [] markerar valfria argument
Argument	• **center** - buffertens mittpunkt (ursprung). Måste vara en punktgeometri. • **azimuth** - vinkel (i grader) som mitten av kilen ska peka mot. • **width** - buffertbredd (i grader). Observera att kilen kommer att sträcka sig till halva vinkelbredden på vardera sidan om azimutriktningen. • **outer_radius** - yttre radie för buffertar • **inner_radius** - valfri inre radie för buffertar
Exempel	• `wedge_buffer(center:=geom_from_wkt('POINT(1 2)'), azimuth:=90,width:=180,outer_radius:=1)` → En kilformad buffert centrerad på punkten (1,2), vänd mot öster, med en bredd på 180 grader och en yttre radie på 1.

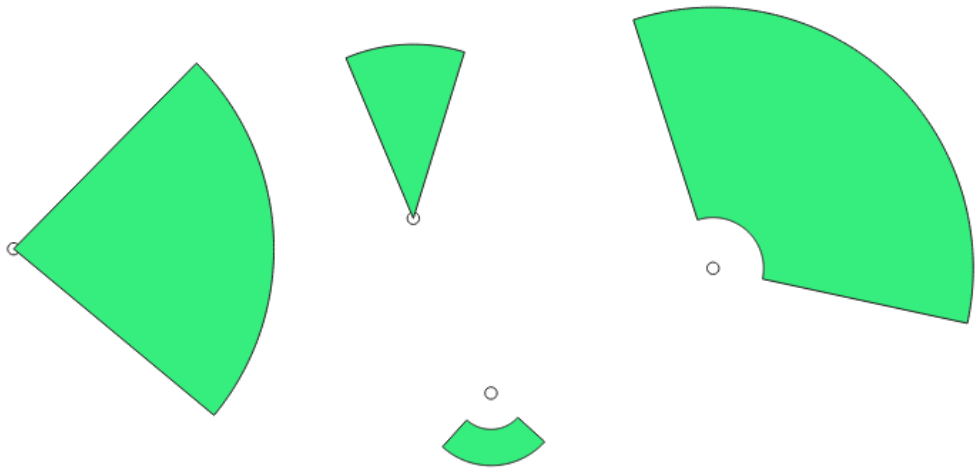


Fig. 9.40: Funktioner för buffring av kilar

För vidare läsning: *Skapa kilbuffertar* algoritm

within

Testar om en geometri är inom en annan. Returnerar TRUE om geometri1 är helt inom geometri2.

Syntax	within(geometry1, geometry2)
Argument	• **geometri1** - en geometri • **geometry2** - en geometri
Exempel	• `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

Ytterligare läsning: *overlay_within*

$x

Returnerar x-koordinaten för den aktuella punktfunktionen. Om funktionen är en flerpunktsfunktion returneras x-koordinaten för den första punkten. **VARNING: Denna funktion är föråldrad. Vi rekommenderar att du använder ersättningsfunktionen x() med variabeln @geometry i stället.**

Syntax	$x
Exempel	• `$x` → 42

Ytterligare läsning: *x*

x

Returnerar x-koordinaten för en punktgeometri, eller x-koordinaten för centroiden för en icke-punktgeometri.

Syntax	x(geometry)
Argument	• **geometri** - en geometri
Exempel	• `x( geom_from_wkt( 'POINT(2 5)' ) )` → 2 • `x( @geometri )` → x-koordinat för den aktuella funktionens tyngdpunkt

$x_at

Hämtar en x-koordinat för den aktuella funktionens geometri. **VARNING: Denna funktion är föråldrad. Vi rekommenderar att du använder ersättningsfunktionen x_at med variabeln @geometry i stället.**

Syntax	$x_at(vertex)
Argument	• **vertex** - index för vertexen i den aktuella geometrin (index börjar på 0; negativa värden gäller från det sista indexet, med början på -1)
Exempel	• `$x_at(1)` → 5

Ytterligare läsning: *x_at*

x_at

Hämtar en x-koordinat för geometrin.

Syntax	x_at(geometry, vertex)
Argument	• **geometry** - geometriobjekt • **vertex** - index för geometrins vertex (index börjar på 0; negativa värden gäller från det sista indexet, med början på -1)
Exempel	• `x_at( geom_from_wkt( 'POINT(4 5)' ), 0 )` → 4

x_max

Returnerar den maximala x-koordinaten för en geometri. Beräkningarna görs i geometrins spatiala referenssystem.

Syntax	x_max(geometry)
Argument	• **geometri** - en geometri
Exempel	• `x_max( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 4

x_min

Returnerar den minsta x-koordinaten för en geometri. Beräkningarna görs i geometrins spatiala referenssystem.

Syntax	x_min(geometry)
Argument	• **geometri** - en geometri
Exempel	• `x_min( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 2

$y

Returnerar y-koordinaten för den aktuella punktfunktionen. Om funktionen är en flerpunktsfunktion returneras y-koordinaten för den första punkten. **VARNING: Denna funktion är föråldrad. Vi rekommenderar att du använder ersättningsfunktionen y() med variabeln @geometry i stället.**

Syntax	$y
Exempel	• `$y` → 42

Ytterligare läsning: *y*

y

Returnerar y-koordinaten för en punktgeometri, eller y-koordinaten för centroiden för en icke-punktgeometri.

Syntax	y(geometry)
Argument	• **geometri** - en geometri
Exempel	• `y( geom_from_wkt( 'POINT(2 5)' ) )` → 5 • `y( @geometri )` → y-koordinat för den aktuella funktionens tyngdpunkt

$y_at

Hämtar en y-koordinat för den aktuella funktionens geometri. **VARNING: Denna funktion är föråldrad. Vi rekommenderar att du använder ersättningsfunktionen y_at med variabeln @geometry i stället.**

Syntax	$y_at(vertex)
Argument	• **vertex** - index för vertexen i den aktuella geometrin (index börjar på 0; negativa värden gäller från det sista indexet, med början på -1)
Exempel	• `$y_at(1)` → 2

Ytterligare läsning: *y_at*

y_at

Hämtar en y-koordinat för geometrin.

Syntax	y_at(geometry, vertex)
Argument	• **geometry** - geometriobjekt • **vertex** - index för geometrins vertex (index börjar på 0; negativa värden gäller från det sista indexet, med början på -1)
Exempel	• `y_at( geom_from_wkt( 'POINT(4 5)' ), 0 )` → 5

y_max

Returnerar den maximala y-koordinaten för en geometri. Beräkningarna görs i geometrins spatiala referenssystem.

Syntax	y_max(geometry)
Argument	• **geometri** - en geometri
Exempel	• `y_max( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 8

y_min

Returnerar den minsta y-koordinaten för en geometri. Beräkningarna görs i geometrins spatiala referenssystem.

Syntax	y_min(geometry)
Argument	• **geometri** - en geometri
Exempel	• `y_min( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 5

$z

Returnerar z-värdet för den aktuella punktfunktionen om den är 3D. Om funktionen är en flerpunktsfunktion returneras z-värdet för den första punkten. **VARNING: Denna funktion är föråldrad. Vi rekommenderar att du använder ersättningsfunktionen z() med variabeln @geometry istället.**

Syntax	$z
Exempel	• `$z` → 123

z

Returnerar z-koordinaten för en punktgeometri, eller NULL om geometrin inte har något z-värde.

Syntax	z(geometry)
Argument	• **geometri** - en punktgeometri
Exempel	• `z( geom_from_wkt( 'POINTZ(2 5 7)' ) )` → 7

z_at

Hämtar en z-koordinat för geometrin, eller NULL om geometrin inte har något z-värde.

Syntax	z_at(geometry, vertex)
Argument	• **geometry** - geometriobjekt • **vertex** - index för geometrins vertex (index börjar på 0; negativa värden gäller från det sista indexet, med början på -1)
Exempel	• `z_at(geom_from_wkt('LineStringZ(0 0 0, 10 10 5, 10 10 0)'), 1)` → 5

z_max

Returnerar den maximala z-koordinaten för en geometri, eller NULL om geometrin inte har något z-värde.

Syntax	z_max(geometry)
Argument	• **geometri** - en geometri med z-koordinat
Exempel	• `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

Returnerar den lägsta z-koordinaten för en geometri, eller NULL om geometrin inte har något z-värde.

Syntax	z_min(geometry)
Argument	• **geometri** - en geometri med z-koordinat
Exempel	• `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

9.2.14 Layoutfunktioner

Denna grupp innehåller funktioner för att manipulera egenskaper för utskriftslayoutobjekt.

item_variables

Returnerar en karta med variabler från ett layoutobjekt i den här utskriftslayouten.

Syntax	item_variables(id)
Argument	• **id** - ID för layoutobjekt
Exempel	• `map_get( item_variables('Map 0'), 'map_scale')` → skala för objektet 'Map 0' i den aktuella utskriftslayouten

Ytterligare läsning: Lista över standard *variabler*

map_credits

Returnerar en lista med kreditsträngar (användningsrättigheter) för de lager som visas i en layout eller ett specifikt layoutkartobjekt.

Syntax	map_credits([id], [include_layer_names=false], [layer_name_separator=': ']) [] markerar valfria argument
Argument	• **id** - Kartobjektets ID. Om inget anges kommer lagren från alla kartor i layouten att användas. • **include_layer_names** - Ställ in till true för att inkludera lagernamn före deras kreditsträngar • **layer_name_separator** - Sträng som ska infogas mellan lagernamn och deras kreditsträngar, om include_layer_names är true
Exempel	• `array_to_string( map_credits() )` → kommaseparerad lista med lagerkrediter för alla lager som visas i alla kartobjekt i layouten, t.ex. ”CC-BY-NC, CC-BY-SA • `array_to_string( map_credits( 'Huvudkarta' ) )` → kommaseparerad lista över lagerkrediter för lager som visas i layoutobjektet 'Huvudkarta', t.ex. 'CC-BY-NC, CC-BY-SA' • `array_to_string( map_credits( "Huvudkarta", include_layer_names := true, layer_name_separator := ': ' ) )` → kommaseparerad lista över lagernamn och deras krediter för lager som visas i layoutobjektet ”Huvudkarta”, t.ex. ”Järnvägslinjer: CC-BY-NC, Baskarta: CC-BY-SA

Denna funktion kräver att *Access metadata properties* för lagren har fyllts i.

9.2.15 Kartlager

Denna grupp innehåller en lista över tillgängliga lager i det aktuella projektet och, för varje lager, deras fält (lagrade i datasetet, virtuella eller extra samt från sammanfogningar). Fälten kan interageras på samma sätt som nämns i *Fält och värden*, förutom att ett dubbelklick lägger till namnet som en sträng (med enkla citattecken) i uttrycket i stället för som en fältreferens, eftersom de inte tillhör det aktiva lagret. Detta är ett bekvämt sätt att skriva uttryck som hänvisar till olika lager, t.ex. när man utför *aggregates*, *attribute* eller *spatial*.

Det ger också några praktiska funktioner för att manipulera lager.

decode_uri

Tar ett lager och avkodar uri för den underliggande dataleverantören. Det beror på dataleverantören vilka data som är tillgängliga.

Syntax	decode_uri(layer, [part]) [] markerar valfria argument
Argument	• **layer** - Det lager för vilket uri ska avkodas. • **part** - Den del av uri som ska returneras. Om inget anges kommer en karta med alla uri-delar att returneras.
Exempel	• `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

Returnerar ett matchande värde för lageregenskaper eller metadata.

Syntax	layer_property(layer, property)
Argument	• **lager** - en sträng som representerar antingen ett lagernamn eller ett lager-ID • **property** - en sträng som motsvarar den egenskap som ska returneras. Giltiga alternativ är: – namn: lagrets namn – id: lager-ID – title: titelsträng för metadata – abstrakt: metadata abstrakt sträng – nyckelord: nyckelord för metadata – data_url: URL för metadata – attribution: attributionssträng för metadata – attribution_url: URL för tilldelning av metadata – källa: lager källa – min_scale: minsta visningsskala för lager – max_scale: maximal visningsskala för lager – is_editable: om lagret är i redigeringsläge – crs: lager CRS – crs_definition: lager CRS fullständig definition – crs_description: beskrivning av CRS-lager – crs_ellipsoid: akronym för lagret CRS ellipsoid – extent: lagerutbredning (som ett geometriobjekt) – distance_units: lagrets avståndsenheter – type: typ av lager, t.ex. vektor eller raster – storage_type: lagringsformat (endast vektorlager) – geometry_type: geometrityp, t.ex. punkt (endast vektorlager) – feature_count: ungefärligt antal funktioner för lagret (endast vektorlager) – sökväg: Filsökväg till lagrets datakälla. Endast tillgängligt för filbaserade lager.
Exempel	• `layer_property('streets','title')` → 'Basemap Streets' • `layer_property('airports','feature_count')` → 120 • `layer_property('landsat','crs')` → 'EPSG:4326'

Läs mer: *vector*, *raster* och *mesh* lageregenskaper

load_layer

Laddar ett lager med käll-URI och leverantörsnamn.

Syntax	load_layer(uri, provider)
Argument	• **uri** - URI-sträng för källan till lagret • **provider** - namn på leverantör av lagerdata
Exempel	• `layer_property(load_layer('c:/data/roads.shp', 'ogr'), 'feature_count')` → antal funktioner från vektorlagret c:/data/roads.shp

9.2.16 Kartor Funktioner

Den här gruppen innehåller funktioner för att skapa eller manipulera nycklar och värden i map-datastrukturer (även kända som ordboksobjekt, nyckelvärdepar eller associativa arrayer). Till skillnad från *list-datastrukturen* där värdenas ordning spelar roll, är ordningen på nyckel-värdeparen i map-objektet inte relevant och värdena identifieras av sina nycklar.

from_json

Läser in en JSON-formaterad sträng.

Syntax	from_json(string)
Argument	• **sträng** - JSON-sträng
Exempel	• `from_json('{"1":"one","2":"two"}')` → { '1': 'en', '2': 'två' } • `from_json('[1,2,3]')` → [1,2,3]

hstore_to_map

Skapar en karta från en hstore-formaterad sträng.

Syntax	hstore_to_map(string)
Argument	• **sträng** - inmatningssträngen
Exempel	• `hstore_to_map('qgis=>rocks')` → { 'qgis': 'rocks' }

map

Returnerar en karta som innehåller alla nycklar och värden som skickas som ett par parametrar.

Syntax	map(nyckel1, värde1, nyckel2, värde2, ...)
Argument	• **nyckel** - en nyckel (sträng) • **value** - ett värde
Exempel	• `map('1','ett','2','två')` → {'1': 'ett', '2': 'två' } • `map('1','ett','2','två')['1']` → 'ett'

map_akeys

Returnerar alla nycklar i en map som en array.

Syntax	map_akeys(map)
Argument	• **map** - en karta
Exempel	• `map_akeys(map('1','ett','2','två'))` → ['1', '2']

map_avals

Returnerar alla värden i en map som en array.

Syntax	map_avals(map)
Argument	• **map** - en karta
Exempel	• `map_avals(map('1','ett','2','två'))` → ['ett', 'två']

map_concat

Returnerar en karta som innehåller alla poster i de angivna kartorna. Om två kartor innehåller samma nyckel, används värdet i den andra kartan.

Syntax	map_concat(map1, map2, …)
Argument	• **map** - en karta
Exempel	• `map_concat(map('1','ett','2','åsidosatt'),map('2','två','3','tre'))` → {'1': 'ett', '2': 'två', '3': 'tre' }

map_delete

Returnerar en karta med den angivna nyckeln och dess motsvarande värde raderat.

Syntax	map_delete(map, key)
Argument	• **map** - en karta • **nyckel** - den nyckel som ska raderas
Exempel	• `map_delete(map('1','ett','2','två'),'2')` → {'1': 'ett' }

map_exist

Returnerar TRUE om den angivna nyckeln finns i kartan.

Syntax	map_exist(map, key)
Argument	• **map** - en karta • **nyckel** - den nyckel som ska sökas upp
Exempel	• `map_exist(map('1','ett','2','två'),'3')` → FALSE

map_get

Returnerar värdet av en map, givet dess nyckel. Returnerar NULL om nyckeln inte finns.

Syntax	map_get(map, key)
Argument	• **map** - en karta • **nyckel** - den nyckel som ska sökas upp
Exempel	• `map_get(map('1','ett','2','två'),'2')` → 'två' • `map_get( item_variables('Map 0'), 'map_scale')` → skala för objektet 'Map 0' (om det finns) i den aktuella utskriftslayouten

Råd

Du kan också använda *indexoperatorn ([])* för att hämta ett värde från en map.

map_insert

Returnerar en map med en tillagd nyckel/värde. Om nyckeln redan finns åsidosätts dess värde.

Syntax	map_insert(map, key, value)
Argument	• **map** - en karta • **nyckel** - den nyckel som ska läggas till • **value** - det värde som ska läggas till
Exempel	• `map_insert(map('1','ett'),'3','tre')` → {'1': 'ett', '3': 'tre' } • `map_insert(map('1','one','2','overridden'),'2','two')` → {'1': 'one', '2': 'two' }

map_prefix_keys

Returnerar en karta med alla nycklar som har en given sträng som prefix.

Syntax	map_prefix_keys(map, prefix)
Argument	• **map** - en karta • **prefix** - en sträng
Exempel	• `map_prefix_keys(map('1','ett','2','två'), 'prefix-')` → { 'prefix-1': 'ett', 'prefix-2': 'två' }

map_to_hstore

Sammanfoga kartelement till en hstore-formaterad sträng.

Syntax	map_to_hstore(map)
Argument	• **map** - inmatningskartan
Exempel	• `map_to_hstore(map('1','ett','2','två'))` → '"1"=>"ett","2"=>"två"'

map_to_html_dl

Sammanfoga kartelement till en HTML-definitionsliststräng.

Syntax	map_to_html_dl(map)
Argument	• **map** - inmatningskartan
Exempel	• `map_to_html_dl(map('1','ett','2','två'))` → <dl><dt>1</dt><dd>ett</dd><dt>2</dt><dd>två</dd></dl>

map_to_html_table

Sammanfoga kartelement till en HTML-tabellsträng.

Syntax	map_to_html_table(map)
Argument	• **map** - inmatningskartan
Exempel	• `map_to_html_table(map('1','ett','2','två'))` → <table><thead><tr><th>1</th><th>2</th></tr></thead><tbody><tr><td>ett</td><td>två</td></tr></tbo

to_json

Skapa en JSON-formaterad sträng från en map, array eller annat värde.

Syntax	to_json(value)
Argument	• **värde** - Inmatningsvärdet
Exempel	• `to_json(map('1','ett','2','två'))` → {”1”:”one”,”2”:”two”} • `to_json(array(1,2,3))` → [1,2,3]

url_encode

Returnerar en URL-kodad sträng från en karta. Transformerar alla tecken i sin korrekt kodade form och producerar en helt kompatibel frågesträng.

Observera att plustecknet ”+” inte konverteras.

Syntax	url_encode(map)
Argument	• **map** - en karta.
Exempel	• `url_encode(map('a&+b','a och plus b','a=b','a är lika med b'))` → ’a%26+b=a%20and%20plus%20b&a%3Db=a%20equals%20b’

9.2.17 Matematiska funktioner

Den här gruppen innehåller matematiska funktioner (t.ex. kvadratrot, sin och cos).

abs

Returnerar det absoluta värdet av ett tal.

Syntax	abs(value)
Argument	• **värde** - ett tal
Exempel	• `abs(-2)` → 2

acos

Returnerar den inversa cosinus för ett värde i radianer.

Syntax	acos(value)
Argument	• **värde** - cosinus för en vinkel i radianer
Exempel	• `acos(0.5)` → 1.0471975511966

asin

Returnerar den inversa sinus för ett värde i radianer.

Syntax	asin(value)
Argument	• **värde** - sinus för en vinkel i radianer
Exempel	• `asin(1.0)` → 1.5707963267949

atan

Returnerar den inversa tangenten för ett värde i radianer.

Syntax	atan(value)
Argument	• **värde** - tan av en vinkel i radianer
Exempel	• `atan(0.5)` → 0.463647609000806

atan2

Returnerar den inversa tangenten av dy/dx genom att använda tecknen på de två argumenten för att bestämma kvadranten för resultatet.

Syntax	atan2(dy, dx)
Argument	• **dy** - skillnad i y-koordinat • **dx** - skillnad i x-koordinat
Exempel	• `atan2(1.0, 1.732)` → 0.523611477769969

ceil

Rundar ett tal uppåt.

Syntax	ceil(value)
Argument	• **värde** - ett tal
Exempel	• `ceil(4.9)` → 5 • `ceil(-4.9)` → -4

clamp

Begränsar ett inmatningsvärde till ett angivet intervall.

Syntax	clamp(minimum, input, maximum)
Argument	• **minimum** - det minsta värde som *input* får anta. • **input** - ett värde som begränsas till det intervall som anges av *minimum* och *maximum* • **maximum** - det största värde som *input* får anta
Exempel	• `clamp(1,5,10)` → 5 *input* är mellan 1 och 10 och returneras därför oförändrad • `clamp(1,0,10)` → 1 *inmatningen* är mindre än minimivärdet 1, så funktionen returnerar 1 • `clamp(1,11,10)` → 10 *inmatningen* är större än maxvärdet 10, så funktionen returnerar 10

cos

Returnerar cosinus för en vinkel.

Syntax	cos(angle)
Argument	• **vinkel** - vinkel i radianer
Exempel	• `cos(1,571)` → 0,000796326710733263

degrees

Konverterar från radianer till grader.

Syntax	degrees(radians)
Argument	• **radianer** - numeriskt värde
Exempel	• `grader(3.14159)` → 180 • `grader(1)` → 57,2958

exp

Returnerar exponentialen av ett värde.

Syntax	exp(value)
Argument	• **värde** - tal att returnera exponent för
Exempel	• `exp(1,0)` → 2.71828182845905

floor

Rundar ett tal nedåt.

Syntax	floor(value)
Argument	• **värde** - ett tal
Exempel	• `golv(4,9)` → 4 • `golv(-4,9)` → -5

ln

Returnerar den naturliga logaritmen av ett värde.

Syntax	ln(value)
Argument	• **värde** - numeriskt värde
Exempel	• `ln(1)` → 0 • `ln(2,7182818284590452354)` → 1

log

Returnerar värdet av logaritmen av det angivna värdet och basen.

Syntax	log(base, value)
Argument	• **bas** - valfritt positivt tal • **värde** - valfritt positivt tal
Exempel	• `log(2, 32)` → 5 • `log(0,5, 32)` → -5

log10

Returnerar värdet av bas 10-logaritmen för det passerade uttrycket.

Syntax	log10(value)
Argument	• **värde** - valfritt positivt tal
Exempel	• `log10(1)` → 0 • `log10(100)` → 2

max

Returnerar det största värdet i en uppsättning värden.

Syntax	max(value1, value2, …)
Argument	• **värde** - ett tal
Exempel	• `max(2,10,2,5,5)` → 10,2 • `max(20.5,NULL,6.2)` → 20.5

min

Returnerar det minsta värdet i en uppsättning värden.

Syntax	min(value1, value2, …)
Argument	• **värde** - ett tal
Exempel	• `min(20.5,10,6.2)` → 6.2 • `min(2,-10.3,NULL)` → -10.3

pi

Returnerar värdet av pi för beräkningar.

Syntax	pi()
Exempel	• `pi()` → 3.14159265358979

radians

Omvandlar från grader till radianer.

Syntax	radians(degrees)
Argument	• **grader** - numeriskt värde
Exempel	• `radianer(180)` → 3,14159 • `radianer(57,2958)` → 1

rand

Returnerar ett slumpmässigt heltal inom det intervall som anges av minimum- och maximum-argumentet (inklusive). Om ett frö anges kommer det returnerade talet alltid att vara detsamma, beroende på fröet.

Syntax	rand(min, max, [seed=NULL]) [] markerar valfria argument
Argument	• **min** - ett heltal som representerar det minsta möjliga slumpmässiga talet som önskas • **max** - ett heltal som representerar det största möjliga slumpmässiga talet som önskas • **seed** - valfritt värde att använda som seed
Exempel	• `rand(1, 10)` → 8

randf

Returnerar en slumpmässig float inom det intervall som anges av minimum- och maximum-argumentet (inklusive). Om ett frö anges kommer det returnerade värdet alltid att vara detsamma, beroende på fröet.

Syntax	randf([min=0,0], [max=1,0], [seed=NULL]) [] markerar valfria argument
Argument	• **min** - en flottör som representerar det minsta möjliga slumpmässiga talet som önskas • **max** - en float som representerar det största möjliga slumpmässiga talet som önskas • **seed** - valfritt värde att använda som seed
Exempel	• `randf(1, 10)` → 4.59258286403147

round

Avrundar ett tal till ett antal decimaler.

Syntax	round(värde, [platser=0]) [] markerar valfria argument
Argument	• **värde** - decimaltal som ska avrundas • **places** - Valfritt heltal som representerar antal platser att avrunda decimaler till. Kan vara negativt.
Exempel	• `rund(1234,567, 2)` → 1234,57 • `rund(1234,567)` → 1235 • `rund(1234,567, -1)` → 1230

scale_exponential

Transformerar ett givet värde från en inmatningsdomän till ett utmatningsområde med hjälp av en exponentiell kurva. Denna funktion kan användas för att underlätta värden i eller utanför det angivna utmatningsområdet.

Syntax	scale_exponential(value, domain_min, domain_max, range_min, range_max, exponent)
Argument	• **värde** - Ett värde i inmatningsdomänen. Funktionen returnerar ett motsvarande skalat värde i utdataområdet. • **domain_min** - Anger minimivärdet i indatadomänen, det minsta värde som indatavärdet bör anta. • **domain_max** - Anger det maximala värdet i indatadomänen, det största värde som indatavärdet bör anta. • **range_min** - Anger det lägsta värdet i utmatningsintervallet, det minsta värde som ska matas ut av funktionen. • **range_max** - Anger maxvärdet i utmatningsintervallet, det största värde som ska matas ut av funktionen. • **exponent** - Ett positivt värde (större än 0) som bestämmer hur ingångsvärdena ska mappas till utdataområdet. Stora exponenter gör att utdatavärdena ”lättar in” och börjar långsamt för att sedan accelerera när inmatningsvärdena närmar sig domänens maximum. Mindre exponenter (mindre än 1) gör att utdatavärdena ”lättar ut”, där mappningen börjar snabbt men saktar in när den närmar sig domänens maximum.
Exempel	• skalning exponentiell(5,0,10,0,100,2)→ 3,030 lätta på trycket, med en exponent på 2 • skala_exponentiell(3,0,10,0,100,0,5)`` → 87,585 lättnader, med en exponent på 0,5

skala_linjär

Transformerar ett givet värde från en inmatningsdomän till ett utmatningsområde med hjälp av linjär interpolation.

Syntax	scale_linear(värde, domän_min, domän_max, intervall_min, intervall_max)
Argument	• **värde** - Ett värde i inmatningsdomänen. Funktionen returnerar ett motsvarande skalat värde i utdataområdet. • **domain_min** - Anger minimivärdet i indatadomänen, det minsta värde som indatavärdet bör anta. • **domain_max** - Anger det maximala värdet i indatadomänen, det största värde som indatavärdet bör anta. • **range_min** - Anger det lägsta värdet i utmatningsintervallet, det minsta värde som ska matas ut av funktionen. • **range_max** - Anger maxvärdet i utmatningsintervallet, det största värde som ska matas ut av funktionen.
Exempel	• skalning_linjär(5,0,10,0,100)`` → 50 • skalning_linjär(0,2,0,1,0,360)`` → 72 skalning av ett värde mellan 0 och 1 till en vinkel mellan 0 och 360 • `skala_linjär(1500,1000,10000,9,20)` → 9,6111111 skala en population som varierar mellan 1000 och 10000 till en teckenstorlek mellan 9 och 20

scale_polynomial

Transformerar ett givet värde från en inmatningsdomän till ett utmatningsområde med hjälp av en polynomkurva. Denna funktion kan användas för att underlätta värden inom eller utanför det angivna utmatningsområdet.

Syntax	scale_polynomial(value, domain_min, domain_max, range_min, range_max, exponent)
Argument	• **värde** - Ett värde i inmatningsdomänen. Funktionen returnerar ett motsvarande skalat värde i utdataområdet. • **domain_min** - Anger minimivärdet i indatadomänen, det minsta värde som indatavärdet bör anta. • **domain_max** - Anger det maximala värdet i indatadomänen, det största värde som indatavärdet bör anta. • **range_min** - Anger det lägsta värdet i utmatningsintervallet, det minsta värde som ska matas ut av funktionen. • **range_max** - Anger maxvärdet i utmatningsintervallet, det största värde som ska matas ut av funktionen. • **exponent** - Ett positivt värde (större än 0) som bestämmer hur ingångsvärdena ska mappas till utdataområdet. Stora exponenter gör att utdatavärdena ”lättar in” och börjar långsamt för att sedan accelerera när inmatningsvärdena närmar sig domänens maximum. Mindre exponenter (mindre än 1) gör att utdatavärdena ”lättar ut”, där mappningen börjar snabbt men saktar in när den närmar sig domänens maximum.
Exempel	• `skala_polynom(5,0,10,0,100,2)` → 25 lätta på trycket, med en exponent på 2 • `skala_polynom(3,0,10,0,100,0,5)` → 54,772 lättnader, med en exponent på 0,5

synd

Returnerar sinus för en vinkel.

Syntax	sin(vinkel)
Argument	• **vinkel** - vinkel i radianer
Exempel	• `sin(1,571)` → 0,999999682931835

kvadrat

Returnerar kvadratroten av ett värde.

Syntax	sqrt(värde)
Argument	• **värde** - ett tal
Exempel	• `sqrt(9)` → 3

solbränna

Returnerar tangenten för en vinkel.

Syntax	tan(vinkel)
Argument	• **vinkel** - vinkel i radianer
Exempel	• `tan(1.0)` → 1.5574077246549

9.2.18 Meshes Funktioner

Denna grupp innehåller funktioner som beräknar eller returnerar nätrelaterade värden.

$yta_area

Returnerar arean för den aktuella meshytan. Den area som beräknas med denna funktion respekterar både ellipsoidinställningen och inställningarna för areaenhet i det aktuella projektet. Om t.ex. en ellipsoid har ställts in för projektet kommer den beräknade ytan att vara ellipsoidisk, och om ingen ellipsoid har ställts in kommer den beräknade ytan att vara planimetrisk.

Syntax	$yta_area
Exempel	• `$face_area` → 42

$face_index

Returnerar indexet för den aktuella meshytan.

Syntax	$face_index
Exempel	• `$face_index` → 4581

$vertex_som_punkt

Returnerar den aktuella vertexen som en punktgeometri.

Syntax	$vertex_som_punkt
Exempel	• `geom_to_wkt( $vertex_as_point )` → 'POINT(800 1500 41)'

$vertex_index

Returnerar indexet för den aktuella mesh-vertexen.

Syntax	$vertex_index
Exempel	• `$vertex_index` → 9874

$vertex_x

Returnerar X-koordinaten för den aktuella mesh-vertexen.

Syntax	$vertex_x
Exempel	• `$vertex_x` → 42.12

$vertex_y

Returnerar Y-koordinaten för den aktuella mesh-vertexen.

Syntax	$vertex_y
Exempel	• `$vertex_y` → 12.24

$vertex_z

Returnerar Z-värdet för den aktuella mesh-vertexen.

Syntax	$vertex_z
Exempel	• `$vertex_z` → 42

9.2.19 Operatörer

Denna grupp innehåller operatorer (t.ex. +, -, *). Observera att för de flesta av de matematiska funktionerna nedan gäller att om en av indata är NULL blir resultatet NULL.

%

Återstoden av divisionen. Tar tecknet för utdelningen.

Syntax	a % b
Argument	• **a** - värde • **b** - värde
Exempel	• `9 % 2` → 1 • `9 % -2` → 1 • `-9 % 2` → -1 • `5 % NULL` → NULL

*

Multiplikation av två värden

Syntax	a * b
Argument	• **a** - värde • **b** - värde
Exempel	• `5 * 4` → 20 • `5 * NULL` → NULL

+

Addition av två värden. Om ett av värdena är NULL blir resultatet NULL.

Syntax	a + b
Argument	• **a** - värde • **b** - värde
Exempel	• `5 + 4` → 9 • `5 + NULL` → NULL • `'QGIS ' + 'STENAR'` → 'QGIS STENAR' • `to_datetime('2020-08-01 12:00:00') + '1 dag 2 timmar'` → 2020-08-02T14:00:00

Ytterligare läsning: *concat*, ||

-

Subtraktion av två värden. Om ett av värdena är NULL blir resultatet NULL.

Syntax	a - b
Argument	• **a** - värde • **b** - värde
Exempel	• `5 - 4` → 1 • `5 - NULL` → NULL • `to_datetime('2012-05-05 12:00:00') - to_interval('1 dag 2 timmar')` → 2012-05-04T10:00:00

/

Division av två värden

Syntax	a / b
Argument	• **a** - värde • **b** - värde
Exempel	• `5 / 4` → 1.25 • `5 / NULL` → NULL

//

Golvdelning av två värden

Syntax	a // b
Argument	• **a** - värde • **b** - värde
Exempel	• `9 // 2` → 4

<

Jämför två värden och utvärderar till 1 om det vänstra värdet är mindre än det högra värdet.

Syntax	a < b
Argument	• **a** - värde • **b** - värde
Exempel	• `5 < 4` → FALSE • `5 < 5` → FALSE • `4 < 5` → TRUE

<=

Jämför två värden och utvärderar till 1 om det vänstra värdet är mindre eller lika med det högra värdet.

Syntax	a <= b
Argument	• **a** - värde • **b** - värde
Exempel	• `5 <= 4` → FALSE • `5 <= 5` → TRUE • `4 <= 5` → TRUE

<>

Jämför två värden och utvärderar till 1 om de inte är lika.

Syntax	a <> b
Argument	• **a** - värde • **b** - värde
Exempel	• `5 <> 4` → TRUE • `4 <> 4` → FALSE • `5 <> NULL` → NULL • `NULL <> NULL` → NULL

=

Jämför två värden och värderar dem till 1 om de är lika.

Syntax	a = b
Argument	• **a** - värde • **b** - värde
Exempel	• `5 = 4` → FALSE • `4 = 4` → TRUE • `5 = NULL` → NULL • `NULL = NULL` → NULL

>

Jämför två värden och utvärderar till 1 om det vänstra värdet är större än det högra värdet.

Syntax	a > b
Argument	• **a** - värde • **b** - värde
Exempel	• `5 > 4` → TRUE • `5 > 5` → FALSE • `4 > 5` → FALSE

>=

Jämför två värden och utvärderar till 1 om det vänstra värdet är större eller lika med det högra värdet.

Syntax	a >= b
Argument	• **a** - värde • **b** - värde
Exempel	• `5 >= 4` → TRUE • `5 >= 5` → TRUE • `4 >= 5` → FALSE

OCH

Returnerar TRUE när villkoren a och b är sanna.

Syntax	a OCH b
Argument	• **a** - skick • **b** - skick
Exempel	• `SANT OCH SANT` → SANT • `SANT OCH FALSKT` → FALSKT • `4 = 2+2 AND 1 = 1` → TRUE • `4 = 2+2 AND 1 = 2` → FALSE

BETWEEN

Returnerar TRUE om värdet ligger inom det angivna intervallet. Intervallet anses inkludera gränserna. För att testa för uteslutning kan NOT BETWEEN användas.

Syntax	värde MELLAN lägre_bound OCH högre_bound
Argument	• **value** - det värde som ska jämföras med ett intervall. Det kan vara en sträng, ett tal eller ett datum. • **lägre_gräns OCH högre_gräns** - gränser för intervall
Exempel	• `'B' BETWEEN 'A' AND 'C'` → TRUE • `2 MELLAN 1 OCH 3` → SANT • `2 MELLAN 2 OCH 3` → SANT • `'B' BETWEEN 'a' AND 'c'` → FALSE • `lower('B') MELLAN 'a' OCH 'b'` → TRUE

Observera

value BETWEEN lower_bound AND higher_bound är detsamma som ”*value* >= *lower_bound AND value* <= *higher_bound*”.

Ytterligare läsning: *NOT BETWEEN* (uttrycksfunktionens operatorer)

ILIKE

Returnerar TRUE om den första parametern matchar det angivna mönstret utan att skilja på versaler och gemener. LIKE kan användas istället för ILIKE för att göra matchningen skiftlägeskänslig. Fungerar även med siffror.

Syntax	sträng/tal ILIKE mönster
Argument	• **sträng/siffra** - sträng att söka i • **pattern** - mönster att hitta, du kan använda ”%” som jokertecken, ”_” som enstaka tecken och ”\\” för att undkomma dessa specialtecken.
Exempel	• `'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

Returnerar TRUE om värdet hittas i en lista med värden.

Syntax	a IN b
Argument	• **a** - värde • **b** - lista över värden
Exempel	• `'A' IN ('A','B')` → TRUE • `'A' IN ('C','B')` → FALSE

IS

Returnerar TRUE om a är samma sak som b.

Syntax	a IS b
Argument	• **a** - valfritt värde • **b** - valfritt värde
Exempel	• `'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, om din geometri inte är NULL

ÄR INTE

Returnerar TRUE om a inte är samma sak som b.

Syntax	a ÄR INTE b
Argument	• **a** - värde • **b** - värde
Exempel	• `'a' IS NOT 'b'` → TRUE • `'a' IS NOT 'a'` → FALSE • `4 IS NOT 2+2` → FALSE

LIKAMED

Returnerar TRUE om den första parametern matchar det angivna mönstret. Fungerar även med siffror.

Syntax	sträng/siffra LIKE-mönster
Argument	• **sträng/siffra** - värde • **pattern** - mönster att jämföra värdet med, du kan använda ”%” som jokertecken, ”_” som enstaka tecken och ”\\” för att undkomma dessa specialtecken.
Exempel	• `'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

ICKE

Negerar ett villkor.

Syntax	INTE en
Argument	• **a** - skick
Exempel	• `INTE 1` → FALSK • `INTE 0` → SANT

NOT BETWEEN

Returnerar TRUE om värdet inte ligger inom det angivna intervallet. Intervallet anses inkludera gränserna.

Syntax	värde INTE MELLAN lägre_bound OCH högre_bound
Argument	• **value** - det värde som ska jämföras med ett intervall. Det kan vara en sträng, ett tal eller ett datum. • **lägre_gräns OCH högre_gräns** - gränser för intervall
Exempel	• `'B' NOT BETWEEN 'A' AND 'C'` → FALSE • `1.0 INTE MELLAN 1.1 OCH 1.2` → TRUE • `2 INTE MELLAN 2 OCH 3` → FALSE • `'B' NOT BETWEEN 'a' AND 'c'` → TRUE • `lower('B') INTE MELLAN 'a' OCH 'b'` → FALSE

Observera

value NOT BETWEEN lower_bound AND higher_bound är samma sak som ”*value* < *lower_bound OR value* > *higher_bound*”.

Ytterligare läsning: *BETWEEN*

ELLER

Returnerar TRUE när villkor a eller b är sant.

Syntax	a ELLER b
Argument	• **a** - skick • **b** - skick
Exempel	• `4 = 2+2 OR 1 = 1` → TRUE • `4 = 2+2 OR 1 = 2` → TRUE • `4 = 2 OR 1 = 2` → FALSE

[]

Index-operator. Returnerar ett element från en array eller ett map-värde.

Syntax	[index]
Argument	• **index** - index för array eller nyckelvärde i karta
Exempel	• `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

Ytterligare läsning: *array_get*, *map_get*

^

Makt av två värden.

Syntax	a ^ b
Argument	• **a** - värde • **b** - värde
Exempel	• `5 ^ 4` → 625 • `5 ^ NULL` → NULL

||

Sammanfogar två värden till en sträng.

Om ett av värdena är NULL blir resultatet NULL. Se CONCAT-funktionen för ett annat beteende.

Syntax	a || b
Argument	• **a** - värde • **b** - värde
Exempel	• `` `'Här' || ' och ' || 'där' `` → 'Här och där' • `` `Inget' || NULL `` → NULL • `'Dia: ' || "Diameter"` → 'Dia: 25' • `1 || 2` → '12'

För vidare läsning: *concat*, *+*

~

Utför en matchning med reguljära uttryck på ett strängvärde. Backslash-tecken måste dubbeleskapsas (t.ex. ”\\s” för att matcha ett tecken med vitt utrymme).

Syntax	sträng ~ regex
Argument	• **sträng** - Ett strängvärde • **regex** - Ett reguljärt uttryck. Bindestreck måste escapas, t.ex. \\d.
Exempel	• `'hello' ~ 'll'` → TRUE • `'hello' ~ '^ll'` → FALSE • `'hello' ~ 'llo$'` → TRUE • `'abc123' ~ '\\d+'` → TRUE

Ytterligare läsning: *regexp_match*

9.2.20 Bearbetningsfunktioner

Den här gruppen innehåller funktioner som arbetar med bearbetningsalgoritmer.

parameter

Returnerar värdet på en inmatningsparameter för en bearbetningsalgoritm.

Syntax	parameter(namn)
Argument	• **namn** - namn på motsvarande inmatningsparameter
Exempel	• `parameter('BUFFER_SIZE')` → 5.6

9.2.21 Rasterfunktioner

Denna grupp innehåller funktioner för att arbeta med rasterlager.

raster_attributes

Returnerar en karta med fältnamnen som nycklar och rasterattributtabellens värden som värden från den attributtabellspost som matchar det angivna rastervärdet.

Syntax	raster_attributes(layer, band, value)
Argument	• **layer** - namn eller id för ett rasterlager • **band** - bandnumret för den associerade attributtabelluppslagningen. • **värde** - rastervärde
Exempel	• `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_statistik

Returnerar statistik från ett rasterlager.

Syntax	raster_statistic(layer, band, property)
Argument	• **layer** - en sträng som representerar antingen ett rasterlagernamn eller ett lager-ID • **band** - heltal som representerar bandnumret från rasterlagret, med början på 1 • **property** - en sträng som motsvarar den egenskap som ska returneras. Giltiga alternativ är: – min: lägsta värde – max: maximalt värde – avg: genomsnittligt värde (medelvärde) – stdev: standardavvikelse för värden – intervall: värdeintervall (max - min) – summa: summan av alla värden från raster
Exempel	• `raster_statistic('lc',1,'avg')` → Medelvärde från band 1 från rasterlagret 'lc • `raster_statistic('ac2010',3,'min')` → Minsta värde från band 3 från rasterlagret "ac2010

raster_value

Returnerar det rastervärde som finns vid den angivna punkten.

Syntax	raster_value(layer, band, point)
Argument	• **layer** - namn eller id för ett rasterlager • **band** - det bandnummer som värdet ska hämtas från. • **point** - punktgeometri (för flerdelade geometrier som har mer än en del returneras ett NULL-värde)
Exempel	• `raster_value('dem', 1, make_point(1,1))` → 25 • `raster_value('ndvi', 2, @layer_cursor_point)` → 25

9.2.22 Funktioner för registrering och attribut

Denna grupp innehåller funktioner som arbetar med identifikatorer för poster.

attribute

Returnerar ett attribut från en funktion.

Variant 1

Returnerar värdet för ett attribut från den aktuella funktionen.

Syntax	attribute(attribute_name)
Argument	• **attributnamn** - namn på det attribut som ska returneras
Exempel	• `attribute( 'name' )` → värde lagrat i attributet 'name' för den aktuella funktionen

Variant 2

Gör det möjligt att ange målfunktionens och attributets namn.

Syntax	attribute(feature, attribute_name)
Argument	• **feature** - en funktion • **attributnamn** - namn på det attribut som ska returneras
Exempel	• `attribut( @atlas_feature, 'name' )` → värde lagrat i attributet 'name' för den aktuella atlasfunktionen

attributes

Returnerar en karta som innehåller alla attribut från en funktion, med fältnamn som kartnycklar.

Variant 1

Returnerar en karta med alla attribut från den aktuella funktionen.

Syntax	attributes()
Exempel	• `attributes()['name']` → värde lagrat i attributet 'name' för den aktuella funktionen

Variant 2

Gör det möjligt att ange målfunktionen.

Syntax	attributes(feature)
Argument	• **feature** - en funktion
Exempel	• `attributes( @atlas_feature )['name']` → värde lagrat i attributet ”name” för den aktuella atlasfunktionen

Ytterligare läsning: *Kartor Funktioner*

$currentfeature

Returnerar den aktuella funktionen som utvärderas. Detta kan användas med funktionen ’attribute’ för att utvärdera attributvärden från den aktuella funktionen. **VARNING: Denna funktion är föråldrad. Vi rekommenderar att du använder ersättningsvariabeln @feature i stället.**

Syntax	$currentfeature
Exempel	• `attribute( $currentfeature, 'name' )` → värde lagrat i attributet ’name’ för den aktuella funktionen

display_expression

Returnerar visningsuttrycket för en given funktion i ett lager. Uttrycket utvärderas som standard. Kan användas med noll, ett eller flera argument, se nedan för mer information.

Inga parametrar

Om funktionen anropas utan parametrar utvärderar den displayuttrycket för den aktuella funktionen i det aktuella lagret.

Syntax	display_expression()
Exempel	• `display_expression()` → Visningsuttrycket för den aktuella funktionen i det aktuella lagret.

En parameter för ”funktion”

Om funktionen endast anropas med parametern ”feature” utvärderas den angivna egenskapen från det aktuella lagret.

Syntax	display_expression(feature)
Argument	• **feature** - Den funktion som ska utvärderas.
Exempel	• `display_expression(@atlas_feature)` → Visningsuttrycket för den aktuella atlasfunktionen.

Parametrar för lager och funktioner

Om funktionen anropas med både ett lager och en funktion, kommer den att utvärdera den angivna funktionen från det angivna lagret.

Syntax	display_expression(lager, funktion, [evaluate=true]) [] markerar valfria argument
Argument	• **layer** - Lagret (eller dess ID eller namn) • **feature** - Den funktion som ska utvärderas. • **evaluate** - Om uttrycket måste utvärderas. Om false returneras uttrycket endast som en strängbokstav (som eventuellt kan utvärderas senare med hjälp av funktionen ”eval”).
Exempel	• `display_expression( 'streets', get_feature_by_id('streets', 1))` → Visningsuttrycket för funktionen med ID 1 i lagret ’streets’. • `display_expression('a_layer_id', @feature, 'False')` → Displayuttrycket för den angivna funktionen utvärderades inte.

feature_id

Returnerar ett unikt ID för en funktion, eller NULL om funktionen inte är giltig.

Syntax	feature_id(feature)
Argument	• **feature** - ett funktionsobjekt
Exempel	• `feature_id( @feature )` → ID för den aktuella funktionen

Ytterligare läsning: *get_feature_by_id*

get_feature

Returnerar den första egenskapen i ett lager som matchar ett givet attributvärde.

Variant med ett enda värde

Tillsammans med lagrets ID anges en enda kolumn och ett värde.

Syntax	get_feature(layer, attribute, value)
Argument	• **layer** - namn eller ID för lagret • **attribut** - attributnamn som ska användas för matchningen • **värde** - attributvärde som ska matchas
Exempel	• `get_feature('streets','name','main st')` → första objektet som hittas i lagret ”streets” med värdet ”main st” i fältet ”name”

Kartvariant

Tillsammans med lagrets ID, en karta som innehåller de kolumner (nyckel) och deras respektive värde som ska användas.

Syntax	get_feature(layer, attribute)
Argument	• **layer** - namn eller ID för lagret • **attribute** - Map som innehåller de kolumn- och värdepar som ska användas
Exempel	• `get_feature('streets',map('name','main st','lane_num','4'))` → första funktionen som hittades i lagret ”streets” med värdet ”main st” i fältet ”name” och värdet ”4” i fältet ”lane_num”

get_feature_by_id

Returnerar objektet med ett id på ett lager.

Syntax	get_feature_by_id(layer, feature_id)
Argument	• **layer** - lager, lagernamn eller lager-ID • **feature_id** - id för den funktion som ska returneras
Exempel	• `get_feature_by_id('streets', 1)` → funktionen med id 1 på lagret ”streets”

Ytterligare läsning: *feature_id*

$id

Returnerar funktions-ID för den aktuella raden. **VARNING: Denna funktion är föråldrad. Vi rekommenderar att du använder ersättningsvariabeln @id i stället.**

Syntax	$id
Exempel	• `$id` → 42

Ytterligare läsning: *feature_id*, *get_feature_by_id*

is_attribute_valid

Returnerar TRUE om ett specifikt funktionsattribut uppfyller alla begränsningar.

Syntax	is_attribute_valid(attribute, [feature], [layer], [strength]) [] markerar valfria argument
Argument	• **attribut** - ett attributnamn • **feature** - En funktion. Om den inte anges kommer den aktuella funktionen att användas. • **layer** - Ett vektorlager. Om det inte anges kommer det aktuella lagret att användas. • **strength** - Ställs in på ”hard” eller ”soft” för att begränsa till en specifik begränsningstyp. Om den inte är inställd returnerar funktionen FALSE om antingen en hård eller en mjuk begränsning misslyckas.
Exempel	• `is_attribute_valid('HECTARES')` → TRUE om den aktuella egenskapens värde i fältet ”HECTARES” uppfyller alla begränsningar. • `is_attribute_valid('HOUSES',get_feature('my_layer', 'FID', 10), 'my_layer')` → FALSE om värdet i fältet ”HOUSES” från funktionen med ”FID”=10 i ’my_layer’ inte uppfyller alla begränsningar.

Ytterligare läsning: *Begränsningar*

is_feature_valid

Returnerar TRUE om en funktion uppfyller alla fältbegränsningar.

Syntax	is_feature_valid([feature], [layer], [strength]) [] markerar valfria argument
Argument	• **feature** - En funktion. Om den inte anges kommer den aktuella funktionen att användas. • **layer** - Ett vektorlager. Om det inte anges kommer det aktuella lagret att användas. • **strength** - Ställs in på ”hard” eller ”soft” för att begränsa till en specifik begränsningstyp. Om den inte är inställd returnerar funktionen FALSE om antingen en hård eller en mjuk begränsning misslyckas.
Exempel	• `is_feature_valid(strength:='hard')` → TRUE om alla fält från den aktuella funktionen uppfyller sina hårda begränsningar. • `is_feature_valid(get_feature('my_layer', 'FID', 10), 'my_layer')` → FALSE om alla fält från funktionen med ”FID”=10 i ’my_layer’ inte uppfyller alla begränsningar.

Ytterligare läsning: *Begränsningar*

is_selected

Returnerar TRUE om en funktion är vald. Kan användas med noll, ett eller två argument, se nedan för mer information.

Inga parametrar

Om funktionen anropas utan parametrar returnerar den TRUE om den aktuella funktionen i det aktuella lagret är vald.

Syntax	is_selected()
Exempel	• `is_selected()` → TRUE om den aktuella funktionen i det aktuella lagret är vald.

En parameter för ”funktion”

Om funktionen endast anropas med en ”feature”-parameter returnerar den TRUE om den angivna funktionen från det aktuella lagret har valts.

Syntax	is_selected(feature)
Argument	• **feature** - Den funktion som ska kontrolleras för urval.
Exempel	• `is_selected(@atlas_feature)` → TRUE om den aktuella atlasfunktionen är vald. • `is_selected(get_feature('streets', 'name', 'Main St.'))` → TRUE om den unika funktionen med namnet ”Main St.” i det aktiva ”streets”-lagret är vald. • `is_selected(get_feature_by_id('streets', 1))` → TRUE om funktionen med id 1 på det aktiva lagret ”streets” är vald.

Två parametrar

Om funktionen anropas med både ett lager och en funktion returnerar den TRUE om den angivna funktionen från det angivna lagret har valts.

Syntax	is_selected(layer, feature)
Argument	• **layer** - Det lager (dess ID eller namn) som urvalet ska kontrolleras på. • **feature** - Den funktion som ska kontrolleras för urval.
Exempel	• `is_selected( 'streets', get_feature('streets', 'name', "street_name"))` → TRUE om den aktuella byggnadens gata är vald (förutsatt att byggnadslagret har ett fält med namnet ”street_name” och att lagret ”streets” har ett fält med namnet ”name” med unika värden). • `is_selected( 'streets', get_feature_by_id('streets', 1))` → TRUE om funktionen med id 1 på lagret ”streets” är vald.

maptip

Returnerar karttipset för en given funktion i ett lager. Uttrycket utvärderas som standard. Kan användas med noll, ett eller flera argument, se nedan för detaljer.

Inga parametrar

Om funktionen anropas utan parametrar utvärderar den karttipset för den aktuella funktionen i det aktuella lagret.

Syntax	maptip()
Exempel	• `maptip()` → Karttipset för den aktuella funktionen i det aktuella lagret.

En parameter för ”funktion”

Om funktionen endast anropas med parametern ”feature” utvärderas den angivna egenskapen från det aktuella lagret.

Syntax	maptip(feature)
Argument	• **feature** - Den funktion som ska utvärderas.
Exempel	• `maptip(@atlas_feature)` → Karttipset för den aktuella atlasfunktionen.

Parametrar för lager och funktioner

Om funktionen anropas med både ett lager och en funktion, kommer den att utvärdera den angivna funktionen från det angivna lagret.

Syntax	maptip(layer, feature, [evaluate=true]) [] markerar valfria argument
Argument	• **layer** - Lagret (eller dess ID eller namn) • **feature** - Den funktion som ska utvärderas. • **evaluate** - Om uttrycket måste utvärderas. Om false returneras uttrycket endast som en strängbokstav (som eventuellt kan utvärderas senare med hjälp av funktionen ”eval_template”).
Exempel	• `maptip('streets', get_feature_by_id('streets', 1))` → Karttipset för funktionen med ID 1 i lagret ”streets”. • `maptip('a_layer_id', @feature, 'False')` → Karttipset för den angivna funktionen utvärderades inte.

num_selected

Returnerar antalet valda funktioner på ett visst lager. Arbetar som standard på det lager som uttrycket utvärderas på.

Syntax	num_selected([lager=aktuellt lager]) [] markerar valfria argument
Argument	• **layer** - Det lager (eller dess id eller namn) som urvalet ska kontrolleras på.
Exempel	• `num_selected()` → Antalet valda funktioner i det aktuella lagret. • `num_selected('streets')` → Antalet utvalda funktioner i lagret streets

represent_attributes

Returnerar en karta med attributnamnen som nycklar och de konfigurerade representationsvärdena som värden. Representationsvärdet för attributen beror på den konfigurerade widget-typen för varje attribut. Kan användas med noll, ett eller flera argument, se nedan för mer information.

Inga parametrar

Om funktionen anropas utan parametrar returnerar den representationen av attributen för den aktuella funktionen i det aktuella lagret.

Syntax	represent_attributes()
Exempel	• `represent_attributes()` → Representationen av attributen för den aktuella funktionen.

En parameter för ”funktion”

Om funktionen endast anropas med en ”feature”-parameter returnerar den representationen av attributen för den angivna funktionen från det aktuella lagret.

Syntax	represent_attributes(feature)
Argument	• **feature** - Den funktion som ska utvärderas.
Exempel	• `represent_attributes(@atlas_feature)` → Representationen av attributen för den angivna funktionen från det aktuella lagret.

Parametrar för lager och funktioner

Om funktionen anropas med en ”layer”- och en ”feature”-parameter returnerar den representationen av attributen för den angivna funktionen från det angivna lagret.

Syntax	represent_attributes(layer, feature)
Argument	• **layer** - Lagret (eller dess ID eller namn). • **feature** - Den funktion som ska utvärderas.
Exempel	• `represent_attributes('atlas_layer', @atlas_feature)` → → Representationen av attributen för den angivna funktionen från det angivna lagret.

Ytterligare läsning: *represent_value*

represent_value

Returnerar det konfigurerade representationsvärdet för ett fältvärde. Det beror på den konfigurerade widget-typen. Detta är ofta användbart för widgetar av typen ”Value Map”.

Syntax	represent_value(value, [fieldName]) [] markerar valfria argument
Argument	• **value** - Det värde som ska lösas. Mest troligt ett fält. • **fieldName** - Det fältnamn för vilket widgetkonfigurationen ska laddas.
Exempel	• `represent_value("field_with_value_map")` → Beskrivning för värde • `represent_value('statiskt värde', 'fältnamn')` → Beskrivning för statiskt värde

Ytterligare läsning: *widget-typer*, *represent_attributes*

sqlite_fetch_and_increment

Hantera automatiskt ökande värden i sqlite-databaser.

SQlite-standardvärden kan endast tillämpas vid infogning och inte förskjutas.

Detta gör det omöjligt att skaffa en inkrementerad primärnyckel via AUTO_INCREMENT innan raden skapas i databasen. Sidenote: med postgres fungerar detta via alternativet *evaluate default values*.

När man lägger till nya funktioner med relationer är det väldigt trevligt att redan kunna lägga till barn för en förälder, medan förälderns formulär fortfarande är öppet och förälderns funktion därmed inte är bekräftad.

För att komma runt denna begränsning kan denna funktion användas för att hantera sekvensvärden i en separat tabell i sqlite-baserade format som gpkg.

Sekvenstabellen filtreras efter ett sekvens-ID (filter_attribute och filter_value) och det aktuella värdet för id_field ökas med 1 och det ökade värdet returneras.

Om ytterligare kolumner kräver att värden anges kan default_values-kartan användas för detta ändamål.

Anmärkning

Denna funktion modifierar målets sqlite-tabell. Den är avsedd att användas med standardvärdeskonfigurationer för attribut.

När databasparametern är ett lager och lagret är i transaktionsläge hämtas värdet bara en gång under transaktionens livstid och cachelagras och inkrementeras. Detta gör det osäkert att arbeta med samma databas från flera parallella processer.

Syntax	sqlite_fetch_and_increment(databas, tabell, id_fält, filter_attribut, filter_värde, [standardvärden]) [] markerar valfria argument
Argument	• **database** - Sökväg till sqlite-filen eller geopackage-lagret • **table** - Namn på den tabell som hanterar sekvenserna • **id_field** - Namnet på det fält som innehåller det aktuella värdet • **filter_attribute** - Namnge det fält som innehåller en unik identifierare för denna sekvens. Måste ha ett UNIQUE-index. • **filter_value** - Namn på den sekvens som ska användas. • **default_values** - Karta med standardvärden för ytterligare kolumner i tabellen. Värdena måste vara fullständigt citerade. Funktioner är tillåtna.
Exempel	• `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`

Ytterligare läsning: *Egenskaper för datakällor*, *Inställning av relationer mellan flera lager*

uuid

Genererar en universellt unik identifierare (UUID) för varje rad med hjälp av Qt-metoden QUuid::createUuid.

Syntax	uuid([format='WithBraces']) [] markerar valfria argument
Argument	• **format** - Det format som UUID kommer att formateras i. "WithBraces", "WithoutBraces" eller "Id128".
Exempel	• `uuid()` → '{0bd2f60f-f157-4a6d-96af-d4ba4cb366a1}' • `uuid('WithoutBraces')` → '0bd2f60f-f157-4a6d-96af-d4ba4cb366a1' • `uuid('Id128')` → '0bd2f60ff1574a6d96afd4ba4cb366a1'

9.2.23 Relationer

Den här gruppen innehåller en lista över de *relationer* som finns tillgängliga i det aktuella projektet, med beskrivning. Den ger snabb åtkomst till relations-ID:t för att skriva ett uttryck (med t.ex. funktionen *relation_aggregate*) eller anpassa ett formulär.

9.2.24 Sensorer Funktioner

Denna grupp innehåller funktioner för att interagera med sensorer.

sensor_data

Returnerar det senast insamlade värdet (eller värden som en karta för sensorer som rapporterar flera värden) från en registrerad sensor.

Syntax	sensor_data(name, [expiration]) [] markerar valfria argument
Argument	• **namn** - sensorns namn • **expiration** - högsta tillåtna millisekund sedan det senast registrerade värdet
Exempel	• `sensor_data('geiger_1')` → '2000'

9.2.25 Strängfunktioner

Den här gruppen innehåller funktioner som arbetar med strängar (t.ex. som ersätter, konverterar till versaler).

ascii

Returnerar den unicode-kod som är associerad med det första tecknet i en sträng.

Syntax	ascii(string)
Argument	• **sträng** - den sträng som ska konverteras till Unicode-kod
Exempel	• `ascii('Q')` → 81

char

Returnerar tecknet som är associerat med en unicode-kod.

Syntax	char(code)
Argument	• **code** - ett unicode-kodnummer
Exempel	• `char(81)` → 'Q'

concat

Sammanfogar flera strängar till en. NULL-värden konverteras till tomma strängar. Andra värden (t.ex. siffror) konverteras till strängar.

Syntax	concat(sträng1, sträng2, …)
Argument	• **sträng** - ett strängvärde
Exempel	• `concat('sun', 'set')` → 'solnedgång' • `concat('a','b','c','d','e')` → 'abcde' • `concat('Anno ', 1984)` → 'Anno 1984' • `concat('The Wall', NULL)` → 'The Wall'

Om sammanlänkning av fält

Du kan också sammanfoga strängar eller fältvärden med hjälp av antingen `||` eller + operatorer, med vissa speciella egenskaper:

- Operatorn + betyder också summera uttryck, så om du har ett heltal (fält eller numeriskt värde) operand, kan detta vara felaktigt och du använder bättre de andra:

```
'My feature id is: ' + "gid" => triggers an error as gid returns an integer
```

- När något av argumenten är ett NULL-värde, kommer antingen `||` eller + att returnera ett NULL-värde. För att returnera de andra argumenten oavsett NULL-värdet kan du använda funktionen `concat` eller `array_to_string`:

```
'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: '
```

vidare läsning: *array_to_string*, ||, +

format

Formatera en sträng med hjälp av angivna argument.

Syntax	format(sträng, arg1, arg2, …)
Argument	• **sträng** - En sträng med platshållarna %1, %2 osv. för argumenten. Platshållarna kan upprepas. Den lägst numrerade platshållaren ersätts av arg1, nästa av arg2, osv. • **arg** - valfri typ. Valfritt antal argument.
Exempel	• `format('Detta %1 är %2','är', 'test')` → 'Detta är ett test' • `format('Detta är %2','lite oväntat men 2 är det lägsta talet i strängen','normal')` → 'Detta är lite oväntat men 2 är det lägsta talet i strängen'

format_date

Formaterar en datatyp eller sträng till ett anpassat strängformat. Använder Qt datum/tid formatsträngar. Se QDateTime::toString.

Syntax	format_date(datetime, format, [language]) [] markerar valfria argument
Argument	• **datetime** - datum, tid eller datetime-värde • **format** - Strängmall som används för att formatera strängen.

Uttryck	Utmatning
d	dagen som ett nummer utan inledande nolla (1 till 31)
dd	dagen som ett nummer med en inledande nolla (01 till 31)
ddd	det förkortade lokala dagsnamnet (t.ex. ”mån” till ”sön”)
dddd	det långa lokaliserade dagsnamnet (t.ex. ”måndag” till ”söndag”)
M	månaden som ett nummer utan inledande nolla (1-12)
MM	månaden som ett nummer med en inledande nolla (01-12)
MMM	det förkortade lokala månadsnamnet (t.ex. ”Jan” till ”Dec”)
MMMM	det långa lokaliserade månadsnamnet (t.ex. ”januari” till ”december”)
yy	årtalet som tvåsiffrigt nummer (00-99)
yyyy	året som ett fyrsiffrigt tal

Dessa uttryck kan användas för tidsdelen av formatsträngen:

Uttryck	Utmatning
h	timmen utan inledande nolla (0 till 23 eller 1 till 12 om AM/PM-visning)
hh	timmen med en inledande nolla (00 till 23 eller 01 till 12 om AM/PM-visning)
H	timmen utan inledande nolla (0 till 23, även med AM/PM-visning)
HH	timmen med en inledande nolla (00 till 23, även med AM/PM-visning)
m	minuten utan inledande nolla (0 till 59)
mm	minuten med en inledande nolla (00 till 59)
s	den andra siffran utan inledande nolla (0 till 59)
ss	den andra siffran med en inledande nolla (00 till 59)
z	millisekunder utan nollor efter (0 till 999)
zzz	millisekunder med nollor efter (000 till 999)
AP eller A	tolkas som en AM/PM-tid. *AP* måste vara antingen ’AM’ eller ’PM’.
ap eller en	Tolkas som en AM/PM-tid. *ap* måste vara antingen ’am’ eller ’pm’.

- **language** - språk (gemener, två eller tre bokstäver, ISO 639 språkkod) som används för att formatera datumet till en anpassad sträng. Som standard används den aktuella QGIS-användarens språk.

Exempel

- `format_date('2012-05-15','dd.MM.åååå')` → ’15.05.2012’
- `format_date('2012-05-15','d MMMM åååå','fr')` → ’15 maj 2012’
- `format_date('2012-05-15','dddd')` → ”Tuesday”, om den aktuella locale är en engelsk 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

Returnerar ett tal formaterat med den lokala separatorn för tusentals. Som standard används den aktuella QGIS-användarens locale. Decimalerna trunkeras också till det antal decimaler som anges.

Syntax	format_number(number, [places=0], [language], [omit_group_separators=false], [trim_trailing_zeroes=false]) [] markerar valfria argument
Argument	• **number** - nummer som ska formateras • **places** - heltal som representerar antalet decimaler som strängen ska trunkeras till. • **language** - språk (gemener, två eller tre bokstäver, ISO 639 språkkod) som används för att formatera talet till en sträng. Som standard används den aktuella QGIS-användarens språk. • **omit_group_separators** - om inställningen är true kommer gruppavgränsare inte att inkluderas i strängen • **trim_trailing_zeroes** - om värdet är true kommer nollor efter decimaltecknet att tas bort från strängen
Exempel	• `format_number(10000000.332,2)` → '10,000,000.33' om t.ex. den aktuella locale är en engelsk variant • `format_nummer(10000000.332,2,'fr')` → '10 000 000,33'

left

Returnerar en delsträng som innehåller de *n* tecken som ligger längst till vänster i strängen.

Syntax	left(string, length)
Argument	• **sträng** - en sträng • **length** - heltal. Antalet tecken från vänster i strängen som ska returneras.
Exempel	• `vänster('Hello World',5)` → 'Hello'

Ytterligare läsning: *right*

length

Returnerar antalet tecken i en sträng eller längden på en geometrisk linjestring.

String variant

Returnerar antalet tecken i en sträng.

Syntax	length(string)
Argument	• **sträng** - sträng att räkna längden på
Exempel	• `längd('hello')` → 5

Geometrivariant

Beräknar längden på ett geometriskt linjeobjekt. Beräkningar är alltid planimetriska i det spatiala referenssystemet (SRS) för denna geometri, och enheterna för den returnerade längden kommer att matcha enheterna för SRS. Detta skiljer sig från de beräkningar som utförs av funktionen $length, som utför ellipsoidiska beräkningar baserat på projektets inställningar för ellipsoid och avståndsenheter.

Syntax	length(geometry)
Argument	• **geometry** - objekt för linjegeometri
Exempel	• `längd(geom_from_wkt('LINESTRING(0 0, 4 0)'))` → 4,0

lower

Konverterar en sträng till små bokstäver.

Syntax	lower(string)
Argument	• **sträng** - strängen som ska konverteras till gemener
Exempel	• `lower('HELLO World')` → 'hej värld'

Ytterligare läsning: *upper*, *title*

lpad

Returnerar en sträng som är utfylld till vänster till den angivna bredden, med ett fyllnadstecken. Om målbredden är mindre än strängens längd avkortas strängen.

Syntax	lpad(string, width, fill)
Argument	• **sträng** - sträng att fylla i • **bredd** - längden på den nya strängen • **fill** - tecken att fylla ut det återstående utrymmet med
Exempel	• `lpad('Hello', 10, 'x')` → 'xxxxxHello' • `lpad('Hello', 3, 'x')` → 'Hel'

Ytterligare läsning: *rpad*

ltrim

Tar bort den längsta strängen som bara innehåller de angivna tecknen (ett mellanslag som standard) från början av strängen.

Syntax	ltrim(string, [characters=' ']) [] markerar valfria argument
Argument	• **sträng** - sträng som ska trimmas • **tecken** - tecken som ska trimmas
Exempel	• `ltrim(' hello world ')` → 'hello world ' • `ltrim('zzzytest', 'xyz')` → 'test'

Ytterligare läsning: *rtrim*, *trim*

regexp_match

Returnerar den första matchande positionen som matchar ett reguljärt uttryck i en unikodsträng, eller 0 om delsträngen inte hittas.

Syntax	regexp_match(input_string, regex)
Argument	• **input_string** - den sträng som ska testas mot det reguljära uttrycket • **regex** - Det reguljära uttryck som ska testas mot. Backslash-tecken måste dubbeleskapsas (t.ex. "\\s" för att matcha ett tecken med vitt utrymme eller "\\b" för att matcha en ordgräns).
Exempel	• `regexp_match('QGIS ROCKS','\\sROCKS')` → 5 • `regexp_match('Budač','udač\\b')` → 2

Ytterligare läsning: *strpos*

regexp_replace

Returnerar en sträng med det angivna reguljära uttrycket ersatt.

Syntax	regexp_replace(input_string, regex, replacement)
Argument	• **input_string** - strängen som ska ersätta matchningar i • **regex** - Det reguljära uttryck som ska ersättas. Backslash-tecken måste dubbeleskapsas (t.ex. "\\s" för att matcha ett tecken med vitt utrymme). • **replacement** - Den sträng som ska ersätta alla matchande förekomster av det reguljära uttryck som anges. Fångade grupper kan infogas i ersättningssträngen med hjälp av001, 002, etc.
Exempel	• `regexp_replace('QGIS SKA ROCKERA','\\sSHOULD\\s',' DOES ')` → 'QGIS DOES ROCKERA' • `regexp_replace('ABC123','\\d+','')` → 'ABC' • `regexp_replace('mitt namn är John','(.*) är (.*)','\\2 är \\1')` → 'John är mitt namn'

Ytterligare läsning: *replace*, *array_replace*

regexp_substr

Returnerar den del av en sträng som matchar ett angivet reguljärt uttryck.

Syntax	regexp_substr(input_string, regex)
Argument	• **input_string** - den sträng som ska sökas efter matchningar i • **regex** - Det reguljära uttryck som ska matchas mot. Backslash-tecken måste dubbeleskapsas (t.ex. ”\\s” för att matcha ett tecken med vitt utrymme).
Exempel	• `regexp_substr('abc123','(\\d+)')` → ’123’

Ytterligare läsning: *substr*, *regexp_matches*

replace

Returnerar en sträng med den angivna strängen, arrayen eller kartan med strängar ersatt.

String & array-variant

Returnerar en sträng med den angivna strängen eller strängarray ersatt av en sträng eller en strängarray.

Syntax	replace(string, before, after)
Argument	• **sträng** - inmatningssträngen • **before** - den sträng eller array av strängar som ska ersättas • **after** - den sträng eller array av strängar som ska användas som ersättning
Exempel	• `replace('QGIS SKA ROCKERA','SKA','DOES')` → ’QGIS DOES ROCKERA’ • `replace('QGIS ABC',array('A','B','C'),array('X','Y','Z'))` → ’QGIS XYZ’ • `replace('QGIS',array('Q','S'),'')` → ’GI’

Kartvariant

Returnerar en sträng med de angivna map-nycklarna ersatta av parade värden. Längre map-nycklar utvärderas först.

Syntax	replace(string, map)
Argument	• **sträng** - inmatningssträngen • **map** - kartan som innehåller nycklar och värden
Exempel	• `replace('APP SKULLE ROCKERA',map('APP','QGIS','SKULLE','GÖR'))` → ’QGIS GÖR ROCKERA’ • `replace('fyrtiotvå',map('för','4','två','2','fyrtiotvå','42'))` → ’42’

Ytterligare läsning: *regexp_replace*, *array_replace*

right

Returnerar en delsträng som innehåller de *n* tecken som ligger längst till höger i strängen.

Syntax	right(string, length)
Argument	• **sträng** - en sträng • **length** - heltal. Antalet tecken från höger i strängen som ska returneras.
Exempel	• `rätt('Hello World',5)` → 'World'

Ytterligare läsning: *left*

rpad

Returnerar en sträng som är utfylld till höger till den angivna bredden med hjälp av ett fyllnadstecken. Om målbredden är mindre än strängens längd, avkortas strängen.

Syntax	rpad(string, width, fill)
Argument	• **sträng** - sträng att fylla i • **bredd** - längden på den nya strängen • **fill** - tecken att fylla ut det återstående utrymmet med
Exempel	• `rpad('Hello', 10, 'x')` → 'Hellooxxxxx' • `rpad('Hello', 3, 'x')` → 'Hel'

För vidare läsning: *lpad*

rtrim

Tar bort den längsta strängen som bara innehåller de angivna tecknen (ett mellanslag som standard) från slutet av strängen.

Syntax	rtrim(string, [characters=' ']) [] markerar valfria argument
Argument	• **sträng** - sträng som ska trimmas • **tecken** - tecken som ska trimmas
Exempel	• `rtrim(' hello world ')` → ' hello world' • `rtrim('testxxzx', 'xyz')` → 'test'

Ytterligare läsning: *ltrim*, *trim*

strpos

Returnerar den första matchande positionen för en delsträng i en annan sträng, eller 0 om delsträngen inte hittas.

Syntax	strpos(haystack, needle)
Argument	• **haystack** - sträng som ska genomsökas • **nål** - sträng att söka efter
Exempel	• `strpos('HELLO WORLD','WORLD')` → 7 • `strpos('HELLO WORLD','GOODBYE')` → 0

Ytterligare läsning: *regexp_match*

substr

Returnerar en del av en sträng.

Syntax	substr(string, start, [length]) [] markerar valfria argument
Argument	• **sträng** - den fullständiga inmatningssträngen • **start** - heltal som representerar startpositionen för att extrahera med början på 1; om start är negativ kommer returnsträngen att börja i slutet av strängen minus startvärdet • **length** - heltal som representerar längden på den sträng som ska extraheras; om length är negativ kommer returnsträngen att utelämna den angivna längden tecken från slutet av strängen
Exempel	• `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'

Ytterligare läsning: *regexp_substr*, *regexp_matches*

title

Konverterar alla ord i en sträng till title case (alla ord gemener med inledande versal).

Syntax	title(string)
Argument	• **sträng** - strängen som ska konverteras till titelsak
Exempel	• `title('hello WOrld')` → 'Hello World'

Ytterligare läsning: *lower*, *upper*

to_string

Konverterar ett tal till en sträng.

Syntax	to_string(number)
Argument	• **number** - Heltal eller verkligt värde. Det tal som ska konverteras till sträng.
Exempel	• `to_string(123)` → '123'

Ytterligare läsning: *format_number*

trim

Tar bort alla inledande och avslutande blanksteg (mellanslag, tabbar etc.) från en sträng.

Syntax	trim(string)
Argument	• **sträng** - sträng som ska trimmas
Exempel	• `trim(' hello world ')` → 'hello world'

Ytterligare läsning: *ltrim*, *rtrim*

upper

Konverterar en sträng till stora bokstäver.

Syntax	upper(string)
Argument	• **sträng** - strängen som ska konverteras till versaler
Exempel	• `upper('hello WOrld')` → 'HELLO WORLD'

För vidare läsning: *lower*, *title*

wordwrap

Returnerar en sträng som är omslagen till ett maximalt/minimalt antal tecken.

Syntax	wordwrap(sträng, wrap_längd, [avgränsare_sträng]) [] markerar valfria argument
Argument	• **sträng** - den sträng som ska omslutas • **wrap_length** - ett heltal. Om wrap_length är positivt representerar talet det ideala maximala antalet tecken som ska brytas; om det är negativt representerar talet det minsta antalet tecken som ska brytas. • **delimiter_string** - Valfri avgränsningssträng som ska brytas till en ny rad.
Exempel	• `wordwrap('UNIVERSITY OF QGIS',13)` → 'UNIVERSITY OF QGIS' • `wordwrap('UNIVERSITY OF QGIS',-3)` → 'UNIVERSITY OF QGIS'

9.2.26 Användaruttryck

Den här gruppen innehåller de uttryck som sparats som *user expressions*.

9.2.27 Variabler

Den här gruppen innehåller dynamiska variabler som är relaterade till programmet, projektfilen och andra inställningar. Variablernas tillgänglighet beror på sammanhanget:

- från dialogrutan Välj efter uttryck
- från dialogrutan Fältkalkylator
- från dialogrutan för lageregenskaper
- från trycklayouten

Om du vill använda dessa variabler i ett uttryck måste de föregås av tecknet `@` (t.ex. `@row_number`).

Variable	Beskrivning
algorithm_id	Det unika ID:t för en algoritm
animation_end_time	Slut på animationens totala tidsintervall (som ett datetime-värde)
animation_interval	Varaktighet för animationens totala temporala tidsintervall (som ett intervallvärde)
animation_start_time	Början på animationens övergripande tidsintervall (som ett datetime-värde)
atlas_feature	Den aktuella atlasfunktionen (som funktionsobjekt)
atlas_featureid	Den aktuella atlasens funktion ID
atlas_featurenumber	Den aktuella atlasens funktionsnummer i layouten
atlas_filename	Aktuellt filnamn för atlasen
atlas_geometry	Den aktuella atlasens geometri
atlas_layerid	Atlasens aktuella täckningsskikts ID
atlas_layername	Det aktuella namnet på atlasens täckningslager
atlas_pagename	Den aktuella atlasens sidnamn
atlas_totalfeatures	Det totala antalet funktioner i atlas
band	Numret på bandet i rasterlagret
band_description	Beskrivningen av bandet i rasterlagret
band_name	Namnet på bandet i rasterlagret
canvas_cursor_point	Den senaste markörpositionen på duken i projektets geografiska koordinater
cluster_color	Färgen på symbolerna inom ett kluster, eller NULL om symbolerna har blandade färger
cluster_size	Antalet symboler som ingår i ett kluster
current_feature	Den funktion som för närvarande redigeras i attributformuläret eller tabellraden

continues on next page

Table 9.1 – fortsättning från föregående sida

Variable	Beskrivning
current_geometry	Geometrin för den funktion som för närvarande redigeras i formuläret eller tabellraden
current_parent_feature	representerar den funktion som för närvarande redigeras i det överordnade formuläret. Kan endast användas i en inbäddad formulärkontext.
current_parent_geometry	representerar geometrin för den funktion som för närvarande redigeras i det överordnade formuläret. Endast användbar i en inbäddad formulärkontext.
form_mode	Vad formuläret används för, t.ex. AddFeatureMode, SingleEditMode, MultiEditMode, SearchMode, AggregateSearchMode eller IdentifyMode som sträng.
feature	Den aktuella funktionen som utvärderas. Detta kan användas tillsammans med funktionen ”attribute” för att utvärdera attributvärden från den aktuella funktionen.
frame_duration	Tidsmässig varaktighet för varje animationsbild (som ett intervallvärde)
frame_number	Aktuellt bildnummer under uppspelning av animation
frame_rate	Antal bilder per sekund under uppspelning av animationer
fullextent_maxx	Maximalt x-värde från hela dukens utbredning (inklusive alla lager)
fullextent_maxy	Maximalt y-värde från hela dukens utsträckning (inklusive alla lager)
fullextent_minx	Minsta x-värde från hela dukens utsträckning (inklusive alla lager)
fullextent_miny	Minsta y-värde från hela dukens utsträckning (inklusive alla lager)
geometry	Geometrin för den aktuella funktionen som utvärderas
geometry_part_count	Antalet delar i geometrin för den återgivna funktionen
geometry_part_num	Det aktuella geometriska partnumret för objektet som återges
geometry_point_count	Antalet punkter i den renderade geometrins del
geometry_point_num	Det aktuella punktnumret i den renderade geometrins del
geometry_ring_num	Aktuellt geometriringnummer för objektet som renderas (endast för polygonobjekt). Den yttre ringen har ett värde på 0.
grid_axis	Aktuellt rutnäts annoteringsaxel (t.ex. ’x’ för longitud, ’y’ för latitud)
grid_number	Det aktuella värdet för rutnätets annotation
id	ID för den aktuella funktionen som utvärderas
item_id	Användar-ID för layoutobjektet (inte nödvändigtvis unikt)
item_uuid	Layoutobjektets unika ID
layer	Aktuella lagret
layer_crs	Det aktuella lagrets auktoritets-ID för koordinatsystemets referenssystem
layer_crs_ellipsoid	Det aktuella lagrets ellipsoid Authority ID CRS
layer_cursor_point	Punktgeometri under musens position i kartbilden (eller GetFeatureInfo-positionen i QGIS Server), i det aktiva lagrets CRS
layer_id	ID för aktuellt lager
layer_ids	ID:n för alla kartlager i det aktuella projektet i form av en lista
layer_name	Namnet på det aktuella lagret
layer_vertical_crs	Identifieraren för lagrets referenssystem för vertikala koordinater (t.ex. ”EPSG:5703”)
layer_vertical_crs_definition	Den fullständiga definitionen av lagrets vertikala referenssystem för koordinater
layer_vertical_crs_descriptio	Namnet på lagrets vertikala referenssystem för koordinater
layer_vertical_crs_wkt	WKT-definitionen av det vertikala koordinatreferenssystemet för det aktuella lagret
layers	Alla kartlager i det aktuella projektet som en lista
layout_dpi	Kompositionens upplösning (DPI)
layout_name	Layoutens namn
layout_numpages	Antalet sidor i layouten
layout_page	Sidnumret för det aktuella objektet i layouten
layout_pageheight	Den aktiva sidhöjden i layouten (i mm för standardpappersstorlekar, eller den enhet som användes för anpassade pappersstorlekar)
layout_pageoffsets	Array av Y-koordinater för toppen av varje sida. Gör det möjligt att dynamiskt placera objekt på sidor i ett sammanhang där sidstorleken kan ändras
layout_pagewidth	Den aktiva sidbredden i layouten (i mm för standardpappersstorlekar, eller den enhet som användes för anpassade pappersstorlekar)
legend_column_count	Antal kolumner i teckenförklaringen

continues on next page

Table 9.1 – fortsättning från föregående sida

Variable	Beskrivning
legend_filter_by_map	Anger om innehållet i teckenförklaringen filtreras av kartan
legend_filter_out_atlas	Anger om atlasen filtreras bort från teckenförklaringen
legend_split_layers	Anger om lager kan delas upp i teckenförklaringen
legend_title	Titeln på förklaringen
legend_wrap_string	Det eller de tecken som används för att omsluta legendtexten
map_crs	Den aktuella kartans referenssystem för koordinater
map_crs_acronym	Akronymen för den aktuella kartans koordinatreferenssystem
map_crs_definition	Den fullständiga definitionen av den aktuella kartans koordinatreferenssystem
map_crs_description	Namnet på den aktuella kartans referenssystem för koordinater
map_crs_ellipsoid	Akronymen för ellipsoiden i den aktuella kartans referenssystem för koordinater
map_crs_proj4	Proj4-definitionen av koordinatreferenssystemet för den aktuella kartan
map_crs_projection	Det beskrivande namnet på den projektionsmetod som används av kartans koordinatreferenssystem (t.ex. ”Albers Equal Area”)
map_crs_wkt	WKT-definitionen av den aktuella kartans referenssystem för koordinater
map_end_time	Slutet på kartans temporala tidsintervall (som ett datetime-värde)
map_extent	Den geometri som representerar kartans aktuella utbredning
map_extent_center	Punktfunktionen i mitten av kartan
map_extent_height	Kartans aktuella höjd
map_extent_width	Kartans aktuella bredd
map_id	ID för aktuell kartdestination. Detta kommer att vara ”canvas” för canvas-renderingar, och objekt-ID för layoutkartrenderingar
map_interval	Varaktigheten för kartans temporala tidsintervall (som ett intervallvärde)
map_layer_ids	Listan över kartskikts-ID:n som är synliga i kartan
map_layers	Listan över kartlager som är synliga i kartan
map_rotation	Kartans aktuella rotation
map_scale	Kartans aktuella skala
map_start_time	Början på kartans temporala tidsintervall (som ett datetime-värde)
map_units	Enheter för kartmätningar
map_z_range_lower	Lägsta höjden i kartans höjdintervall
map_z_range_upper	Övre höjden av kartans höjdintervall
model_path	Fullständig sökväg (inklusive filnamn) till den aktuella modellen (eller projektsökväg om modellen är inbäddad i ett projekt).
model_folder	Mapp som innehåller aktuell modell (eller projektmapp om modellen är inbäddad i ett projekt).
model_name	Namn på aktuell modell
model_group	Grupp för nuvarande modell
notification_message	Innehållet i det meddelande som skickas av leverantören (endast tillgängligt för åtgärder som utlöses av meddelanden från leverantören).
parent	Hänvisar till den aktuella funktionen i det överordnade lagret, vilket ger tillgång till dess attribut och geometri vid filtrering av en *aggregate*-funktion
plot_axis	Den associerade plotaxeln, t.ex. ”x” eller ”y”.
plot_axis_value	Det aktuella värdet för plotaxeln.
project_abstract	Projektets sammanfattning, hämtad från projektets metadata
project_area_units	Arealenheten för det aktuella projektet, används vid beräkning av geometrins area
project_author	Projektets författare, hämtat från projektets metadata
project_basename	Basnamnet på det aktuella projektets filnamn (utan sökväg och tillägg)
project_creation_date	Datum då projektet skapades, hämtat från projektets metadata
project_crs	Identifierare för projektets referenssystem för koordinater (t.ex. ”EPSG:4326”)
project_crs_arconym	Akronymen för projektets koordinatreferenssystem
project_crs_definition	Den fullständiga definitionen av projektets referenssystem för koordinater
project_crs_description	Beskrivning av projektets referenssystem för koordinater
project_crs_ellipsoid	Ellipsoiden för projektets koordinatreferenssystem
project_crs_proj4	Proj4-bilden av projektets koordinatreferenssystem
project_crs_wkt	WKT-representationen (välkänd text) av projektets koordinatreferenssystem

continues on next page

Table 9.1 – fortsättning från föregående sida

Variable	Beskrivning
project_distance_units	Avståndsenheten för det aktuella projektet, används vid beräkning av geometrilängder och avstånd
project_ellipsoid	Namnet på ellipsoiden för det aktuella projektet, som används vid beräkning av geodetiska områden eller geometrilängder
project_filename	Filnamnet för det aktuella projektet
project_folder	Mappen för det aktuella projektet
project_home	Hemvägen för det aktuella projektet
project_identifier	Projektets identifierare, hämtad från projektets metadata
project_keywords	Projektets nyckelord, hämtade från projektets metadata
project_last_saved	Datum/tid när projektet senast sparades.
project_path	Den fullständiga sökvägen (inklusive filnamn) till det aktuella projektet
project_title	Titeln på det aktuella projektet
project_units	Enheterna i projektets CRS
project_vertical_crs	Identifierare för projektets referenssystem för vertikala koordinater (t.ex. ”EPSG:5703”)
project_vertical_crs_definitic	Den fullständiga definitionen av projektets referenssystem för vertikala koordinater
project_vertical_crs_descript	Beskrivning av projektets referenssystem för vertikala koordinater
project_vertical_crs_wkt	WKT (välkänd text)-representation av projektets vertikala koordinatreferenssystem
qgis_locale	Det aktuella språket i QGIS
qgis_os_name	Namnet på det aktuella operativsystemet, t.ex. ”windows”, ”linux” eller ”osx
qgis_platform	QGIS-plattformen, t.ex. ”desktop” eller ”server
qgis_release_name	Det aktuella namnet på QGIS-versionen
qgis_short_version	Den aktuella QGIS-versionen kort sträng
qgis_version	Den aktuella QGIS-versionens sträng
qgis_version_no	Det aktuella versionsnumret för QGIS
row_number	Lagrar numret på den aktuella raden
snapping_results	Ger tillgång till fotograferingsresultat när du digitaliserar ett objekt (endast tillgängligt i Add Feature)
scale_value	Det aktuella värdet för skalstapelns avstånd
selected_file_path	Vald filsökväg från filwidgetväljaren vid uppladdning av en fil med ett externt lagringssystem
symbol_angle	Vinkeln på den symbol som används för att återge objektet (gäller endast markörsymboler)
symbol_color	Färgen på den symbol som används för att rendera funktionen
symbol_count	Antalet funktioner som representeras av symbolen (i layoutlegenden)
symbol_frame	Bildnummer (endast för animerade symboler)
symbol_id	Symbolens interna ID (i layoutlegenden)
symbol_label	Symbolens etikett (antingen en användardefinierad etikett eller den autogenererade standardetiketten - i layoutlegenden)
symbol_layer_count	Totalt antal symbollager i symbolen
symbol_layer_index	Aktuellt symbolskiktsindex
symbol_marker_column	Kolumnnummer för markören (gäller endast för fyllning av punktmönster).
symbol_marker_row	Radnummer för markören (gäller endast för punktmönsterfyllningar).
user_account_name	Den aktuella användarens kontonamn i operativsystemet
user_full_name	Den aktuella användarens användarnamn för operativsystemet
value	Det aktuella värdet
vector_tile_zoom	Exakt zoomnivå för vektorplattor på den karta som återges (härledd från den aktuella kartskalan). Normalt i intervallet [0, 20]. Till skillnad från @zoom_level är denna variabel ett flyttalsvärde som kan användas för att interpolera värden mellan två heltalszoomnivåer.
with_variable	Gör det möjligt att ställa in en variabel för användning inom ett uttryck och undvika att samma värde räknas om flera gånger
zoom_level	Zoomnivå för vektorplattor för den karta som återges (härledd från den aktuella kartskalan). Normalt i intervallet [0, 20].

Några exempel:

- Returnera X-koordinaten för ett kartobjekts centrum i layout:

```
x( map_get( item_variables( 'map1'), 'map_extent_center' ) )
```

- Returnera, för varje funktion i det aktuella lagret, antalet överlappande flygplatsfunktioner:

```
aggregate( layer:='airport', aggregate:='count', expression:="code",
               filter:=intersects( $geometry, geometry( @parent ) ) )
```

- Hämta objekt_id för den första snäppta punkten på en linje:

```
with_variable(
  'first_snapped_point',
  array_first( @snapping_results ),
  attribute(
    get_feature_by_id(
      map_get( @first_snapped_point, 'layer' ),
      map_get( @first_snapped_point, 'feature_id' )
    ),
    'object_id'
  )
)
```

9.2.28 Senaste funktionerna

Denna grupp innehåller nyligen använda funktioner. Beroende på i vilket sammanhang den används (funktionsval, fältkalkylator, generisk) läggs nyligen använda uttryck till i motsvarande lista (upp till tio uttryck), sorterade från mer till mindre aktuella. Detta gör det enkelt att snabbt hämta och återanvända tidigare använda uttryck.

KAPITEL 10

Stilbiblioteket

10.1 Den stilansvarige

10.1.1 Dialogrutan Stilhanterare

I *Style Manager* kan du hantera och skapa generiska stilobjekt. Dessa är symboler, färgramper, textformat eller etikettinställningar som kan användas för att symbolisera funktioner, lager eller utskriftslayouter. De lagras i databasen `symbology-style.db` under den aktiva *användarprofilen* och delas med alla projektfiler som öppnas med den profilen. Stilobjekt kan också delas med andra tack vare export/importfunktionerna i dialogrutan *Style Manager*.

Du kan inte heller öppna den modellösa dialogen:

- från menyn *Inställningar* ► *Style Manager…* menyn
- med knappen Style Manager från verktygsfältet Project
- eller med Stilhanterare-knappen från en vektor *Lageregenskaper* ►-menyn (medan du *konfigurerar en symbol* eller *formaterar en text*).

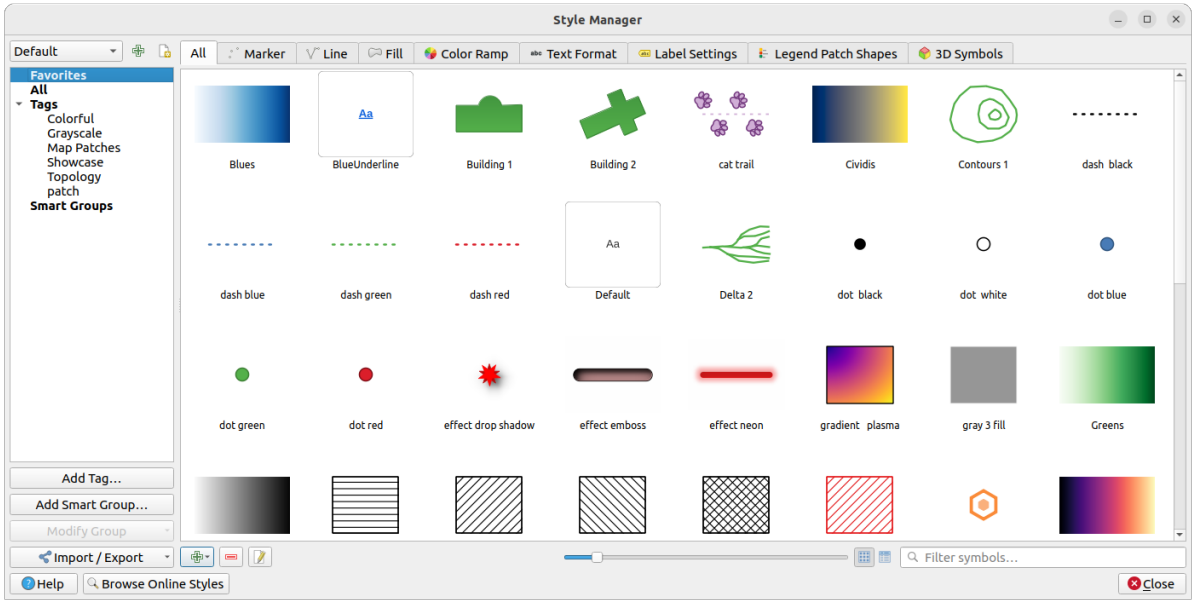


Fig. 10.1: Den stilansvarige

Organisering av stilföremål

I det övre vänstra hörnet av dialogrutan *Style Manager* kan du i rullgardinsmenyn välja vilken stildatabas du vill ansluta till. Om du väljer *Default* kommer du att anslutas till standardstildatabasen där du kan hitta alla tillgängliga standardstilobjekt och de som du har sparat i den här databasen. Om du väljer *Project Styles* kommer du att kopplas till Project Style-databasen där du bara hittar symboler som du har sparat i den här databasen. Om du har skapat fler stildatabaser kommer de att listas i rullgardinsmenyn. Det finns också alternativ för att Lägga till befintlig stildatabas till projektet och Skapa ny stildatabas (se mer *Styledatabas*).

För varje stildatabas kan du organisera elementen i olika kategorier, som listas i panelen till vänster:

- **Favoriter**: visas som standard när du konfigurerar ett objekt och visar en utökad uppsättning objekt;
- **All**: listar alla tillgängliga objekt för den aktiva typen;
- **Tags**: visar en lista med etiketter som du kan använda för att identifiera objekten. Ett objekt kan märkas mer än en gång. Välj en tagg i listan så uppdateras flikarna så att de bara visar de objekt som hör till den. Om du vill skapa en ny tagg som du senare kan koppla till en uppsättning objekt använder du knappen *Add Tag…* eller väljer *Add Tag…* från en taggs kontextuella meny;
- **Smart Group**: en smart grupp hämtar dynamiskt sina symboler enligt fastställda villkor (se t.ex. Fig. 10.2). Klicka på knappen *Add Smart Group…* för att skapa smarta grupper. I dialogrutan kan du ange ett uttryck för att filtrera de objekt som ska väljas (har en viss tagg, har en sträng i sitt namn, etc.). Alla symboler, färgramper, textformat eller etikettinställningar som uppfyller det eller de angivna villkoren läggs automatiskt till i den smarta gruppen.

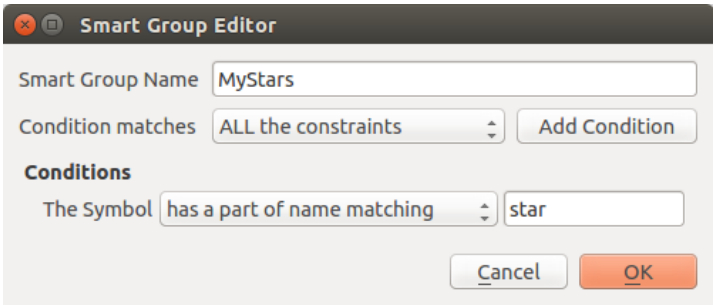


Fig. 10.2: Skapa en smart grupp

Taggar och smarta grupper är inte ömsesidigt uteslutande: de är helt enkelt två olika sätt att organisera dina stilelement. Till skillnad från de smarta grupperna som automatiskt hämtar sina tillhörande objekt baserat på inmatningsbegränsningarna, fylls taggar i av användaren. För att redigera någon av dessa kategorier kan du antingen:

- markera objekten, högerklicka och välj *Add to Tag* ► och välj sedan taggnamnet eller skapa en ny tagg;
- markera taggen och tryck på *Modify group…* ► *Attach Selected Tag to Symbols*. En kryssruta visas bredvid varje objekt för att hjälpa dig att välja eller välja bort det. När valet är klart trycker du på *Modify group…* ► *Finish Tagging*.
- välj smartgruppen, tryck på *Modifiera grupp…* ► *Redigera smartgrupp…* och konfigurera en ny uppsättning begränsningar i dialogrutan *Smartgruppsredigerare*. Detta alternativ är också tillgängligt i den kontextuella menyn för smartgruppen.

Om du vill ta bort en tagg eller en smart grupp högerklickar du på den och väljer knappen *Remove*. Observera att detta inte tar bort de objekt som är grupperade i kategorin.

I dialogrutan *Style Manager* visas i mitten en ram med förhandsgranskade objekt som är organiserade i flikar:

- *All* för en komplett samling av punkt-, linjär- och ytsymboler och etikettinställningar samt fördefinierade färgramper och textformat;
- *Marker* endast för punktsymboler;
- *Line* endast för linjära symboler;
- *Fyll* endast för ytsymboler;
- *Färgramp*;
- abc *Textformat* för att hantera *textformat*, som lagrar teckensnitt, färg, buffertar, skuggor och bakgrunder för texter (dvs. alla formateringsdelar av etikettinställningarna, som t.ex. kan användas i layouter);
- abc *Etikettinställningar* för att hantera *Etikettinställningar*, som inkluderar textformat och vissa lagertypspecifika inställningar som etikettplacering, prioritet, bildtexter, rendering…
- *Legend Patch Shapes* för att hantera anpassade förklaring patch shapes, som inkluderar geometrierna *Marker*, *Line* och *Fill*.
- *3D Symbols* för att konfigurera symboler med *3D egenskaper* (extrudering, skuggning, höjd, …) för de funktioner som ska återges i en *3D kartvy*

Du kan ordna stilarna i *Icon View* eller i *List View* längst ned till höger. I båda vyerna visar verktygstipset en större instans av stilen. Skjutreglaget för miniatyrstorlek till vänster om ikonerna hjälper dig att justera de faktiska miniatyrstorlekarna i dialogrutan, så att du får en bättre förhandsgranskning av symbolerna!

Lägga till, redigera eller ta bort ett objekt

Som du såg tidigare listas stilelement under olika flikar vars innehåll beror på den aktiva kategorin (tagg, smartgrupp, favoriter …). När en flik är aktiverad kan du:

- Lägg till nya objekt: tryck på knappen Add item och konfigurera objektet enligt *symboler*, *färgramper* eller *textformat och etikett* byggarbeskrivning.
- Ändra ett befintligt objekt: välj ett objekt och tryck på Edit item-knappen och konfigurera enligt ovan.
- Ta bort befintliga objekt: Om du vill ta bort ett element som du inte längre behöver markerar du det och klickar på :sup:`Remove item` (även tillgängligt via högerklick). Objektet raderas från den lokala databasen.

Observera att fliken *All* ger tillgång till dessa alternativ för alla typer av objekt.

Om du högerklickar på ett urval av objekt kan du också göra följande:

- *Lägg till i favoriter*;
- *Avlägsna från favoriter*;
- *Add to Tag* ► och välj lämplig tagg eller skapa en ny att använda; de taggar som för närvarande är tilldelade markeras;
- *Clear Tags*: avlägsnar symbolerna från valfri tagg;
- *Remove Item(s)*;
- *Edit Item*: gäller för det objekt som du högerklickar på;
- *Kopiera objekt*;
- *Paste Item* …: klistra in i en av kategorierna i stilhanteraren eller någon annanstans i QGIS (symbol- eller färgknappar)
- *Exportera valda symboler som PNG…* (endast tillgängligt med symboler);
- *Exportera valda symboler som SVG…* (endast tillgängligt med symboler);

Delning av stilobjekt

Verktyget *Import/Export*, längst ned till vänster i dialogen Style Manager, erbjuder alternativ för att enkelt dela symboler, färgramper, textformat och etikettinställningar med andra. Dessa alternativ är också tillgängliga genom att högerklicka på objekten.

Exportera artiklar

Du kan exportera en uppsättning objekt till en `.XML`-fil:

1. Expandera rullgardinsmenyn *Import/Export* och välj *Export Item(s)…*
2. Välj de objekt som du vill integrera. Valet kan göras med musen eller med hjälp av en tagg eller en grupp som ställts in tidigare.
3. Tryck på *Export* när du är klar. Du kommer att uppmanas att ange destinationen för den sparade filen. XML-formatet genererar en enda fil som innehåller alla valda objekt. Den här filen kan sedan importeras till en annan användares stilbibliotek.

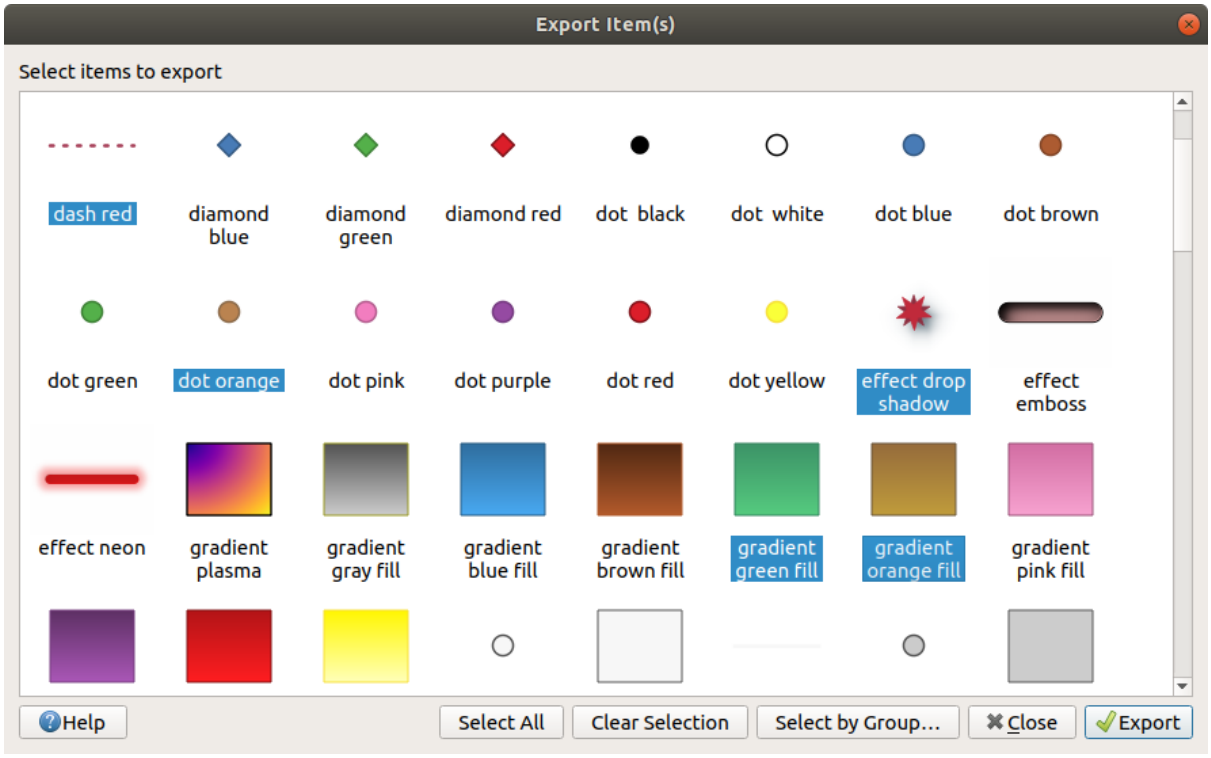


Fig. 10.3: Exportera stilobjekt

När symboler är markerade kan du även exportera dem till `.PNG` eller `.SVG`. När du exporterar till `.PNG` eller `.SVG` (båda är inte tillgängliga för andra typer av stilobjekt) skapas en fil för varje markerad symbol i en viss mapp. SVG-mappen kan läggas till i *SVG paths* i *Inställningar* ► *Alternativ* ► *System*-menyn för en annan användare, så att han får direkt tillgång till alla dessa symboler.

Importera artiklar

Du kan utöka ditt stilbibliotek genom att importera nya objekt:

1. Expandera rullgardinsmenyn *Importera/Exportera* och välj *Importera objekt* längst ned till vänster i dialogrutan.
2. I den nya dialogrutan anger du källan till stilelementen (det kan vara en `.xml`-fil på disken eller en webbadress).
3. Ange om objekten som ska importeras ska läggas till i favoriterna enligt *Add to favorites*.
4. Check *Do not import embedded tags* för att undvika import av taggar som är kopplade till de objekt som importeras.
5. Ange namnet på eventuella *Addditional tag(s)* som ska tillämpas på de nya objekten.
6. Välj från förhandsgranskningen de symboler som du vill lägga till i ditt bibliotek.
7. Tryck sedan på *Import*.

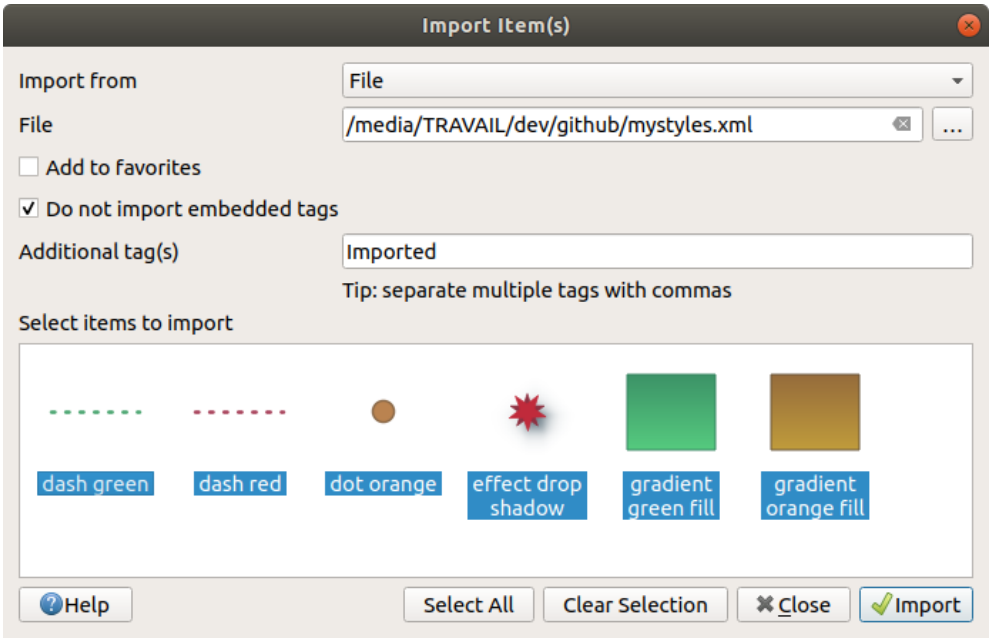


Fig. 10.4: Importera stilobjekt

Använda bläddrarpanelen

Det är också möjligt att importera stilobjekt till den aktiva användarprofilens stildatabas direkt från *Browser*-panelen:

1. Välj stilen `.xml`-fil i bläddraren
2. Dra och släpp den över kartbilden eller högerklicka och välj *Importera stil…*
3. Fyll i dialogrutan *Import Items* enligt *Importera artiklar*
4. Tryck på *Import* så läggs de valda stilelementen till i stildatabasen

Genom att dubbelklicka på stilfilen i bläddraren öppnas dialogrutan *Style Manager* som visar objekten i filen. Du kan markera dem och trycka på *Copy to Default Style…* för att importera dem till den aktiva stildatabasen. Objekten kan tilldelas taggar. Även tillgängligt genom högerklick, kommandot *Öppna stil…*.

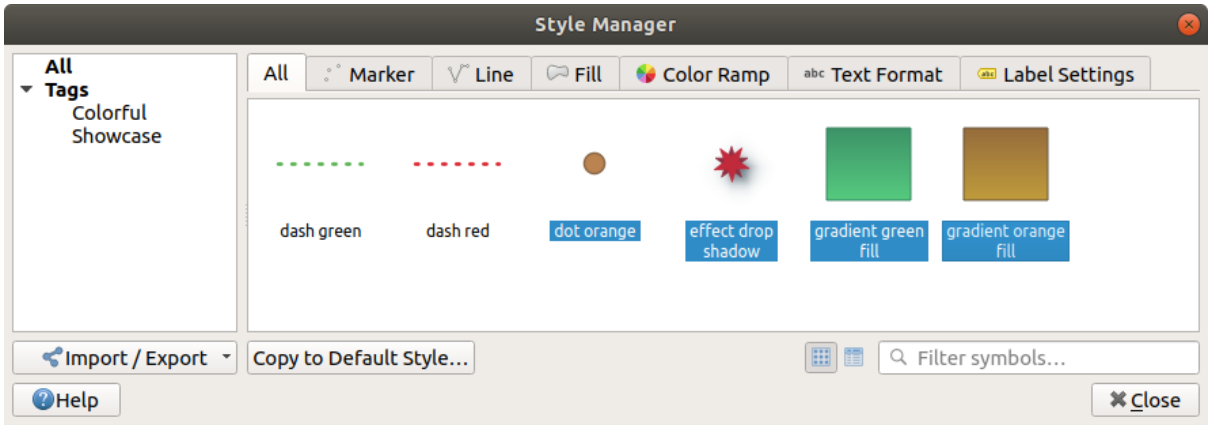


Fig. 10.5: Öppna en fil med stilobjekt

Dialogrutan gör det också möjligt att exportera enskilda symboler som `.PNG`- eller `.SVG`-filer.

Använda online-arkivet

QGIS-projektet upprätthåller ett arkiv med en samling stilar som delas av QGIS-användare. Den finns på https://hub.qgis.org/styles och kan nås från dialogrutan *Style Manager* genom att trycka på knappen *Browse Online Styles* längst ned.

Från det förvaret kan du:

1. Bläddra bland och sök efter alla stilobjekt, baserat på typ eller namn
2. Ladda ner stilfilen och packa upp den
3. Ladda den `.xml`-baserade filen till din stildatabas i QGIS med hjälp av någon av de ovannämnda importmetoderna.

10.1.2 Inställning av en färgramp

Fliken Color ramp i dialogrutan *Style Manager* hjälper dig att förhandsgranska olika färgramper baserat på den kategori som valts i den vänstra panelen.

För att skapa en anpassad färgramp aktiverar du fliken Color ramp och klickar på knappen Add item. Knappen visar en rullgardinslista där du kan välja typ av ramp:

- *Gradient*: ger en start- och slutfärg och genererar en färgramp som kan vara **kontinuerlig** eller **diskret**. Genom att dubbelklicka på förhandsgranskningen av rampen kan du lägga till så många mellanliggande färgstopp som du vill. Klicka på färgstoppsindikatorn och under *Gradient stop* kan du:
  - justera dess *Relativ position* från färgrampens start. Det är också möjligt att dra indikatorn med musen eller trycka på piltangenterna (kombinera med `Shift`-tangenten för en större förflyttning)
  - ange den färgmodell som ska användas vid interpolering mellan färger: den kan vara *RGB*, *HSL* eller *HSV*. Under vissa omständigheter kan detta alternativ hjälpa till att undvika mättade mellantoner, vilket resulterar i mer visuellt tilltalande gradienter.
  - anger den riktning som interpoleringen ska följa för **Hue**-komponenten i en *HSL* eller *HSV* färgspecifikation. Den kan vara *Clockwise* eller *Counterclockwise*.
  - ställa in *färgens egenskaper*
  - ta bort färgstoppet genom att trycka på *Delete stop* eller `DEL`

  Gruppen *Plots* ger ett annat grafiskt sätt att utforma färgrampen genom att ändra färgstoppens position eller opacitet och HSL-komponenter.

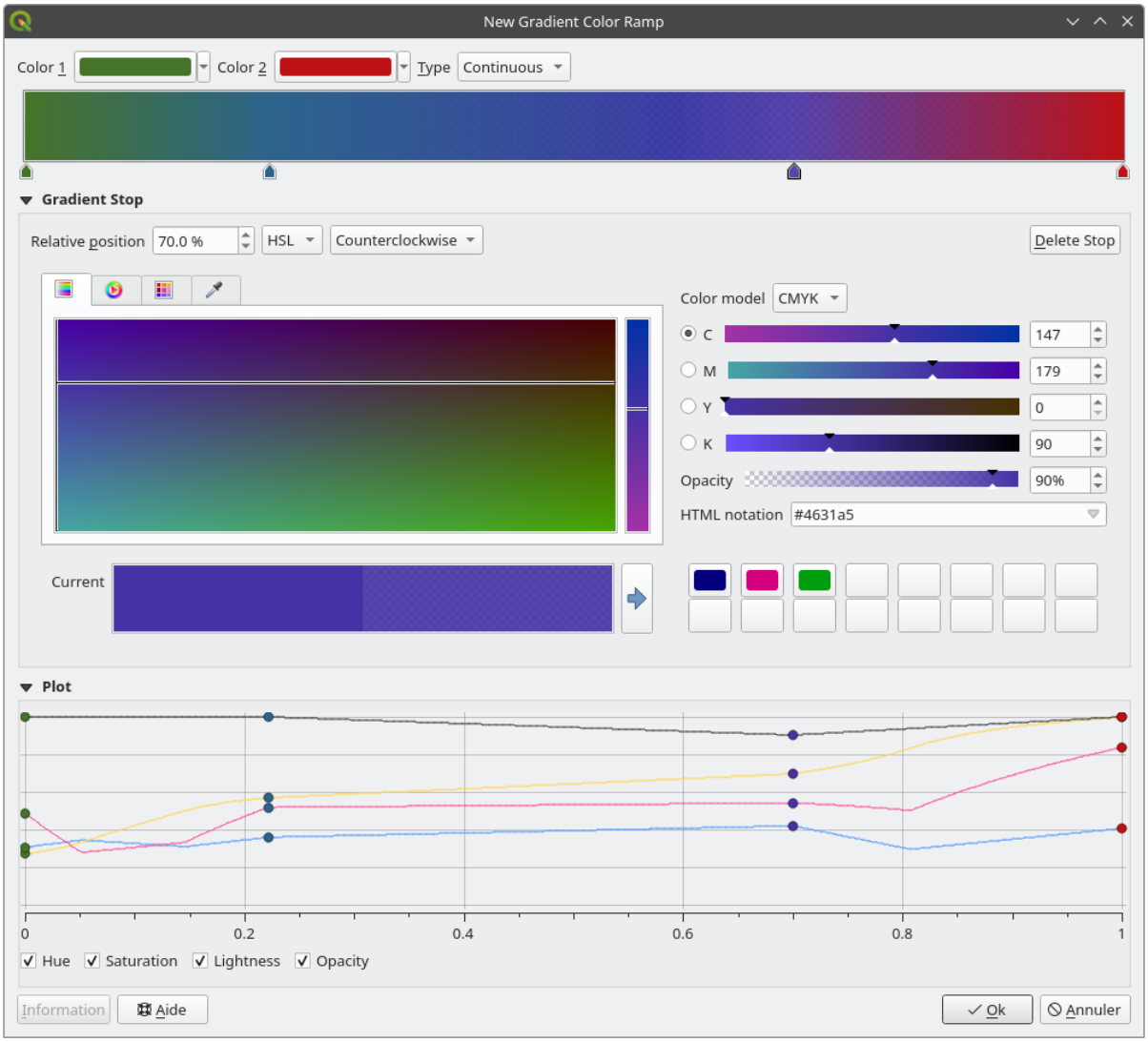


Fig. 10.6: Exempel på anpassad gradientfärgsramp med flera stopp

> **Råd**
>
> Dra och släpp en färg från en färgpunkt på förhandsgranskningen av lutningsrampen och lägg till ett nytt färgstopp.

- *Color presets*: gör det möjligt att skapa en färgramp som består av en lista med färger som användaren har valt;
- *Random*: skapar en slumpmässig uppsättning färger baserat på värdeintervall för *Hue*, *Saturation*, *Value* och *Opacity* och ett antal färger (*Classes*);
- *Katalog: ColorBrewer*: en uppsättning fördefinierade diskreta färggradienter du kan anpassa antalet färger i rampen;
- eller *Catalog: cpt-city*: tillgång till en hel katalog med färggradienter som kan *sparas lokalt som standardgradient*. Alternativet cpt-city öppnar en ny dialog med hundratals teman som ingår ”out of the box”.

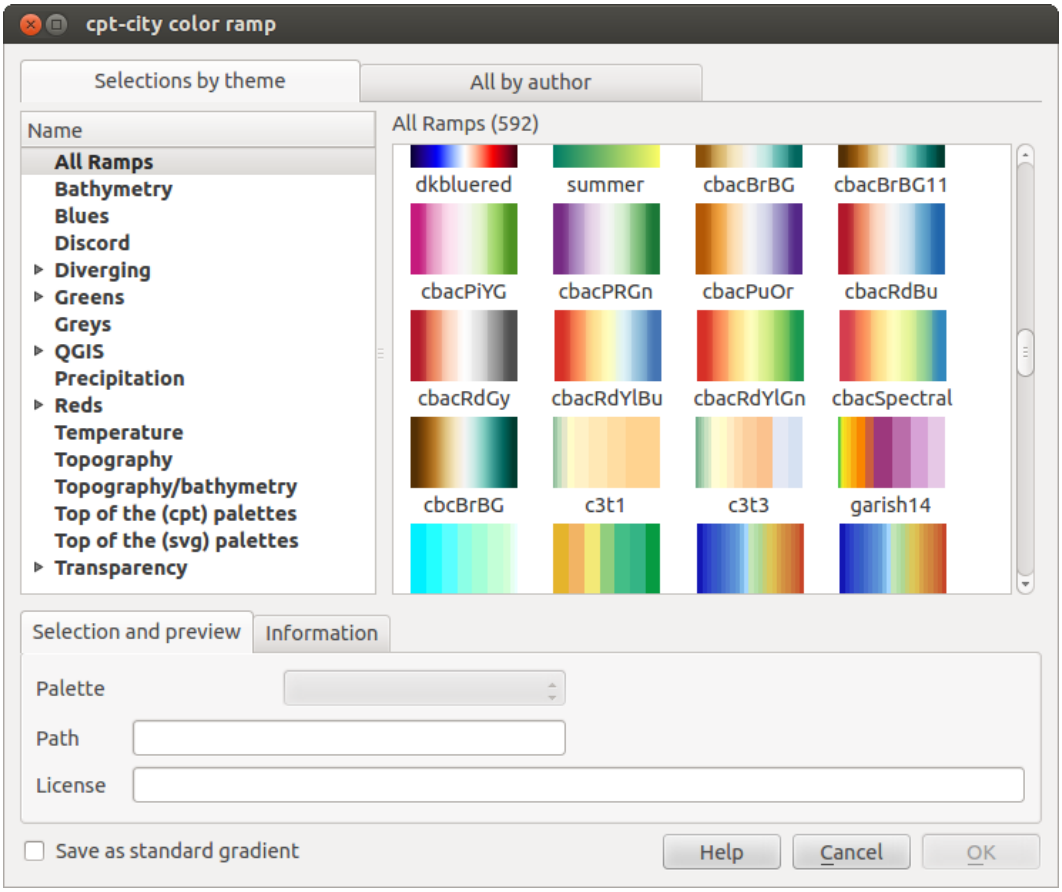


Fig. 10.7: cpt-city dialog med hundratals färgramper

10.1.3 Skapa en Legend Patch Shape

Du skapar en ny Legend Patch Shape genom att aktivera fliken *Legend Patch Shapes* och klicka på knappen Add item. Knappen visar en listruta där du kan välja geometrityp:

- *Marker Legend Patch Shape…*: för användning med punktgeometrier.
- *Line Legend Patch Shape…*: för användning med linjegeometrier.
- *Fill Legend Patch Shape…*: för användning med polygongeometrier.

Alla tre alternativen visar samma dialogruta.

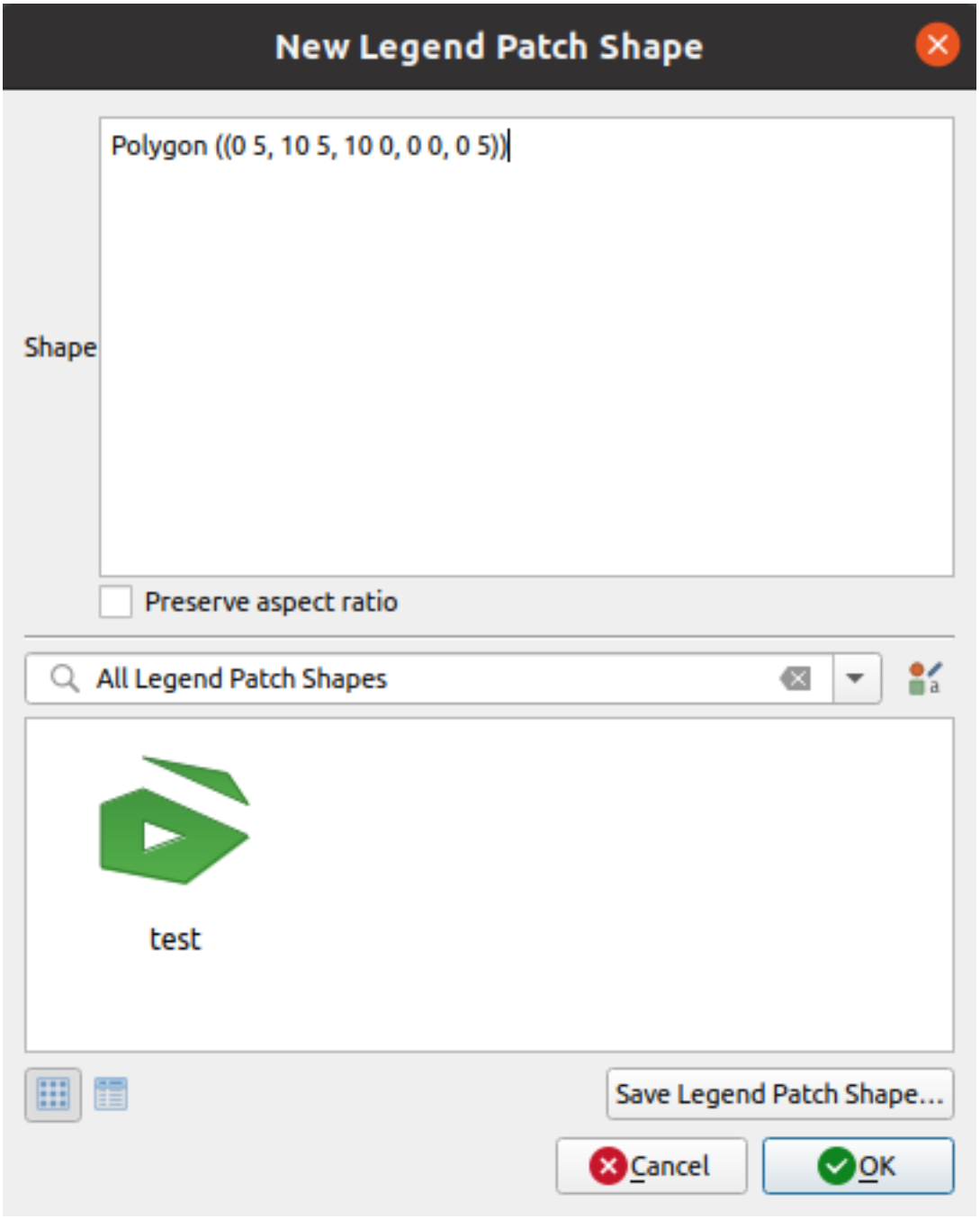


Fig. 10.8: Skapa en ny Legend Patch Shape

Det är endast formtypen och de visade legendpatchformerna som skiljer sig åt beroende på den valda geometritypen. Följande alternativ kommer att vara tillgängliga:

- *Shape*: definierar formen på legendpatchens form som en WKT-sträng. Geometrier med en eller flera delar kan användas, men ingen GeometryCollection.
- *Bevara bildförhållande*
- *Icon View* eller *List View* av tillgängliga former för teckenförklaringar, filtrerade efter taggar.

När den nya formen är definierad kan du *Save Legend Patch Shape…* eller trycka på *OK*, vilket båda leder till samma dialogruta.

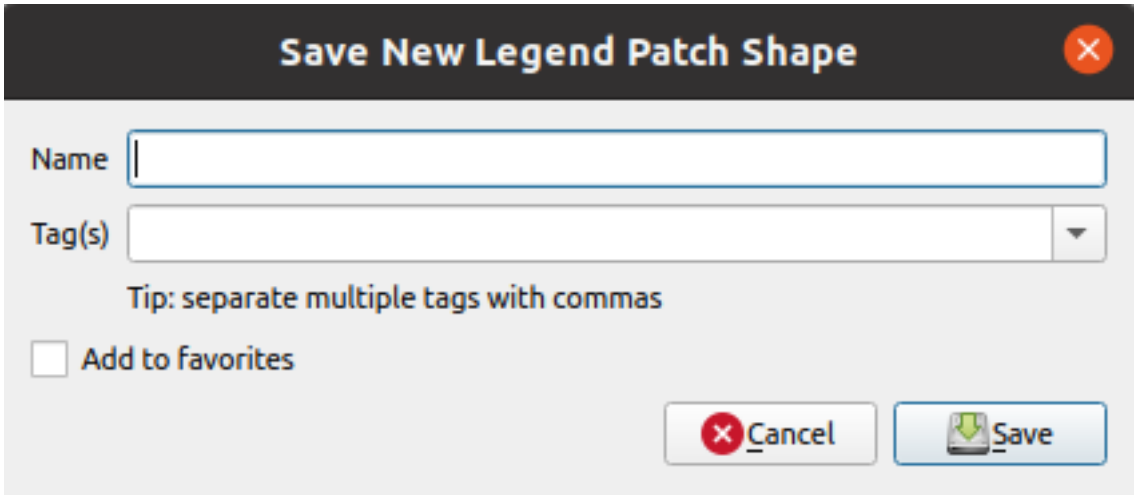


Fig. 10.9: Spara en ny Legend Patch Shape

Här måste du välja ett namn, taggar för att beskriva formen och om den ska läggas till bland favoriterna.

Om du trycker på *Save…* läggs formen till i listan och du kommer tillbaka till dialogrutan *New Legend Patch Shape* där du kan fortsätta skapa nya former.

10.2 Symbolväljaren

Symbolväljaren är den viktigaste dialogrutan för att utforma en symbol. Du kan skapa eller redigera markör-, linje- eller fyllnadssymboler.

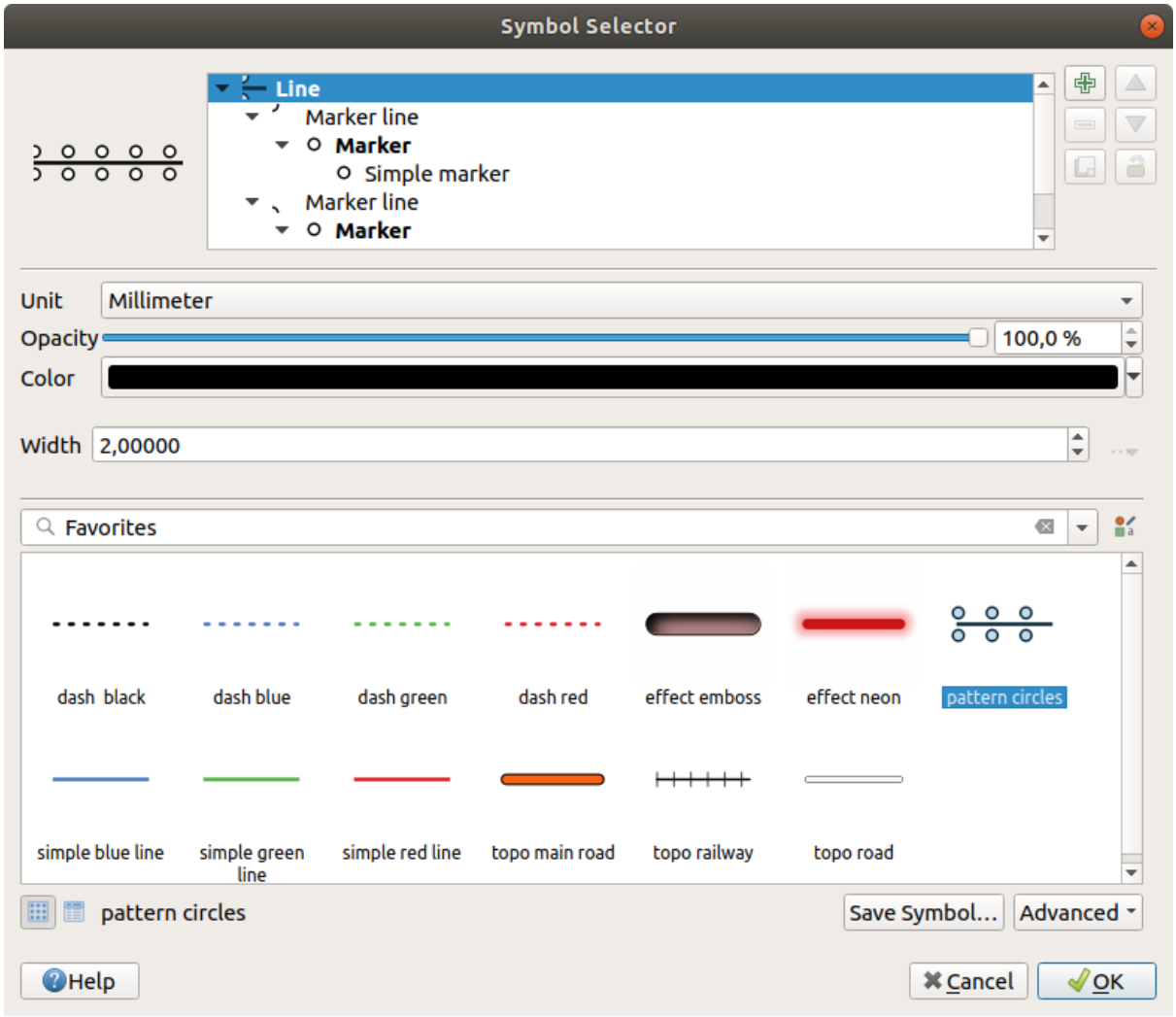


Fig. 10.10: Utformning av en linjesymbol

Två huvudkomponenter strukturerar dialogrutan för symbolväljare:

- symbolträdet, som visar symbollager som kombineras efteråt för att forma en ny global symbol
- och inställningar för att konfigurera det valda symbollagret i trädet.

10.2.1 Symbollagrets träd

En symbol kan bestå av flera *Symbollager*. Symbolträdet visar överlagringen av dessa symbollager som sedan kombineras för att forma en ny global symbol. Dessutom uppdateras en dynamisk symbolrepresentation så snart symbolens egenskaper ändras.

Beroende på vilken nivå som valts i symbolträdet finns olika verktyg tillgängliga för att hjälpa dig att hantera trädet:

- lägg till nytt symbollager: du kan stapla så många symboler du vill
- ta bort det valda symbollagret
- låsa färger i symbollagret: en låst färg förblir oförändrad när användaren ändrar färgen på den globala (eller övre) symbolnivån
- duplicera en (grupp av) symbollager

- flytta upp eller ner symbollagret

10.2.2 Konfigurera en symbol

I QGIS görs konfigurationen av en symbol i två steg: symbolen och sedan symbollagret.

Symbolen

På den översta nivån i trädet beror det på lagrets geometri och kan vara av typen **Marker**, **Line** eller **Fill**. Varje symbol kan bädda in en eller flera symboler (även av annan typ) eller symbollager.

Du kan ställa in vissa parametrar som gäller för den globala symbolen:

- *Unit*: det kan vara **Millimeter**, **Punkter**, **Pixlar**, **Meter i skala**, **Kartenheter** eller **Tum** (se *Väljare för enhet* för mer information)
- *Opacitet*
- *Color*: när denna parameter ändras av användaren, återges dess värde till alla olåsta undersymboler color
- *Size* och *Rotation* för markörsymboler
- *Bredd* för linjesymboler

> **Tips**
>
> Använd egenskaperna *Size* (för markörsymboler) eller *Width* (för linjesymboler) på symbolnivå för att proportionellt ändra storlek på alla dess inbäddade *symbollager*-dimensioner.

> **Observera**
>
> Knappen *Data-defined override* bredvid parametrarna för bredd, storlek eller rotation är inaktiv när symbolen ställs in i dialogrutan Style manager. När symbolen är kopplad till ett kartlager hjälper den här knappen dig att skapa *proportionell eller multivariat analys* rendering.

- En förhandsgranskning av *symbolbiblioteket*: Symboler av samma typ visas och kan filtreras med hjälp av den redigerbara rullgardinslistan ovanför, med fritext eller med *kategorier*. Du kan också uppdatera listan med symboler med hjälp av knappen Style Manager och öppna dialogen eponym. Där kan du använda alla funktioner som beskrivs i avsnittet *Den stilansvarige*.

  Symbolerna visas antingen:

  - i en ikonlista (med miniatyrbild, namn och tillhörande taggar) med hjälp av knappen List View under ramen;
  - eller som förhandsgranskning av ikoner med hjälp av knappen Icon View.

- Tryck på knappen *Save Symbol* för att öppna dialogrutan *Save New Symbol*. Här kan du välja *Destination* där du vill lägga till symbolen som redigeras, ge den ett *Namn* och lägga till *Tag(s)*. Du har också möjlighet att kryssa i Add to favorites för din nya symbol.
- Med alternativet *Advanced* kan du:
  - för linje- och fyllnadssymboler, *Clip features to canvas extent*.
  - för fyllningssymboler, *Force right-hand rule orientation*: gör det möjligt att tvinga renderade fyllningssymboler att följa standard ”högerhandsregeln” för ringorientering (dvs. polygoner där den yttre ringen är medurs och de inre ringarna är alla moturs).

Orienteringsfixeringen tillämpas endast under renderingen och den ursprungliga geometrin för objektet är oförändrad. Detta gör det möjligt att skapa fyllnadssymboler med konsekvent utseende, oavsett vilken dataset som renderas och ringorienteringen för enskilda objekt.

- för markörsymboler gör *Buffertinställningar…* det möjligt att lägga till en haloeffekt runt markörsymbolen för att göra den mer läsbar mot olika bakgrunder. Bufferten beräknas och ritas med hjälp av formen på ALLA symbollager i markören. Du kan justera *Size*, *Join style* och *Symbol* för visning.
- Beroende på *symbology* för det lager som en symbol används på finns ytterligare inställningar tillgängliga i menyn *Advanced*:
  - *Symbolnivåer…* för att definiera ordningen på de symboler som återges
  - *Legende för datadefinierad storlek*
  - *Matcha med sparade symboler…* och *Matcha med symboler från fil…* för att automatiskt *tilldela symboler till klasser*
  - *Animationsinställningar*

Symbollagret

På en lägre nivå i trädet kan du anpassa symbollagren. Vilka typer av symbollager som är tillgängliga beror på den övre symboltypen. Du kan använda symbollagret *paint effects* för att förbättra renderingen.

Eftersom det inte är möjligt att beskriva alla alternativ för alla typer av symbollager, nämns endast särskilda och viktiga alternativ nedan.

Vanliga parametrar

Det finns några vanliga alternativ och widgetar för att bygga ett symbollager, oavsett om det är av typen markör, linje eller fyllning:

- widgeten *färgväljaren* för att underlätta färgmanipulering
- *Units*: det kan vara **Millimeters**, **Points**, **Pixels**, **Meters at Scale**, **Map units** eller **Inches** (se *Väljare för enhet* för mer information)
- widgeten Data-defined override nära nästan alla alternativ, vilket utökar möjligheterna att anpassa varje symbol (se *Inställning av datadefinierad åsidosättning* för mer information)
- alternativet *Enable symbol layer* styr symbollagrets synlighet. Inaktiverade symbollager ritas inte när symbolen renderas, men sparas i symbolen. Att kunna dölja symbollager är praktiskt när man letar efter den bästa designen för sin symbol eftersom man inte behöver ta bort något för testningen. Den datadefinierade åsidosättningen gör det möjligt att dölja eller visa olika symbollager baserat på uttryck (t.ex. med hjälp av funktionsattribut).
- *Rita effekter*-knappen för *effektrendering*.

Observera

Beskrivningen nedan förutsätter att symbolskiktstypen är bunden till objektets geometri, men tänk på att du kan bädda in symbollager i varandra. I så fall kan parametern för symbollagret på lägre nivå (placering, offset…) vara bunden till symbolen på högre nivå, och inte till själva geometrin för objektet.

Symboler för markörer

Markörsymboler är lämpliga för punktgeometriska objekt och har flera *Symbollagertyper*:

- **Enkel markör** (standard)

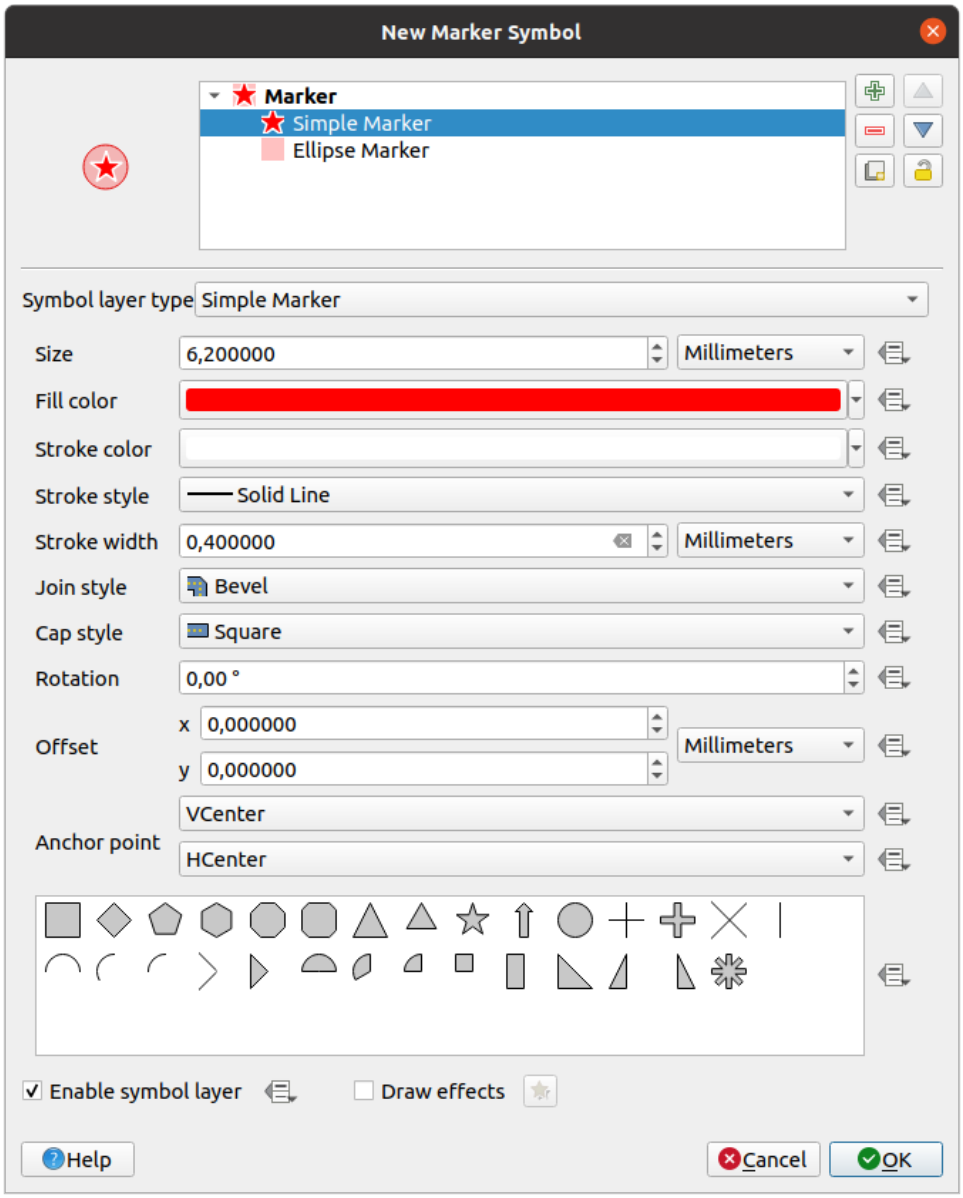


Fig. 10.11: Utformning av en enkel markörsymbol

Lagertypen enkel markörsymbol har följande egenskaper:

- *Storlek* i olika enheter som stöds
- *Fyllnadsfärg*
- *Stroke color*, *Stroke style* från en fördefinierad lista och *Stroke size*
- *Join style*: det kan vara **Bevel**, **Miter** eller **Round**
- *Cap style*: den kan vara **Square**, **Flat** eller **Round**
- *Rotation*

  - *Offset* i riktningarna *X* och *Y* från objektet
  - *Anchor point*: definierar den kvadrantpunkt på symbolen som ska användas som placeringsursprung. Detta är den punkt som *Offset* appliceras på.

- **Animerad markör** (se *Animerad markör*)
- **Ellipsmarkör**: ett enkelt lager med markörsymboler, med anpassningsbar bredd och höjd
- **Fylld markör**: liknar det enkla symbollagret för markörer, förutom att det använder en *fill sub symbol* för att rendera markören. Detta möjliggör användning av alla befintliga QGIS-stilar för fyllning (och streck) för rendering av markörer, t.ex. gradient- eller shapeburst-fyllningar.
- **Font marker**: liknar det enkla symbollagret för markörer, förutom att det använder installerade teckensnitt för att återge markören. Dess ytterligare egenskaper är:
  - *Fontfamilj*
  - *Fontstil*
  - *Tecken*, som representerar den text som ska visas som symbol. De kan skrivas in eller väljas från widgeten för teckensamling och du kan live *Preview* dem med de valda inställningarna.
- **Geometri-generator** (se *Geometri-generator*)
- **Mask**: dess undersymbol definierar en maskform vars färgegenskaper ignoreras och endast opaciteten används. Detta är praktiskt när markörsymbolen överlappar etiketter eller andra symboler vars färger ligger nära varandra, vilket gör det svårt att tyda dem. Mer information finns på *Masker Egenskaper*.
- **Rasterbildsmarkör**: använd en bild (`PNG`, `JPG`, `BMP` …) som markörsymbol. Bilden kan vara en fil på disken, en fjärr-URL, inbäddad i stildatabasen (*mer detaljer*) eller så kan den kodas som en base64-sträng. Bildens bredd och höjd kan ställas in oberoende av varandra eller med hjälp av Lock aspect ratio. Storleken kan ställas in med hjälp av någon av *common units* eller som en procentandel av bildens originalstorlek (skalad med bredden).
- **Vektorfältmarkör** (se *Vector Field-markören*)
- ** SVG-markör**: ger dig bilder från dina SVG-sökvägar (inställda i :menuselection: *Inställningar* –> *Alternativ* … –> *System* meny) för att återges som markörsymbol. Symbolens bredd och höjd kan ställas in oberoende eller med hjälp av Lock aspect ratio. Varje SVG-fils färger och streck kan också anpassas. Bilden kan vara en fil på disken, en fjärr-URL, inbäddad i stildatabasen (*mer detaljer*) eller så kan den kodas som en base64-sträng.

  Symbolen kan också ställas in med *Dynamiska SVG-parametrar*. Se avsnittet *Parametriserbar SVG* för att parametrisera en SVG-symbol.

  Observera

  Krav på SVG-version

  QGIS renderar SVG-filer som följer profilen SVG Tiny 1.2, avsedd för implementering på en rad olika enheter, från mobiltelefoner och handdatorer till bärbara och stationära datorer, och innehåller därför en delmängd av de funktioner som ingår i SVG 1.1 Full, tillsammans med nya funktioner för att utöka SVG:s kapacitet.

  Vissa funktioner som inte ingår i dessa specifikationer kanske inte återges korrekt i QGIS.

Linjesymboler

Linjesymboler är lämpliga för linjegeometriska funktioner och har följande typer av symbollager:

- **Enkel linje** (standard)

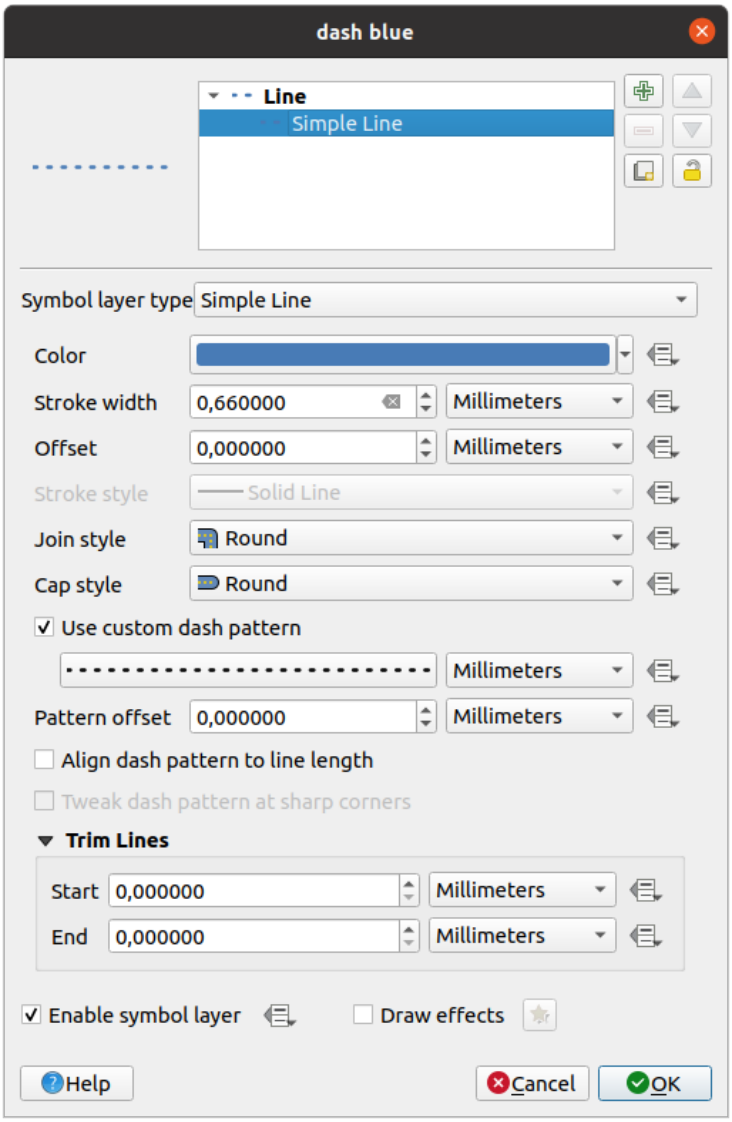


Fig. 10.12: Utformning av en enkel linjesymbol

Lagertypen enkel linjesymbol har många av samma egenskaper som *enkel markeringssymbol*, och dessutom:

- *Use custom dash pattern*: åsidosätter inställningen *Stroke style* med ett anpassat streck. Du måste definiera längden på de på varandra följande strecken och mellanslagen som formar modellen, i vald enhet. Den totala längden på mönstret visas längst ner i dialogrutan.
- *Pattern offset*: positioneringen av strecken/utrymmena i linjen kan justeras, så att de kan placeras på snyggare positioner för att ta hänsyn till hörn i linjen (kan också potentiellt användas för att ”anpassa” intilliggande streckmönsterkanter)
- *Align dash pattern to line length*: streckmönstrets längd kommer att justeras så att raden slutar med ett komplett streckelement, istället för ett mellanrum.
- *Tweak dash pattern at sharp corners*: justerar dynamiskt streckmönstrets placering så att skarpa hörn

representeras av ett helt streckelement som går in i och ut ur det skarpa hörnet. Beroende av *Align dash pattern to line length*.

  - *Trim lines* from *Start* and/or *End*: gör det möjligt för linjeåtergivningen att trimma bort de första x mm och sista y mm från den faktiska linjesträngen när linjen ritas. Den stöder en rad olika *enheter*, inklusive procent av den totala linjelängden, och kan datadefinieras för extra kontroll. Trimavståndet för start/slut kan t.ex. användas när man skapar komplexa symboler där ett linjelager inte ska överlappa markörsymbollager som placerats i början och slutet av linjen.

- **Arrow**: ritar linjer som böjda (eller inte böjda) pilar med ett enkelt eller dubbelt huvud med konfigurerbar (och datadefinierad):
  - *Huvudtyp*
  - *Piltyp*
  - *Pilens bredd*
  - *Pilens bredd vid start*
  - *Huvudets längd*
  - *Huvudets tjocklek*
  - *Offset*

  Det är möjligt att skapa *Böjda pilar* (linjefunktionen måste ha minst tre hörn) och *Upprepa pilen på varje segment*. Den använder också en *fill-symbol* som t.ex. gradienter eller shapeburst för att rendera pilkroppen. I kombination med geometri-generatorn hjälper den här typen av lagersymbol dig att representera flödeskartor.

- **Geometri-generator** (se *Geometri-generator*)

- **Interpolerad linje**: gör det möjligt att rendera en linje vars *Stroke width* och/eller *Color* kan vara konstant (givet parametrarna *Fixed width* och *Single color*) eller variera längs geometrin. Vid variation är nödvändiga indata:
  - *Startvärde* och *Slutvärde*: Värden som kommer att användas för interpolering vid ytterpunkterna i objektets geometri. De kan vara fasta värden, objektets attribut eller baserade på ett uttryck.
  - *Min. värde* och *Max. värde*: Värden mellan vilka interpoleringen utförs. Tryck på knappen Läs in för att automatiskt fylla i dem baserat på de lägsta och högsta start-/slutvärdena för lagret.
  - Endast tillgängligt för stroke-alternativet:
    * *Min. width* och *Max. width*: definierar intervallet för den varierande bredden. *Min. width* tilldelas *Min. value* och *Max. width* tilldelas *Max. value*. En *unit* kan associeras.
    * *Use absolute value*: tar endast hänsyn till absolut värde vid interpolering (negativa värden används som positiva).
    * *Ignore out of range*: som standard, när intervallet `[startvärde – slutvärde]` för en funktion inte ingår i intervallet `[min. värde – max. värde]`, återges de delar av funktionens geometri som ligger utanför gränserna med min- eller maxbredden. Markera detta alternativ för att inte rendera dem alls.
  - För varierande färg kan du använda någon av interpoleringsmetoderna i *color ramp classification*

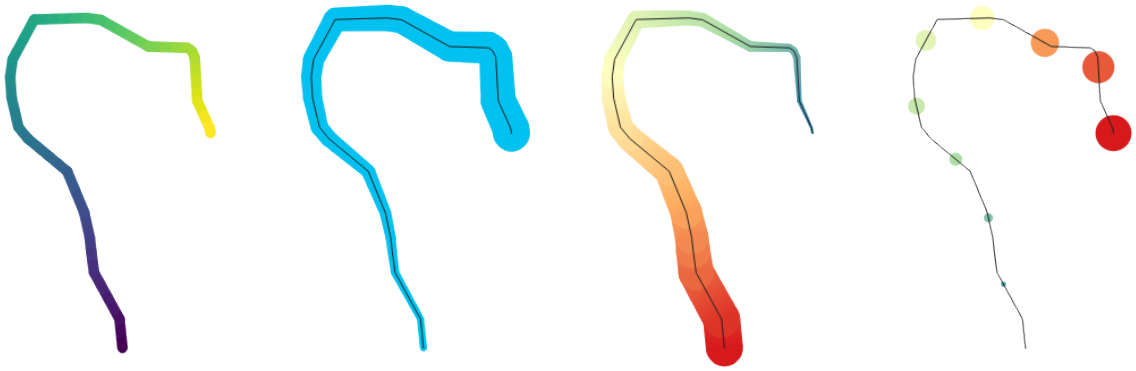


Fig. 10.13: Exempel på interpolerade linjer

- **Markörlinje**: upprepar en *markörsymbol* över längden på en linje.
  - *Markeringens placering* kan ställas in med hjälp av en regelbunden intervallinställning (med början från första toppunkten) och/eller linjens geometriegenskap (på första eller sista toppunkten, inre toppunkter, linjens eller varje segments mittpunkt, eller på varje kurvpunkt).
  - När placering vid första eller sista vertexen är aktiverad kommer alternativet *Place on every part extremity* att göra att markörerna även återges vid första eller sista vertexen för varje del av flerdelade geometrier.
  - *Offset along line*: markörernas placering kan också ges en förskjutning längs linjen, i ref:*unit <unit_selector>* efter eget val (millimeter, punkter, kartans enhet, meter i skala, procent, …):
    - Ett positivt värde förskjuter markörsymbolerna i linjeriktningen (med *On first vertex* och *With interval* placeringar) och bakåt (med *On last vertex* placering).
    - Ett negativt värde på en icke sluten linje resulterar i ingen förskjutning (för *På första toppunkten* och *På sista toppunkten* placeringar) eller bakåtförskjutning av symbolerna (från sista toppunkten).
    - Med en sluten ring behandlar QGIS förskjutningen som att den fortsätter att slingra sig runt ringen (framåt eller bakåt). Om du t.ex. ställer in offset till 150% (resp. -10% or -110%) behandlas offset som 50% (resp. 90%) av längden på den slutna ringen.
  - Alternativet *Rotera markören för att följa linjens riktning* anger om varje markörsymbol ska orienteras i förhållande till linjens riktning eller inte.

    Eftersom en linje ofta är en följd av segment med olika riktningar beräknas markörens rotation genom ett genomsnitt över ett angivet avstånd längs linjen. Om du till exempel ställer in egenskapen *Average angle over* till `4mm` innebär det att de två punkter längs linjen som ligger `2mm` före och efter symbolplaceringen används för att beräkna linjevinkeln för den aktuella markörsymbolen. Detta har effekten att utjämna (eller ta bort) alla små lokala avvikelser från den övergripande linjeriktningen, vilket resulterar i mycket trevligare visuella orienteringar av markörlinjesymbolerna.
  - *Line offset*: Markörsymbolerna kan också förskjutas från linjefunktionen.
- **Hashed line**: upprepar ett linjesegment (en hash) över längden på en linjesymbol, med en linjesubsymbol som används för att återge varje enskilt segment. Med andra ord är en hashad linje som en markörlinje där markörsymboler ersätts med segment. Som sådan har de hashade linjerna *samma egenskaper* som markörlinjesymboler, tillsammans med:
  - *Hash-längd*
  - *Hash rotation*

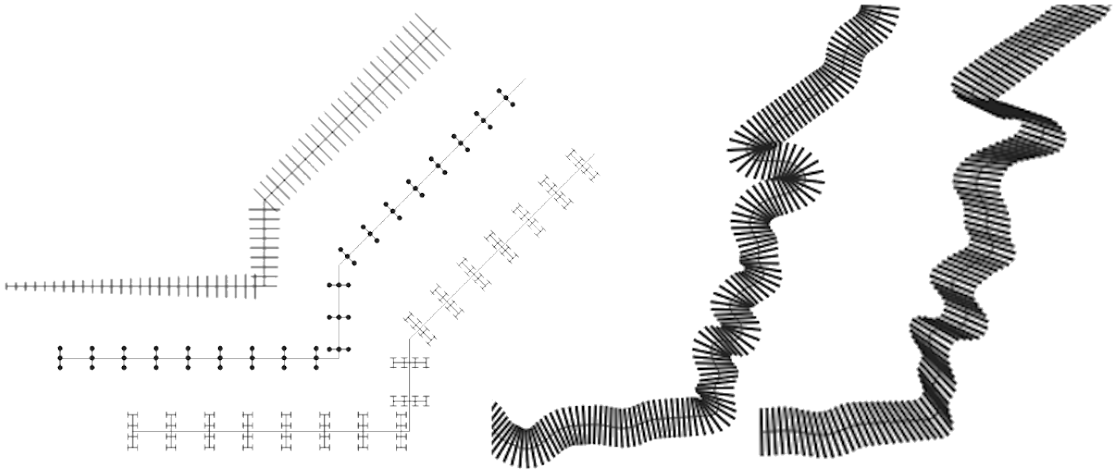


Fig. 10.14: Exempel på hashade linjer

- **Raster line**: Renderar och upprepar en rasterbild som följer längden på en linje. Justeringar kan göras för *Stroke width*, *Offset*, *Join style*, *Cap style* och *Opacity*.

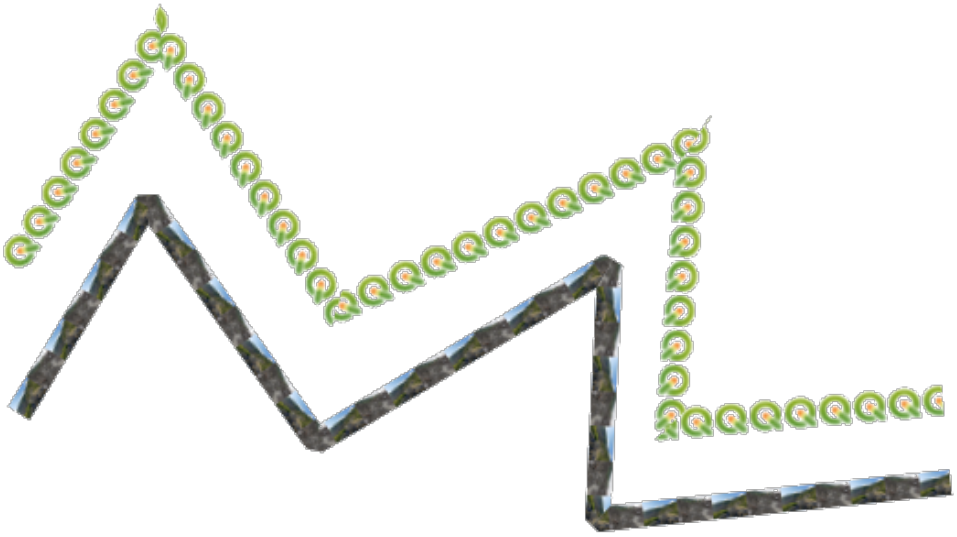


Fig. 10.15: Exempel på rasterlinjer

- **Linear referencing**: gör det möjligt att placera textetiketter med jämna mellanrum längs en linje eller på positioner som motsvarar befintliga hörn. Positionerna kan beräknas med hjälp av kartesiska avstånd eller interpoleras från z/m-värden. Parametrar inkluderar:
  - *Mätning av placering*: Etiketter kan placeras på vanliga kartesiska 2d-avstånd, på vanliga linjärt interpolerade avstånd som beräknas med hjälp av Z- eller M-värdena i geometrier, eller på befintliga hörn. Ett fast eller datadefinierat *Interval* anges.
  - *Kvantitet*: Etiketten som visas vid varje position representerar löpavståndet längs linjen, eller det linjärt interpolerade Z- eller M-värdet. Etiketter renderas med hjälp av hela utbudet av formateringsfunktionerna *text* och *number* som finns tillgängliga i QGIS.

> **Observera**
>
> När du använder avståndsbaserad placering eller etiketter beräknas avstånden endast med 2D. Kartesiska beräkningar baseras på CRS för originallagret.

- *Hoppa över multiplar av*: Om värdet är inställt kommer etiketter som är en multipel av detta värde att hoppas över. Detta gör det möjligt att konstruera komplexa referensetiketter, t.ex. när en symbol har två linjära referenssymbollager, ett för att märka var 100:e meter i ett litet typsnitt och hoppa över multiplar av 1000, och ett andra för att märka var 1000:e meter i ett stort fett typsnitt.
- *Genomsnittlig vinkel över*: Etiketter renderas med en vinkel som beräknas genom att ta medelvärdet av linjesträngen, så att skarpa små jaggies inte resulterar i obetydlig rotation av etiketten.
- *Visa markörsymboler*, vid refererade punkter i linjefunktionen, med hjälp av en fullständig QGIS-markörsymbol. Detta gör det möjligt att t.ex. visa en korsteckning vid den märkta punkten för en linje i ”linjalstil”.

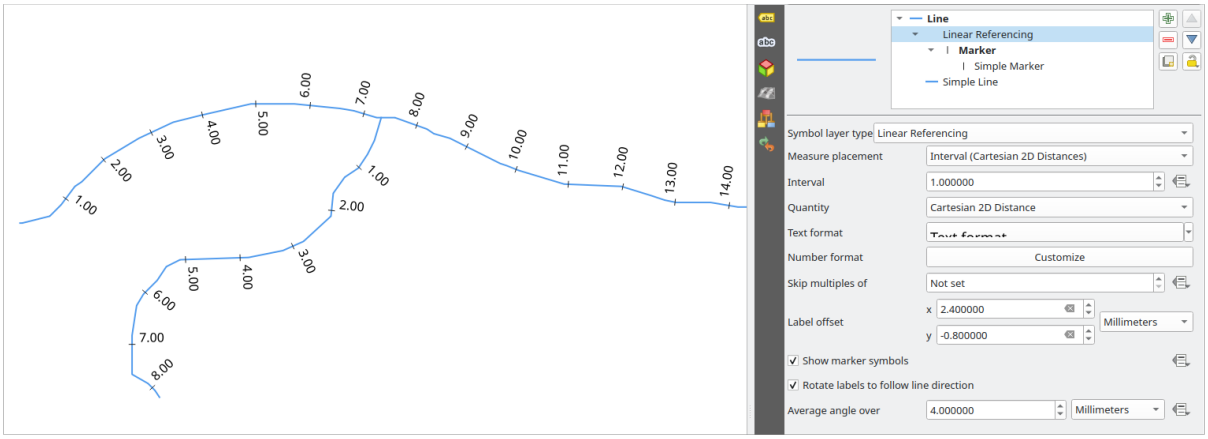


Fig. 10.16: Ritning av linjära etiketter längs linjefunktionen

- **Lineburst**: Renderar en gradient längs bredden på en linje. Du kan välja mellan *Two color* eller *Color ramp* och *Stroke width*, *Offset*, *Join style*, *Cap style* kan justeras.

Fig. 10.17: Exempel på lineburst-linjer

- **Fylld linje**: renderar linjernas inre med hjälp av en fyllningssymbol, vilket möjliggör linjer fyllda med gradienter, linjeluckor etc. Justeringar kan göras för *Stroke width*, *Offset*, *Join style*, *Cap style*.

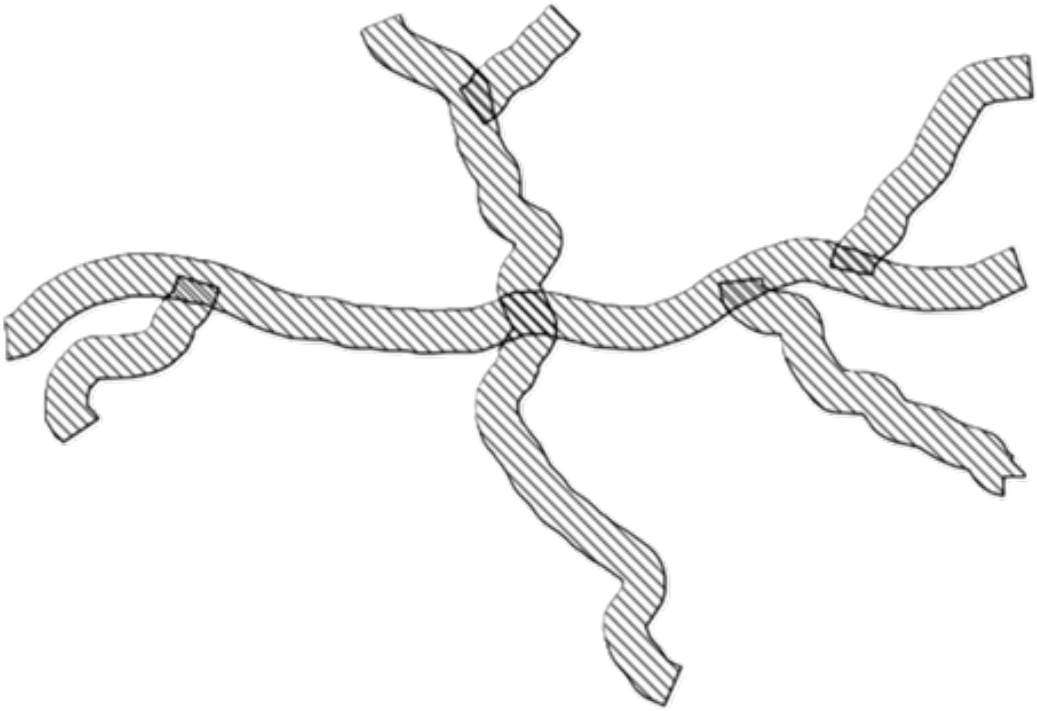


Fig. 10.18: Exempel på fyllda linjer

Fyll symboler

Fyllningssymboler är lämpliga för polygongeometriska funktioner och har också flera symbollagertyper:

- **Simple fill** (standard): fyller en polygon med en enhetlig färg

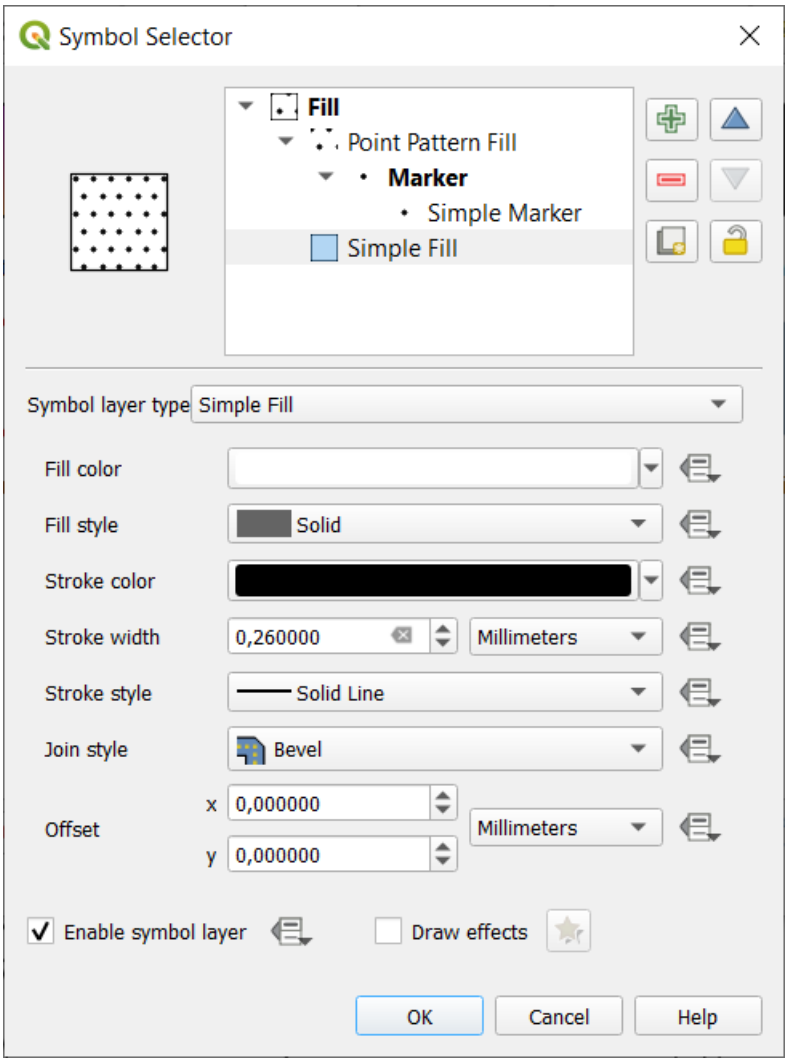


Fig. 10.19: Utforma en enkel fyllningssymbol

- **Centroid fill**: placerar en *markörsymbol* vid centroiden för den synliga funktionen. Markörens position kanske inte är den verkliga centroiden för objektet, eftersom beräkningen tar hänsyn till polygonerna som klippts till det område som är synligt i kartbilden för rendering och ignorerar hål. Använd *geometri-generator-symbolen* om du vill ha den exakta centroiden.

  Det kan du:

  - *Tvinga fram placering av markörer inuti polygoner*
  - *Rita markörer på varje del av flerdelade objekt* eller placera punkten endast på dess största del
  - visa markörsymbolen/markörerna helt eller delvis, och behåll de delar som överlappar den aktuella geometrin (*Clip markers to polygon boundary*) eller den geometriska del som symbolen tillhör (*Clip markers to current part boundary only*)
- **Geometri-generator** (se *Geometri-generator*)
- **Gradient fill**: använder en radiell, linjär eller konisk gradient, baserad på antingen enkla tvåfärgsgradienter eller en fördefinierad *gradient color ramp* för att fylla polygoner. Lutningen kan roteras och tillämpas på en

enskild funktion eller över hela kartans utsträckning. Start- och slutpunkter kan också ställas in via koordinater eller med hjälp av centroiden (för objektet eller kartan). En datadefinierad förskjutning kan definieras.

- **Line pattern fill**: fyller polygonen med ett skrafferingsmönster av *linjesymbolskikt*. Du kan ställa in:
  - *Alignment*: definierar hur mönstret är placerat i förhållande till objektet/objekten:
    - *Align pattern to feature*: linjer återges inom varje feature
    - *Align pattern to map extent*: ett mönster återges över hela kartutbredningen, vilket gör att linjerna kan riktas in snyggt över funktioner
  - *Rotation* av linjerna, moturs
  - *Spacing*: avstånd mellan på varandra följande linjer
  - *Offset* linjernas avstånd från objektgränsen
  - *Clipping*: gör det möjligt att styra hur linjer i fyllningen ska klippas till polygonformen. Alternativen är:
    - *Clip During Render Only*: linjer skapas som täcker hela funktionens avgränsningsbox och klipps sedan under ritningen. Linjens ytterpunkter (början och slut) syns inte.
    - *Clip Lines Before Render*: linjer klipps till polygonens exakta form före rendering. Linjernas ytterpunkter (inklusive lockstilar, start/slut-markörlinjeobjekt, …) kommer att vara synliga och kan ibland sträcka sig utanför polygonen (beroende på inställningarna för linjesymbolen).
    - *No Clipping*: ingen klippning alls görs - linjerna täcker hela objektets avgränsande box
- **Point pattern fill**: fyller polygonen med ett rutnätsmönster av *markörsymbol*. Du kan ställa in:
  - *Alignment*: definierar hur mönstret är placerat i förhållande till objektet/objekten:
    - *Align pattern to feature*: markörlinjer återges inom varje objekt
    - *Align pattern to map extent*: ett mönster återges över hela kartans utsträckning, vilket gör att markörer kan riktas in snyggt över funktioner

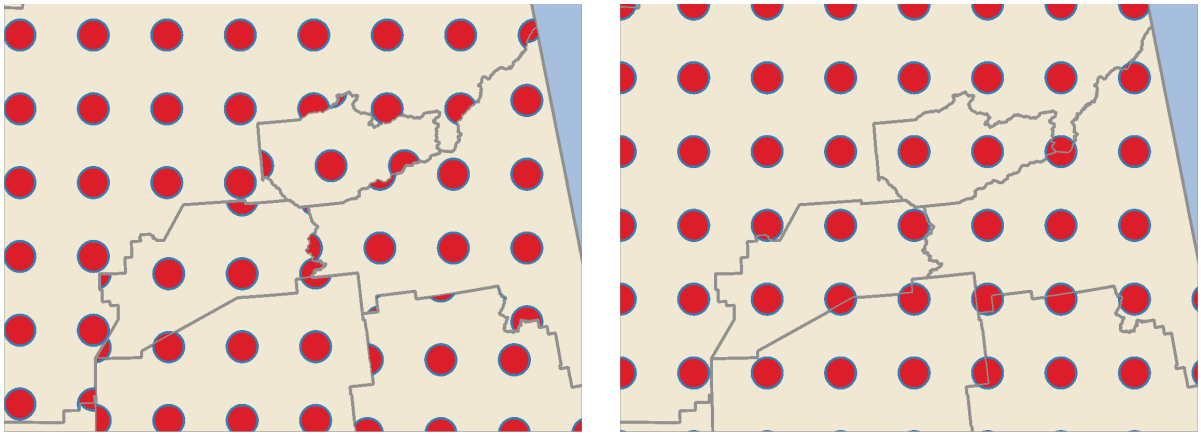


Fig. 10.20: Inriktning av punktmönster mot objekt (vänster) och mot kartutbredning (höger)

  - *Distance*: *Horizontal* och *Vertical* avstånd mellan på varandra följande markörer
  - *Displacement*: en *Horizontal* (resp. *Vertical*) förskjutning av inriktningen mellan på varandra följande markörer i en kolumn (resp. i en rad)
  - *Offset*: *Horisontella* och *Vertikala* avstånd från objektets gräns
  - *Clipping*: gör det möjligt att styra hur markörer i fyllningen ska klippas till polygonformen. Alternativen är:
    - *Clip to shape*: markörerna klipps så att endast delarna inuti polygonen är synliga

    * *Markeringscentrum inom formen*: endast markörer där markörens centrum faller inom polygonen ritas, men dessa markörer kommer inte att klippas till polygonens utsida
    * *Marker completely within shape*: endast markörer som helt och hållet faller inom polygonen visas
    * *Ingen klippning*: alla markörer som överhuvudtaget korsar polygonen kommer att återges fullständigt (strängt taget är det ”korsar markörens avgränsande box”)

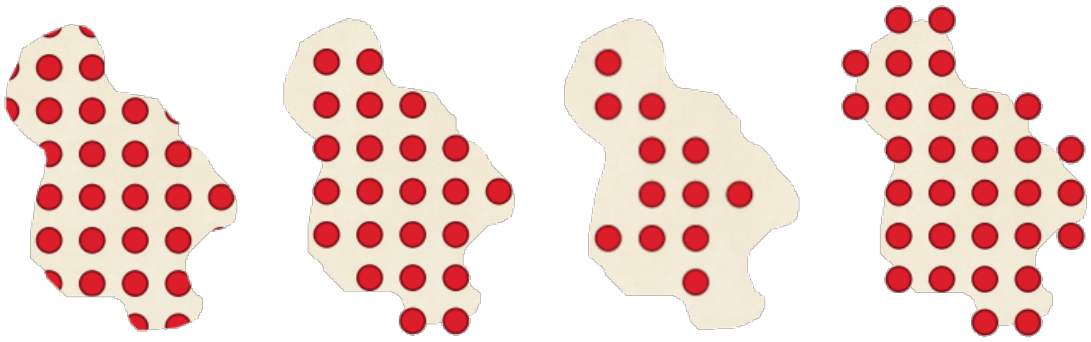


Fig. 10.21: Klippning av markörer i fyllning - Från vänster till höger: Klipp till formen, Markörens tyngdpunkt inom formen, Markören helt inom formen, Ingen klippning

  – *Rotation* av hela mönstret, medurs
  – Med gruppinställningen *Randomize pattern* kan varje punkt i en punktmönsterfyllning slumpmässigt förskjutas upp till det angivna maximala avståndet *Horizontally* eller *Vertically*. Du kan ange den maximala förskjutningen i alla enheter som stöds, t.ex. millimeter, punkter, kartenheter eller till och med ”procent” (där procent är i förhållande till mönstrets bredd eller höjd).

    Du kan ställa in ett valfritt slumptalsfrö för att undvika att symbolmönstren ”hoppar” runt mellan kartuppdateringar. Datadefinierade åsidosättningar stöds också.

    Observera

    Den huvudsakliga skillnaden mellan symboltypen *Randomize pattern* och *random marker fill* är att den slumpmässiga förskjutningen med ett punktmönster möjliggör en kvasi-”regelbunden” placering av markörer - eftersom punkterna i mönstret i praktiken är begränsade till ett rutnät, möjliggör detta skapandet av halvslumpmässiga fyllningar som inte har tomma områden eller överlappande markörer. (I motsats till den slumpmässiga markörfyllningen, som alltid placerar punkter helt slumpmässigt … vilket ibland resulterar i visuella kluster av punkter eller oönskade tomma områden).

- **Random marker fill**: fyller polygonen med en *markörsymbol* som placeras på slumpmässiga platser inom polygonens gränser. Du kan ställa in:
  – *Count method*: om antalet markörsymboler som ska återges ska betraktas som ett absolut antal eller täthetsbaserat
  – *Point count*: antalet markörsymboler som ska återges,
  – ett valfritt slumptal *seed*, för att ge konsekvent placering
  – *Density area*: i händelse av densitetsbaserad räknemetod, säkerställer att fyllningstätheten för markörer förblir densamma på olika skalor / zoomnivåer för markörer när kartor uppdateras (tillåter också slumpmässig placering för att spela bra med QGIS-server och kakelbaserad rendering)
  – *Clip markers to polygon boundary*: om markörer som återges nära polygonens kanter ska klippas till polygonens gräns eller inte
- **Raster image fill**: fyller polygonen med brickor från en rasterbild (`PNG JPG`, `BMP` …). Bilden kan vara en fil på disken, en fjärr-URL eller en inbäddad fil kodad som en sträng (*mer detaljer*). Alternativen

omfattar (datadefinierad) opacitet, storlek, koordinatläge (objekt eller vy), rotation och förskjutning. Under alternativet Size kan du självständigt justera bredden och höjden på fyllningsmönstret, vilket möjliggör utsträckta rasterfyllningar i antingen horisontell eller vertikal riktning. Bildens bredd och höjd kan ställas in med hjälp av någon av de *vanliga enheterna* eller som en procentandel av originalstorleken.

- **SVG-fyllning**: fyller polygonen med *SVG-markörer* av en given storlek (*Texturbredd*).
- **Shapeburst fill**: buffrar en gradientfyllning, där en gradient ritas från polygonens gräns mot polygonens mitt. Konfigurerbara parametrar inkluderar avstånd från gränsen till skuggan, användning av färgramper eller enkla tvåfärgsgradienter, valfri oskärpa i fyllningen och offsets.
- **Outline: Arrow**: använder ett linje *pilsymbol*-lager för att representera polygongränsen. Inställningarna för konturpilen är desamma som för pilens linjesymboler.
- **Outline: Hashed line**: använder ett *hash line symbol*-lager för att representera polygongränsen (*Rings*) som kan vara endast de inre ringarna, endast den yttre ringen eller alla ringarna). De övriga inställningarna för den hashade linjen är desamma som för hashade linjesymboler.
- **Outline: Marker line**: använder ett *marker line symbol*-lager för att representera polygongränsen (*Ringar*) som kan vara endast de inre ringarna, endast den yttre ringen eller alla ringarna). De andra inställningarna för den konturmarkeringslinje är desamma som för markörlinjesymboler.
- **Outline: simple line**: använder ett *simple line symbol*-lager för att representera polygongränsen (*Rings*) som kan vara endast de inre ringarna, endast den yttre ringen eller alla ringarna). Alternativet *Draw line only inside polygon* visar polygongränserna inuti polygonen och kan vara användbart för att tydligt representera angränsande polygongränser. De övriga inställningarna för outline simple line är desamma som för enkla linjesymboler.

Observera

När geometritypen är polygon kan du välja att inaktivera den automatiska klippningen av linjer/polygoner till kanvasens utsträckning. I vissa fall resulterar denna klippning i en ogynnsam symbologi (t.ex. centroidfyllningar där centroiden alltid måste vara den faktiska funktionens centroid).

Parametriserbar SVG

Du har möjlighet att ändra färgerna på en *SVG-markör*. Du måste lägga till platshållarna `param(fill)` för fyllnadsfärg, `param(fill-opacity)` för fyllnadsopacitet, `param(outline)` och `param(outline-opacity)` för streckfärg respektive opacitet och `param(outline-width)` för streckbredd. Dessa platshållare kan eventuellt följas av ett standardvärde, t.ex:

```
<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>
```

Mer allmänt kan SVG parametriseras fritt med hjälp av `param(param_name)`. Denna param kan antingen användas som ett attributvärde eller en nodtext:

```
<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>
```

Parametrarna kan sedan definieras som uttryck i tabellen *Dynamic SVG parameters*.

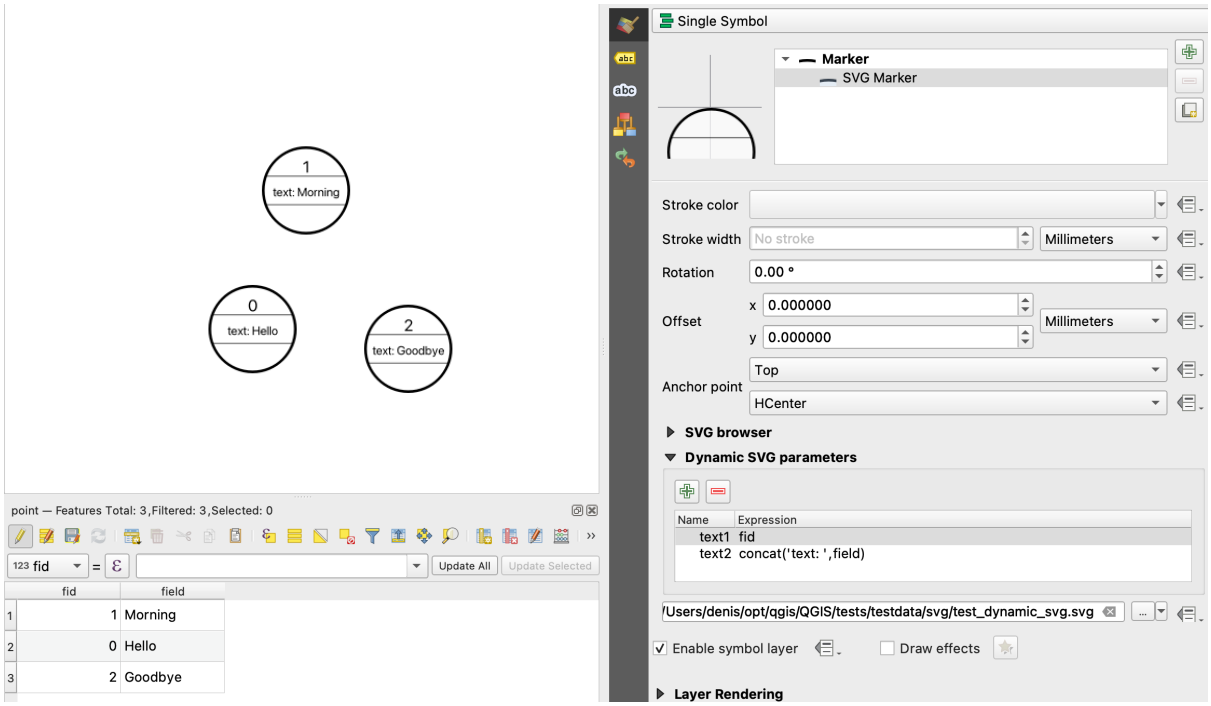


Fig. 10.22: Tabell över dynamiska SVG-parametrar

Observera

QGIS letar efter en komplett SVG-nod. Så om din parameter finns i en mer komplex nod måste du injicera den kompletta noden med uttrycket. Till exempel kommer `transform="rotate(param(angle)"` inte att fungera. Istället måste du göra `transform="param(rotation)"` och parametern `rotation` kommer att definieras med uttrycket `'rotate(' || coalesce(my_field, 0) || ')'`.

Geometri-generator

Symbollagret *geometry generator* är tillgängligt för alla typer av symboler och gör det möjligt att använda *uttryckssyntax* för att generera en geometri i farten under renderingsprocessen. Den resulterande geometrin behöver inte matcha den ursprungliga *Geometrityp* och du kan lägga till flera olika modifierade symbollager ovanpå varandra.

En *Units*-egenskap kan ställas in: när geometri-generatorsymbolen inte används på ett lager (t.ex. om den används på ett layoutobjekt), ger detta mer kontroll över den genererade utdata.

Några exempel:

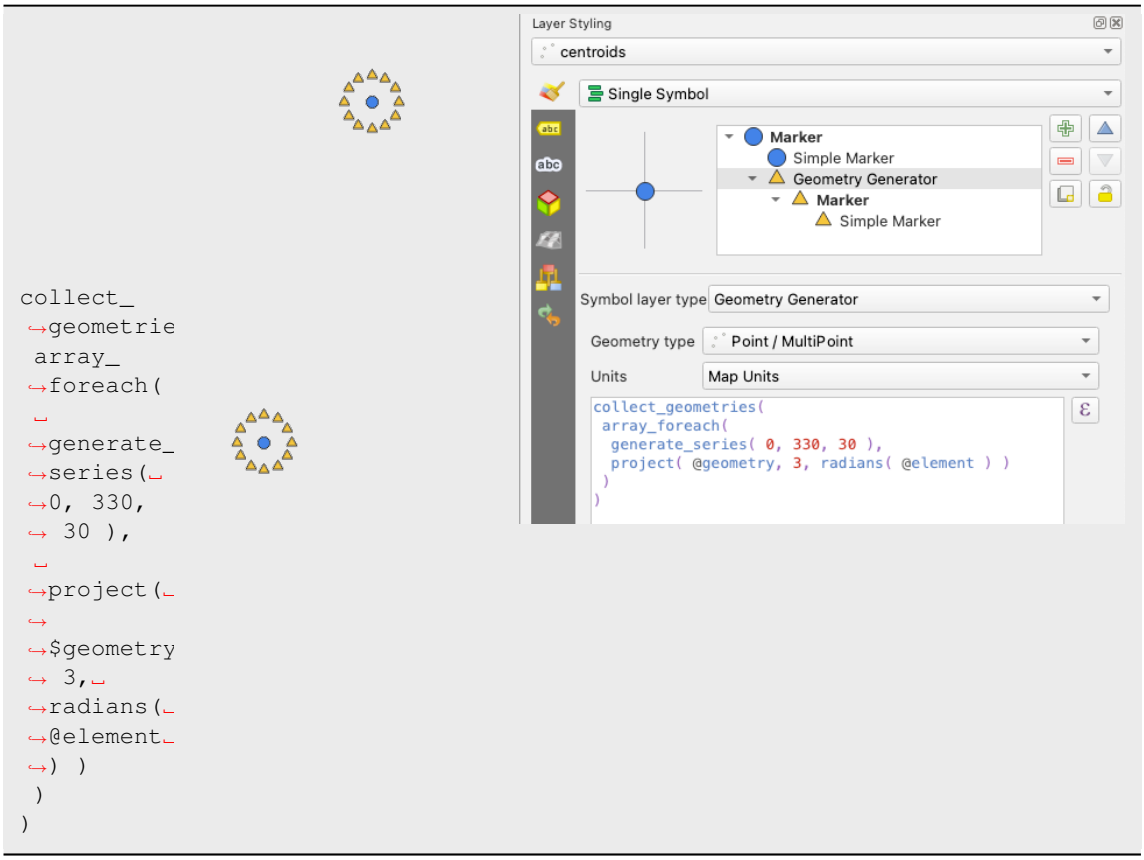
- Återge symbolen som tyngdpunkten för ett objekt

```
centroid( $geometry )
```

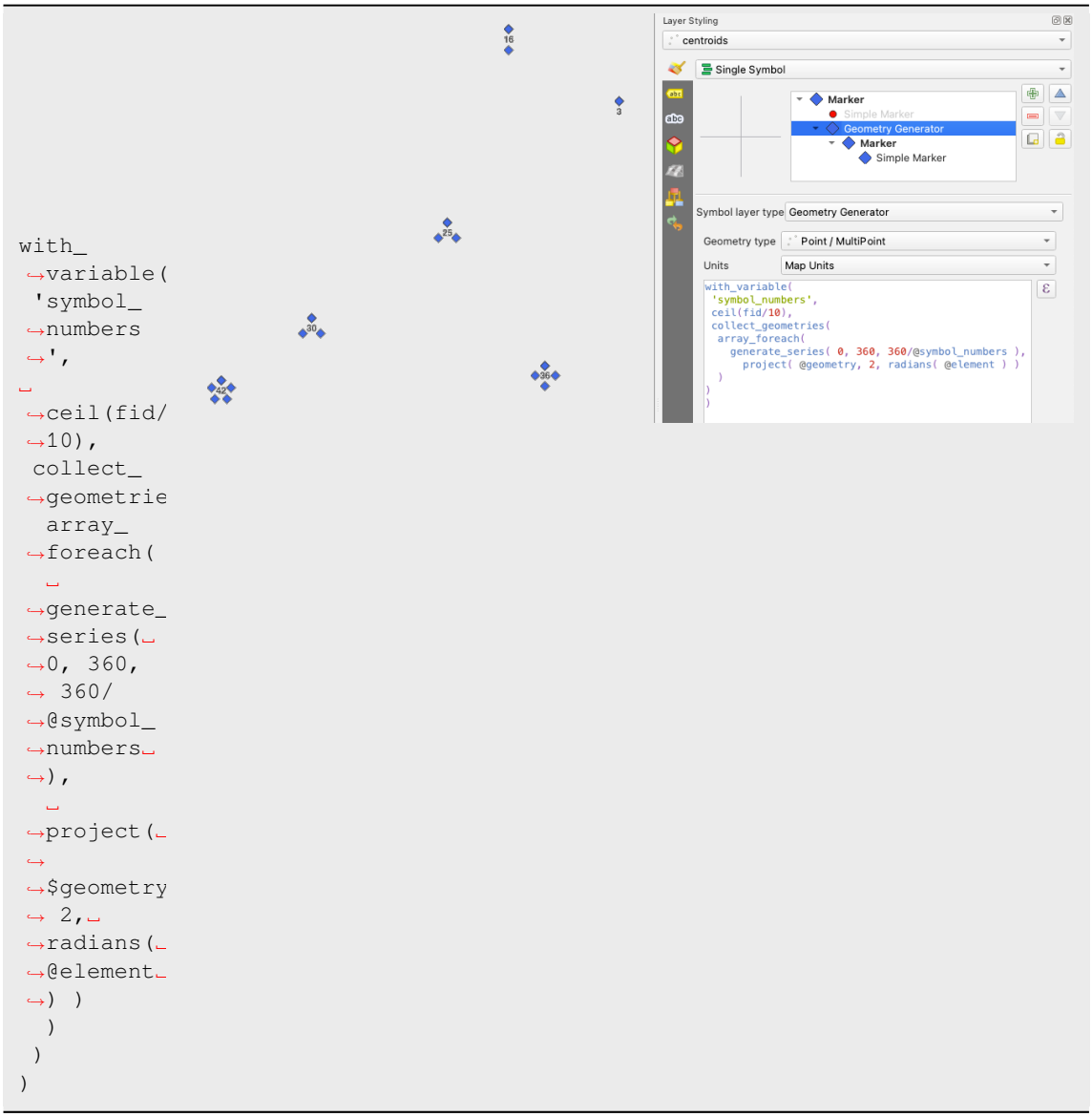
- Visuell överlappning av funktioner inom ett avstånd på 100 kartenheter från en punktfunktion, dvs. skapa en buffert på 100 m runt punkten

```
buffer( $geometry, 100 )
```

- Skapa en radiell effekt av punkter som omger den centrala funktionspunkten när den används som en MultiPoint geometri-generator

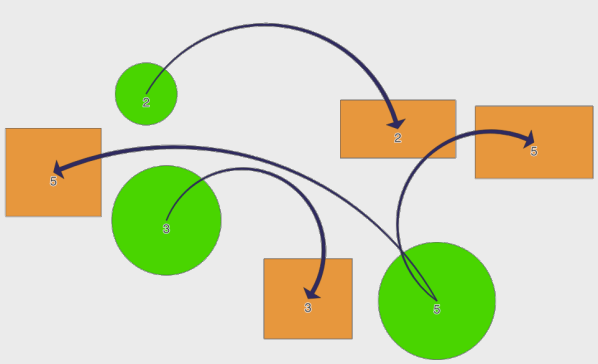


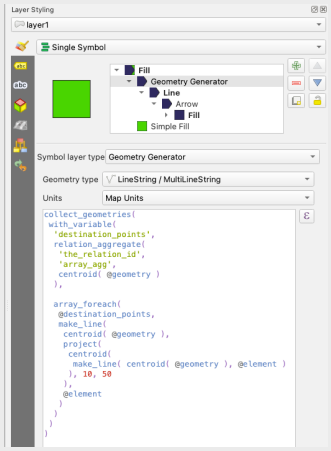
- Skapa en radiell effekt av punkter som omger den centrala funktionspunkten. Antalet punkter varierar beroende på fält.




```
with_
↪variable(
 'symbol_
↪numbers
↪',
␣
↪ceil(fid/
↪10),
 collect_
↪geometrie
  array_
↪foreach(
  ␣
↪generate_
↪series(␣
↪0, 360,
↪ 360/
↪@symbol_
↪numbers␣
↪),
  ␣
↪project(␣
↪
↪$geometry
↪ 2,␣
↪radians(␣
↪@element␣
↪) )
  )
 )
)
```

- Skapa en böjd pil som förbinder funktioner i två lager baserat på deras *relation*






```
collect_
↪geometrie
 with_
↪variable(

↪'destinat
↪points
↪',
 ␣
↪relation_
↪aggregate
   'the_
↪relation_
↪id',

↪'array_
↪agg',
  ␣
↪centroid(
↪
↪$geometry
↪)
  ),
  array_
↪foreach(
  ␣
↪@destinat
↪points,
   make_
↪line(
   ␣
↪centroid(
↪@geometry
↪),
   ␣
↪project(
    ␣
↪centroid(
     ␣
↪make_
↪line(␣
↪centroid(
↪@geometry
↪),␣
↪@element␣
↪)
     ),
     10,␣
↪50
    ),
   ␣
↪@element
   )
  )
 )
)
```

Vector Field-markören

Vektorfältmarkören används för att visa vektorfältdata såsom jorddeformation, tidvattenflöden och liknande. Vektorerna visas som linjer (helst pilar) som är skalade och orienterade enligt valda attribut för datapunkter. Den kan endast användas för att återge punktdata; linje- och polygonlager ritas inte med denna symbologi.

Vektorfältet definieras av attribut i data, som kan representera fältet antingen genom:

- **kartesiska** komponenter (fältets x- och y-komponenter)
- eller **polära** koordinater: i detta fall definierar attributen `Längd` och `Vinkel`. Vinkeln kan mätas antingen medurs från norr eller moturs från öster och kan anges antingen i grader eller radianer.
- eller som data med endast **höjd**, vilket visar en vertikal pil som skalats med hjälp av ett attribut för data. Detta är lämpligt för att till exempel visa den vertikala komponenten av deformation.

Fältets storlek kan skalas upp eller ner till en lämplig storlek för visning av fältet.

Animerad markör

Med symboltypen Animated marker kan du använda en animationsfil av typen `.GIF`, `.WebP`, `.MNG` etc. för att representera punkter på kartan. Du kan ange:

- *File* sökväg,
- *Frame rate*: antal steg som visas per sekund, vilket anger hur snabbt animationen spelas upp,
- *Storlek* i valfri *understödd enhet*,
- *Opacity*,
- *Rotation*,
- *Offset* i riktningarna *x* och *y* från markörens position,
- *Anchor point*

Det finns två sätt att hantera animerade symboler:

- **När din karta inte är konfigurerad som en animation** (dvs. det är ett standard QGIS-projekt utan animationer), kommer ramen för de animerade markörerna att bestämmas enbart av den aktuella tidsstämpeln.
- **När din karta är** *konfigurerad som en animation*, synkroniseras de animerade markörerna med animationens tidslinje. Det innebär att de animerade markörerna pausar när animationen pausas, fortsätter med animationen och så vidare. Kartan ritas också om i enlighet med den bildfrekvens som fastställts för den temporala animeringen. Detta läge tillämpas även när du exporterar en animering med hjälp av den temporala styrenheten.

10.3 Ställa in en etikett

Etiketter är textinformation som du kan visa på vektorfunktioner, mesh-lager eller kartor. De lägger till detaljer som du inte nödvändigtvis kan representera med hjälp av symboler. Två typer av textrelaterade objekt finns tillgängliga i QGIS:

- *Textformat*: definierar utseendet på texten, inklusive *font, storlek, färger*, *skugga*, *bakgrund*, *buffert*, …

  De kan användas för att rendera texter över kartan (layout/karttitel, dekorationer, skalfält, …), vanligtvis genom widgeten *font*.

  Så här skapar du en *Text Format*-artikel:

  1. Öppna dialogrutan *Stilhanterare*

2. Aktivera fliken *Textformat*

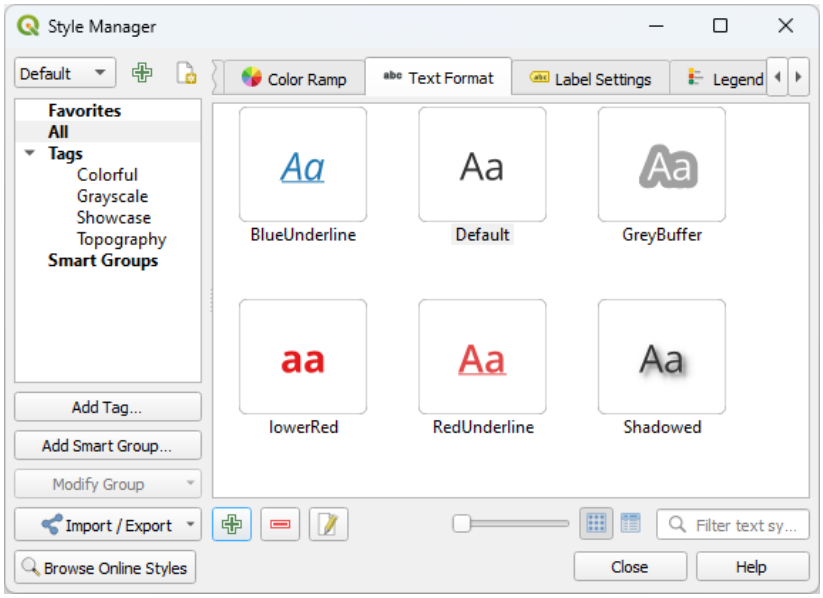


Fig. 10.23: Textformat i dialogen Style Manager

3. Tryck på knappen Add item. Dialogrutan *Text Format* öppnas för *konfiguration*. Som vanligt är dessa egenskaper *data-definierbar*.

- *Label Settings*: utöka inställningarna för textformat med egenskaper relaterade till platsen eller interaktionen med andra texter eller funktioner (*callouts*, *placement*, *overlay, scale visibility*, mask …).

  De används för att konfigurera smart märkning för vektor- och mesh-lager via fliken *Labels* i vektor- eller mesh-dialogrutan *Lageregenskaper* eller panelen *Layer Styling* eller med knappen Layer Labeling Options i verktygsfältet *Label*.

  För att skapa ett *Label Settings*-objekt:

  1. Öppna dialogrutan *Stilhanterare*
  2. Aktivera fliken *Etikettinställningar*

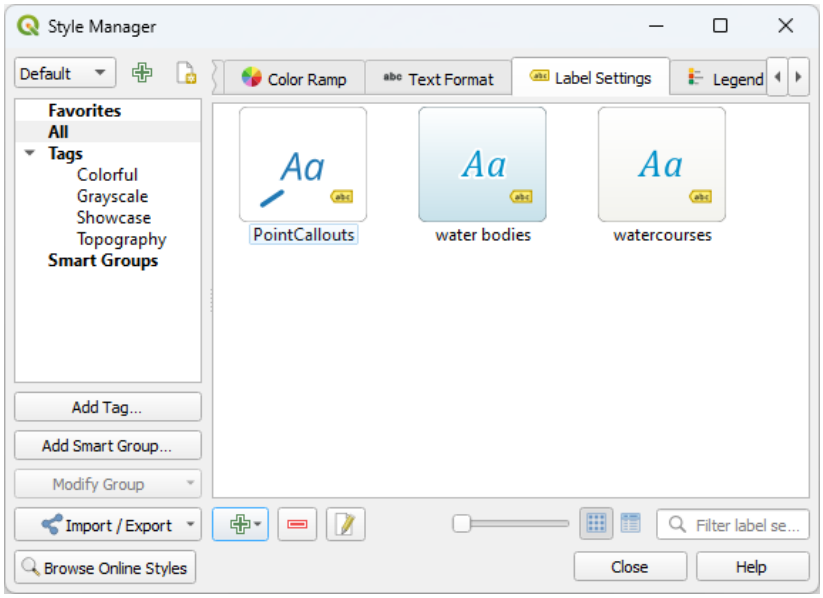


Fig. 10.24: Etikettinställningar i dialogen Style Manager

3. Tryck på menyn Add item och välj den post som motsvarar geometritypen för de objekt som du vill märka.

Dialogrutan *Label Settings* öppnas med följande egenskaper. Som vanligt är dessa egenskaper *data-definierbar*.

10.3.1 Formatering av etikettexten

Oavsett om du konfigurerar ett *Text Format*- eller *Label Settings*-objekt får du följande alternativ:

Fliken Egenskaper	Textformat	Inställningar för etiketter
Text	☑	☑
Formatering	☑	☑
Buffer	☑	☑
Mask		☑
Bakgrund	☑	☑
Shadow	☑	☑
Callout		☑
Placering		☑
Rendering		☑

Fliken Text

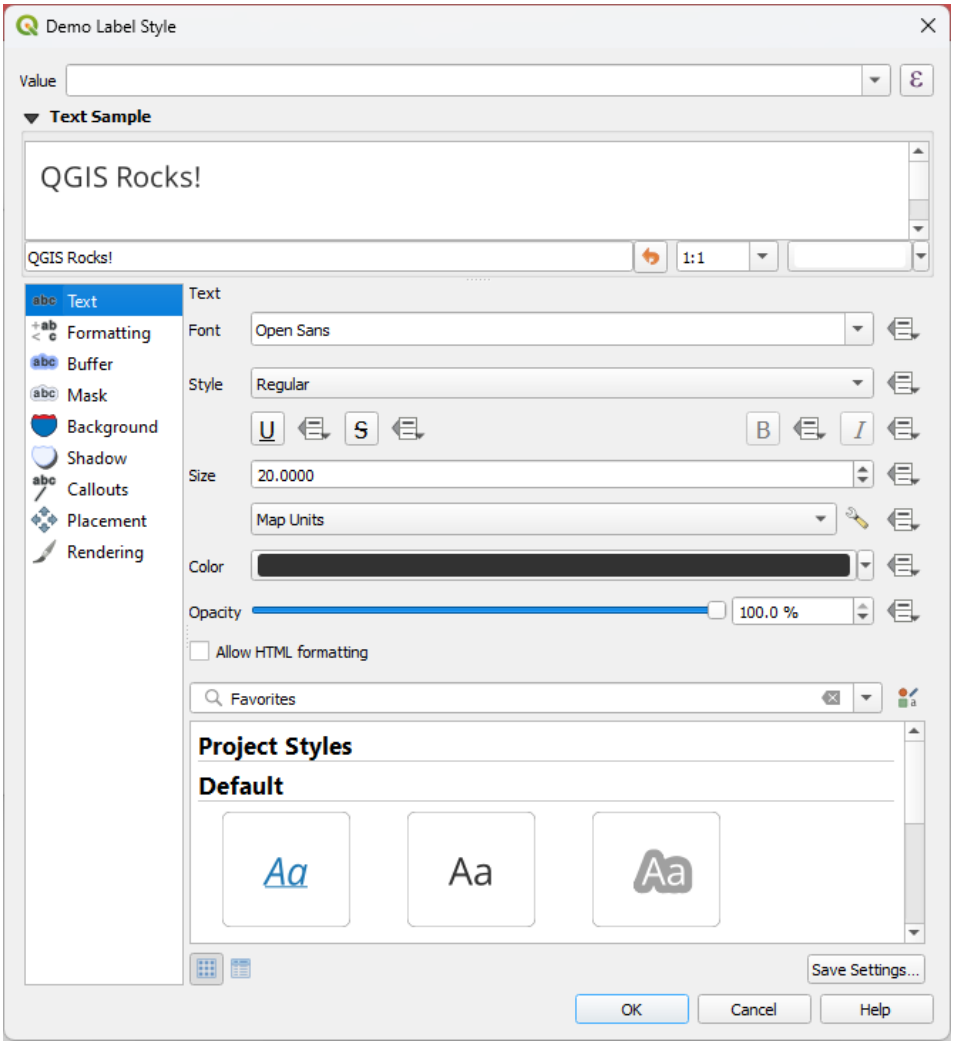


Fig. 10.25: Inställningar för etiketter - fliken Text

På fliken abc *Text* kan du ställa in:

- *Font*, bland de som finns tillgängliga på din maskin
- *Style*: tillsammans med de vanliga stilarna för teckensnittet kan du ställa in om texten ska vara understruken eller genomstruken
- *Size* i valfri *stödd enhet*
- *Färg*
- den *Opacity*
- och *Allow HTML Formatting* gör det möjligt att använda en delmängd HTML-taggar och CSS-regler för att anpassa etiketten.

Längst ner på fliken finns en widget som visar en filtrerbar lista över kompatibla objekt som finns lagrade i din *stilhanteringsdatabas*. Detta gör att du enkelt kan konfigurera det aktuella textformatet eller etikettinställningen baserat på en befintlig inställning och även spara ett nytt objekt i stildatabasen: Tryck på knappen *Spara format…* eller *Spara inställningar…* och ange ett namn och en eller flera taggar.

Observera

När du konfigurerar ett *Etikettinställningar*-objekt är textformatsobjekt också tillgängliga i den här widgeten. Välj ett för att snabbt skriva över den aktuella *textuella egenskaper* för etiketten. På samma sätt kan du skapa/skriva över ett textformat därifrån.

Tillåt HTML-formatering

När *Allow HTML Formatting* är aktiverat måste du ange HTML-koden i fältet *Value*. Använd blanksteg i stället för tabbar för alla typer av indrag. Uttrycket analyseras och alla HTML-taggar som stöds åsidosätter motsvarande inställning i egenskaperna för etiketter. Eftersom det är omöjligt att lista och beskriva varje HTML-tagg och CSS-egenskap som QGIS för närvarande stöder, uppmanar vi dig att utforska och testa i dina etiketter de som stöds av det underliggande Qt-biblioteket.

Exempel på HTML-taggar som stöds:

- Textformatering, t.ex. kursiv eller fet stil, t.ex:

```
<i>QGIS</i> <b>rocks!</b>
```

- Superscript och subscript, där texten kommer att vara vertikalt $^{\text{super}}$ eller $_{\text{sub}}$ justerad och automatiskt dimensionerad till 2/3 av den överordnade teckensnittsstorleken. Du kan också ställa in en fast teckenstorlek för superscript/subscript genom att inkludera css-regler, t.ex:

```
<sup style="font-size:33pt">my superscript text</sup>
```

- Horisontell justering av text, med hjälp av antingen HTML-attributet `align="xxx"` eller taggen `center`. För HTML5-kompatibilitet, använd hellre CSS-egenskapen `text-align`.

Uppmärksamma

Horisontell justering kan inte användas för böjda etiketter.

- Strukturering av ett textblock med hjälp av rubriktaggar (t.ex. `h1`, `h2`, …) eller stycken (med `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>
```

- Bildinläggning: alla bildformat som kan läsas av QGIS kan användas i HTML-etikettinnehåll. De kan hämtas från lokala filsökvägar, HTTP-länkar eller inbäddat innehåll i base64 med attributet `src="xxx"`. Bildstorlekar kan anges via attributen `width="##"` och `height="##"`, i enheten `points`. Om bredd eller höjd inte anges kommer den automatiskt att beräknas utifrån originalbildens storlek. Bilder placeras endast inline, inte som flytande bilder och inte på böjda textetiketter.

```
<img src="qgis.png" width=40 height=60>
```

Exempel på CSS-egenskaper som stöds:

- Font-egenskaper (`färg`, `font-familj`, `font-storlek`, `font-vikt`, `font-stil`, `ordavstånd`). Observera att `word-spacing` alltid kommer att använda enhetspunkter.
- Textdekorationer som understrykning, överstrykning och genomgående linje (`text-decoration`)
- Textjustering (`vertical-align`, `text-align`). Horisontell justering kan inte användas för böjda etiketter.

CSS-egenskaper kan ställas in på HTML-taggar med attributet `style`. HTML-taggen `span` tillämpar inte någon formatering av text i sig och är idealisk om du bara vill tillämpa CSS-styling. Ett namn på en CSS-egenskap och dess värde separeras med ett kolon (`:`). Flera CSS-egenskaper separeras med semikolon (`;`), t.ex:

```
<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>
```

Nedan följer ett exempel på ett HTML-baserat uttryck och rendering (använder olika färger och understrykning för samma etikett):

```
format(
  '<span style="color:blue">%1</span> ( <span style="color:red"><u>%2 ft</u></span>
↪ )',
  title( lower( "Name" ) ),
  round($length)
)
```

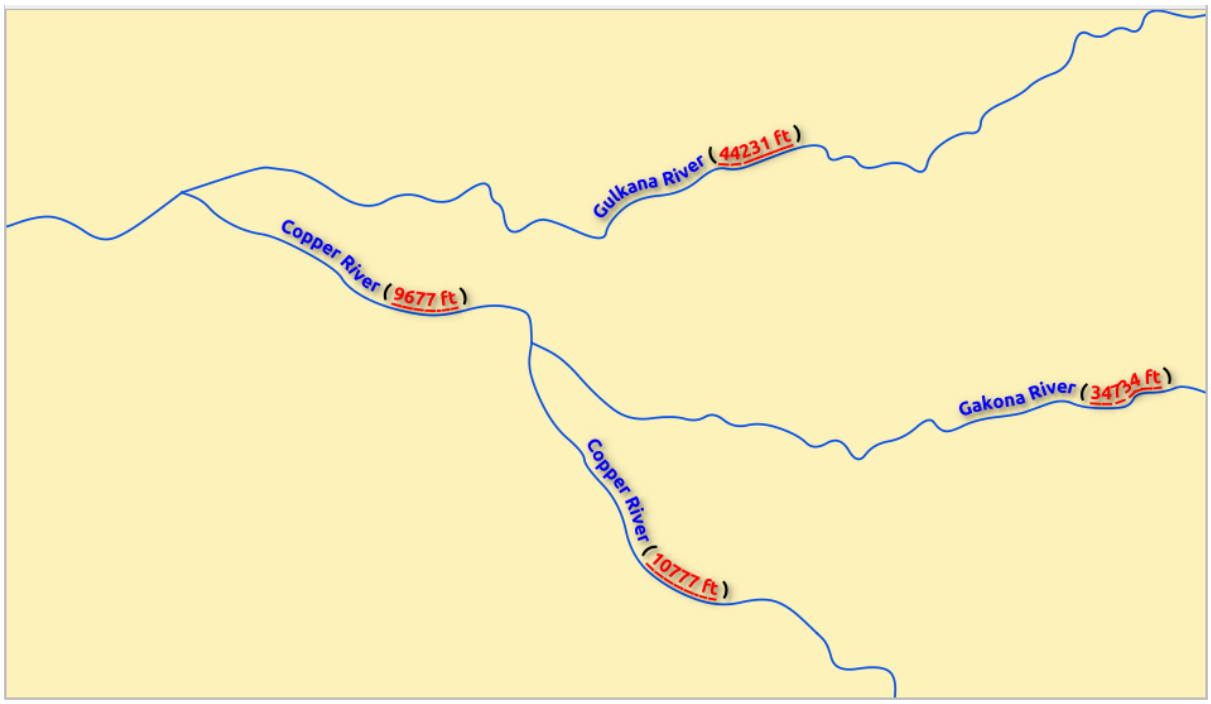


Fig. 10.26: Etikettering med HTML-formatering aktiverad

Fliken Formatering

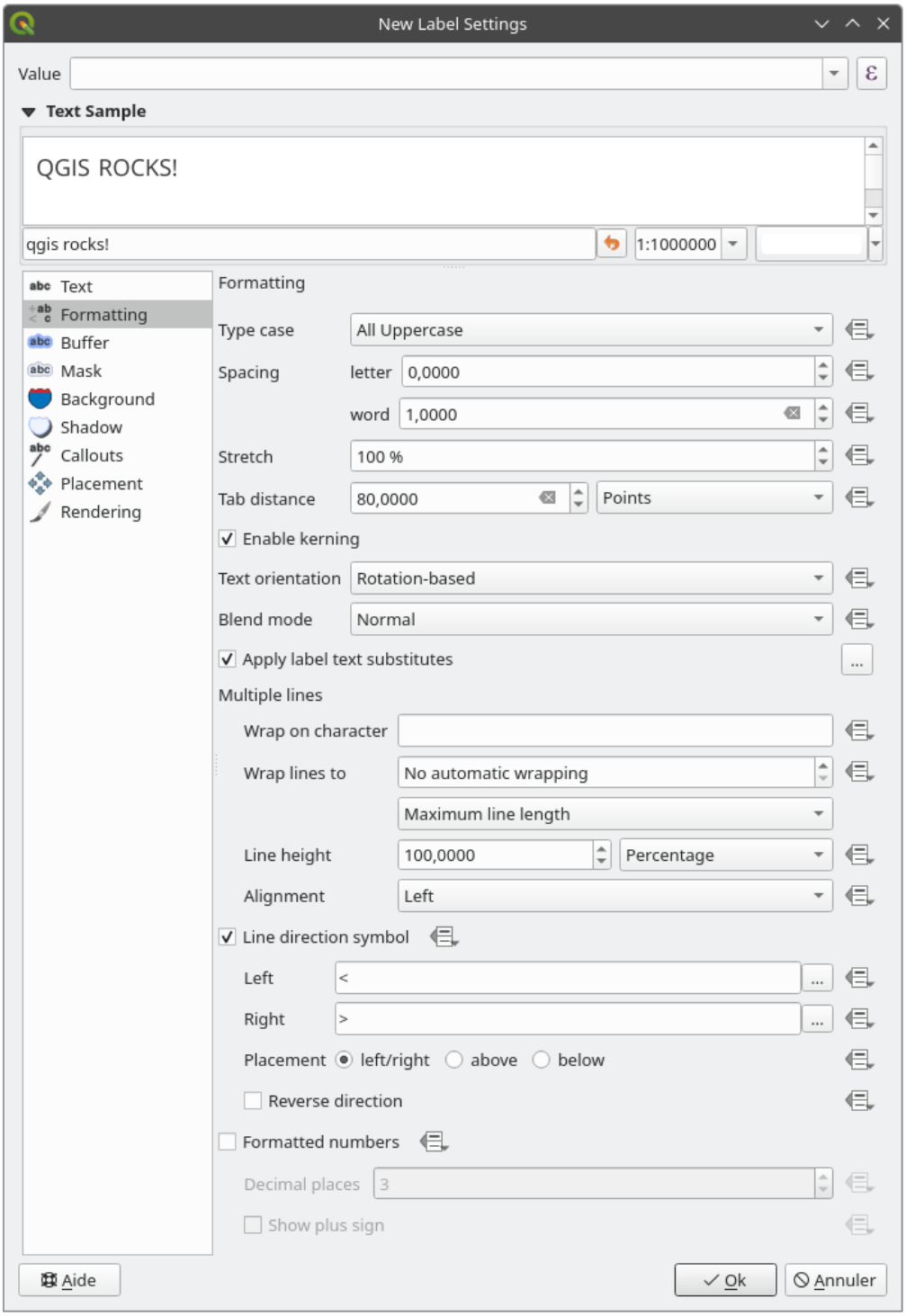


Fig. 10.27: Etikettinställningar - fliken Formatering

På fliken *Formatting* kan du göra följande:

- Använd alternativet *Type case* för att ändra stilen för versaler i texten. Du har möjlighet att återge texten som:
  - *Ingen förändring*
  - *Alla versaler*

  - *Alla gemener*
  - *Title case*: ändrar den första bokstaven i varje ord till versal och gör de andra bokstäverna till gemener om originaltexten använder en enda typ av versaler. Om det finns blandade versaler i texten lämnas de andra bokstäverna orörda.
  - *Tvinga första bokstaven till versal*: ändrar första bokstaven i varje ord till versal och lämnar övriga bokstäver i texten orörda.
- Under *Spacing* kan du ändra avståndet mellan ord och mellan enskilda bokstäver.
- Med *Tab distance* kan du justera ersättningsavståndet för `Tab`-tecken i etiketten, i alla enheter som stöds. Detta gör det t.ex. möjligt att separera eller vertikalt anpassa element i en layouttext.
- *Stretch* ratio: gör att texten kan sträckas ut eller förminskas horisontellt med en faktor. Praktiskt för att justera bredden på teckensnitt för att få plats med lite extra text i etiketter.
- *Aktivera kerning* av texttypsnittet
- Ställ in *Text orientation* som kan vara *Horizontal* eller *Vertical*. Den kan också vara *Rotation-based* när du ställer in en etikett (t.ex. för att korrekt märka linjefunktioner i *parallel* placeringsläge).
- Använd alternativet *Blend mode* för att bestämma hur dina etiketter ska blandas med kartfunktionerna under dem (mer information finns i *Blandningslägen*).
- Med alternativet *Apply label text substitutes* (ej markerat) kan du ange en lista med texter som ska ersätta texter i funktionsetiketter (t.ex. förkortning av gatutyper). Ersättningstexter används när etiketter visas på kartan. Användare kan även exportera och importera listor med ersättningstexter för att underlätta återanvändning och delning.
- Konfigurera *Multiple lines*:
  - Ange ett tecken som tvingar fram en radbrytning i texten med alternativet *Wrap on character*
  - Ange en idealisk linjestorlek för automatisk lindning med alternativet *Wrap lines to*. Storleken kan representera antingen *Maximala linjelängden* eller *Minimala linjelängden*.
  - Bestäm *Line Height*: värden kan anges i *Millimeters*, *Points*, *Pixels*, *Percentage* eller *Inches*. När radhöjden är inställd på procent är det procentandelen av standardradavståndet för texten i den teckensnittsfamiljen. Vanligtvis 1,2 till 1,5 gånger textstorleken.
  - Formatera *Alignment*: typiska tillgängliga värden är *Left*, *Right*, *Justify* och *Center*.

    När du ställer in egenskaper för punktetiketter kan textjusteringen också vara *Följ etikettplacering*. I så fall kommer justeringen att bero på etikettens slutliga placering i förhållande till punkten. Om etiketten t.ex. placeras till vänster om punkten kommer den att vara högerjusterad, medan den kommer att vara vänsterjusterad om den placeras till höger.

> **Observera**
>
> Formateringen *Multiple lines* stöds ännu inte av kurvbaserade *label placement*. Alternativen kommer då att avaktiveras.

- För linjeetiketter kan du inkludera *Line direction symbol* för att hjälpa till att bestämma linjens riktning, med symboler att använda för att ange *Left* eller *Right*. De fungerar särskilt bra när de används med placeringsalternativen *curved* eller *Parallel* från fliken *Placement*. Det finns alternativ för att ställa in symbolens position och för att avmarkera *Reverse direction*.
- Använd alternativet *Formatted numbers* för att formatera numeriska texter. Du kan ställa in antalet *Decimaler*. Som standard kommer `3` decimaler att användas. Använd kryssrutan *Visa plustecken* om du vill visa plustecknet för positiva tal.

Fliken Buffert

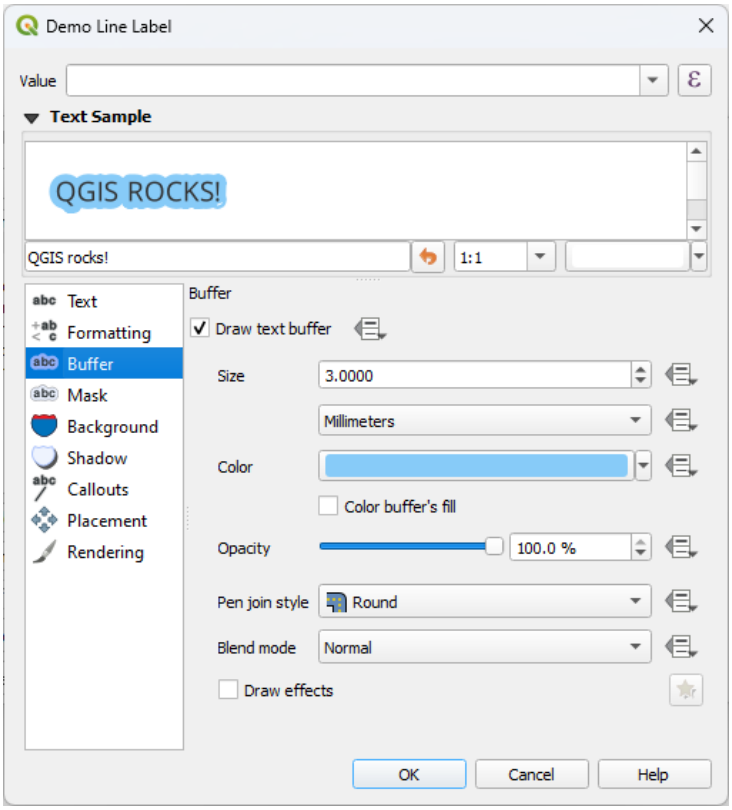


Fig. 10.28: Etikettinställningar - fliken Buffert

För att skapa en buffert runt etiketten aktiverar du kryssrutan *Draw text buffer* på fliken *Buffer*. Sedan kan du göra det:

- Ställ in buffertens *Size* i valfri *understödd enhet*
- Välj buffertens *Färg*
- *Färg buffertens fyllning*: Bufferten expanderar från etikettens kontur, så om alternativet är aktiverat fylls etikettens inre. Detta kan vara relevant när du använder delvis transparenta etiketter eller med icke-normala blandningslägen, vilket gör det möjligt att se bakom etikettens text. Om du avmarkerar alternativet (när du använder helt transparenta etiketter) kan du skapa konturerade textetiketter.
- Definiera buffertens *Opacity*
- Använd en *Pen join style*: den kan vara *Round*, *Miter* eller *Bevel*
- Använd alternativet *Blend mode* för att bestämma hur etikettens buffert ska blandas med kartkomponenterna under dem (mer information finns i *Blandningslägen*).
- Check *Draw effects* för att lägga till avancerade *paint effects* för att förbättra textens läsbarhet, t.ex. genom yttre glöd och oskärpa.

Fliken Bakgrund

På fliken *Background* kan du konfigurera en form som ligger under varje etikett. För att lägga till en bakgrund aktiverar du kryssrutan *Draw Background* och väljer typen *Shape*. Den kan vara:

- en regelbunden form som *Rectangle*, *Square*, *Circle* eller *Ellipse* med fullständiga egenskaper hos en *fill symbol*
- en *SVG*-symbol från en fil, en URL eller inbäddad i projekt- eller stildatabasen (*mer information*)
- eller en *Markeringssymbol* som du kan skapa eller välja från *symbolbiblioteket*.

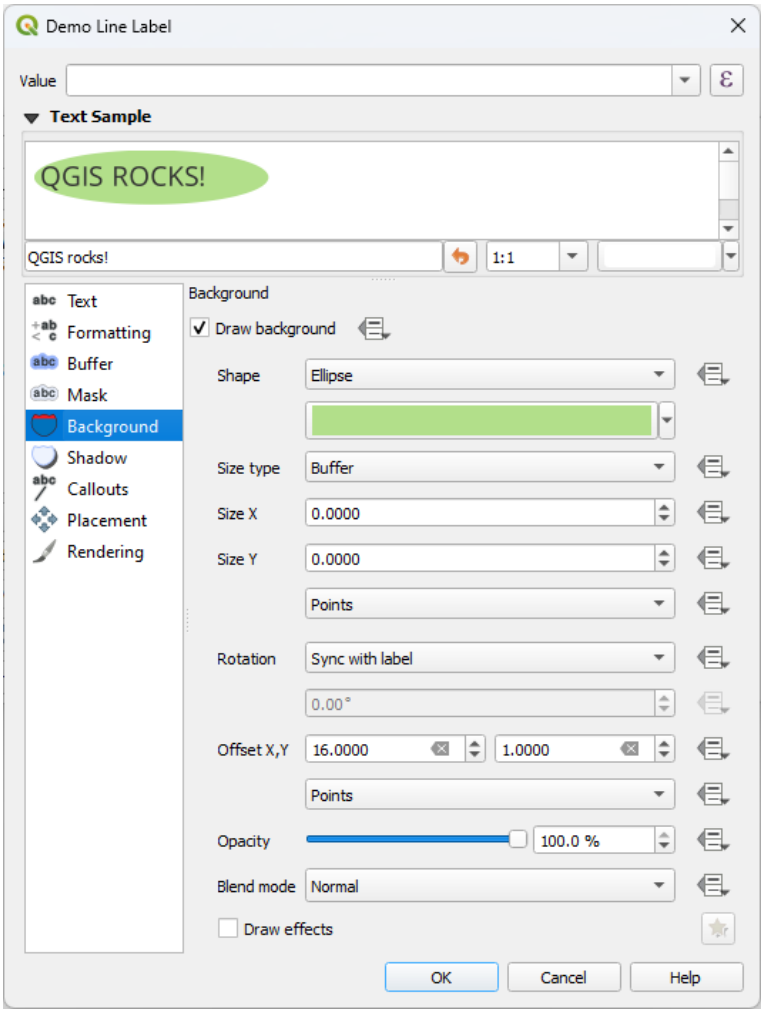


Fig. 10.29: Etikettinställningar - fliken Bakgrund

Beroende på vilken form du har valt måste du konfigurera några av följande egenskaper:

- Ramens *Storlekstyp*, som kan vara:
  - *Fixed*: använder samma storlek för alla etiketter, oavsett textens storlek
  - eller en *Buffer* över textens begränsningsruta
- Ramens *Storlek* i X- och Y-riktningarna, med hjälp av alla *enheter som stöds*
- En *Rotation* av bakgrunden, mellan *Sync with label*, *Offset of label* och *Fixed*. De två sista kräver en vinkel i grader.
- En *Offset X,Y* för att flytta bakgrundsobjektet i X- och/eller Y-riktningen
- En *Radius X,Y* för att runda hörnen på bakgrundsformen (gäller endast rektanglar och kvadratiska former)

- En *Opacity* för bakgrunden
- En *Blend mode* för att blanda bakgrunden med de andra objekten i renderingen (se *Blandningslägen*).
- För SVG-symboler kan du använda dess standardegenskaper (*Load symbol parameters*) eller ange en egen *Fill color*, *Stroke color* och *Stroke width*.
- *Draw effects* för att lägga till avancerade *paint effects* för att förbättra textens läsbarhet, t.ex. genom yttre glöd och oskärpa.

Fliken Skugga

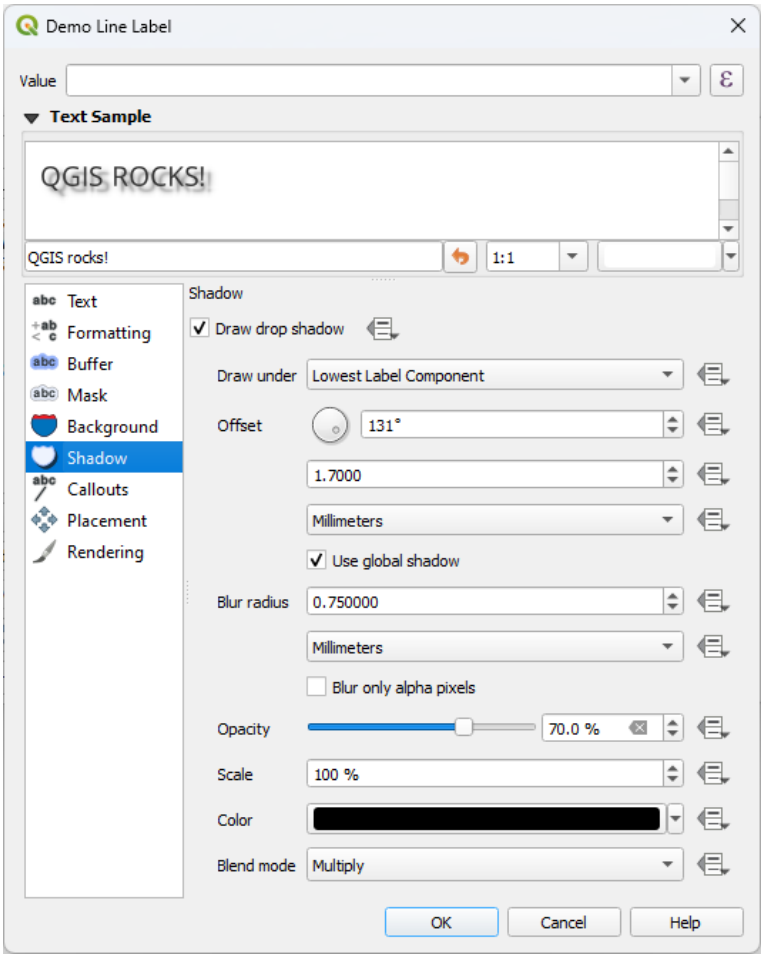


Fig. 10.30: Etikettinställningar - fliken Skugga

För att lägga till en skugga till texten, aktivera fliken *Skugga* och aktivera kryssrutan *Rita släppskugga*. Då kan du göra:

- Ange det objekt som används för att generera skuggan med *Draw under*. Det kan vara *Lowest label component* eller en särskild komponent som *Text* själv, *Buffer* eller *Background*.
- Ställ in skuggans *Offset* från det objekt som skuggas, dvs:
  - Vinkeln: medurs, det beror på det underliggande objektets orientering
  - Förskjutningens avstånd från det objekt som skuggas
  - Enheterna för offset

Om du markerar kryssrutan *Use global shadow*, är vinkelns nollpunkt alltid orienterad mot norr och beror inte på orienteringen av etikettens objekt.

- Påverka skuggans utseende med *Blur radius*. Ju högre siffra, desto mjukare blir skuggorna, i de enheter du väljer.
- Definiera skuggans *Opacity*
- Skala om skuggans storlek med hjälp av faktorn *Scale*
- Välj skuggans *Färg*
- Använd alternativet *Blend mode* för att bestämma hur skuggan av din etikett ska blandas med kartkomponenterna under dem (mer information finns i *Blandningslägen*).

10.3.2 Konfigurera interaktion med etiketter

Utöver de inställningar för textformatering som beskrivs ovan kan du också ställa in hur etiketter ska interagera med varandra eller med funktionerna.

Fliken Mask

På fliken *Mask* kan du definiera ett maskeringsområde runt etiketterna. Denna funktion är mycket användbar när du har överlappande symboler och etiketter med liknande färger och du vill göra etiketterna synliga. En etikettmask hindrar angivna funktioner från att ritas inom den gräns som anges för masken. Du kan t.ex. ställa in en etikettmask så att ett visst lager inte ritar inom 2 mm från etiketten, men tillåta att funktioner från ett annat lager fortfarande visas. Etikettmasker liknar etikettbuffertar på så sätt att de gör det möjligt att kontrollera läsbarheten hos etiketter som täcker andra funktioner. Etikettbufferten ritar ovanpå alla underliggande funktioner, medan etikettmasken selektivt hindrar andra lager från att rita.

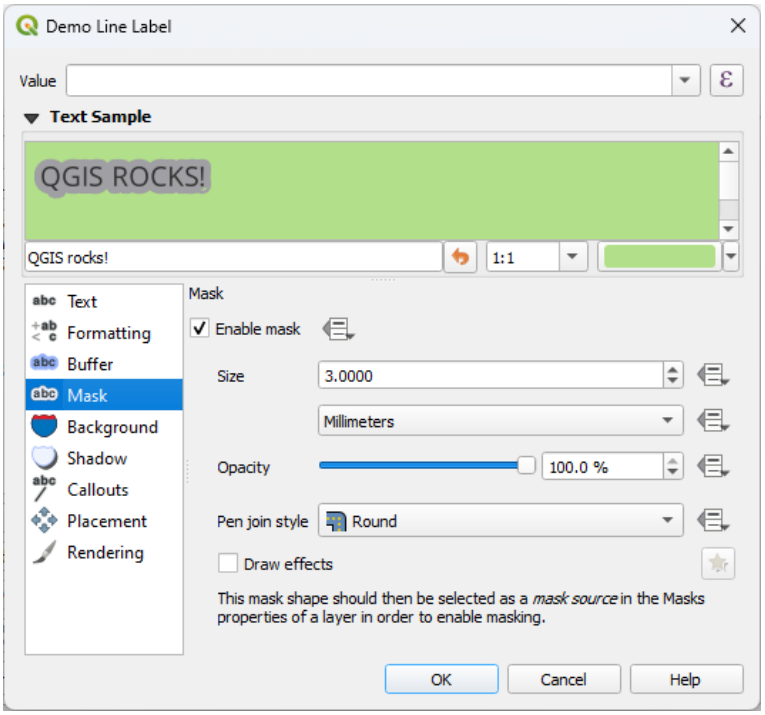


Fig. 10.31: Etiketter - Inställningar - fliken Mask (med textexemplet som visar en grön bakgrund som representerar ett annat lager som utesluts)

För att skapa maskeringseffekter på etiketter:

1. Aktivera kryssrutan *Enable mask* i fliken .
2. Sedan kan du ställa in:
   - maskens *Storlek* i *enheter som stöds*
   - *Opacity* för maskområdet runt etiketten
   - a *Pen Join Style*
   - *målningseffekter* genom kryssrutan *Rita effekter*.
3. Välj denna maskform som maskkälla i egenskaperna för det överlappande lagret *Mask* (se *Masker Egenskaper*).

Fliken Callouts

En vanlig metod för att placera ut etiketter på en fullsatt karta är att använda **callouts** - etiketter som placeras utanför (eller förskjuts från) sin tillhörande funktion identifieras med en dynamisk linje som förbinder etiketten och funktionen. Om en av de två ändarna (antingen etiketten eller objektet) flyttas, beräknas formen på kopplingen på nytt.

Fig. 10.32: Etiketter med olika inställningar för callouts

Om du vill lägga till en bildtext på en etikett aktiverar du fliken *Callouts* och kryssrutan *Draw callouts*. Då kan du göra det:

1. Välj *Style* av kontaktdon, en av:
   - *Enkla linjer*: en rak linje, den kortaste vägen
   - *Manhattan style*: en 90° bruten linje
   - *Böjda linjer*: en böjd linje
   - *Balloons*: en pratbubbla som omger etiketten och pekar på funktionen. Den kan ha rundade hörn.
2. Ställ in egenskaperna för callouten. I följande tabell visas de olika egenskaperna, med beskrivning och kompatibel anslutningstyp.

Table 10.2: Egenskaper för etikettutrop

Egenskap	Typ av bildtext	Beskrivning
Fyllnadsstil	Ballonger	En *fill-symbol* med fullständiga visningsmöjligheter, inklusive lagereffekter, datadefinierade inställningar, … för att rita ballongformen.
Hörnradie		Hörnradie för pratbubblan
Kantbredd		Ställer in hur stor bubblan talanslutning med funktionens pekare ska vara
Marginer		Marginaler runt etikettens text, i den enhet du väljer
Linjestil	Allt utom ballonger	En *linjesymbol* med fullständiga visningsmöjligheter, inklusive lagereffekter, datadefinierade inställningar, … för att rita anslutningslinjen.
Kurvatur	Böjda linjer	Den procentuella krökningen av anslutningslinjen
Orientation		Orientering, med början från etiketten till objektet. Den kan vara *Medurs*, *Moturs* eller *Automatisk* (bestämmer en optimal orientering för varje etikett).
Minsta längd	Allt utom ballonger	Minsta längd på anslutningsledningen
Rita linjer till alla delar av funktionen		Om en geometri består av flera delar anger du om en anslutningslinje ska dras från etiketten till varje geometrisk del.
Labelns ankarpunkt		Styr var anslutningslinjen ska ansluta till etikettexten. Tillgängliga alternativ: • *Närmaste punkten* • *Centroid* • Fast position vid kanten (*Top left*, *Top center*, *Top right*, *Left middle*, *Right middle*, *Bottom left*, *Bottom center* och *Bottom right*).
Offset från etikettområdet		Styr avståndet från etikettens ankarpunkt (där callout-linjen slutar). På så sätt undviker man att dra linjer rakt upp mot texten.
Offset från funktion	Alla	Styr avståndet från objektet (eller dess ankarpunkt om det är en polygon) där callout-linjerna slutar. På så sätt undviker man t.ex. att dra linjer precis intill kanterna på objekten.
Funktionens ankarpunkt		Där anslutningslinjen slutar på (polygon)objektet. Tillgängliga alternativ: • *Stolpe av otillgänglighet* • *Punkt på utsidan* • *Punkt på ytan* • *Centroid*
Markör för slutpunkt	Ballonger	En *markörsymbol* med fullständiga visningsfunktioner

Under gruppen *Datadefinierad placering* kan koordinaterna för punkterna *Origin* (på etikett-sidan) och/eller *Destination* (på feature-sidan) för callouten kontrolleras. Callouts kan också styras manuellt med hjälp av verktyget Move Label, Diagram or Callout i *Labeling Toolbar*. Start- och slutpunkterna för varje callout kan flyttas på detta sätt. Noderna bör vara markerade när muspekaren är i närheten. Om det behövs kan `Shift`-tangenten hållas nedtryckt under förflyttningen. Då snäpps punkten på ett sådant sätt att vinkeln mellan de två callout-punkterna ökar med 15 grader.

Fliken Placering

Välj fliken *Placement* för att konfigurera etikettplacering och etikettprioritet. Observera att placeringsalternativen skiljer sig åt beroende på typen av vektor- eller mesh-lager, dvs. punkt, linje eller polygon, och påverkas av den globala *PAL-inställningen*.

Placering för punktlager

Tillgängliga placeringslägen för punktetiketter är:

- *Cartographic*: Punktetiketter genereras med ett bättre visuellt förhållande till punktfunktionen, enligt idealiska kartografiska placeringsregler. Etiketter kan placeras:
  - på ett inställt *Distance* i *supported units*, antingen från själva punktfunktionen eller från gränserna för den symbol som används för att representera funktionen (inställd i *Distance offset from*). Det senare alternativet är särskilt användbart när symbolstorleken inte är fast, t.ex. om den är inställd på en datadefinierad storlek eller när olika symboler används i en *categorized*-renderare.
  - inom ett *Maximum Distance* från objektet, vilket är en valfri inställning som gör att du kan styra hur långt en etikett kan placeras från objektet som den märker. Detta fungerar tillsammans med inställningen *Distance* för att skapa ett intervall för etikettplacering, vilket ger flexibilitet för att placera etiketter mer effektivt, särskilt på upptagna kartor, och se till att de passar snyggt runt sina motsvarande funktioner.
  - med alternativet *Prioritera placering*, som avgör vad som är viktigast när etiketter placeras ut. Det finns två alternativ:
    - *Företräda närmare etiketter*: Som standard hålls etiketter nära objektet.
    - *Företräda positionsordning*: Etiketten försöker hålla sig i en viss position (t.ex. längst upp till vänster eller längst upp till höger), även om den är lite längre bort från funktionen. Etiketten flyttas bara till andra positioner om det inte finns plats inom det maximala avståndet på din föredragna position.
  - efter en *Position priority* som dikterar placeringskandidater för att förankra etiketter runt och (centrerat) över punktfunktionen, och den ordning i vilken positionerna testas. Standardordningen, som baseras på riktlinjer från Krygier och Wood (2011) och andra kartografiska läroböcker, är följande:
    1. övre höger
    2. övre vänster
    3. nedre höger
    4. nedre vänster
    5. mitten höger
    6. mitten vänster
    7. överst, något till höger
    8. längst ner, något till vänster.

    Med hjälp av knappen Data-defined override kan du ange en kommaseparerad lista med placeringar i prioritetsordning. Detta gör det också möjligt att endast använda vissa placeringar, endast för vissa funktioner, så t.ex. för kustnära funktioner kan du förhindra att etiketter placeras över land.

- *Around Point*: etiketter placeras i en cirkel runt objektet med en lika stor radie som anges i *Distance*. Dessutom kan du ställa in *Maximum Distance* från objektet för att styra hur långt en etikett kan placeras från objektet som den märker. Placeringsprioriteten är medurs från ”längst upp till höger”. Positionen kan begränsas med hjälp av det datadefinierade alternativet *Quadrant*.
- *Offset from Point*: etiketter placeras på ett *Offset X,Y*-avstånd från punktobjektet, i olika enheter, eller helst över objektet. Du kan använda en datadefinierad *Quadrant* för att begränsa placeringen och du kan tilldela etiketten en *Rotation*.

Placering för linjelager

Etiketteringslägen för linjelager inkluderar:

- *Parallel*: ritar etiketten parallellt med en generaliserad linje som representerar objektet, med preferens för placering över rakare delar av linjen. Du kan definiera:
  - *Tillåtna positioner*: *Above line*, *On line*, *Below line* och *Line orientation dependent position* (placerar etiketten till vänster eller höger om linjen). Det är möjligt att välja flera alternativ samtidigt. I så fall kommer QGIS att leta efter den optimala etikettpositionen.
  - *Avståndet* mellan etiketten och linjen
- *Curved*: ritar etiketten så att den följer linjens krökning. Förutom de parametrar som är tillgängliga med *Parallel*-läget kan du ställa in *Maximal vinkel mellan böjda tecken*, antingen inuti eller utanför.
- *Horizontal*: ritar etiketter horisontellt längs längden på linjefunktionen.

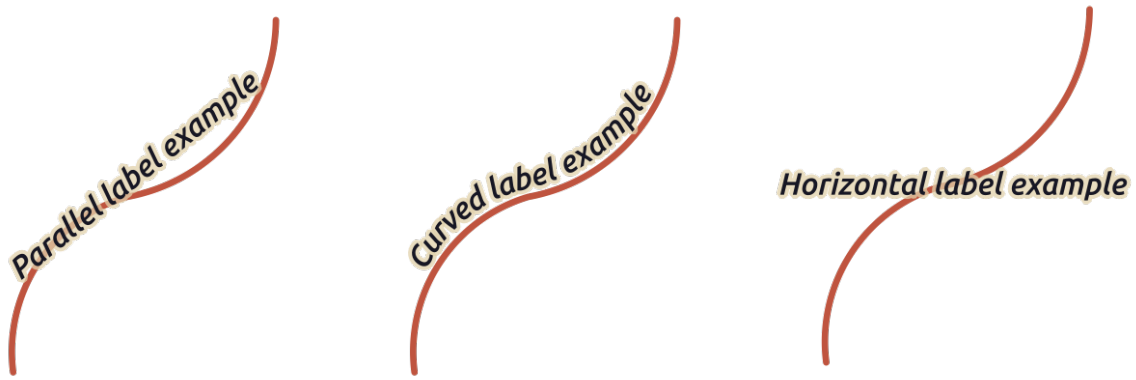


Fig. 10.33: Exempel på placering av etiketter för linjer

Förutom placeringslägen kan du ställa in:

- *Repeating Labels Distance* för att visa etiketten flera gånger över längden på objektet. Avståndet kan anges i `Millimeter`, `Punkter`, `Pixlar`, `Meter i skala`, `Kartenheter` och `Tum`.
- En *Label Overrun Distance* (inte tillgänglig för horisontellt läge): anger det maximala tillåtna avståndet som en etikett får passera förbi slutet (eller början) av linjefunktionerna. Om detta värde ökas kan etiketter visas för kortare linjefunktioner.
- *Label Anchoring*: styr placeringen av etiketterna längs den linjefunktion som de hänvisar till. Klicka på *Settings* … för att välja:
  - positionen längs linjen (som ett förhållande) som etiketterna ska placeras nära. Den kan vara datadefinierad och möjliga värden är:
    - *Linjens mittpunkt*
    - *Start på linjen*

    * *Slutet på linjen*
    * eller *Custom…*.
  - *Klippning*: Bestämmer hur etikettens placering på en linje ska beräknas. Som standard används endast den synliga delen av linjen, men hela linjen kan användas för att få ett mer konsekvent resultat.
  - *Anchor text*: styr vilken del av texten (början, mitten eller slutet) som ska ligga i linje med ankarpunkten. Om du använder *Automatic* anchoring innebär det att:
    * För etiketter som är förankrade nära början av raden (0-25%) kommer förankringsplatsen att vara **starten** av etikettexten
    * För etiketter som är förankrade nära slutet av raden (75-100%) kommer förankringsplatsen att vara **slutet** av etikettexten
    * För etiketter som är förankrade nära mitten av raden (25-75%) kommer förankringsplatsen att vara **centrum** av etikettexten
  - *Placement Behavior*: använd *Preferred Placement Hint* för att behandla etikettankaret endast som en ledtråd för etikettplaceringen. Genom att välja *Strict* placeras etiketter exakt på etikettankaret.

Placering för polygonlager

Du kan välja ett av följande lägen för att placera etiketter på polygoner:

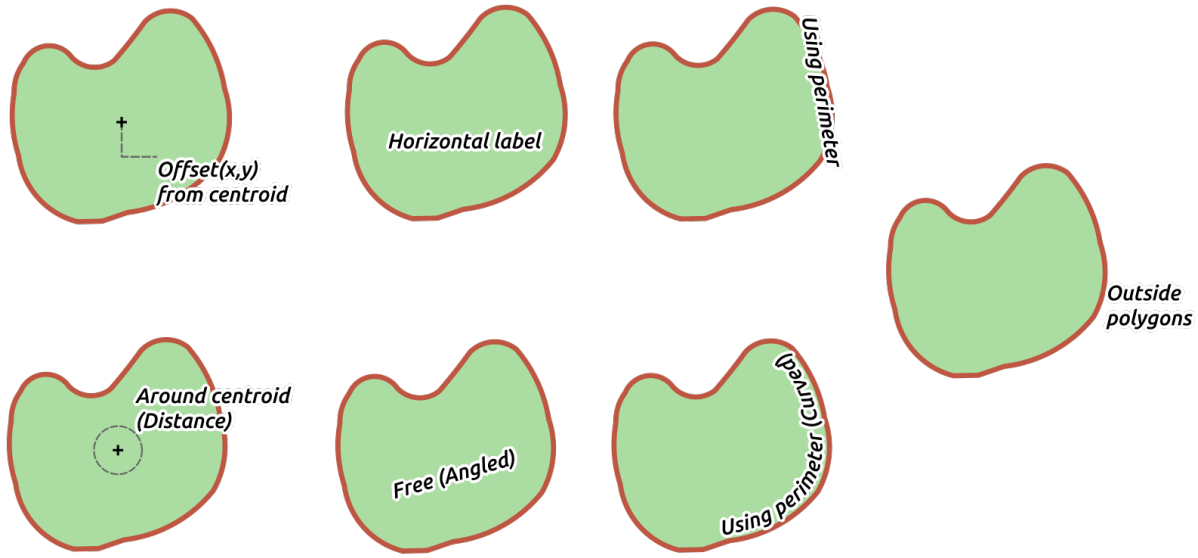


Fig. 10.34: Exempel på placering av etiketter för polygoner

- *Offset from Centroid*: etiketter placeras över objektets centroid eller på ett fast *Offset X,Y*-avstånd (i *stödda enheter*) från centroiden. Referenscentroiden kan bestämmas baserat på den del av polygonen som återges i kartbilden (*visible polygon*) eller *whole polygon*, oavsett om du kan se den. Det kan du också göra:
  - tvinga centroidpunkten att ligga inom deras polygon
  - placera etiketten inom en specifik kvadrant
  - tilldela en rotation
  - *Tillåt placering av etiketter utanför polygoner* när det inte är möjligt att placera dem inuti polygonen. Tack vare datadefinierade egenskaper är det möjligt att antingen tillåta externa etiketter, förhindra externa etiketter eller tvinga fram externa etiketter för varje enskild funktion.

- *Around Centroid*: placerar etiketten inom ett förinställt avstånd runt centroiden, med en preferens för placering direkt över centroiden. Återigen kan du definiera om centroiden är den i *visible polygon* eller *whole polygon*, och om du vill tvinga centroidpunkten inuti polygonen.
- *Horizontal*: placerar en horisontell etikett på bästa möjliga plats inuti polygonen. Den föredragna placeringen är längre från polygonens kanter. Det är möjligt att *Allow placing labels outside of polygons*.
- *Free (Angled)*: placerar en roterad etikett inuti polygonen i bästa möjliga position. Rotationen respekterar polygonens orientering och den föredragna placeringen är längre från polygonens kanter. Det är möjligt att *Allow placing labels outside of polygons*.
- *Using Perimeter*: ritar etiketten parallellt med en generaliserad linje som representerar polygongränsen, med preferens för rakare delar av omkretsen. Du kan definiera:
  - *Allowed positions*: *Above line*, *On line*, *Below line* och *Line orientation dependent position* (placerar etiketten till vänster eller höger om polygonens gräns). Det är möjligt att välja flera alternativ samtidigt. I så fall kommer QGIS att leta efter den optimala etikettpositionen.
  - *Avståndet* mellan etiketten och polygonens kontur
  - *Repeating Labels Distance* för att visa etiketten flera gånger över omkretsens längd.
- *Using Perimeter (Curved)*: ritar etiketten så att den följer krökningen i polygonens gräns. Förutom de parametrar som är tillgängliga i läget *Using Perimeter* kan du ställa in *Maximala vinkeln mellan polygonens krökta tecken*, antingen inuti eller utanför.
- *Outside Polygons*: placerar alltid etiketter utanför polygonerna, på ett bestämt *Distance*

Vanliga placeringsinställningar

Vissa inställningar för etikettplacering är tillgängliga för alla typer av lagergeometrier:

Geometri-generator

Avsnittet *Geometry Generator* låter användaren ändra den underliggande geometrin som används för att placera och rendera etiketten, genom att använda *expressions*. Detta kan vara användbart för att utföra dynamisk förskjutning av geometrin eller för att konvertera den till en annan geometri (typ).

För att kunna använda geometri-generatorn:

1. Markera alternativet *Geometrigenerator*
2. Ange det uttryck som genererar geometrin att förlita sig på
3. Om det är relevant väljer du geometrityp för uttrycksutmatningen: etikettens geometribaserade inställningar, t.ex. placering eller rendering, uppdateras för att matcha den nya geometritypens kapacitet.

Några exempel på användningsområden är:

- Använd en geometri som sparas i ett annat fält ”label_position”
- Använd *genererad geometri* från symbologin även för märkning
- Använd variabeln @map_scale för att beräkna avstånd/storlekar oberoende av zoomnivå.
- I kombination med läget för böjd placering skapas en cirkulär etikett runt en punktfunktion:

```
exterior_ring(make_circle($geometry, 20))
```

- Lägg till en etikett i början och slutet av en radfunktion:

```
collect_geometries( start_point($geometry), end_point($geometry) )
```

- Förlita dig på en utjämnad linje av en flod för att få mer utrymme för etikettplacering:

```
smooth( $geometry, iterations:=30, offset:=0.25, min_length:=10 )
```

Data definierad

Gruppen *Data Defined* ger direkt kontroll över etiketternas placering, funktion för funktion. Den förlitar sig på deras attribut eller ett uttryck för att ställa in:

- koordinaterna *X* och *Y*
- textinriktningen över den anpassade positionen som anges ovan:
  - *Horizontal*: den kan vara **Left**, **Center** eller **Right**
  - texten *Vertical*: den kan vara **Bottom**, **Base**, **Half**, **Cap** eller **Top**
- texten *Rotation*. Rotation definieras som en vinkel medurs med 0° pekande mot öster för ”horisontellt” orienterad text och med 0° pekande mot norr för ”vertikalt” orienterad text. Olika enheter kan definieras för märkningens rotation (t.ex. ”grader”, ”bågminuter”, ”vändningar”). QGIS prioriterar som standard rotationsvinklar för skärmvisning och roterar alltid text så att den blir mest läsbar på skärmen, så rotationsvinklar som är ”upp och ner” åsidosätts vid rendering. För att ändra detta beteende och tvinga QGIS att rendera exakta datadefinierade rotationsvinklar finns det ett alternativ i fliken *Rendering*, under *Show upside-down labels*. Markera *Bevara rotationsvärden för data* om du vill behålla rotationsvärdet i det associerade fältet och tillämpa det på etiketten, oavsett om etiketten är fäst eller inte. Om det inte är markerat återställs etikettens rotationsvärde och dess värde raderas från attributtabellen när etiketten tas bort.

  > **Observera**
  >
  > Datadefinierad rotation med polygonfunktioner stöds för närvarande endast med placeringsläget *Around centroid*.

> **Observera**
>
> Uttryck kan inte användas i kombination med kartverktygen för etiketter (dvs. verktygen *Rotate label* och *Move label*) för att *data-definiera* etikettplacering. Widgeten kommer att återställas till motsvarande *auxiliary storage field*.

Prioritet

I avsnittet *Priority* kan du definiera placeringsprioriteten för varje etikett, dvs. om det finns olika diagram eller etiketter som är kandidater för samma plats, kommer objektet med den högsta prioriteten att visas och de andra kan utelämnas.

Prioritetsrangordningen används också för att utvärdera om en etikett kan utelämnas på grund av en högre viktad *obstacle feature*.

Hinder

I vissa sammanhang (t.ex. etiketter med hög densitet, överlappande funktioner …) kan etikettplaceringen leda till att etiketter placeras över orelaterade funktioner.

Ett hinder är en funktion över vilken QGIS undviker att placera andra funktioners etiketter eller diagram. Detta kan styras från avsnittet *Obstacles*:

1. Aktivera alternativet *Funktioner fungerar som hinder* för att bestämma att funktioner i lagret ska fungera som hinder för alla etiketter och diagram (inklusive objekt från andra funktioner i samma lager).

   I stället för hela lagret kan du välja en delmängd av funktioner som ska användas som hinder med hjälp av kontrollen Data-defined override bredvid alternativet.

2. Använd knappen *Settings* för att justera hindrets viktning.

   - För varje potentiell hinderfunktion kan du tilldela en *Obstacle weight*: alla *label* eller *diagram* vars placeringsprioritet är större än detta värde kan placeras över. Etiketter eller diagram med lägre rang kommer att utelämnas om ingen annan placering är möjlig.

     Denna viktning kan också vara datadefinierad, så att det inom samma lager är mer sannolikt att vissa funktioner täcks än andra.

   - För polygonlager kan du välja vilken typ av hinder som funktionen är:

     – **over the feature's interior**: undviker att placera etiketter över polygonens inre (föredrar att placera etiketter helt utanför eller bara något innanför polygonen)

     – or **over the feature's boundary**: undviker att placera etiketter över polygonens gräns (föredrar att placera etiketter utanför eller helt inuti polygonen). Detta kan vara användbart för lager där funktionerna täcker hela området (administrativa enheter, kategoriska täckningar, …). I detta fall är det omöjligt att undvika att placera etiketter inom dessa funktioner, och det ser mycket bättre ut när man undviker att placera dem över gränserna mellan funktionerna.

Fliken Rendering

På fliken *Rendering* kan du ställa in när etiketterna ska renderas och hur de ska interagera med andra etiketter och funktioner.

Alternativ för etiketter

Under *Label options*:

- Du hittar inställningarna för synlighet *skalbaserad* och *pixelstorleksbaserad*.
- *Label z-index* bestämmer i vilken ordning etiketter återges, både i förhållande till andra funktionsetiketter i lagret (med hjälp av datadefinierade åsidosättningsuttryck) och till etiketter från andra lager. Etiketter med ett högre z-index återges ovanpå etiketter (från vilket lager som helst) med lägre z-index.

  Dessutom har logiken justerats så att om två etiketter har matchande z-index, så:

  – om de är från samma lager, kommer den mindre etiketten att ritas ovanför den större etiketten

  - om de kommer från olika lager kommer etiketterna att ritas i samma ordning som lagren själva (dvs. med hänsyn till den ordning som anges i kartlegenden).

> **Observera**
>
> Den här inställningen gör inte att etiketterna ritas under funktionerna från andra lager, utan styr bara i vilken ordning etiketterna ritas ovanpå alla lagrens funktioner.

- *Tillåt sämre reservplaceringar*: Som standard försöker QGIS att rendera etiketter på bästa möjliga plats, enligt dina inställningar. Markera detta läge för att tillåta funktioner att fallbacka till sämre placeringsalternativ när det inte finns något annat val, t.ex. när en linje är för kort för att passa en böjd etikettext kan etiketten placeras horisontellt precis över funktionens mittpunkt.
- Med datadefinierade uttryck i *Show label* och *Always Show* kan du finjustera vilka etiketter som ska återges.
- Tillåt att *Visa upp-och-nedvända etiketter*: alternativen är **aldrig**, **när rotation definieras** eller **alltid**.
  - **aldrig** - standardinställning, skärmens läsbarhet prioriteras,
  - **när rotation definieras** - etikettrotation ska definieras under *Placement*-fliken, inom *Data Defined*-gruppen
  - **alltid** - upp-och-nedvända etiketter är tillåtna

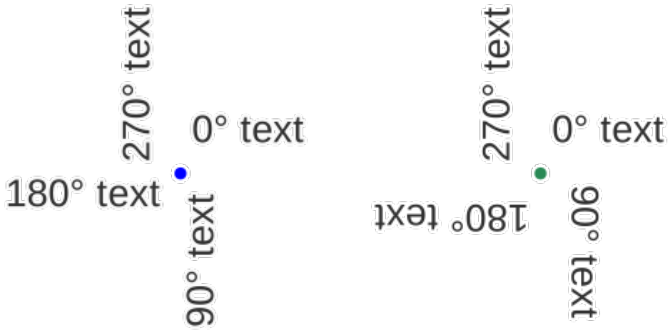


Fig. 10.35: Datadefinierad etikettrotation med alternativet visa upp-och-nedvända etiketter inställt på: ”aldrig” (vänster), ”när rotationen är definierad” (höger)

- Med gruppen *Overlappande etiketter* kan du styra om överlappande etiketter ska tillåtas för objekt i lagret och hur var och en av dem ska hanteras:
  - *Never overlap*: placera aldrig överlappande etiketter för lagret, även om det innebär att vissa etiketter kommer att saknas
  - *Tillåt överlappningar vid behov*: om etiketten inte kan placeras på annat sätt, rita en överlappande etikett. Detta läge gör att etiketten flyttas till en mindre idealisk placering om det är möjligt, t.ex. att flytta etiketten längre bort från mitten av en linje eller polygon, OM detta gör att överlappande etiketter undviks. Men om det inte finns några andra möjliga positioner, rita då etiketten överlappande.
  - *Tillåt överlappningar utan bestraffning*: Det spelar ingen som helst roll om etiketten överlappar andra etiketter eller hinder, det går bra att göra och den bästa placeringen (t.ex. den mest centrala placeringen) ska alltid användas även om en alternativ placering är möjlig som helt undviker överlappningar.

  Genom att tillåta både överlappande etiketter och alternativ för reservplaceringar garanteras att alla funktioner i lagret är märkta … inte nödvändigtvis med sin bästa rendering!

Alternativ för funktioner

Under *Funktionsalternativ*:

- Du kan välja att *Märka varje del av en flerdelad funktion* och *Limitera antalet funktioner som ska märkas till*.
- Både linje- och polygonlager ger möjlighet att ange en minsta storlek för de objekt som ska märkas, med *Undertryck märkning av objekt som är mindre än*.
- För polygonfunktioner kan du också filtrera de etiketter som ska visas beroende på om de passar helt inom funktionen eller inte.
- För linjefunktioner kan du välja att *Merge connected lines to avoid duplicate labels*, vilket ger en ganska luftig karta i kombination med alternativen *Distance* eller *Repeat* på fliken *Placement*.

10.4 Skapa 3D-symboler

Med *Style Manager* kan du skapa och lagra 3D-symboler för varje geometrityp som ska återges i *3D-kartvyn*.

Som för de andra objekten, aktivera fliken *3D Symbols* och expandera menyn för att skapa:

- *3D-punktsymboler*
- *3D linjesymboler*
- *3D-polygonsymboler*

10.4.1 Punktlager

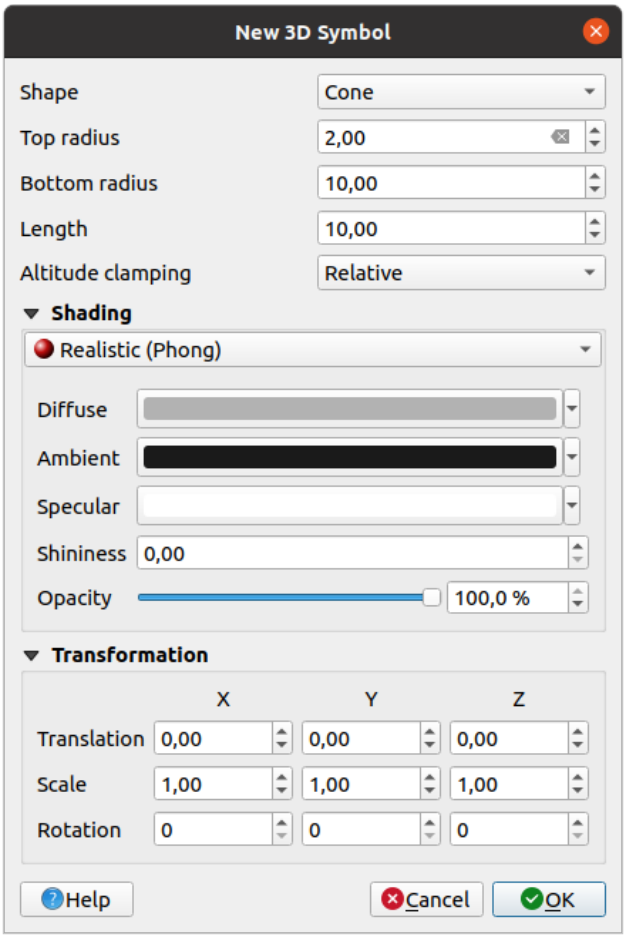


Fig. 10.36: Egenskaper för en 3D-punktsymbol

- Du kan definiera olika typer av 3D *Shape* som ska användas för punktsymboler. De definieras huvudsakligen av sina dimensioner, vars enhet hänvisar till projektets CRS. Tillgängliga typer är:
  - *Sfär* definierad av en *Radius*
  - *Cylinder* definierad av en *Radius* och *Längd*
  - *Cube* definieras av en *Size*
  - *Cone* definierad av en *Top radius*, en *Bottom radius* och en *Length*
  - *Plane* definieras av en *Size*
  - *Torus* definierad av en *Radius* och en *Minor radius*
  - *3D Model*, använder en 3D-modellfil: format som stöds är wavefront `.obj`, `.glTF` och `.fbx`. Modeller kan vara en fil på en disk, en fjärr-URL eller *inbäddad i projektet*. Modeller som skapats av gemenskapen delas på QGIS Hub.
  - *Billboard*, definieras av *Billboard height* och *Billboard symbol* (vanligtvis baserad på en *markörsymbol*). Symbolen kommer att ha en stabil storlek. Lämplig för visualisering av 3D-punktmoln Shapes.
- Inställningen *Altitude clamping* kan ställas in på *Absolute*, *Relative* eller *Terrain*. Inställningen *Absolute* kan användas när höjdvärdena för 3d-vektorerna anges som absoluta mått från 0. *Relative* och *Terrain* lägger till angivna höjdvärden till den underliggande terrängens höjd.
- Egenskaperna *shading* kan definieras.

- Under ramen *Transformations* kan du tillämpa affin transformation på symbolen:
  - *Translation* för att flytta objekt i x-, y- och z-axeln.
  - *Scale* för att ändra storlek på 3D-formerna
  - *Rotation* runt x-, y- och z-axeln.

10.4.2 Linjelager

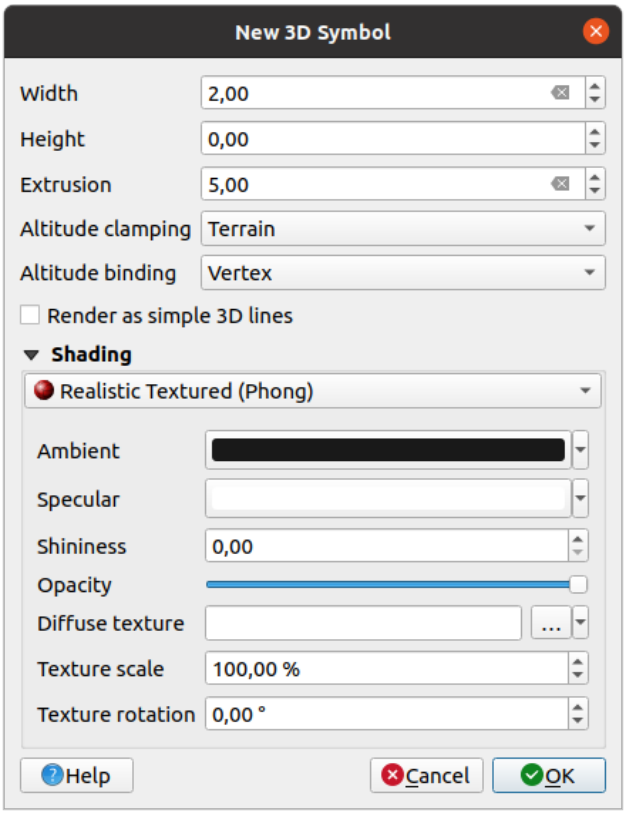


Fig. 10.37: Egenskaper för en 3D-linjesymbol

- Under inställningarna *Width* och *Height* kan du definiera *Extrusion* för vektorlinjerna. Om linjerna inte har z-värden kan du definiera 3d-volymerna med den här inställningen.
- Med *Altitude clamping* definierar du 3D-linjernas position i förhållande till den underliggande terrängytan, om du har inkluderat rasterhöjddata eller andra 3D-vektorer.
- Bindningen *Altitude* definierar hur objektet kläms fast i terrängen. Antingen kommer varje *Vertex* i objektet att klämmas fast i terrängen eller så görs detta av *Centroid*.
- Det är möjligt att *Rendera som enkla 3D-linjer*.
- Egenskaperna *shading* kan definieras.

10.4.3 Polygonlager

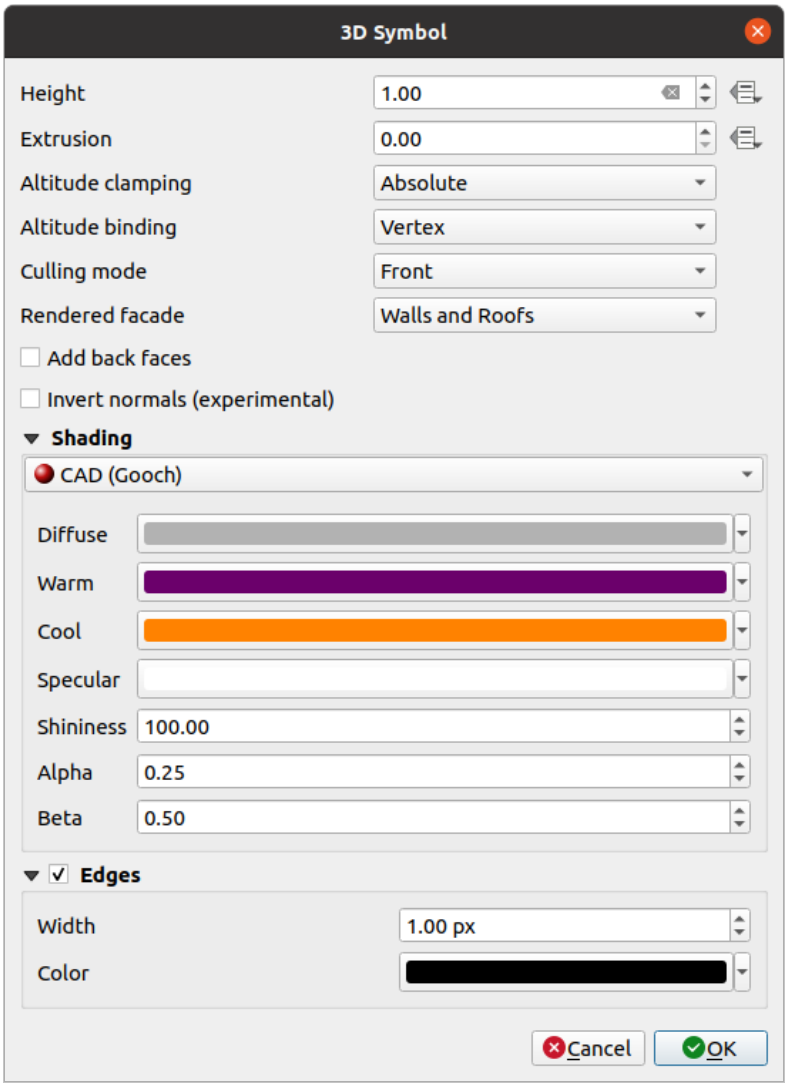


Fig. 10.38: Egenskaper för en 3D-polygonsymbol

- När det gäller de andra kan *Height* definieras i CRS-enheter. Du kan också använda knappen för att skriva över värdet med ett eget uttryck, en variabel eller en post i attributtabellen
- Återigen, *Extrusion* är möjligt för saknade z-värden. Även för extrudering kan du använda knappen för att använda värdena i vektorlagret och få olika resultat för varje polygon:

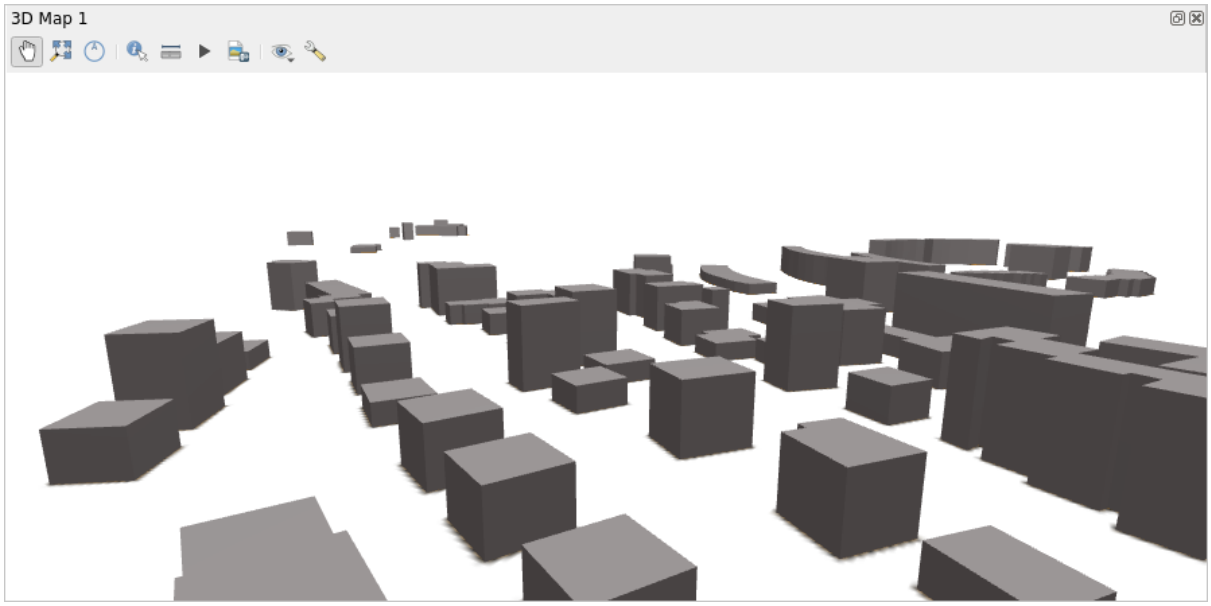


Fig. 10.39: Datadefinierad extrudering

- *Altitude clamping*, *Altitude binding* kan definieras enligt ovanstående förklaring.
- Det *Culling mode* som ska tillämpas på symbolen; det kan vara:
  - *No Culling*: detta kan hjälpa till att undvika till synes saknade ytor när polygonZ/multipatch-data inte har konsekvent ordning av hörn (t.ex. alla klockvis eller mot klockvis)
  - *Front*
  - eller *Back*
- *Rendered facade* bestämmer vilka fasader som ska visas. Möjliga värden är *No facades*, *Walls*, *Roofs*, eller *Walls and roofs*
- ☑ *Add back faces*: för varje triangel skapas både fram- och baksida med korrekta normaler - på bekostnad av ett ökat antal vertexdata. Detta alternativ kan användas för att åtgärda skuggproblem (t.ex. på grund av data med inkonsekvent ordning på topparna).
- ☑ *Invertera normaler (experimentell)*: kan vara användbart för att fixa medurs/moturs ordning på vertex i ansiktet
- Egenskaperna *shading* kan definieras.
- Visning av ☑ *Kanter* för symbolerna kan aktiveras och tilldelas en *Bredd* och *Färg*.

Råd

Kombination för bästa återgivning av 3D-data

Culling mode, *Add back faces* och *Invert normals* är alla avsedda att korrigera utseendet på 3D-data om det inte ser rätt ut. När du laddar data är det oftast bäst att först prova `culling mode=back` och `add back faces=disabled` - det är det mest effektiva. Om renderingen inte ser korrekt ut, prova `add back faces=enabled` och behåll `culling mode=no culling`. Andra kombinationer är mer avancerade och endast användbara i vissa scenarier baserat på hur blandad indatauppsättningen är.

10.4.4 Skuggning av texturen

Shading hjälper dig att avslöja 3d-detaljer på objekt som annars kan vara dolda på grund av scenens belysning. I slutändan är det ett enklare material att arbeta med eftersom du inte behöver oroa dig för att ställa in lämplig scenbelysning för att visualisera funktioner.

Olika tekniker för skuggning används i QGIS och deras tillgänglighet beror på symbolens geometrityp:

- *Realistic (Phong)*: beskriver hur en yta reflekterar ljus som en kombination av *Diffuse*-reflektion från skrovliga ytor och *Specular*-reflektion från blanka ytor (*Shininess*). Den innehåller också ett *Ambient*-alternativ för att ta hänsyn till den lilla mängd ljus som sprids över hela scenen. Använd reglaget *Opacity* för att rendera halvgenomskinliga objekt i 3D. Läs mer på Phong reflection description.
- *Realistic Textured (Phong)*: samma som *Realistic (Phong)* förutom att en bild används som *Diffuse Texture*. Bilden kan vara en fil på hårddisken, en fjärr-URL eller *inbäddad i projektet*. Uppgifterna *Texture scale* och *Texture rotation* krävs. Använd skjutreglaget *Opacity* för att rendera halvgenomskinliga objekt i 3D.
- *CAD (Gooch)*: Denna teknik gör att skuggning endast sker i mellantoner så att kantlinjer och högdagrar förblir visuellt framträdande. Tillsammans med alternativen *Diffuse*, *Specular*, *Shininess* måste du ange en *Warm*-färg (för ytor som vetter mot ljuset) och en *Cool*-färg (för de som vetter bort). De relativa bidragen till de kalla och varma färgerna från den diffusa färgen styrs också av egenskaperna *Alpha* respektive *Beta*. Se även Gooch shading.
- *Metal Roughness*: ett fysikaliskt baserat renderingsmaterial som ger en korrekt återgivning av hur ljus interagerar med ytor. Alternativ finns för att ställa in materialets *Base color*, *Metalness* och *Roughness*.
- *Embedded Textures* med 3D-modeller form

10.4.5 Exempel på tillämpning

För att gå igenom de inställningar som förklaras ovan kan du titta på https://app.merginmaps.com/projects/saber/luxembourg/tree.

KAPITEL 11

Hantera datakälla

11.1 Öppningsdata

Som en del av ett ekosystem för programvara med öppen källkod bygger QGIS på olika bibliotek som, i kombination med de egna leverantörerna, ger möjlighet att läsa och ofta skriva i många olika format:

- Vektordataformat inkluderar GeoPackage, GML, GeoJSON, GPX, KML, Comma Separated Values, ESRI-format (Shapefile, Geodatabase …), MapInfo och MicroStation-filformat, AutoCAD DWG / DXF, GRASS och många fler … Läs den fullständiga listan över vektorformat som stöds <https://gdal.org/en/latest/drivers/vector/index.html>`__.
- Rasterdataformat inkluderar GeoTIFF, JPEG, ASCII Gridded XYZ, MBTiles, R- eller Idrisi-raster, GDAL Virtual, SRTM, Sentinel Data, ERDAS IMAGINE, ArcInfo Binary Grid, ArcInfo ASCII Grid och många fler … Läs den fullständiga listan över rasterformat som stöds <https://gdal.org/en/latest/drivers/raster/index.html>`__.
- Databasformat inkluderar PostgreSQL / PostGIS, SQLite / SpatiaLite, Oracle, MS SQL Server, SAP HANA, MySQL …
- Webbkart- och datatjänster (WM(T)S, WFS, WCS, CSW, XYZ tiles, ArcGIS-tjänster, …) hanteras också av QGIS-leverantörer. Se *Arbeta med OGC/ISO-protokoll* för mer information om några av dessa.
- Du kan läsa filer som stöds från arkiverade mappar och använda QGIS inbyggda format som QML-filer (*QML - QGIS Style File Format*) och virtuella lager och minneslager.

Mer än 80 vektorformat och 140 rasterformat stöds av GDAL och QGIS inbyggda leverantörer.

Observera

Av olika skäl fungerar inte alla de listade formaten i QGIS. Till exempel kräver vissa externa proprietära bibliotek, eller så kanske GDAL/OGR-installationen i ditt operativsystem inte har byggts för att stödja det format du vill använda. För att se listan över tillgängliga format, kör kommandoraden `ogrinfo --formats` (för vektor) och `gdalinfo --formats` (för raster), eller kontrollera menyn :menuselection:` *Inställningar* –> *Alternativ* –> *GDAL* i QGIS.

I QGIS finns det, beroende på dataformat, olika verktyg för att öppna en dataset, huvudsakligen tillgängliga i menyn *Layer* ► *Add Layer* ► eller från verktygsfältet *Manage Layers* (aktiverat via menyn *View* ► *Toolbars*). Alla dessa verktyg pekar dock på en unik dialogruta, dialogrutan *Data Source Manager*, som du kan öppna med knappen
Open Data Source Manager, som finns i verktygsfältet *Data Source Manager*, eller genom att trycka på `Ctrl+L`. Dialogrutan *Data Source Manager* (Fig. 11.1) erbjuder ett enhetligt gränssnitt för att öppna filbaserade data samt databaser eller webbtjänster som stöds av QGIS.

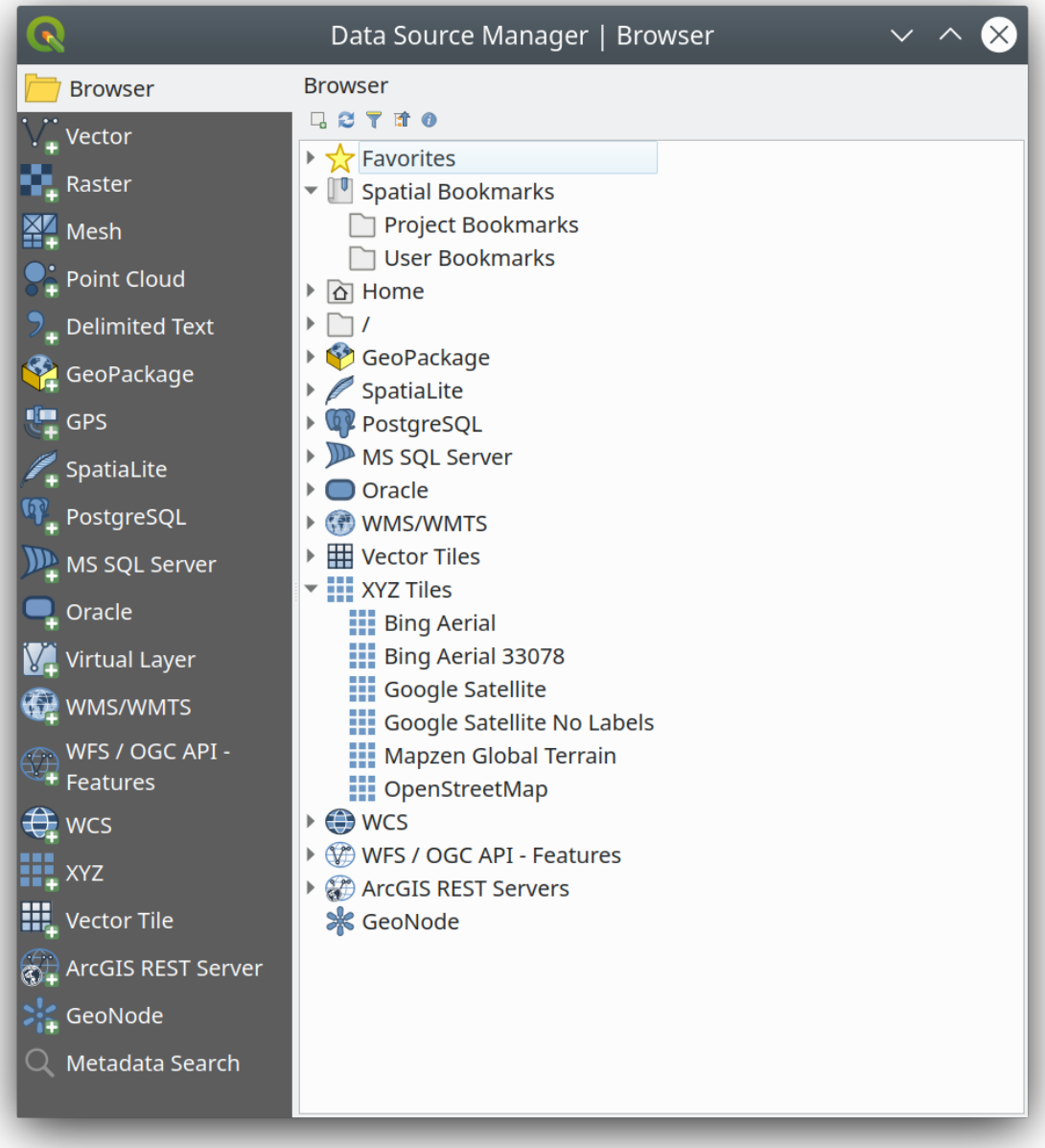


Fig. 11.1: Dialogrutan QGIS Datakällhanterare

Förutom denna huvudingångspunkt har du också tillägget *DB Manager* som erbjuder avancerade funktioner för att analysera och manipulera anslutna databaser. Mer information om DB Manager-funktionerna finns i *DB Manager Plugin*.

Det finns många andra verktyg, inbyggda eller tillägg från tredje part, som hjälper dig att öppna olika dataformat.

I detta kapitel beskrivs endast de verktyg som finns som standard i QGIS för att ladda data. Det kommer huvudsakligen

att fokusera på dialogrutan *Data Source Manager*, men utöver att beskriva varje flik kommer det även att utforska verktygen baserat på dataleverantörens eller formatets specifika egenskaper.

11.1.1 Bläddrarpanelen

Browser är ett av de viktigaste sätten att snabbt och enkelt lägga till dina data i projekt. Den finns tillgänglig som:

- en flik *Data Source Manager*, aktiverad genom att trycka på knappen Open Data Source Manager (`Ctrl+L`);
- som en QGIS-panel kan du öppna från menyn *Visa* ► *Paneler* (eller *Inställningar* ► *Paneler*) eller genom att trycka på `Ctrl+2`.

I båda fallen hjälper *Browser* dig att navigera i ditt filsystem och hantera geodata, oavsett typ av lager (raster, vektor, tabell) eller datakällans format (vanliga eller komprimerade filer, databaser, webbtjänster).

Utforska gränssnittet

Längst upp på panelen Browser finns några knappar som hjälper dig att:

- Add Selected Layers: du kan också lägga till data i kartbilden genom att välja **Add selected layer(s)** från lagrets snabbmeny;
- Uppdatera bläddringsträdet;
- Filtrera bläddraren för att söka efter specifika data. Ange ett sökord eller ett jokertecken så filtrerar bläddraren trädet så att det bara visar sökvägar till matchande DB-tabeller, filnamn eller mappar - andra data eller mappar visas inte. Se exemplet Browser Panel(2) i Fig. 11.2. Jämförelsen kan vara skiftlägeskänslig eller inte. Den kan också ställas in till:
  - *Normal*: visa objekt som innehåller söktexten
  - *Wildcard(s)*: finjustera sökningen med hjälp av tecknen `?` och/eller `*` för att ange söktextens position
  - *Regelbundet uttryck*
- Collapse All hela trädet;
- Aktivera/inaktivera widget för egenskaper: när den är aktiverad läggs en ny widget till längst ned i panelen som visar, om tillämpligt, metadata för det valda objektet.

Posterna i panelen *Browser* är hierarkiskt organiserade och det finns flera poster på högsta nivån:

1. *Favoriter* där du kan placera genvägar till ofta använda platser
2. *Spatial Bookmarks* där du kan lagra ofta använda kartutsnitt (se *Lägga till bokmärken för extenter på kartan*)
3. *Project Home*: för snabb åtkomst till den mapp där (de flesta) data som är relaterade till ditt projekt lagras. Standardvärdet är den katalog där din projektfil finns.
4. *Home*-katalogen i filsystemet och filsystemets rotkatalog.
5. Anslutna lokala enheter eller nätverksenheter
6. Sedan kommer ett antal container-/databastyper och serviceprotokoll, beroende på din plattform och underliggande bibliotek:
   - *GeoPackage*
   - *SpatiaLite*
   - *PostgreSQL*

- *SAP HANA*
- *MS SQL Server*
- *Oracle*
- *WMS/WMTS*
- *Vector Tiles*
- *XYZ Tiles*
- *WCS*
- *WFS/OGC API-funktioner*
- *ArcGIS REST Server*

Interagera med bläddrarobjekten

Bläddraren stöder dra och släpp inom webbläsaren, från bläddraren till duken och *Lager*-panelen, och från *Lager*-panelen till lagerbehållare (t.ex. GeoPackage) i bläddraren.

Projektfilobjekt i bläddraren kan expanderas och visa hela lagerträdet (inklusive grupper) som finns i det projektet. Projektobjekt behandlas på samma sätt som alla andra objekt i bläddraren, så de kan dras och släppas i bläddraren (t.ex. för att kopiera ett lagerobjekt till en geopackage-fil) eller läggas till i det aktuella projektet genom att dra och släppa eller dubbelklicka.

Kontextmenyn för ett element i *Browser*-panelen öppnas genom att högerklicka på det.

För katalogposter i filsystemet erbjuder snabbmenyn följande:

- *New* ► för att i den valda posten skapa en:
  - *Katalog…*
  - *GeoPackage…*
  - *ShapeFile…*
- *Add as a Favorite*: Favoritmappar kan döpas om (*Rename favorite…*) eller tas bort (*Remove favorite*) när som helst.
- *Hide from Browser*: dolda mappar kan göras synliga från :menuselection:` *Inställningar –> Alternativ –> Data Sources –> Hidden browser paths* inställningen
- *Snabbsökning av denna katalog*
- *Öppen katalog*
- *Öppna i terminal*
- *Egenskaper…*
- *Katalogegenskaper…*

För bladposter som kan fungera som lager i projektet kommer snabbmenyn att ha stödposter. Till exempel för icke databas-, icke tjänstebaserade vektor-, raster- och mesh-datakällor:

- *Exportera lager* ► *Till fil…*
- *Lägg till lager i projektet*
- *Lageregenskaper*
- *Öppna med Data Source Manager…*
- *Manage* ► *Byt namn på "<name of file>"…* eller *Ta bort "<name of file>"…*

- *Visa i filer*
- *Filegenskaper*

I posten *Lageregenskaper* hittar du (på samma sätt som du hittar i *vector* och *raster* lageregenskaper när lagren har lagts till i projektet):

- *Metadata* för lagret. Metadatagrupper: *Information från datakälla* (om möjligt kommer *Sökväg* att vara en hyperlänk till källan), *Identifiering*, *Utsträckning*, *Åtkomst*, *Fält* (för vektorlager), *Band* (för rasterlager), *Kontakter*, *Länkar* (för vektorlager), *Referenser* (för rasterlager), *Historik*.
- En *Preview*-panel
- Attributtabellen för vektorkällor (i panelen *Attributes*).

Använd *Open with Data Source Manager…* för att direkt öppna och konfigurera datakällan i *Data Source Manager* med hjälp av datakällans URI. Detta förenklar processen med att lägga till ett lager från *Browser* genom att du kan ställa in specifika öppningsalternativ för datakällan. Den är för närvarande tillgänglig för vektordatakällor (inklusive den dedikerade GeoPackage-posten), rasterdatakällor och SpatiaLite-datakällor.

För att lägga till ett lager i projektet med hjälp av *Browser*:

1. Aktivera *Bläddrare* enligt beskrivningen ovan. Ett bläddrarträd med ditt filsystem, databaser och webbtjänster visas. Du kan behöva ansluta databaser och webbtjänster innan de visas (se särskilda avsnitt).
2. Leta reda på lagret i listan.
3. Använd snabbmenyn, dubbelklicka på dess namn eller dra och släpp det i *map canvas*. Ditt lager läggs nu till i panelen *Layers* och kan visas på kartbilden.

> **Tips**
>
> **Öppna ett QGIS-projekt direkt från bläddraren**
>
> Du kan också öppna ett QGIS-projekt direkt från panelen Browser genom att dubbelklicka på dess namn eller genom att dra och släppa det i kartbilden.

När en fil har laddats kan du zooma runt den med hjälp av kartans navigeringsverktyg. Om du vill ändra stilen för ett lager öppnar du dialogrutan *Lageregenskaper* genom att dubbelklicka på lagernamnet eller genom att högerklicka på namnet i teckenförklaringen och välja *Egenskaper* från snabbmenyn. Se avsnitt *Egenskaper för symbologi* för mer information om hur du ställer in symbologi för vektorlager.

Om du högerklickar på ett objekt i bläddrarträdet kan du göra följande:

- för en fil eller en tabell, visa dess metadata eller öppna den i ditt projekt. Tabeller kan även bytas namn på, raderas eller förkortas.
- för en mapp, lägg den som bokmärke i dina favoriter eller dölj den från bläddrarträdet. Dolda mappar kan hanteras från fliken *Inställningar* ► *Alternativ* ► *Datakällor*.
- hantera dina *spatiala bokmärken*: bokmärken kan skapas, exporteras och importeras som `XML`-filer.
- skapa en anslutning till en databas eller en webbtjänst.
- uppdatera, byta namn på eller ta bort ett schema.

Du kan också importera filer till databaser eller kopiera tabeller från ett schema/en databas till en annan med en enkel dra-och-släpp-funktion. Det finns en andra bläddrarpanel tillgänglig för att undvika lång rullning när du drar. Välj bara filen och dra och släpp den från den ena panelen till den andra.

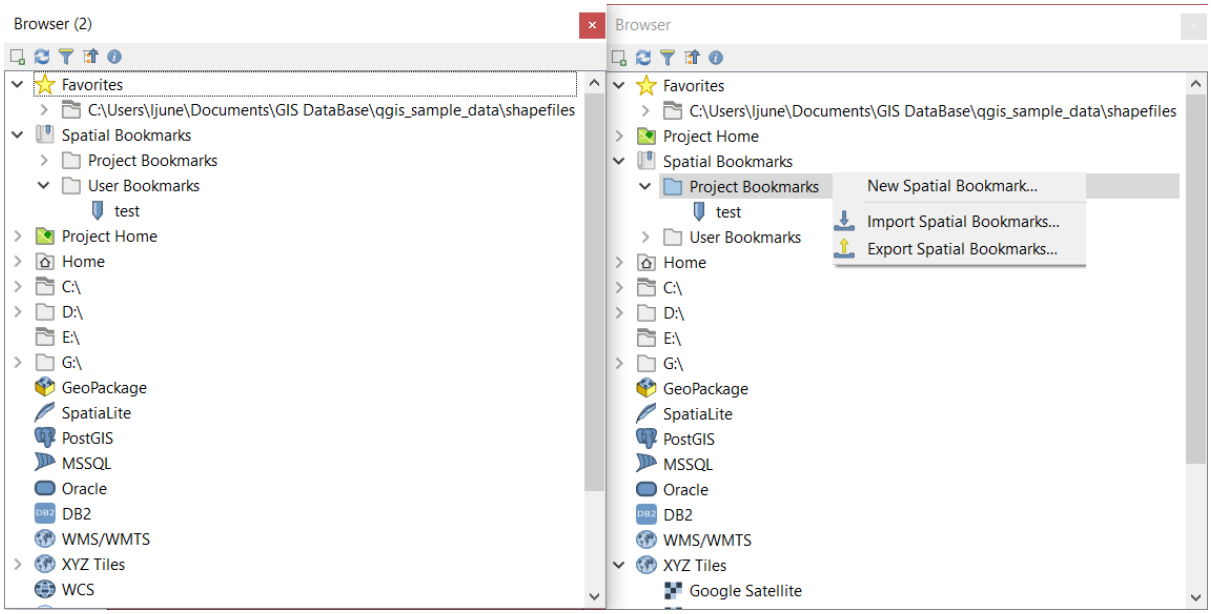


Fig. 11.2: QGIS Browser-paneler sida vid sida

Tips

Lägg till lager i QGIS genom att enkelt dra och släppa dem från OS:ets filbläddrare

Du kan också lägga till filer i projektet genom att dra och släppa dem från filbläddraren i operativsystemet till *Layers Panel* eller kartbilden.

11.1.2 DB-chefen

Tillägget *DB Manager* är ett annat verktyg för att integrera och hantera spatiala databasformat som stöds av QGIS (PostGIS, SpatiaLite, GeoPackage, Oracle Spatial, MS SQL Server, Virtual layers). Det kan aktiveras från *Tillägg* ► *Manage and Install Tillägg…*-menyn.

Tillägget DB Manager innehåller flera funktioner:

- ansluta till databaser och visa deras struktur och innehåll
- förhandsgranska tabeller i databaser
- lägga till lager på kartbilden, antingen genom att dubbelklicka eller dra och släppa.
- lägga till lager i en databas från QGIS Browser eller från en annan databas
- skapa SQL-frågor och lägga till deras resultat i kartbilden
- skapa *virtuella lager*

Mer information om DB Manager-funktioner finns i *DB Manager Plugin*.

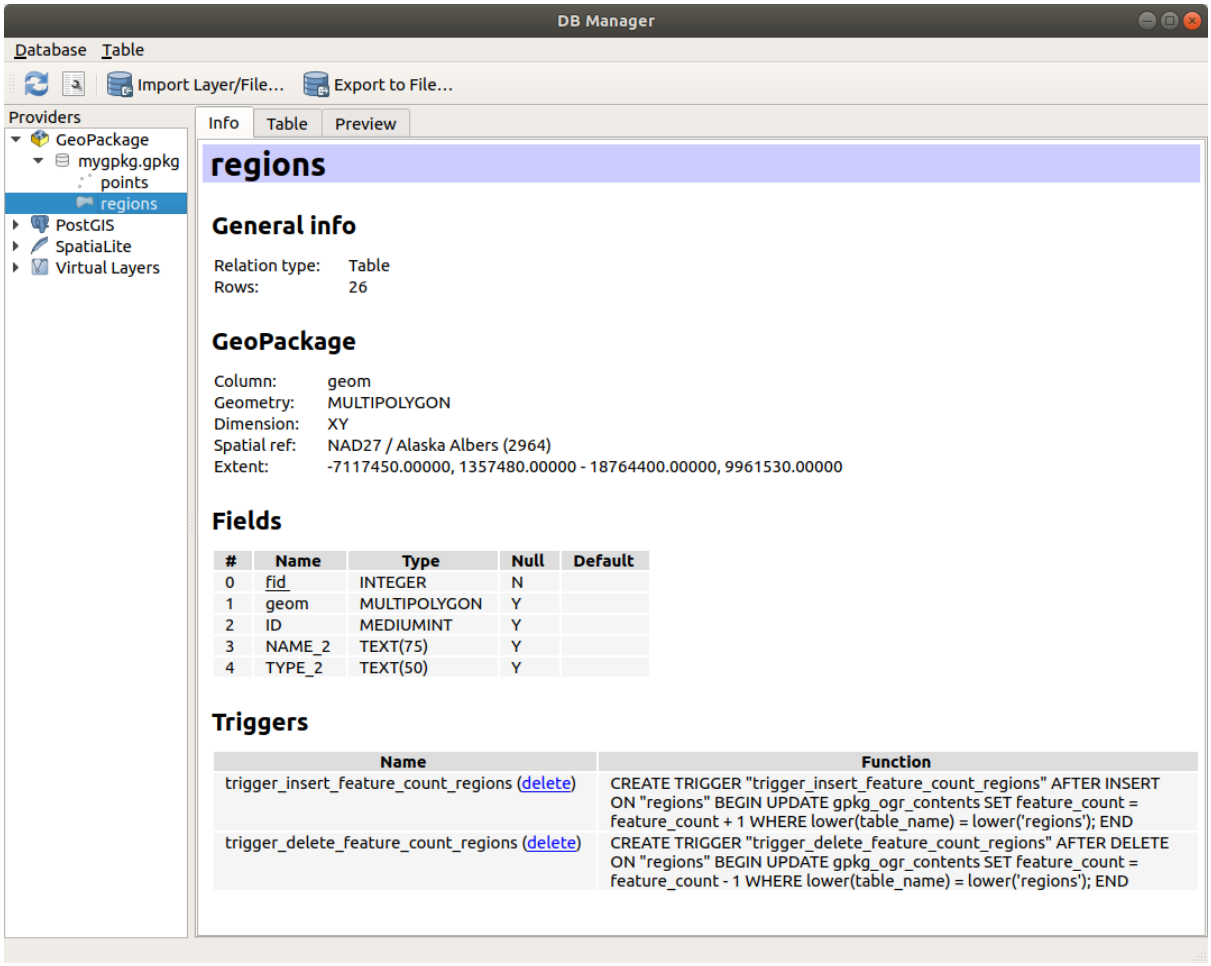


Fig. 11.3: Dialogrutan DB Manager

11.1.3 Leverantörsbaserade laddningsverktyg

Förutom Browser Panel och DB Manager, som är de viktigaste verktygen i QGIS för att lägga till lager, finns det även verktyg som är specifika för dataleverantörer.

> **Observera**
>
> Vissa *externa tillägg* tillhandahåller också verktyg för att öppna filer i specifika format i QGIS.

Läsa in ett lager från en fil

För att ladda ett lager från en fil:

1. Öppna fliken för lagertyp i dialogrutan *Data Source Manager*, dvs. klicka på knappen Open Data Source Manager (eller tryck på `Ctrl+L`) och aktivera målfliken eller:
   - för vektordata (som GML-, ESRI Shapefile-, Mapinfo- och DXF-lager): tryck på `Ctrl+Shift+V`, välj *Lager* ► *Lägg till lager* ► *Lägg till vektorlager* menyalternativ eller klicka på knappen Lägg till vektorlager i verktygsfältet.

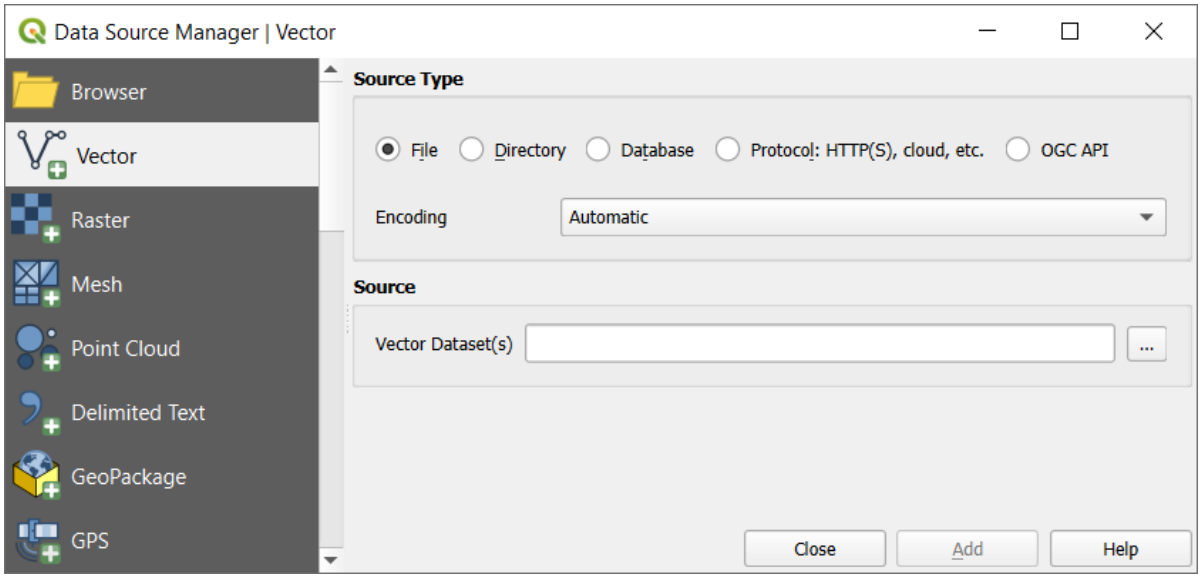


Fig. 11.4: Dialogrutan Lägg till vektorlager

- för rasterdata (som GeoTiff, MBTiles, GRIdded Binary och DWG-lager): tryck på `Ctrl+Shift+R`, välj *Layer* ► *Add Layer* ► *Add Raster Layer* menyalternativ eller klicka på knappen Add Raster Layer i verktygsfältet.

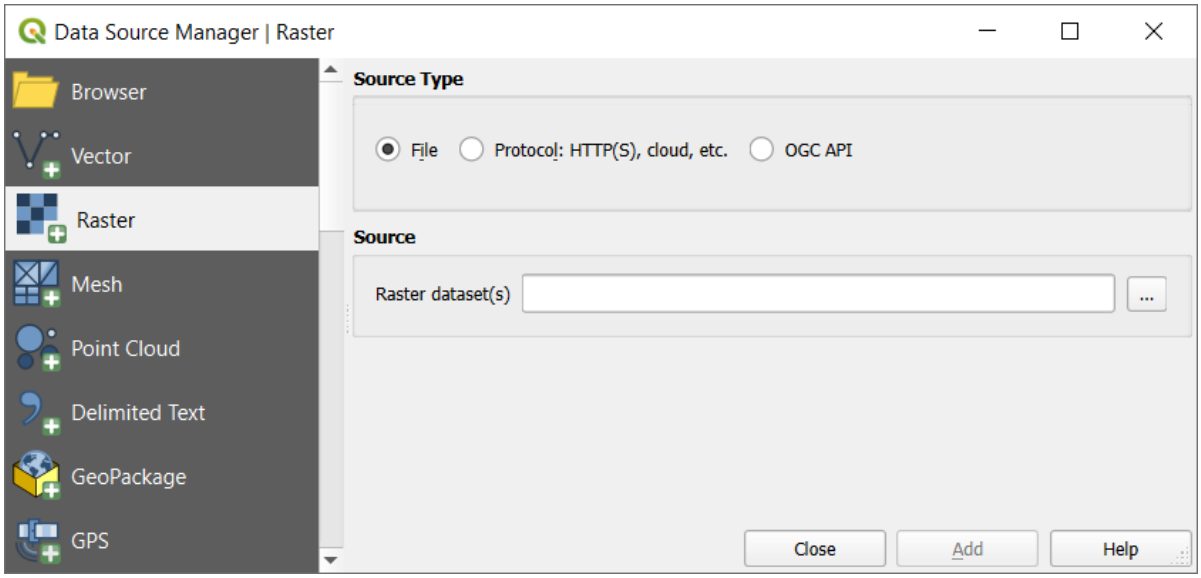


Fig. 11.5: Dialogrutan Lägg till Rasterlager

2. Kontrollera *File* källtyp
3. Klicka på … Browse-knappen
4. Navigera i filsystemet och ladda en datakälla som stöds. Fler än ett lager kan laddas samtidigt genom att hålla ned `Ctrl` och klicka på flera objekt i dialogrutan eller genom att hålla ned `Shift` för att välja ett intervall av objekt genom att klicka på det första och sista objektet i intervallet. Endast format som har testats väl visas i formatfiltret. Andra format kan laddas genom att välja `Alla filer` (det översta alternativet i rullgardinsmenyn).
5. Tryck på *Open* för att ladda den valda filen i dialogrutan *Data Source Manager*.

Beroende på den valda lagertypen finns ytterligare *Options* (kodning, geometrityp, tabellfiltrering, fillåsning, dataformatering …) tillgängliga för konfigurering. Dessa alternativ beskrivs i detalj i den specifika GDAL-dokumentationen för drivrutinerna vector eller raster. Längst upp i alternativen finns en text med hyperlänk som leder direkt till dokumentationen för den drivrutin som är lämplig för det valda filformatet.

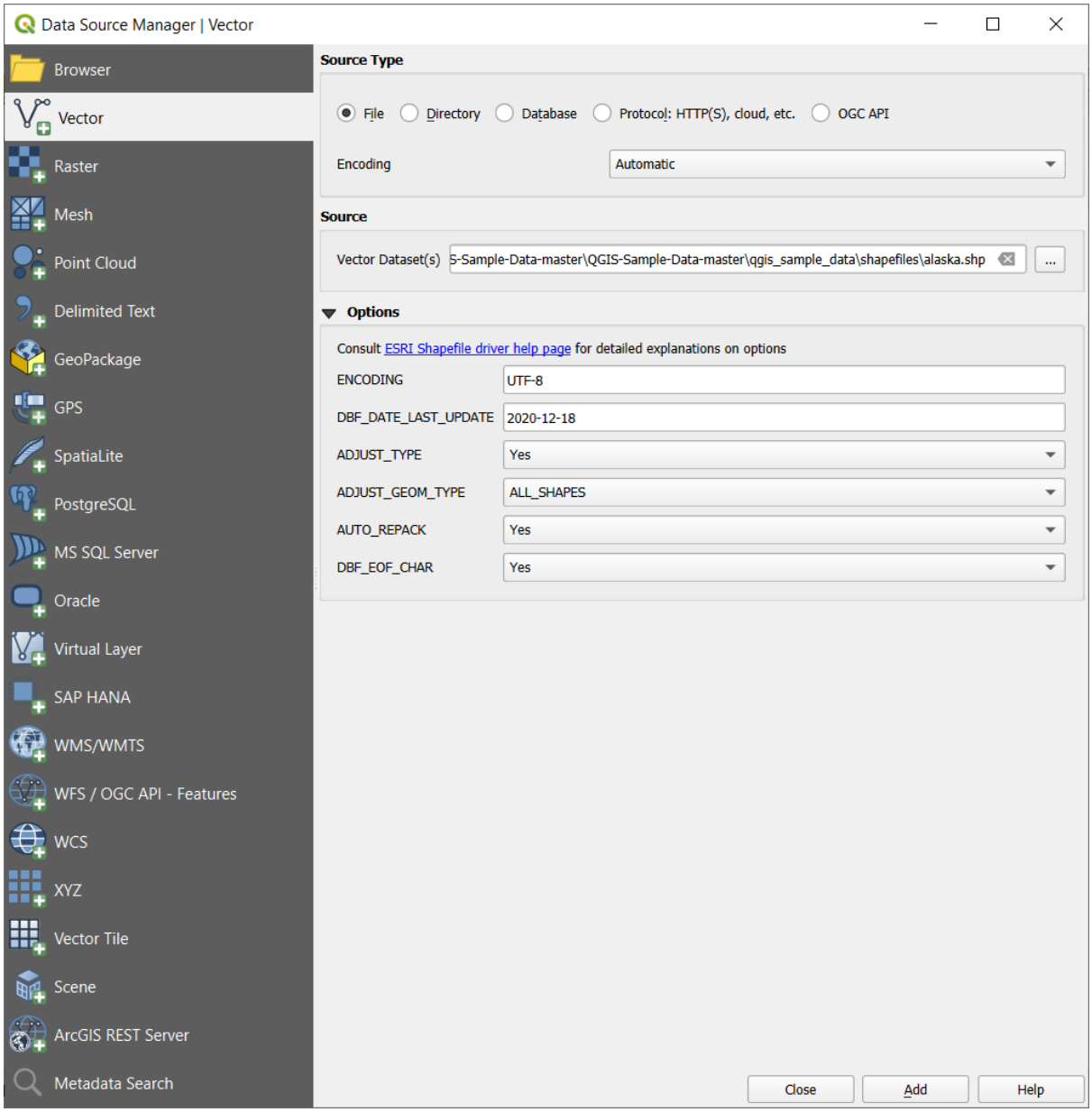


Fig. 11.6: Ladda en Shapefile med öppna alternativ

6. Tryck på *Add* för att läsa in filen i QGIS och visa dem i kartvyn. När du lägger till vektordataset som innehåller flera lager visas dialogrutan *Select Items to Add*. I den här dialogen kan du välja de specifika lager från din dataset som du vill lägga till. Under *Options* kan du också välja att:

   - *Lägg till lager i en grupp*
   - *Visa system och interna tabeller*
   - *Visa tomma vektorlager*.

   Fig. 11.7 visar QGIS efter laddning av filen `alaska.shp`.

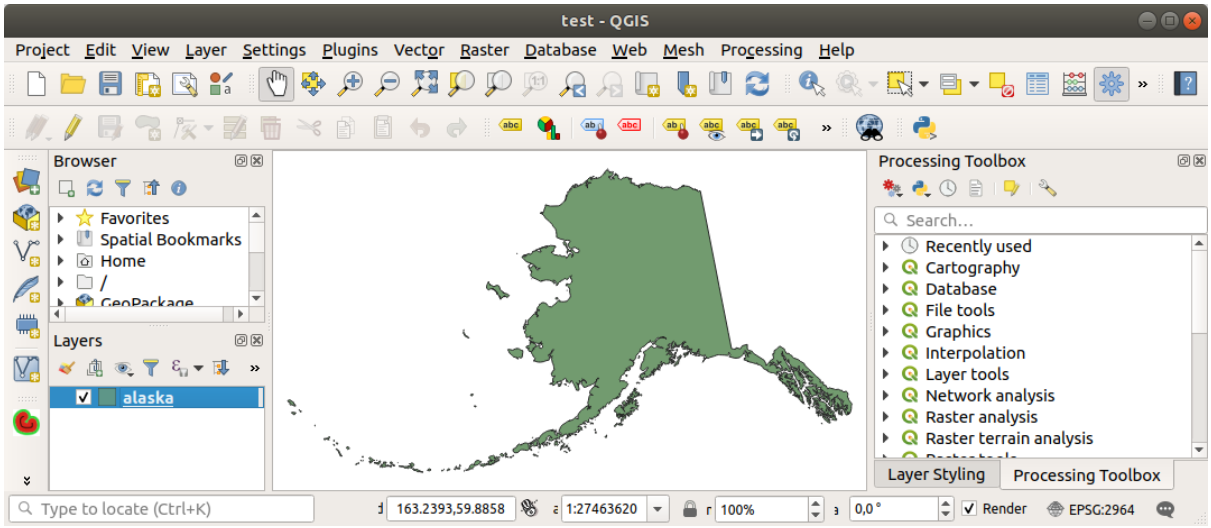


Fig. 11.7: QGIS med Shapefile över Alaska inläst

Observera

Eftersom det i vissa format som MapInfo (t.ex. `.tab`) eller Autocad (`.dxf`) är möjligt att blanda olika typer av geometrier i en och samma fil, öppnas vid inläsning av sådana dataset en dialogruta för att välja vilka geometrier som ska användas så att det blir en geometri per lager.

Flikarna Lägg till vektorlager och Lägg till rasterlager tillåter laddning av lager från andra källtyper än *File*:

- Du kan ladda specifika vektorformat som `ArcInfo Binary Coverage`, `UK. National Transfer Format`, liksom det råa TIGER-formatet från `US Census Bureau` eller `OpenfileGDB`. För att göra det väljer du *Katalog* som *Källtyp*. I detta fall kan en katalog väljas i dialogrutan efter att du tryckt på … Bläddra.

- Med källtypen *Database* kan du välja en befintlig databasanslutning eller skapa en till den valda databastypen. Några möjliga databastyper är `ODBC`, `Esri Personal Geodatabase`, `MS SQL Server` samt `PostgreSQL` eller `MySQL`.

  Om du trycker på knappen *New* öppnas dialogrutan *Create a New OGR Database Connection* vars parametrar är bland dem som du hittar i *Skapa en lagrad anslutning*. Om du trycker på *Open* kan du välja bland de tillgängliga tabellerna, t.ex. i PostGIS-aktiverade databaser.

- Källtypen *Protokoll: HTTP(S), cloud, etc.* källtyp öppnar data som lagras lokalt eller i nätverket, antingen offentligt tillgängligt eller i privata hinkar i kommersiella molnlagringstjänster. Protokolltyper som stöds är:

  - `HTTP/HTTPS/FTP`, med en *URI* och, om så krävs, en *authentication*.
  - Molnlagring såsom `AWS S3`, `Google Cloud Storage`, `Microsoft Azure Blob`, `Microsoft Azure Data Lake Storage`, `Alibaba OSS Cloud` och `Open Stack Swift Storage` stöder direkt kontroll över VSI *Credential Options* när du lägger till OGR-vektor- eller GDAL-rasterlager. Du måste först fylla i *Bucket eller container* och *Object key*. Därefter kan du lägga till de nödvändiga *Credential Options*.

    När du lägger till OGR-vektor- eller GDAL-rasterlager från de molnbaserade protokollen kan du också ställa in ytterligare *Credential options* för den specifika drivrutinen och skopan. När referensalternativ hittas i ett lagers URI kommer de också att ställas in automatiskt. Detta gör att olika lager kan använda olika autentiseringsuppgifter.
  - tjänst som stöder OGC `WFS 3` (fortfarande experimentell), använder `GeoJSON` eller `GEOJSON - Newline Delimited` format eller baserad på `CouchDB` databas. En *URI* krävs, med valfri

authentication.

- För alla vektorkälltyper är det möjligt att definiera *Encoding* eller att använda inställningen *Automatic* ►.

- Med källtypen *OGC API* kan du komma åt vektor och raster-data från servrar som implementerar OGC API-standarder. Så här använder du det här alternativet:

  1. Välj *OGC API* i dialogrutan *Data Source Manager*.
  2. Ange slutpunkten för den OGC API-tjänst som du vill ansluta till. Observera att du inte behöver prefixera endpointen med ”OGCAPI:”.
  3. Klicka på *Connect* för att upprätta en anslutning till servern.

Ladda ett nätlager

Ett mesh är ett ostrukturerat rutnät som vanligtvis har temporala och andra komponenter. Den spatiala komponenten innehåller en samling hörn, kanter och ytor i 2D- eller 3D-rymd. Mer information om mesh-lager finns på *Arbeta med Mesh-data*.

Så här lägger du till ett nätlager i QGIS:

1. Öppna dialogrutan *Data Source Manager*, antingen genom att välja den från menyn *Lager* ► eller genom att klicka på knappen Open Data Source Manager.
2. Aktivera fliken *Mesh* på den vänstra panelen
3. Tryck på knappen … Browse för att välja fil. *Flera format* stöds.
4. Välj filen och tryck på *Add*. Lagret kommer att läggas till med hjälp av den inbyggda mesh-renderingen.
5. Om den valda filen innehåller många mesh-lager kommer du att få en dialogruta där du kan välja vilka underlager som ska laddas. Gör ditt val och tryck på *OK* så laddas lagren med den ursprungliga mesh-renderingen. Det är också möjligt att ladda dem inom en grupp.

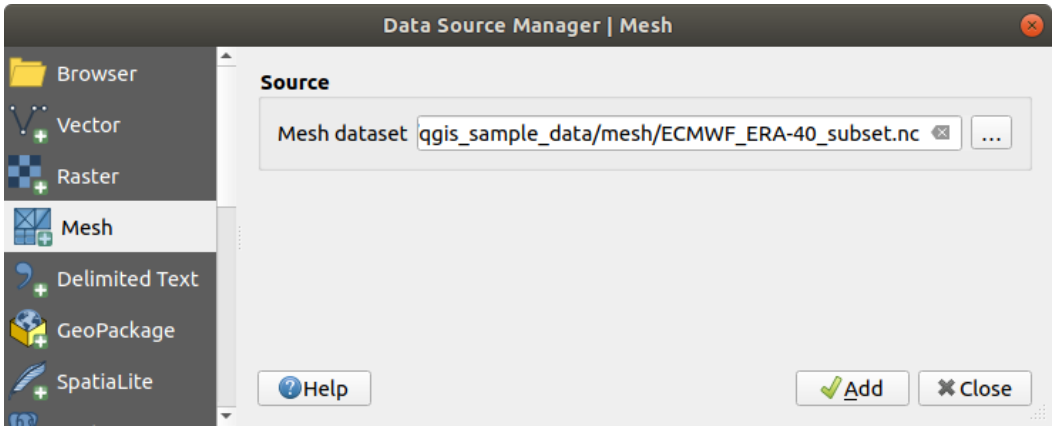


Fig. 11.8: Fliken Mesh i Datakällhanteraren

Importera en avgränsad textfil

Avgränsade textfiler (t.ex. `.txt`, `.csv`, `.dat`, `.wkt`) kan laddas med hjälp av de verktyg som beskrivs ovan. På så sätt kommer de att visas som enkla tabeller. Ibland kan avgränsade textfiler innehålla koordinater/geometrier som du kanske vill visualisera. Det är detta som *Add Delimited Text Layer* är utformat för.

1. Klicka på ikonen Open Data Source Manager för att öppna dialogrutan *Data Source Manager*
2. Aktivera fliken *Delimited Text* (avgränsad text)
3. Välj den avgränsade textfil som ska importeras (t.ex. `qgis_sample_data/csv/elevp.csv`) genom att klicka på knappen … Browse.
4. I fältet *Layer name* anger du det namn som ska användas för lagret i projektet (t.ex. `Elevation`).
5. Konfigurera inställningarna så att de passar ditt dataset och dina behov, enligt vad som förklaras nedan.

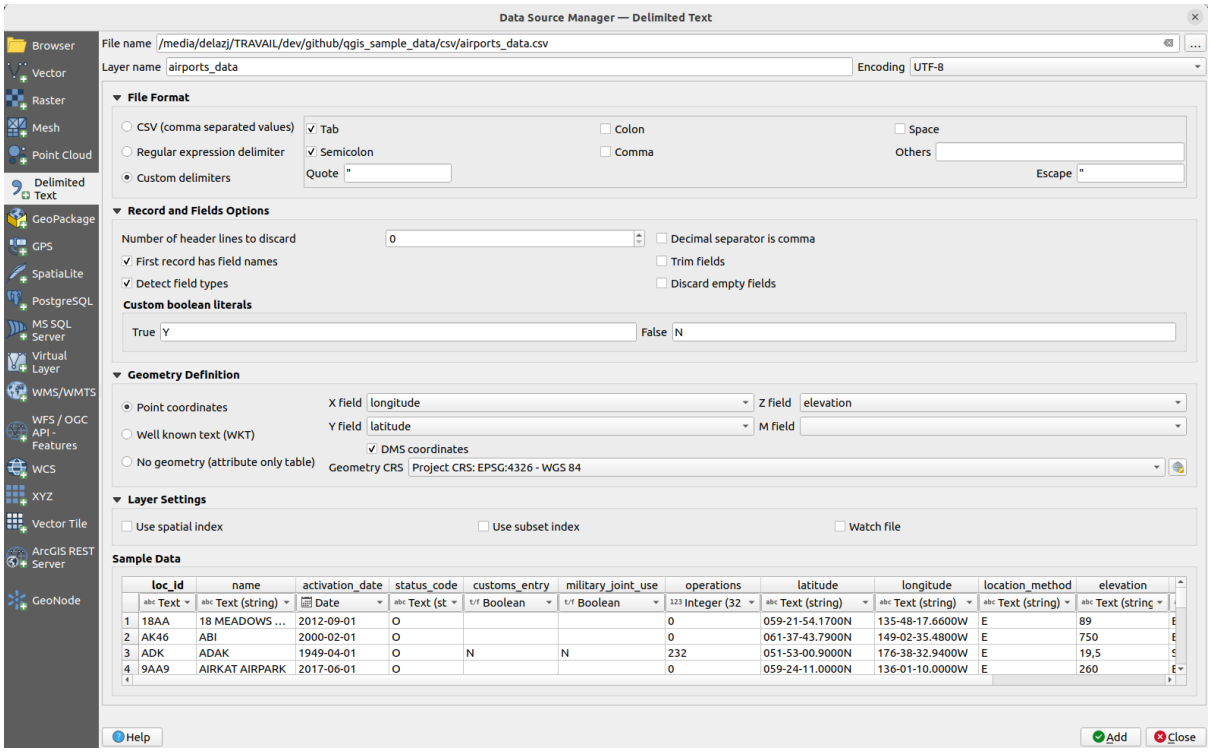


Fig. 11.9: Dialog för avgränsad text

Filformat

När filen väl har valts försöker QGIS analysera filen med den senast använda avgränsaren och identifiera fält och rader. För att QGIS ska kunna analysera filen korrekt är det viktigt att välja rätt avgränsare. Du kan ange en avgränsare genom att välja mellan:

- *CSV (kommaseparerade värden)* för att använda kommatecknet.
- *Regelbundna uttrycks avgränsare* och skriv in text i fältet *Utryck*. Om du t.ex. vill ändra avgränsaren till tabb använder du `t` (detta används i reguljära uttryck för tabbtecknet).
- *Custom delimiters*, välja bland några fördefinierade avgränsare som `comma`, `space`, `tab`, `semicolon`, … .

Poster och fält

Vissa andra praktiska alternativ kan användas för dataigenkänning:

- *Number of header lines to discard*: praktiskt när du vill undvika de första raderna i filen vid importen, antingen för att de är tomma rader eller med annan formatering.
- *Första posten har fältnamn*: värden på första raden används som fältnamn, annars använder QGIS fältnamnen `fält_1`, `fält_2`...
- *Detektera fälttyper*: identifierar automatiskt fälttypen. Om den inte är markerad behandlas alla attribut som textfält.
- *Decimalavgränsare är kommatecken*: du kan tvinga decimalavgränsaren att vara ett kommatecken.
- *Trim fields*: gör att du kan trimma inledande och avslutande mellanslag från fält.
- *Skippa tomma fält*.
- *Custom boolean literals*: gör att du kan lägga till ett anpassat par strängar som kommer att detekteras som booleska värden.

Detektering av fälttyp

QGIS försöker upptäcka fälttyperna automatiskt (om inte *Detect field types* är avmarkerad) genom att undersöka innehållet i en valfri CSVT-fil med sidovagnar (se GeoCSV specification) och genom att skanna hela filen för att se till att alla värden faktiskt kan konverteras utan fel.

Den upptäckta fälttypen visas under fältnamnet i tabellen för förhandsgranskning av exempeldata och kan ändras manuellt vid behov.

Följande fälttyper stöds:

- `Boolean` skiftlägesokänsliga bokstavspar som tolkas som booleska värden är `1`/`0`, `true`/*false*`, `t`/*f*`, `yes`/*no*`
- `Hela talet (heltal)`
- `Helt tal (heltal - 64 bitar)`
- `Decimaltal`: flyttal med dubbel precision
- `Datum`
- `Tid`
- `Datum och tid`
- `Text`

Definition av geometri

När filen har analyserats ställer du in *Geometry definition* till

- *Point coordinates* och ange *X field*, *Y field*, *Z field* (för 3-dimensionella data) och *M field* (för mätdimensionen) om lagret är av punktgeometrityp och innehåller sådana fält. Om koordinaterna definieras som grader/minuter/sekunder, aktivera kryssrutan *DMS coordinates*. Ange lämplig *Geometry CRS* med hjälp av widgeten Select CRS.
- *Well known text (WKT)* alternativ om den spatiala informationen representeras som WKT: välj *Geometry field* som innehåller WKT-geometrin och välj lämpligt *Geometry field* eller låt QGIS automatiskt upptäcka det. Ange lämplig CRS för *Geometry* med hjälp av widgeten Select CRS.

- Om filen innehåller icke-spatiala data, aktivera *Ingen geometri (endast attributtabell)* och den kommer att laddas som en vanlig tabell.

Inställningar för lager

Dessutom kan du aktivera:

- *Use spatial index* för att förbättra prestanda för visning och spatialt urval av funktioner.
- *Use subset index* för att förbättra prestanda för *subset filters* (när det definieras i lagrets egenskaper).
- *Watch file* för att se efter ändringar i filen av andra program medan QGIS körs.

I slutet klickar du på *Lägg till* för att lägga till lagret på kartan. I vårt exempel läggs ett punktlager med namnet `Elevation` till i projektet och beter sig som vilket annat kartlager som helst i QGIS. Det här lagret är resultatet av en fråga på källfilen `.csv` (alltså länkat till den) och kräver *to be saved* för att få ett spatialt lager på disken.

Exempeldata

När du ställer in parseregenskaperna uppdateras förhandsgranskningen av exempeldata med avseende på de tillämpade inställningarna.

Även i Sample Data Table är det möjligt att åsidosätta de automatiskt bestämda kolumntyperna.

Importera en DXF- eller DWG-fil

Filerna `DXF` och `DWG` kan läggas till i QGIS genom att enkelt dra och släppa dem från bläddrarpanelen. Du kommer att uppmanas att välja de underlager som du vill lägga till i projektet. Lagren läggs till med slumpmässiga stilegenskaper.

> **Observera**
>
> För DXF-filer som innehåller flera geometrityper (punkt, linje och/eller polygon) kommer namnet på lagren att genereras som *<filename.dxf> entities <geometry type>*.

För att behålla dxf/dwg-filstrukturen och dess symbologi i QGIS kan du använda det dedikerade *Project* ► *Import/Export* ► *Import Layers from DWG/DXF…*-verktyget som gör att du kan:

1. importera element från ritningsfilen till en GeoPackage-databas.
2. lägga till importerade element i projektet.

I dialogrutan *DWG/DXF Import*, för att importera ritningsfilens innehåll:

1. Ange platsen för *Källritning*, dvs. den DWG/DXF-ritningsfil som ska importeras.
2. Ange referenssystem för koordinater för data i ritningsfilen.
3. Ange platsen för *Target package*, d.v.s. GeoPackage-filen som ska lagra data. Om en befintlig fil anges kommer den att skrivas över.
4. Välj hur du vill importera `block` med den dedikerade kombinationsrutan:
   - *Expand Block Geometries*: importerar blocken i ritningsfilen som normala element.
   - *Expand Block Geometries and Add Insert Points*: importerar blocken i ritningsfilen som normala element och lägger till insättningspunkten som ett punktlager.
   - *Add Only Insert Points*: lägger till blockets insättningspunkt som ett punktlager.
5. Markera *Use curves* för att flytta de importerade lagren till en `curved` geometrityp.

6. Använd knappen *Import* för att importera ritningen till GeoPackage-destinationsfilen. GeoPackage-databasen kommer automatiskt att fyllas med ritningsfilens innehåll. Beroende på filens storlek kan detta ta lite tid.

När data från `.dwg` eller `.dxf` har importerats till GeoPackage-databasen fylls ramen i den nedre halvan av dialogrutan med en lista över lager från den importerade filen. Där kan du välja vilka lager som ska läggas till i QGIS-projektet:

1. Längst upp anger du en *Group name* för att gruppera ritningsfilerna i projektet. Som standard är detta inställt på filnamnet för källritningsfilen.
2. Markera lager som ska visas: Varje valt lager läggs till i en ad hoc-grupp som innehåller vektorlager för punkt-, linje-, etikett- och områdesfunktionerna i ritlagret. Stilen på lagren kommer att likna det utseende de ursprungligen hade i *CAD.
3. Välj om lagret ska vara synligt vid öppning.
4. Om du markerar alternativet *Slå samman lager* placeras alla lager i en enda grupp.
5. Tryck på *OK* för att öppna lagren i QGIS.

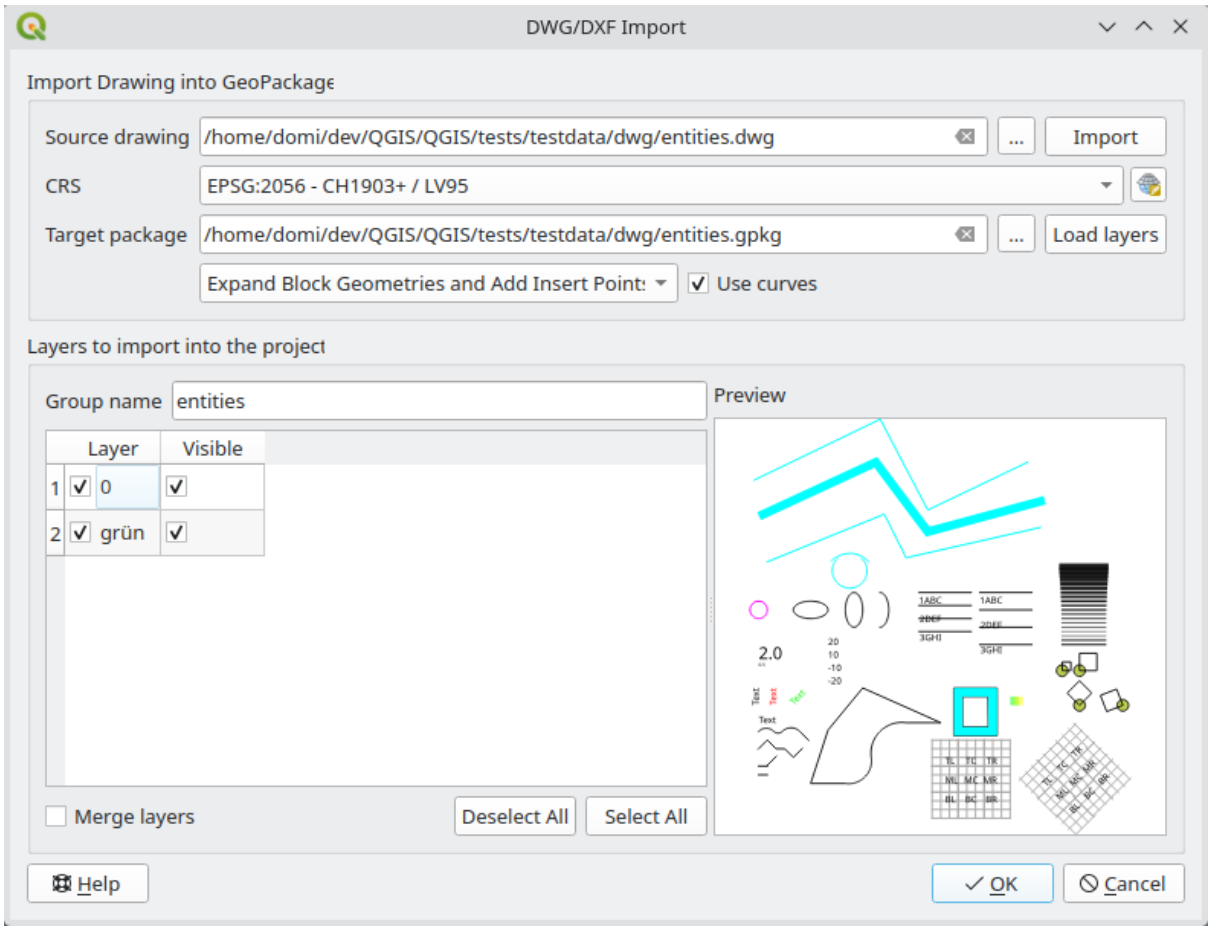


Fig. 11.10: Dialog för import av DWG/DXF-filer

Importera OpenStreetMap-vektorer

OpenStreetMap-projektet är populärt eftersom det i många länder inte finns några kostnadsfria geodata, t.ex. digitala vägkartor. Målet med OSM-projektet är att skapa en gratis redigerbar karta över världen från GPS-data, flygfotografering och lokal kunskap. För att stödja detta mål tillhandahåller QGIS stöd för OSM-data.

Med hjälp av *Browser Panel* kan du ladda en `.osm`-fil till kartbilden, varvid du får en dialogruta för att välja underlager baserat på geometritypen. De laddade lagren kommer att innehålla alla data för den geometritypen i filen `.osm` och behålla datastrukturen för filen `osm`.

SpatiaLite-lager

Första gången du laddar data från en SpatiaLite-databas börjar du med att:

- klicka på knappen Lägg till SpatiaLite-lager i verktygsfältet
- välja alternativet *Add SpatiaLite Layer…* från menyn *Layer* ► *Add Layer*
- eller genom att skriva `Ctrl+Shift+L`

Då öppnas ett fönster där du antingen kan ansluta till en SpatiaLite-databas som QGIS redan känner till (som du väljer från rullgardinsmenyn) eller definiera en ny anslutning till en ny databas. För att definiera en ny anslutning klickar du på *New* och använder filbläddraren för att peka på din SpatiaLite-databas, som är en fil med tillägget `.sqlite`.

QGIS har även stöd för redigerbara vyer i SpatiaLite.

GPS

Det finns dussintals olika filformat för lagring av GPS-data. Det format som QGIS använder kallas GPX (GPS eXchange format), vilket är ett standardiserat utbytesformat som kan innehålla valfritt antal waypoints, rutter och spår i samma fil.

Använd knappen … Browse för att välja GPX-filen och använd sedan kryssrutorna för att välja de funktionstyper som du vill ladda från GPX-filen. Varje funktionstyp laddas i ett separat lager.

Mer om hantering av GPS-data på *Arbeta med GPS-data*.

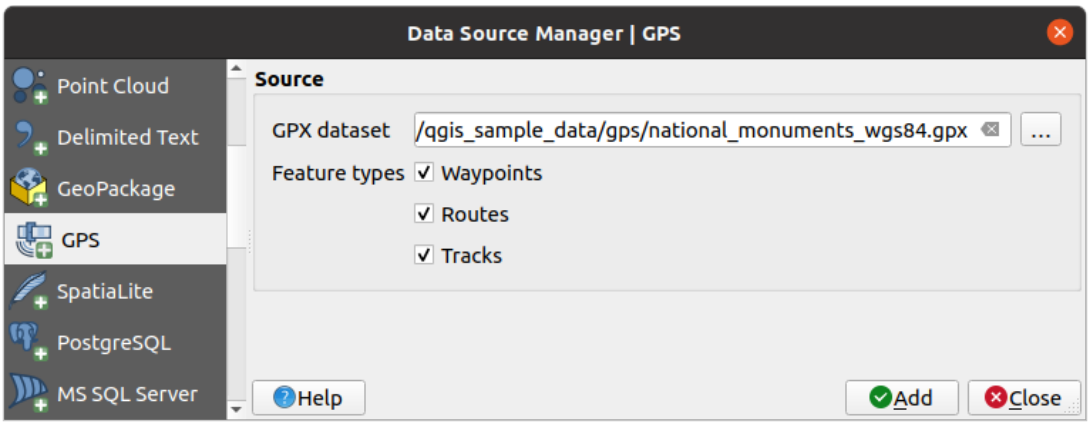


Fig. 11.11: Dialogrutan Ladda GPS-data

GRASS

Att arbeta med GRASS vektordata beskrivs i avsnitt *GRASS GIS-integration*.

Databasrelaterade verktyg

Skapa en lagrad anslutning

För att kunna läsa och skriva tabeller från ett databasformat som QGIS stöder måste du skapa en anslutning till den databasen. Även om *QGIS Browser Panel* är det enklaste och rekommenderade sättet att ansluta till och använda databaser, tillhandahåller QGIS andra verktyg för att ansluta till var och en av dem och ladda deras tabeller:

- *Add PostGIS Layer…* eller genom att skriva `Ctrl+Shift+D`
- *Lägg till MS SQL Server-lager*
- *Add Oracle Spatial Layer…* eller genom att skriva `Ctrl+Shift+O`
- *Add SAP HANA Spatial Layer…* eller genom att skriva `Ctrl+Shift+G`

Dessa verktyg är tillgängliga antingen från *Manage Layers Toolbar* eller från *Layer* ► *Add Layer* ► menyn. Anslutning till SpatiaLite-databasen beskrivs i *SpatiaLite-lager*.

Tips

Skapa en anslutning till databasen från QGIS Browser Panel

Om du väljer motsvarande databasformat i Browser-trädet, högerklickar och väljer connect får du upp dialogrutan för databasanslutning.

De flesta anslutningsdialogerna följer en gemensam struktur:

- ett avsnitt med information om inloggningsuppgifter för att ansluta till databasen
- ett avsnitt med alternativ för att ställa in vilka data som kan begäras i databasen

Anslutning till PostgreSQL

Första gången du använder en PostGIS-datakälla måste du skapa en anslutning till en databas som innehåller data. Tryck på lämplig knapp som visas ovan, vilket öppnar fliken *PostgreSQL* i dialogrutan *Data Source Manager*. För att komma åt anslutningshanteraren klickar du på knappen *New* för att visa dialogrutan *Create a New PostGIS Connection*.

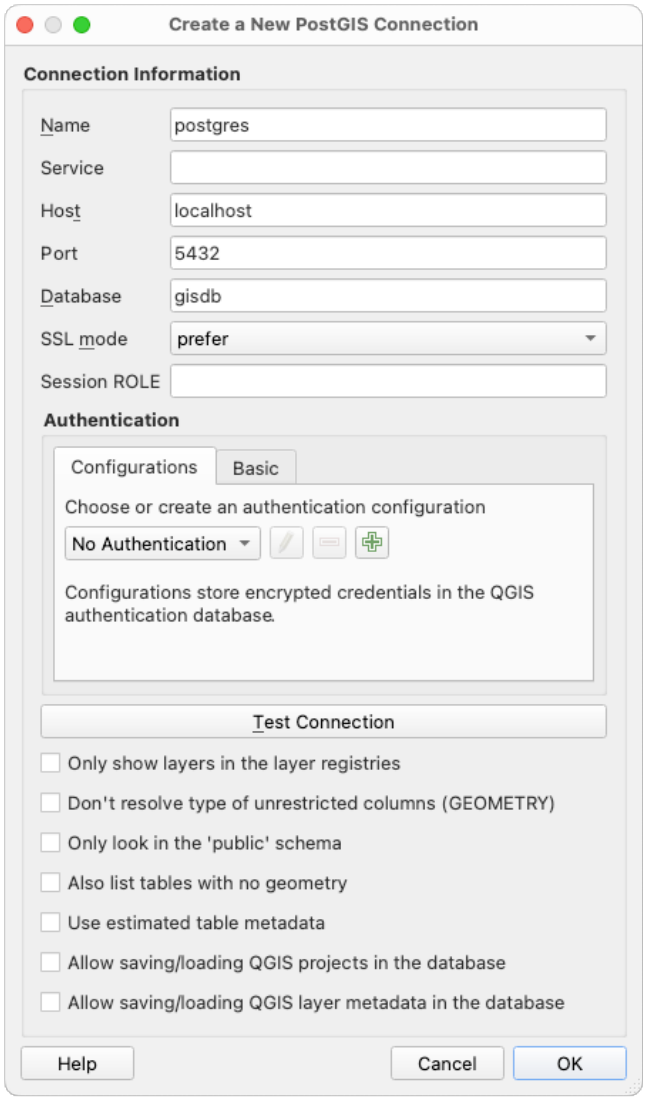


Fig. 11.12: Skapa en ny dialogruta för PostGIS-anslutning

- *Namn*: Ett namn för den här anslutningen. Det kan vara detsamma som *Database*.
- *Service*: Tjänsteparameter som ska användas som alternativ till värdnamn/port (och eventuellt databas). Detta kan definieras i `pg_service.conf`. Se avsnittet *PostgreSQL Service anslutningsfil* för mer information.
- *Host*: Namnet på databasens värd. Detta måste vara ett upplösningsbart värdnamn som kan användas för att öppna en TCP/IP-anslutning eller pinga värden. Om databasen finns på samma dator som QGIS anger du helt enkelt *localhost* här.
- *Port*: Portnummer som PostgreSQL-databasservern lyssnar på. Standardporten för PostGIS är `5432`.
- *Databas*: Namnet på databasen.
- *SSL-läge*: Inställning av SSL-kryptering. Följande alternativ finns tillgängliga:
  - *Prefer* (standardinställningen): Jag bryr mig inte om kryptering, men jag vill betala omkostnaderna för kryptering om servern stöder det.

  - *Kräver*: Jag vill att mina data ska krypteras och jag accepterar kostnaderna för detta. Jag litar på att nätverket kommer att se till att jag alltid ansluter till den server jag vill ha.
  - *Verifiera CA*: Jag vill att mina data ska krypteras och jag accepterar kostnaderna. Jag vill vara säker på att jag ansluter till en server som jag litar på.
  - *Verify Full*: Jag vill att mina data ska krypteras och jag accepterar kostnaden. Jag vill vara säker på att jag ansluter till en server jag litar på och att det är den jag anger.
  - *Allow*: Jag bryr mig inte om säkerhet, men jag är beredd att betala för kryptering om servern insisterar på det.
  - *Disable*: Jag bryr mig inte om säkerhet och jag vill inte betala för kryptering.
- *Session role*: används för att ange den aktuella användaridentifieraren för den aktuella sessionen. Detta är användbart för att automatiskt ge äganderätten till ett nytt objekt (tabell, vy, funktion) till session_role-gruppen och därmed dela äganderätten och tillhörande rättigheter med alla medlemmar i session_role-gruppen. Läs mer om session role.
- *Autentication*, basic.
  - *Användarnamn*: Användarnamn som används för att logga in i databasen.
  - *Lösenord*: Lösenord som används tillsammans med *Username* för att ansluta till databasen.

  Du kan spara någon eller båda av parametrarna `User name` och `Password`, i vilket fall de kommer att användas som standard varje gång du behöver ansluta till den här databasen. Om de inte sparas kommer du att uppmanas att ange autentiseringsuppgifterna för att ansluta till databasen i nästa QGIS-session. De anslutningsparametrar som du angav lagras i en tillfällig intern cache och returneras när ett användarnamn/lösenord för samma databas begärs, tills du avslutar den aktuella QGIS-sessionen.

> **Varning**
>
> **QGIS användarinställningar och säkerhet**
>
> På fliken *Authentication* sparar du **användarnamn** och **lösenord** för att behålla oskyddade uppgifter i anslutningskonfigurationen. Dessa **uppgifter kommer att vara synliga** om du t.ex. delar projektfilen med någon. Därför är det lämpligt att spara dina autentiseringsuppgifter i en *Autentication-konfiguration* istället (*Configurations*-fliken - Se *System för autentisering* för mer information) eller i en serviceanslutningsfil (se *PostgreSQL Service anslutningsfil* för exempel).

- Konfigurationer för *Autentisering*. Välj en autentiseringskonfiguration. Du kan lägga till konfigurationer med hjälp av knappen . Valmöjligheterna är:
  - Grundläggande autentisering
  - PKI PKCS#12-autentisering
  - Autentisering av PKI-sökvägar
  - PKI lagrat identitetscertifikat

Beroende på typ av databas kan du aktivera följande kryssrutor:

- *Visa bara lager i lagerregistren*
- *Lös inte typ av obegränsade kolumner (GEOMETRY)*
- *Titta bara i det ”offentliga” schemat*
- *Också lista tabeller utan geometri*: anger att tabeller utan geometri också ska listas som standard.
- *Use estimated table metadata*: Vid initiering av lager kan det behövas olika frågor för att fastställa egenskaperna hos de geometrier som lagras i databastabellen. När det här alternativet är markerat undersöker dessa frågor endast ett urval av raderna och använder tabellstatistiken, i stället för hela tabellen. Detta kan

drastiskt snabba upp bearbetningen av stora datamängder, men kan leda till felaktig karakterisering av lager (t.ex. kommer antalet funktioner i filtrerade lager inte att fastställas korrekt) och kan även orsaka konstiga beteenden om kolumner som ska vara unika faktiskt inte är det.

- *Tillåt att spara/ladda QGIS-projekt i databasen* - mer information *här*
- *Tillåt att spara/ladda metadata för QGIS-lager i databasen* - mer information *här*

När alla parametrar och alternativ är inställda kan du testa anslutningen genom att klicka på knappen *Test Connection* eller tillämpa den genom att klicka på knappen *OK*.

PostgreSQL Service anslutningsfil

Serviceanslutningsfilen gör att PostgreSQL-anslutningsparametrar kan associeras med ett enda servicenamn. Det servicenamnet kan sedan anges av en klient och de associerade inställningarna kommer att användas.

Den heter `.pg_service.conf` under *nix-system (GNU/Linux, macOS etc.) och `pg_service.conf` under Windows.

Servicefilen kan se ut så här:

```
[water_service]
host=192.168.0.45
port=5433
dbname=gisdb
user=paul
password=paulspass

[wastewater_service]
host=dbserver.com
dbname=water
user=waterpass
```

Observera

Det finns två tjänster i exemplet ovan: `water_service` och `wastewater_service`. Du kan använda dessa för att ansluta från QGIS, pgAdmin etc. genom att bara ange namnet på den tjänst du vill ansluta till (utan de omslutande parenteserna). Om du vill använda tjänsten med `psql` kan du göra `psql service=water_service`.

Du kan hitta alla PostgreSQL-parametrar här <https://www.postgresql.org/docs/current/libpq-connect.html#LIBPQ-PARAMKEYWORDS>`_

Observera

Om du inte vill spara lösenorden i servicefilen kan du använda alternativet .pg_pass.

Observera

QGIS Server och tjänster

När du använder en servicefil och QGIS Server måste du konfigurera tjänsten på serversidan också. Du kan följa QGIS Server-dokumentationen.

På * nix-operativsystem (GNU / Linux, macOS etc.) kan du spara filen `.pg_service.conf` i användarens hemkatalog och PostgreSQL-klienter kommer automatiskt att vara medvetna om den. Till exempel, om den inloggade

användaren är `web`, bör `.pg_service.conf` sparas i `/home/web/` katalogen för att direkt fungera (utan att ange några andra miljövariabler).

Du kan ange platsen för servicefilen genom att skapa miljövariabeln `PGSERVICEFILE` (kör t.ex. kommandot `export PGSERVICEFILE=/home/web/.pg_service.conf` under ditt *nix-operativsystem för att tillfälligt ställa in variabeln `PGSERVICEFILE`).

Du kan också göra servicefilen tillgänglig för hela systemet (alla användare) antingen genom att placera filen `.pg_service.conf` i `pg_config --sysconfdir` eller genom att lägga till miljövariabeln `PGSYSCONFDIR` för att ange den katalog som innehåller servicefilen. Om det finns servicedefinitioner med samma namn i användar- och systemfilen, har användarfilen företräde.

Varning

Det finns vissa förbehåll under Windows:

- Servicefilen ska sparas som `pg_service.conf` och inte som `.pg_service.conf`.
- Servicefilen bör sparas i Unix-format för att fungera. Ett sätt att göra det är att öppna den med Notepad++ och *Edit* ► *EOL Conversion* ► *UNIX Format* ► *File save*.
- Du kan lägga till miljövariabler på olika sätt; ett testat sätt, som är känt för att fungera tillförlitligt, är *Kontrollpanelen* ► *System och säkerhet* ► *System* ► *Avancerade systeminställningar* ► *Miljövariabler* lägga till `PGSERVICEFILE` med sökvägen - t.ex. `C:\Users\John\pg_service.conf`
- När du har lagt till en miljövariabel kan du också behöva starta om datorn.

Ansluta till Oracle Spatial

De spatiala funktionerna i Oracle Spatial hjälper användare att hantera geografiska data och platsdata i en inbyggd typ i en Oracle-databas. Anslutningsdialogen föreslår:

- **Databas**: SID eller SERVICE_NAME för Oracle-instansen;
- **Port**: Portnummer som Oracle-databasservern lyssnar på. Standardporten är `1521`;
- **Options**: Oracle-anslutningsspecifika alternativ (t.ex. OCI_ATTR_PREFETCH_ROWS, OCI_ATTR_PREFETCH_MEMORY). Formatet för options-strängen är en semikolonseparerad lista med optionsnamn eller option=värdepar;
- **Arbetsyta**: Arbetsyta att byta till;
- **Schema**: Schema i vilket data lagras

Alternativt kan du aktivera följande kryssrutor:

- ☑ *Sök endast i metadatatabellen*: begränsar de tabeller som visas till de som finns i vyn `all_sdo_geom_metadata`. Detta kan påskynda den första visningen av spatiala tabeller.
- ☑ *Leta bara efter användarens tabeller*: när du söker efter spatiala tabeller begränsas sökningen till tabeller som ägs av användaren.
- ☑ *Också lista tabeller utan geometri*: anger att tabeller utan geometri också ska listas som standard.
- ☑ *Use estimated table statistics for the layer metadata*: när lagret konfigureras krävs olika metadata för Oracle-tabellen. Detta inkluderar information som tabellens radantal, geometrityp och spatial utbredning av data i geometrikolumnen. Om tabellen innehåller ett stort antal rader kan det vara tidskrävande att fastställa dessa metadata. Genom att aktivera det här alternativet utförs följande snabba operationer för tabellmetadata: Radantalet bestäms från `all_tables.num_rows`. Tabellens utbredning bestäms alltid med funktionen SDO_TUNE.EXTENTS_OF, även om ett lagerfilter används. Tabellgeometrin bestäms utifrån de 100 första geometriska raderna i tabellen som inte är nollrader.
- ☑ *Bara befintliga geometrityper*: listar bara de befintliga geometrityperna och erbjuder inte att lägga till andra.

- *Inkludera ytterligare geometriattribut*.
- *Tillåt att spara/ladda QGIS-projekt i databasen* - mer information *här*
- *Schema*: Gör det möjligt att ange ett enda schema som en anslutning ska begränsas till. När den är inställd kommer endast tabeller från det matchande schemat att inkluderas i bläddrarpanelen och datakällan för anslutningen. Detta kan användas för att begränsa det databasarbete som krävs för att fylla i tabeller för en anslutning som pekar på ett stort databaslager.

Tips

Oracle Spatial Layers

Normalt definieras ett Oracle Spatial-lager av en post i tabellen **USER_SDO_METADATA**.

För att urvalsverktygen ska fungera korrekt rekommenderas att tabellerna har en **primary key**.

Ansluta till MS SQL Server

Som nämnts i *Skapa en lagrad anslutning* låter QGIS dig skapa MS SQL Server-anslutning via *Data Source Manager*.

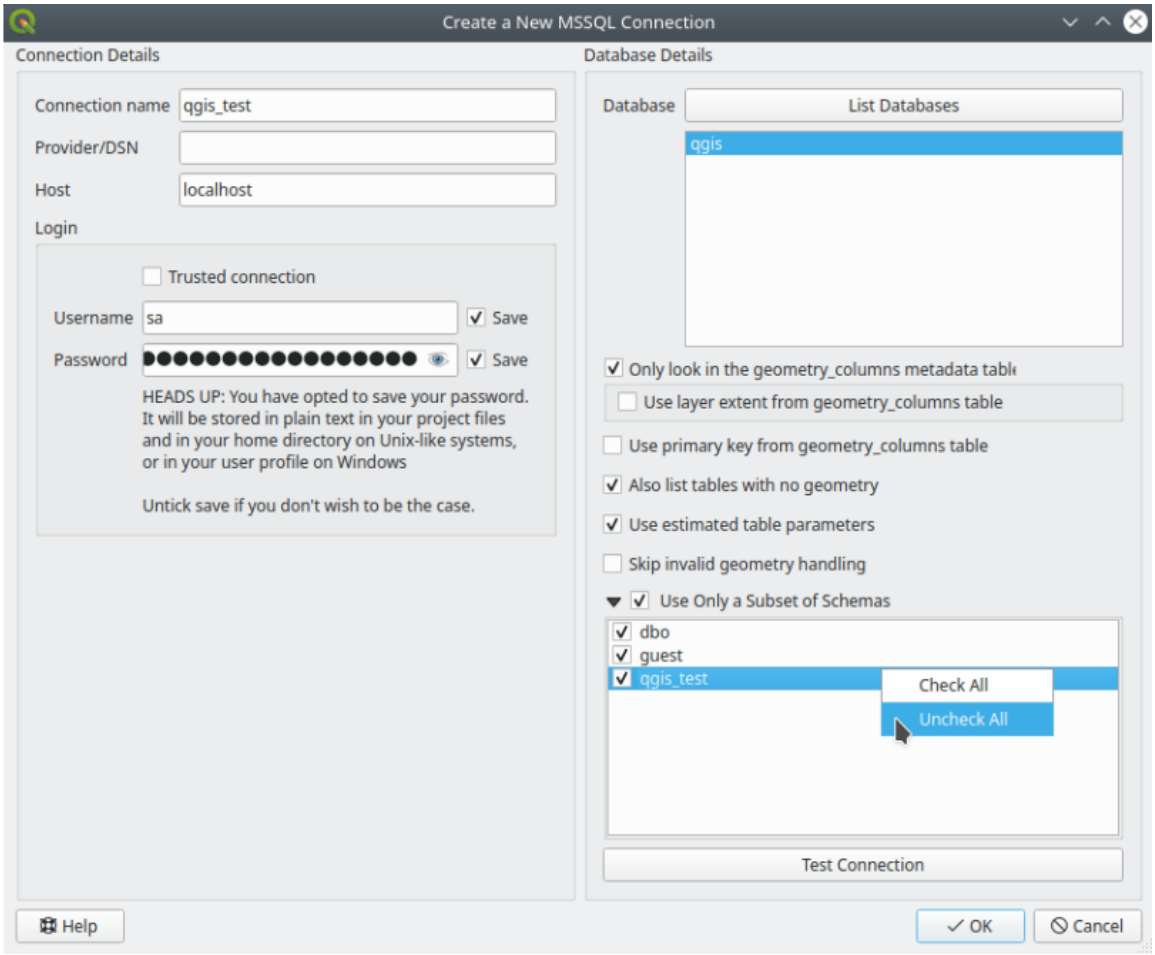


Fig. 11.13: MS SQL Server-anslutning

För att skapa en ny MS SQL Server-anslutning måste du ange några av följande uppgifter i dialogrutan *Connection Details*:

- *Anslutningens namn*

- *Provider/DSN*
- *Host*
- *Login* information. Du kan välja att *Spara* dina inloggningsuppgifter.

Navigera till avsnittet *Database Details* och klicka på knappen *List Databases* för att visa de tillgängliga dataseten. Välj de dataset som du vill ha och tryck sedan på *OK*. Alternativt kan du också utföra en *Test Connection*. När du klickar på *OK* stängs dialogrutan *Create a New MS SQL Server Connection* och i *Data Source Manager* trycker du på *Connect*, väljer ett lager och klickar sedan på *Add*.

Alternativt kan du aktivera följande alternativ:

- *Leta bara efter i metadatatabellen geometry_columns*: begränsar de tillgängliga tabellerna till dem i metadatatabellen `geometry_columns` när du söker efter tabeller. Detta kan påskynda tabellskanningen.
- *Use layer extent from geometry_columns table*: det här alternativet, som är beroende av det föregående, gör att QGIS kan hoppa över beräkningen av utsträckning när lager laddas och därmed minska den tid som krävs för att ladda dem. Det förlitar sig på utsträckning som anges manuellt med hjälp av ytterligare QGIS-specifika kolumner (`qgis_xmin`, `qgis_xmax`, `qgis_ymin`, `qgis_ymax`) i tabellen `geometry_columns`.
- *Use primary key from geometry_columns table*: tillåter QGIS att hoppa över beräkningen av primärnyckeln för vyer när de laddas, vilket minskar den tid som krävs för att ladda dem. Det bygger på namn som fylls i manuellt i en QGIS-specifik kolumnuppsättning `qgis_pkey` i tabellen `geometry_columns`. Om mer än en kolumn används för primärnyckeln ska de fyllas i som kommaseparerade värden.
- *Also list table with no geometry*: tabeller som inte har en geometrikolumn kopplad till sig visas också i listan över tillgängliga tabeller.
- *Use estimated table parameters*: endast uppskattade tabellmetadata kommer att användas. Detta undviker en långsam tabellskanning, men kan resultera i felaktiga lageregenskaper som t.ex. lagerutbredning.
- *Skippa hantering av ogiltig geometri*: all hantering av poster med ogiltig geometri inaktiveras. Detta snabbar upp leverantören, men om det finns några ogiltiga geometrier i en tabell blir resultatet oförutsägbart och kan innehålla saknade poster. Markera endast detta alternativ om du är säker på att alla geometrier som finns i databasen är giltiga och att alla geometrier eller tabeller som läggs till nyligen också är giltiga.
- *Use only a Subset of Schemas* gör att du kan filtrera scheman för MS SQL-anslutning. Om den är aktiverad visas endast markerade scheman. Du kan högerklicka för att *Check* eller *Uncheck* valfritt schema i listan.

Ansluta till SAP HANA

Observera

Du behöver SAP HANA Client för att ansluta till en SAP HANA-databas. Du kan hämta SAP HANA Client för din plattform på webbplatsen SAP Development Tools.

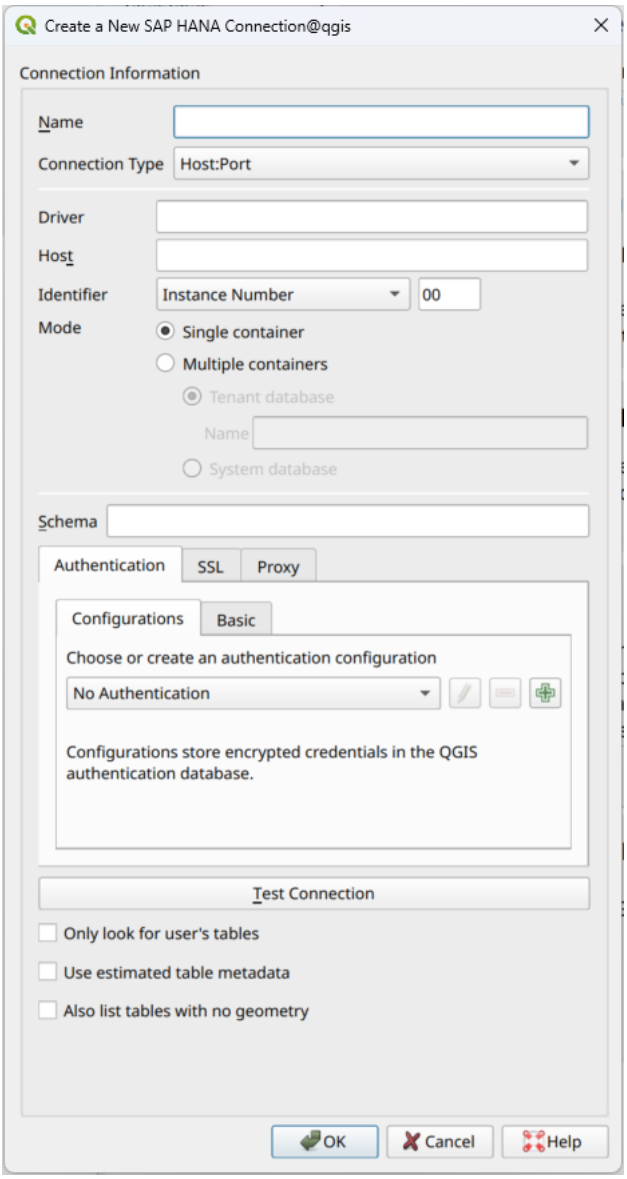


Fig. 11.14: Skapa en ny dialogruta för SAP HANA-anslutning

Följande parametrar kan anges:

- *Namn*: Ett namn för denna anslutning.
- *Drivrutin* : Namnet på HANA ODBC-drivrutinen. Det är `HDBODBC` om du använder 64-bitars QGIS, `HDBODBC32` om du använder 32-bitars QGIS. Det lämpliga drivrutinsnamnet anges automatiskt.
- *Driver* : Antingen det namn under vilket SAP HANA ODBC-drivrutinen har registrerats i `/etc/odbcinst.ini` eller den fullständiga sökvägen till SAP HANA ODBC-drivrutinen.

Installationsprogrammet för SAP HANA-klienten installerar ODBC-drivrutinen till `/usr/sap/hdbclient/libodbcHDB.so` som standard.

- *Host*: Namnet på databasens värd.
- *Identifierare*: Identifierar den instans som ska anslutas till på värden. Detta kan vara antingen *Instance Number* eller *Port Number*. Instansnummer består av två siffror, portnummer ligger i intervallet 1 till 65 535.
- *Mode*: Anger i vilket läge SAP HANA-instansen körs. Denna inställning beaktas endast om *Identifier* är inställd på *Instance Number*. Om databasen är värd för flera behållare kan du antingen ansluta till en klient med det namn som anges i *Tenant database* eller ansluta till systemdatabasen.
- *Schema*: Denna parameter är valfri. Om ett schemanamn anges kommer QGIS endast att söka efter data i det schemat. Om detta fält lämnas tomt kommer QGIS att söka efter data i alla scheman.
- *Autentisering* ► *Grundläggade*.
  - *Användarnamn*: Användarnamn som används för att ansluta till databasen.
  - *Lösenord*: Lösenord som används för att ansluta till databasen.
- *SSL-inställningar*
  - *Aktivera TLS/SSL-kryptering*: Aktiverar TLS 1.1 - TLS1.2-kryptering. Servern väljer den högsta tillgängliga.
  - *Leverantör*: Anger den kryptografiska biblioteksleverantör som används för SSL-kommunikation. *sapcrypto* bör fungera på alla plattformar, *openssl* bör fungera på , *mscrypto* bör fungera på och *commoncrypto* kräver att CommonCryptoLib installeras.
  - *Validera SSL-certifikat*: Om kryssrutan är markerad kommer SSL-certifikatet att valideras med hjälp av den truststore som anges i *Trust store file with public key*.
  - *Overskriv värdnamn i certifikat*: Anger det värdnamn som används för att verifiera serverns identitet. Det värdnamn som anges här verifierar serverns identitet i stället för det värdnamn med vilket anslutningen upprättades. Om du anger `*` som värdnamn valideras inte serverns värdnamn. Andra jokertecken är inte tillåtna.
  - *Keystore-fil med privat nyckel*: Ignoreras för närvarande. Denna parameter kan göra det möjligt att autentisera via certifikat istället för via användare och lösenord i framtiden.
  - *Trust store-fil med offentlig nyckel*: Anger sökvägen till en trust store-fil som innehåller serverns offentliga certifikat om OpenSSL används. Vanligtvis innehåller förtroendelagret rotcertifikatet eller certifikatet för den certifikatutfärdare som signerat serverns offentliga certifikat. Om du använder det kryptografiska biblioteket CommonCryptoLib eller msCrypto ska du lämna den här egenskapen tom.
- *Leta bara efter användarens tabeller*: Om kryssrutan är markerad söker QGIS endast efter tabeller och vyer som ägs av den användare som ansluter till databasen.
- *Use estimated table metadata*: Om markerad kommer uppskattade tabellmetadata att användas om de är tillgängliga. För stora tabeller undviker detta långsamma tabellbelastningar och potentiellt dyra beräkningar, men kan resultera i felaktiga lageregenskaper som lagerutsträckning. Den snabba uppskattningen av omfattningen är tillgänglig från och med QRC1/2024 och SP8 i HANA Cloud respektive HANA On-Premise.
- *Lista även tabeller utan geometrier*: Om kryssrutan är markerad söker QGIS även efter tabeller och vyer som inte innehåller en spatial kolumn.

Tips

Anslutning till SAP HANA Cloud

Om du vill ansluta till en SAP HANA Cloud-instans måste du vanligtvis ställa in *Port Number* till `443` och markera *Enable TLS/SSL encryption*.

Ladda ett databaslager

När du har definierat en eller flera anslutningar till en databas (se avsnitt *Skapa en lagrad anslutning*) kan du ladda lager från den. Detta kräver naturligtvis att data finns tillgängliga. Se avsnitt *Importera data till PostgreSQL* för en diskussion om hur man importerar data till en PostGIS-databas.

Om du vill ladda ett lager från en databas kan du utföra följande steg:

1. Öppna motsvarande flik för databasen i dialogrutan *Data Source Manager*.
2. Välj anslutningsnamn i rullgardinsmenyn och tryck på *Connect*.
3. Markera eller avmarkera *Lista även tabeller utan geometri*.
4. Alternativt kan du använda någon av kryssrutorna i *Search Options* för att minska listan över tabeller till de som matchar din sökning. Du kan också ställa in det här alternativet innan du trycker på knappen *Connect*, vilket snabbar upp databashämtningen.
5. Leta reda på det eller de lager du vill lägga till i listan över tillgängliga lager.
6. Markera det genom att klicka på det. Du kan välja flera lager genom att hålla ned `Shift` eller `Ctrl` medan du klickar.
7. Om tillämpligt, använd knappen *Set Filter* (eller dubbelklicka på lagret) för att starta dialogrutan *Query Builder* (se avsnitt *Frågeverktyg*) och definiera vilka funktioner som ska läsas in från det valda lagret. Filteruttrycket visas i kolumnen `sql`. Denna begränsning kan tas bort eller redigeras i ramen *Lageregenskaper* ► *General* ► *Provider Feature Filter*.
8. Kryssrutan i kolumnen ”Välj vid id” som är aktiverad som standard får fram funktionens id utan attribut och snabbar i allmänhet upp dataladdningen.
9. Klicka på knappen *Add* för att lägga till lagret på kartan.

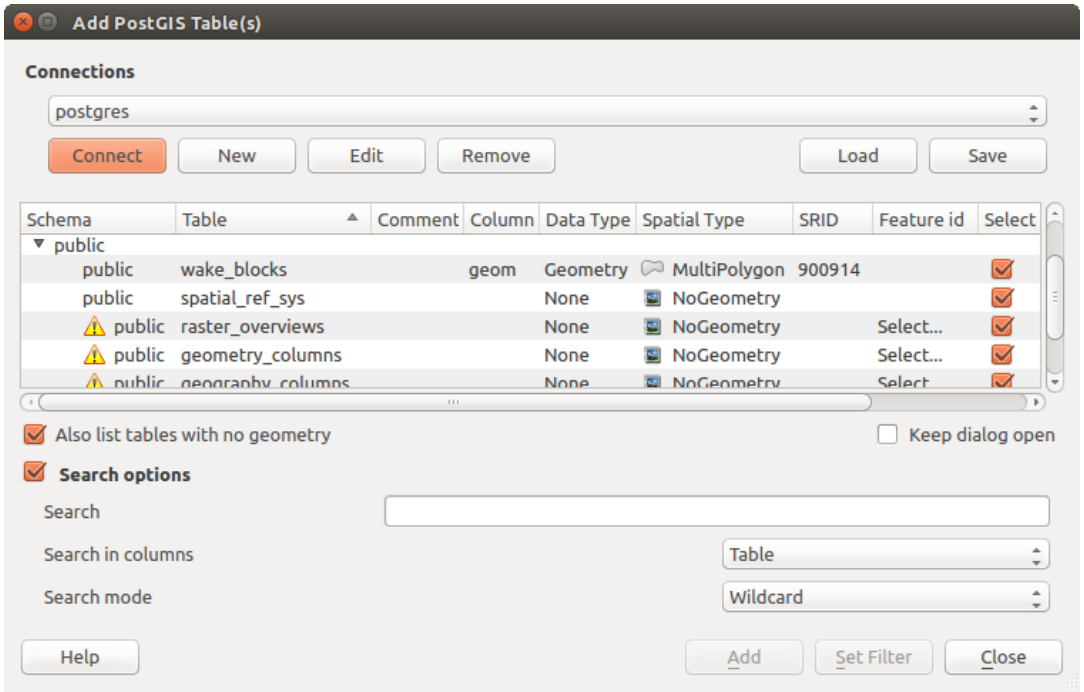


Fig. 11.15: Dialog för att lägga till PostGIS-tabell(er)

Tips

Använd bläddrarpanelen för att påskynda inläsningen av databastabeller

Att lägga till DB-tabeller från *Data Source Manager* kan ibland vara tidskrävande eftersom QGIS hämtar statistik och egenskaper (t.ex. geometrityp och fält, CRS, antal funktioner) för varje tabell i förväg. För att undvika detta är det bättre att använda *Browser Panel* eller *DB Manager* för att dra och släppa databastabellerna i kartbilden när *anslutningen har ställts in*.

11.1.4 Panelen för sökning av metadata för lager

Som standard kan QGIS hämta metadata för lager från anslutningar eller dataleverantörer som tillåter lagring av metadata (mer information om *Spara metadata till databasen*). Panelen *Metadata search* gör det möjligt att bläddra bland lagren efter deras metadata och lägga till dem i projektet (antingen genom att dubbelklicka eller med knappen *Lägg till*). Listan kan filtreras:

- med text, som tittar på en uppsättning metadataegenskaper (identifierare, titel, sammanfattning)
- efter spatial utsträckning, med hjälp av aktuell *projektutsträckning* eller kartans canvasutsträckning
- av lagrets (geometrins) typ

Observera

Källorna till metadata implementeras genom ett system med metadataleverantörer som kan utökas med tillägg.

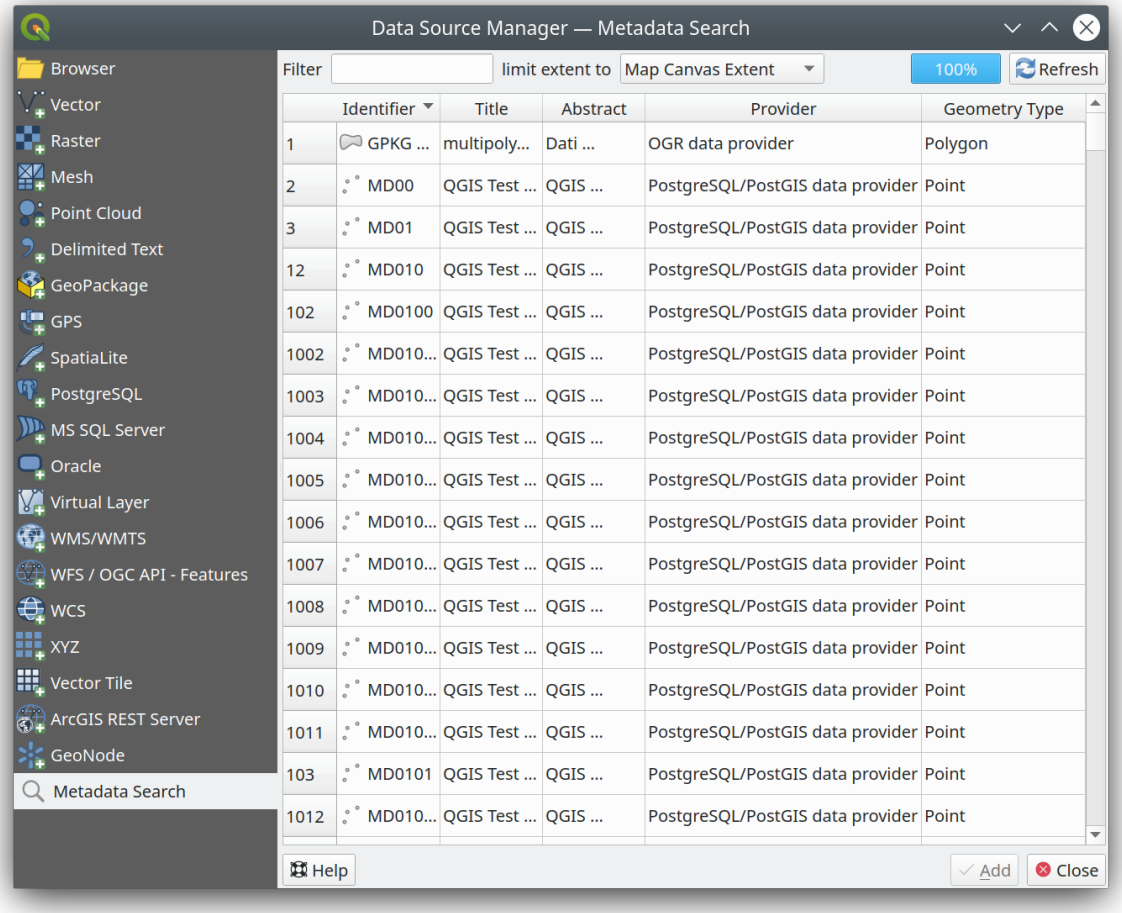


Fig. 11.16: Panel för sökning av metadata för lager

11.1.5 QGIS anpassade format

QGIS föreslår två anpassade format:

- Temporary Scratch Layer: ett minneslager som är knutet till projektet (se *Skapa ett nytt temporärt skrapskikt* för mer information)
- Virtuella lager: ett lager som är resultatet av en fråga på ett eller flera andra lager (se *Skapa virtuella lager* för mer information)

11.1.6 QLR - Definitionsfil för QGIS-lager

Lagerdefinitioner kan sparas som en *Lagerdefinitionsfil* (QLR - `.qlr`) med *Exportera* ► *Spara som lagerdefinitionsfil…* i lagrets kontextmeny.

QLR-formatet gör det möjligt att dela ”kompletta” QGIS-lager med andra QGIS-användare. QLR-filer innehåller länkar till datakällorna och all QGIS-stilinformation som behövs för att utforma lagret.

QLR-filer visas i Browser-panelen och kan användas för att lägga till lager (med sparade stilar) i Layers-panelen. Du kan också dra och släppa QLR-filer från systemets filhanterare till kartbilden.

11.1.7 Ansluta till webbtjänster

Med QGIS kan du få tillgång till olika typer av OGC-webbtjänster (WM(T)S, WFS(-T), WCS, CSW, …). Tack vare QGIS Server kan du också publicera sådana tjänster. QGIS-Server-manual innehåller beskrivningar av dessa funktioner.

Använda Vector Tiles-tjänster

Vector Tile-tjänster kan läggas till via fliken *Vector Tiles* i dialogrutan *Data Source Manager* eller den kontextuella menyn för posten *Vector Tiles* i panelen *Browser*. Tjänster kan vara antingen en *New Generic Connection…* eller en *New ArcGIS Vector Tile Service Connection…*.

Du skapar en tjänst genom att lägga till:

- a *Namn*
- a *Style URL*: en URL till en MapBox GL JSON-stilkonfiguration. Om den tillhandahålls kommer den stilen att tillämpas när lagren från anslutningen läggs till i QGIS. När det gäller anslutningar till Arcgis vector tile-tjänster åsidosätter URL:en den standardstilkonfiguration som anges i serverkonfigurationen.

  Du kan ladda vektorplattor direkt från en *Style URL*. Datakällan analyseras automatiskt från stilen, och URL:er med flera källor stöds. Det gör *Source URL* till ett tillval.
- *Source URL*: av typen `http://example.com/{z}/{x}/{y}.pbf` för generiska tjänster och `http://example.com/arcgis/rest/services/Layer/VectorTileServer` för ArcGIS-baserade tjänster. Tjänsten måste tillhandahålla tiles i formatet `.pbf`.
- *Min. Zoom Level* och *Max. Zoom Level*. Vector Tiles har en pyramidstruktur. Genom att använda dessa alternativ har du möjlighet att individuellt generera lager från kakelpyramiden. Dessa lager kommer sedan att användas för att rendera vektorkartan i QGIS.

  För Mercator-projektion (används av OpenStreetMap Vector Tiles) representerar zoomnivå 0 hela världen i en skala av 1:500.000.000. Zoomnivå 14 representerar skalan 1:35.000.
- konfigurationen *authentication* om det behövs
- a *Referer*

Fig. 11.17 visar dialogrutan med Vector Tiles servicekonfiguration.

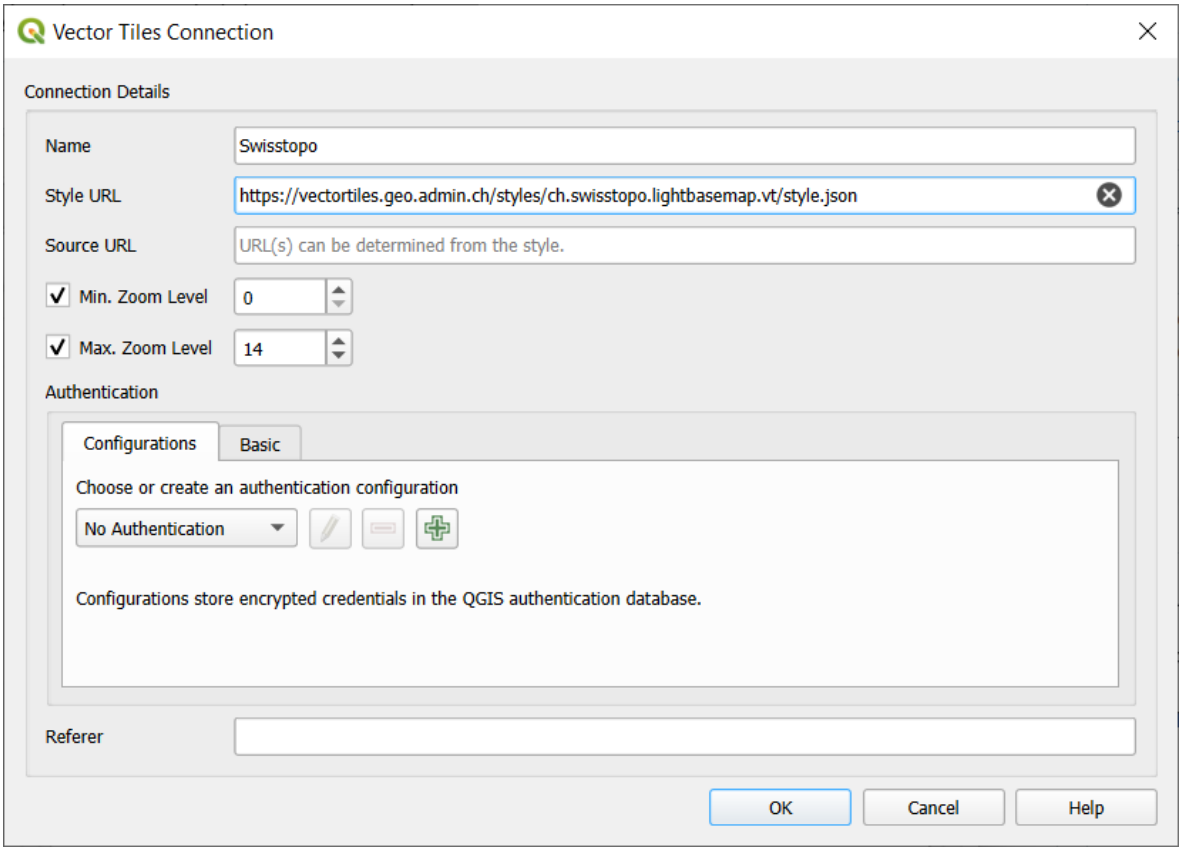


Fig. 11.17: Vector Tiles - Konfiguration av tjänster

Konfigurationer kan sparas i `.XML`-filen (*Save Connections*) via *Vector Tiles* i *Data Source Manager*-dialogrutan eller dess snabbmeny i *Browser*-panelen. På samma sätt kan de läggas till från en fil (*Load Connections*).

När en anslutning till en vector tile-tjänst har upprättats är det möjligt att:

- *Ändra* inställningarna för anslutning av vektorplattor
- *Avlägsna* anslutningen
- Från *Browser*-panelen högerklickar du på posten och du kan också:
  - *Lägg till lager i projektet*: ett dubbelklick lägger också till lagret
  - Visa *Lageregenskaper…* och få tillgång till metadata och en förhandsvisning av de data som tillhandahålls av tjänsten. Fler inställningar är tillgängliga när lagret har laddats in i projektet.

Använda XYZ Tile-tjänster

XYZ Tile-tjänster kan läggas till via fliken *XYZ* i dialogrutan *Data Source Manager* eller via snabbmenyn för posten *XYZ Tiles* i panelen *Browser*. Som standard tillhandahåller QGIS några standardtjänster för XYZ Tiles som är färdiga att använda:

- *Mapzen Global Terrain*, vilket möjliggör en omedelbar tillgång till global DEM-källa för projekten. Mer information och resurser finns på https://registry.opendata.aws/terrain-tiles/
- *OpenStreetMap* för att komma åt världens 2D-karta. Fig. 11.18 visar dialogrutan med konfigurationen av OpenStreetMap XYZ Tile-tjänsten.

För att lägga till en ny tjänst, tryck på *New* (respektive *New Connection* från Browser-panelen) och ange:

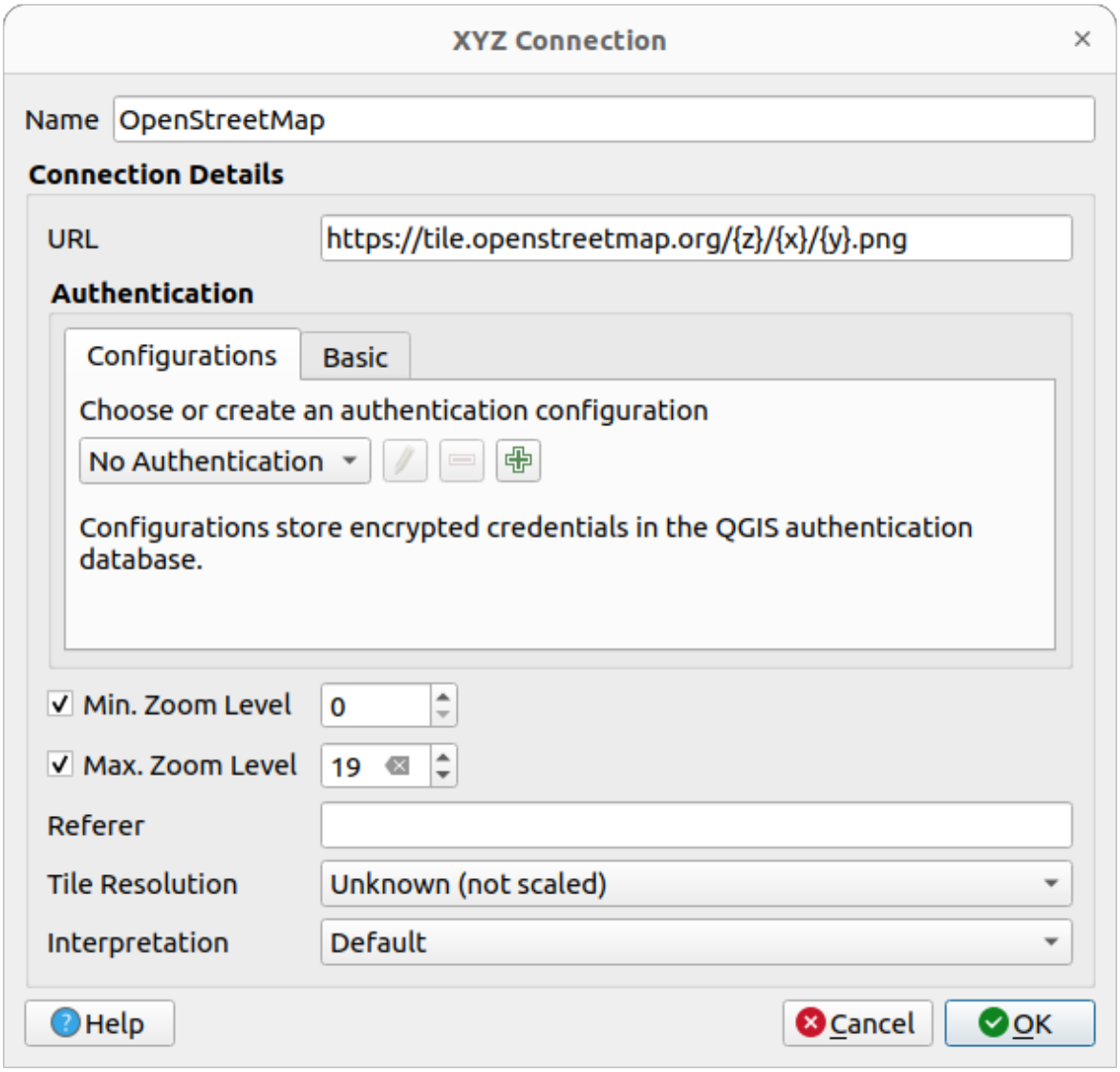


Fig. 11.18: XYZ Tiles - OpenStreetMap-konfiguration

- a *Namn*
- i *URL* kan du lägga till `http://example.com/{z}/{x}/{y}.png` eller `file:///local_path/{z}/{x}/{y}.png`
- konfigurationen *authentication* om det behövs
- inställningarna *Min. Zoomnivå* och *Max. Zoomnivå*
- a *Referer*
- *Tile Resolution*: möjliga värden är *Unknown (ej skalad)*, *Standard (256x256 / 96DPI)* och *High (512x512 / 192DPI)*
- *Interpretation*: konverterar WMTS/XYZ rasterdatauppsättningar till ett rasterlager av typen single band float enligt ett fördefinierat kodningsschema. Scheman som stöds är *Default* (ingen konvertering görs), *MapTiler Terrain RGB* och *Terrarium Terrain RGB*. Den valda konverteraren översätter RGB-källvärdena till float-värden för varje pixel. När lagret har laddats kommer det att presenteras som ett rasterlager med ett enda band med flytande punkter, redo för styling med QGIS vanliga *raster renderers*.

Tryck på *OK* för att upprätta förbindelsen. Det kommer sedan att vara möjligt att:

- *Lägg till* det nya lagret i projektet; det laddas med det namn som anges i inställningarna.
- *Ändra* inställningarna för XYZ-anslutningen
- *Avlägsna* anslutningen
- Från *Browser*-panelen högerklickar du på posten och du kan också:
  - *Exportera lager…* ► *Till fil*, *Spara det som ett raster*
  - *Lägg till lager i projektet*: ett dubbelklick lägger också till lagret
  - Visa *Lageregenskaper…* och få tillgång till metadata och en förhandsvisning av de data som tillhandahålls av tjänsten. Fler inställningar är tillgängliga när lagret har laddats in i projektet.

Konfigurationer kan sparas till `.XML`-filen (*Save Connections*) via *XYZ*-posten i *Data Source Manager*-dialogrutan eller dess kontextuella meny i *Browser*-panelen. På samma sätt kan de läggas till från en fil (*Load Connections*).

XML-filen för OpenStreetMap ser ut så här:

```
<!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>
```

Tips

Laddar XYZ-brickor utan att skapa en anslutning

Det är också möjligt att lägga till XYZ tiles i ett projekt utan att nödvändigtvis lagra dess anslutningsinställningar i din användarprofil (t.ex. för en dataset som du kanske bara behöver en gång). På fliken *Data Source Manager* ► *XYZ* redigerar du alla egenskaper i gruppen *Connection Details*. Fältet *Name* ovan bör ändras till `Custom`. Tryck på *Add* för att ladda lagret i projektet. Det får som standard namnet `XYZ Layer`.

Exempel på tjänster från XYZ Tile:

- OpenStreetMap Monochrome: *URL*: `http://tiles.wmflabs.org/bw-mapnik/{z}/{x}/{y}.png`, *Min. Zoom Level*: 0, *Max. Zoom Level*: 19.
- Google Maps: *URL*: `https://mt1.google.com/vt/lyrs=m&x={x}&y={y}&z={z}`, *Min. Zoom Level*: 0, *Max. Zoom Level*: 19.
- Öppna väderkarta Temperatur: *URL*: `http://tile.openweathermap.org/map/temp_new/{z}/{x}/{y}.png?appid={api_key}` *Min. Zoom Level*: 0, *Max. Zoom Level*: 19.

Använda ArcGIS REST-servrar

ArcGIS REST Servers kan läggas till via fliken *ArcGIS REST Server* i dialogrutan *Data Source Manager* eller via kontextmenyn för posten *ArcGIS REST Servers* i panelen *Bläddrare*. Tryck på *Ny* (respektive *Ny anslutning*) och ange:

- a *Namn*
- den *URL*
- a *Prefix*: Detta används för att ange proxyprefixet i URL:en, vilket är nödvändigt för vissa ArcGIS-servrar som använder webbproxyprefix.
- a *URL för slutpunkt i gemenskapen*
- a *URL för innehållets slutpunkt*

- konfigurationen *authentication* om det behövs
- a *Referer*

Observera

ArcGIS Feature Service-anslutningar som har sina motsvarande URL:er för Portal-slutpunkter kan utforskas av innehållsgrupper i bläddrarpanelen.

Om en anslutning har Portal-slutpunkterna inställda visas en mapp med ”Grupper” och ”Tjänster” när du expanderar anslutningen i bläddraren, istället för den fullständiga listan med tjänster som vanligtvis visas. Om du expanderar gruppmappen visas en lista över alla innehållsgrupper som användaren är medlem i. Varje grupp kan expanderas för att visa de serviceobjekt som tillhör den gruppen.

Konfigurationer kan sparas till `.XML`-fil (*Spara anslutningar*) via *ArcGIS REST Server* i *Data Source Manager*-dialogrutan. På samma sätt kan de läggas till från en fil (*Läs in anslutningar*).

När en anslutning till en ArcGIS REST Server har upprättats är det möjligt att:

- *Redigera* anslutningsinställningarna för ArcGIS REST Server
- *Avlägsna* anslutningen
- *Friskställ* anslutningen
- använda ett filter för de tillgängliga lagren
- välj från en lista över tillgängliga lager med alternativet att *Begära endast funktioner som överlappar den aktuella vyutbredningen*
- Från *Browser*-panelen högerklickar du på anslutningsposten så kan du:
  - *Refresh*
  - *Redigera anslutning…*
  - *Avlägsna anslutning…*
  - *View Service Info* som öppnar standardwebbläsaren och visar serviceinformationen.
- Högerklicka på lagerposten så kan du också göra det:
  - *View Service Info* som öppnar standardwebbläsaren och visar serviceinformationen.
  - *Exportera lager…* ► *Till fil*
  - *Lägg till lager i projektet*: ett dubbelklick lägger också till lagret
  - Visa *Lageregenskaper…* och få tillgång till metadata och en förhandsvisning av de data som tillhandahålls av tjänsten. Fler inställningar är tillgängliga när lagret har laddats in i projektet.

Använda 3D-tjänster för kaklade scener

QGIS stöder flera format av 3D-dataset med mosaik, som grupperas tillsammans som ”mosaikscener”. Dessa inkluderar Cesium 3D Tiles och Quantized Mesh tiles.

Om du vill ladda en datauppsättning med kaklade scener i QGIS använder du fliken *Scene* i dialogrutan *Data Source Manager*.

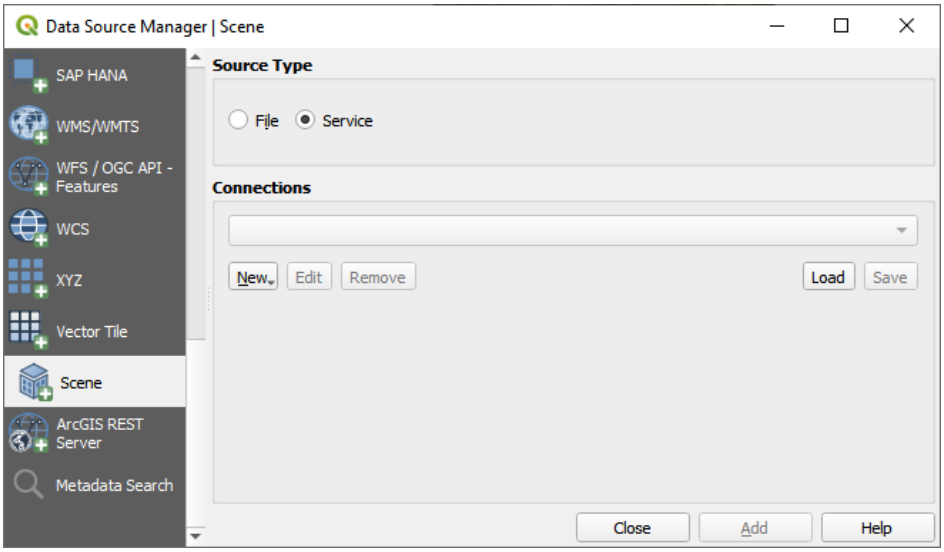


Fig. 11.19: Chef för datakällor - Scene

Skapa en anslutning genom att klicka på *New*. Du kan lägga till en *New Cesium 3D Tiles Connection* eller en *New Quantized Mesh Connection*.

Välj ett *Name* och ange *URL* till URL:en för en JSON-fil med lagerbeskrivning.

URL:en kan vara fjärrbaserad (t.ex. `http://exempel.se/tileset.json`) eller lokal (t.ex. `file:///sökväg/till/tiles/tileset.json`).

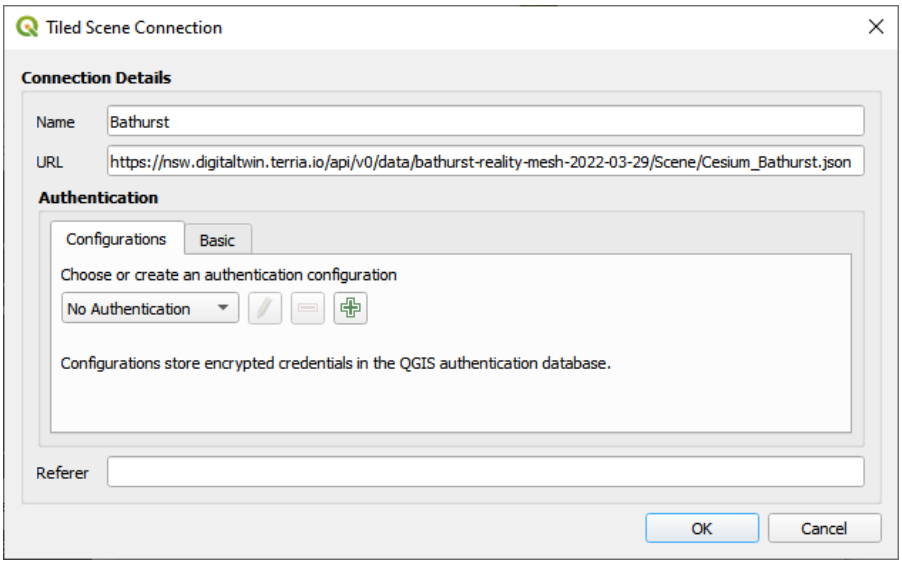


Fig. 11.20: Anslutning för kaklad scen

Du kan också lägga till tjänsten från *Browser Panel*.

När du har skapat en ny anslutning kan du *Lägga till* det nya lagret på din karta.

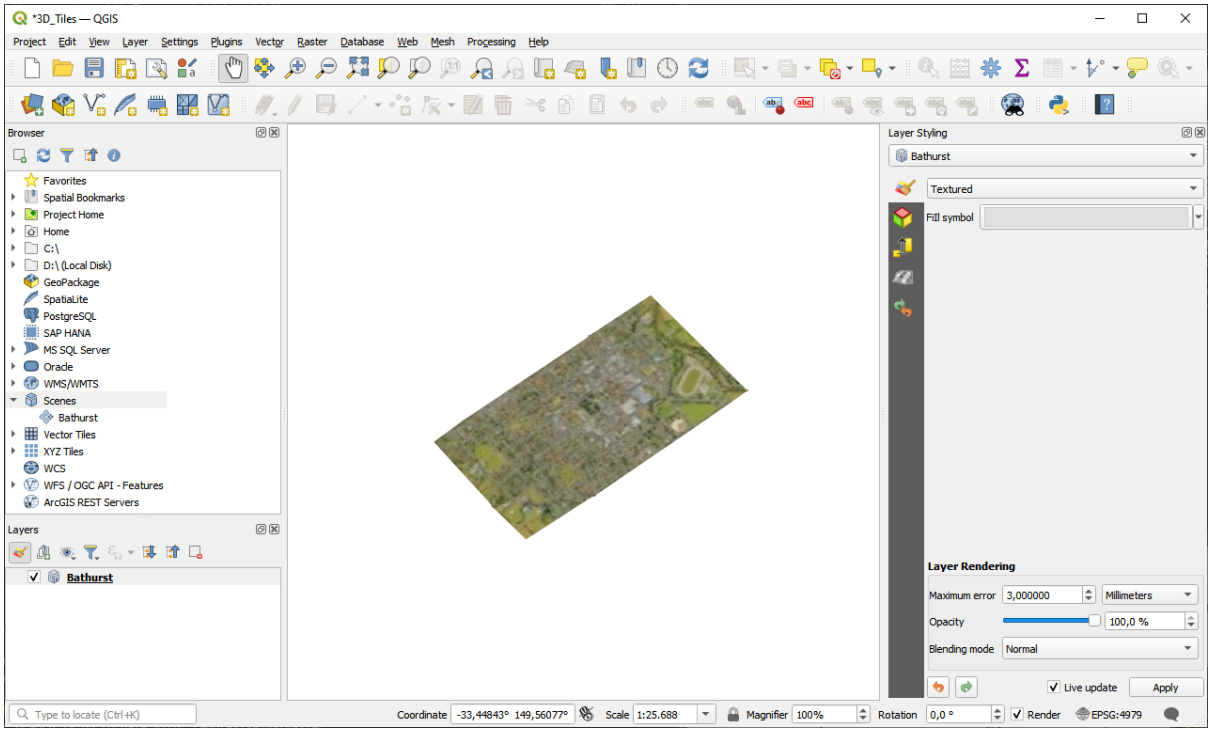


Fig. 11.21: 3D-lager för plattor - strukturerad

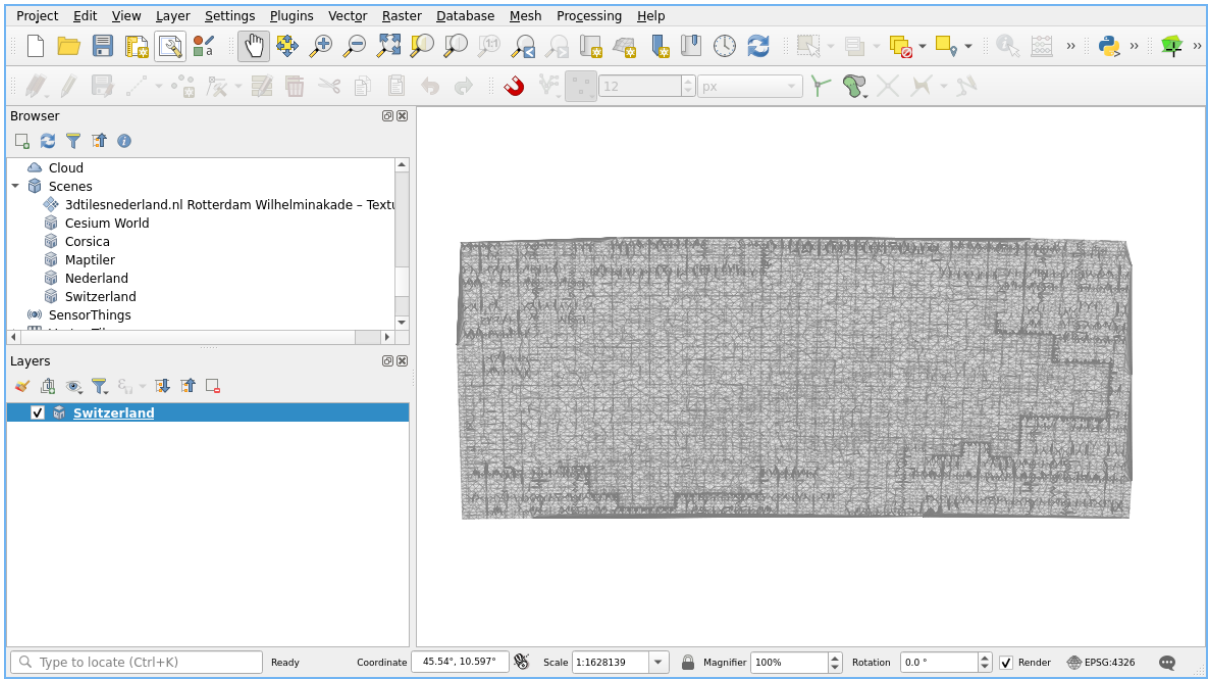


Fig. 11.22: Kvantiserat Mesh-lager

Använda molnanslutningar

QGIS stöder anslutningar till molntjänster som Alibaba Cloud OSS, Amazon S3, Google Cloud Storage, Microsoft Azure Blob Storage, Microsoft Azure Data Lake Storage och OpenStack Swift Object Storage. Du kan ladda vektor- och rasterdata från dessa tjänster till QGIS. Ställ in en ny *Cloud*-anslutning i *Browser*-panelen genom att högerklicka på *Cloud*-posten och välja *New Connection*. Du kommer att se en rullgardinslista med tillgängliga molntjänster. Välj den tjänst som du vill ansluta till och fyll i de obligatoriska fälten:

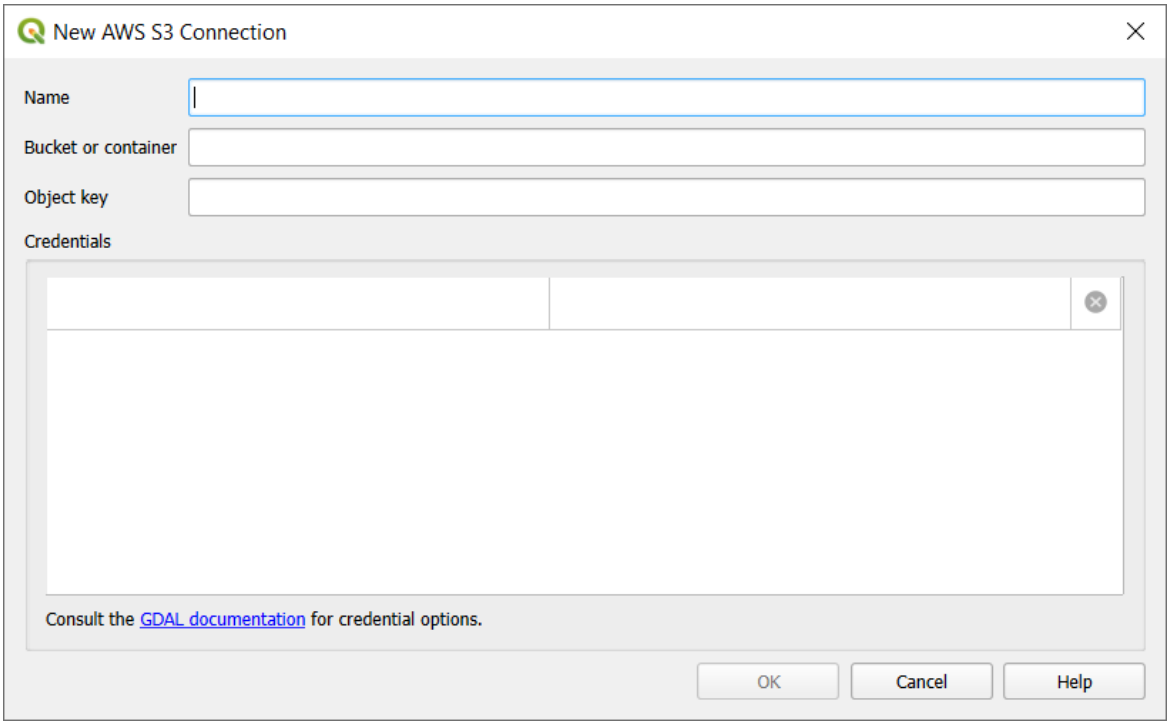


Fig. 11.23: Dialog för molnanslutning

- *Namn*: Ett namn för anslutningen.
- *Bucket eller Container*: Namnet på skopan eller behållaren i molntjänsten.
- *Objektnyckel* (valfritt): Nyckeln till objektet i skopan eller behållaren.
- *Credentials*: Autentiseringsuppgifter för åtkomst till molntjänsten.

Du kan också välja att *Save Connection* till en XML-fil eller *Load Connection* från en XML-fil.

11.2 Skapa lager

Lager kan skapas på många olika sätt, bland annat:

- tomma lager från början
- lager från befintliga lager
- lager från urklipp
- lager som ett resultat av en SQL-liknande fråga baserad på ett eller flera lager (*virtual layers*)

QGIS innehåller också verktyg för att importera/exportera från/till olika format.

11.2.1 Skapa nya vektorlager

I QGIS kan du skapa nya lager i olika format. Det finns verktyg för att skapa GeoPackage-, Shapefile-, SpatiaLite-, GPX-format och Temporary Scratch-lager (även kallade minneslager). Skapande av ett *nytt GRASS-lager* stöds inom GRASS-tillägget.

Skapa ett nytt GeoPackage-lager

Du skapar ett nytt GeoPackage-lager genom att trycka på knappen *New GeoPackage Layer…* i menyn *Layer* ► *Create Layer* ► eller i verktygsfältet *Data Source Manager*. Du kan också skapa ett nytt GeoPackage-lager via *Browser Panel* genom att välja *Create Database and Layer…*. Dialogrutan *New GeoPackage Layer* visas enligt bilden i Fig. 11.24.

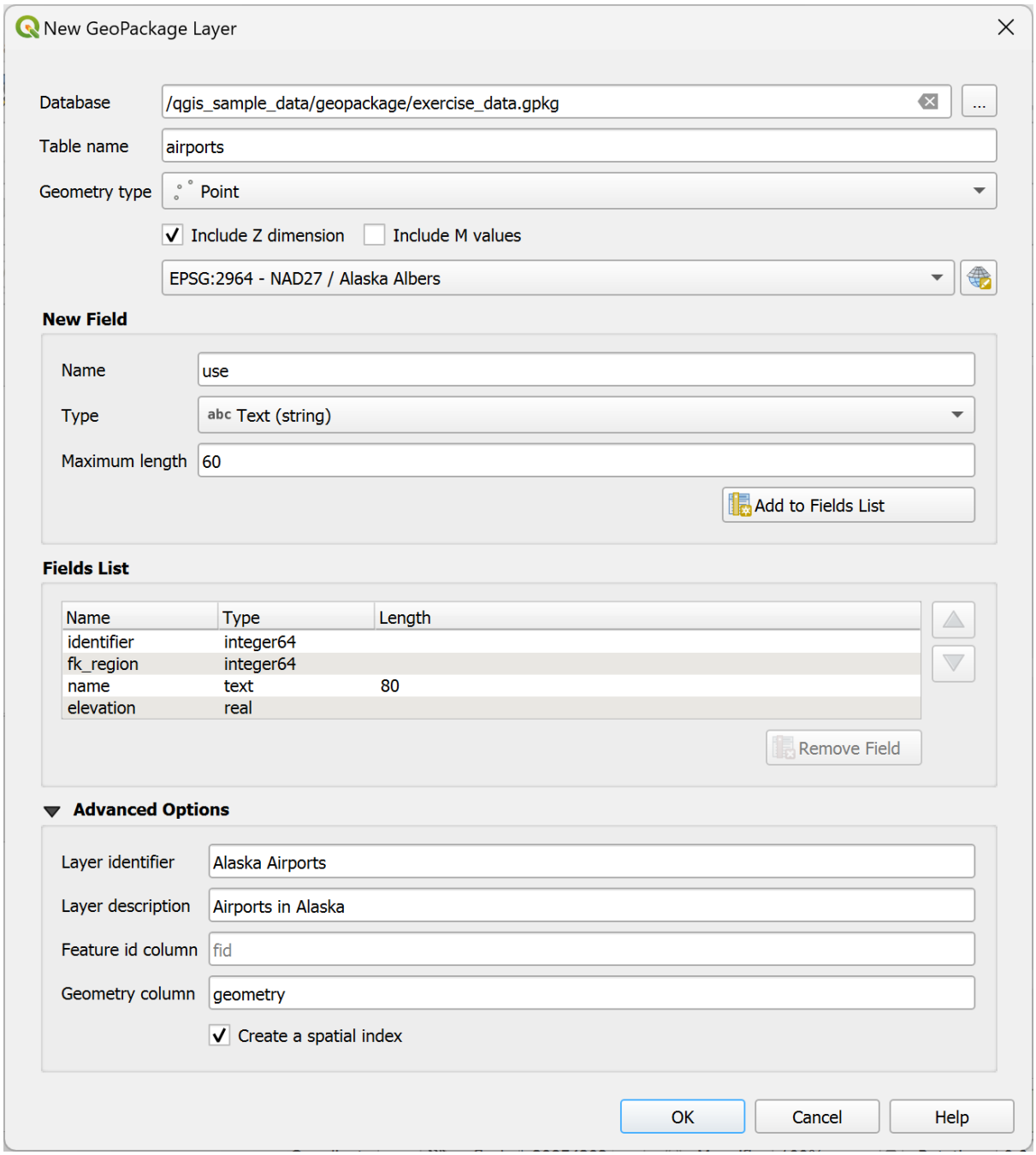


Fig. 11.24: Skapa en ny dialogruta för GeoPackage-lager

1. Det första steget är att ange databasfilens plats. Detta kan göras genom att trycka på knappen … till höger om fältet *Database* och välja en befintlig GeoPackage-fil eller skapa en ny. QGIS kommer automatiskt att lägga till rätt tillägg till det namn du anger.
2. Ge det nya lagret / tabellen ett namn (*Tabellnamn*)
3. Definiera *Geometri typ*. Om det inte är ett geometrilöst lager kan du ange om det ska *Inkludera Z-dimension* och/eller *Inkludera M-värden*.
4. Ange referenssystemet för koordinaterna med hjälp av knappen

För att lägga till fält i det lager du skapar:

1. Ange fältets *Namn*
2. Välj data *Type*. Följande typer stöds: *Textdata*, *Hela tal* (både heltal och heltal64), *Decimaltal*, *Datum* och *Datum och tid*, *Binär (BLOB)* och *Boolean*.
3. Ange värdenas *Maximala längd* beroende på det valda dataformatet.
4. Klicka på knappen *Add to Fields List* (Lägg till i listan över fält)
5. Upprepa stegen ovan för varje fält som du behöver lägga till
6. Du kan senare ändra fältordningen med hjälp av knapparna Flytta upp och Flytta ner
7. När du är nöjd med attributen klickar du på *OK*. QGIS lägger till det nya lagret i teckenförklaringen och du kan redigera det enligt beskrivningen i avsnitt *Digitalisering av ett befintligt lager*.

När ett GeoPackage-lager skapas genererar QGIS som standard en *Feature id column* som kallas `fid` och som fungerar som primärnyckel för lagret. Namnet kan ändras. Geometrifältet, om det finns, får namnet `geometry` och du kan välja att *Create a spatial index* på det. Dessa alternativ finns under *Advanced Options* tillsammans med *Layer identifier* (kort mänskligt läsbart namn på lagret) och *Layer description*.

Ytterligare hantering av GeoPackage-lager kan göras med *DB Manager*.

Skapa ett nytt Shapefile-lager

Om du vill skapa ett nytt lager i ESRI Shapefile-format trycker du på knappen *Nytt Shapefil-lager…* i menyn *Lager* ► *Skapa lager* ► eller i verktygsfältet *Datakällshanterare*. Dialogrutan *Nytt Shapefil-lager* visas som visas i Fig. 11.25.

1. Ange en sökväg och ett filnamn med hjälp av knappen … bredvid *Filnamn*. QGIS kommer automatiskt att lägga till rätt tillägg till det namn du anger.
2. Ange sedan *Filkodning* för data
3. Välj *Geometrityp* för lagret: Ingen geometri (resulterar i en fil i formatet `.DBF`), punkt, multipunkt, linje eller polygon
4. Ange om geometrin ska ha ytterligare dimensioner: *None*, *Z (+ M värden)* eller *M värden*
5. Ange referenssystemet för koordinaterna med hjälp av knappen

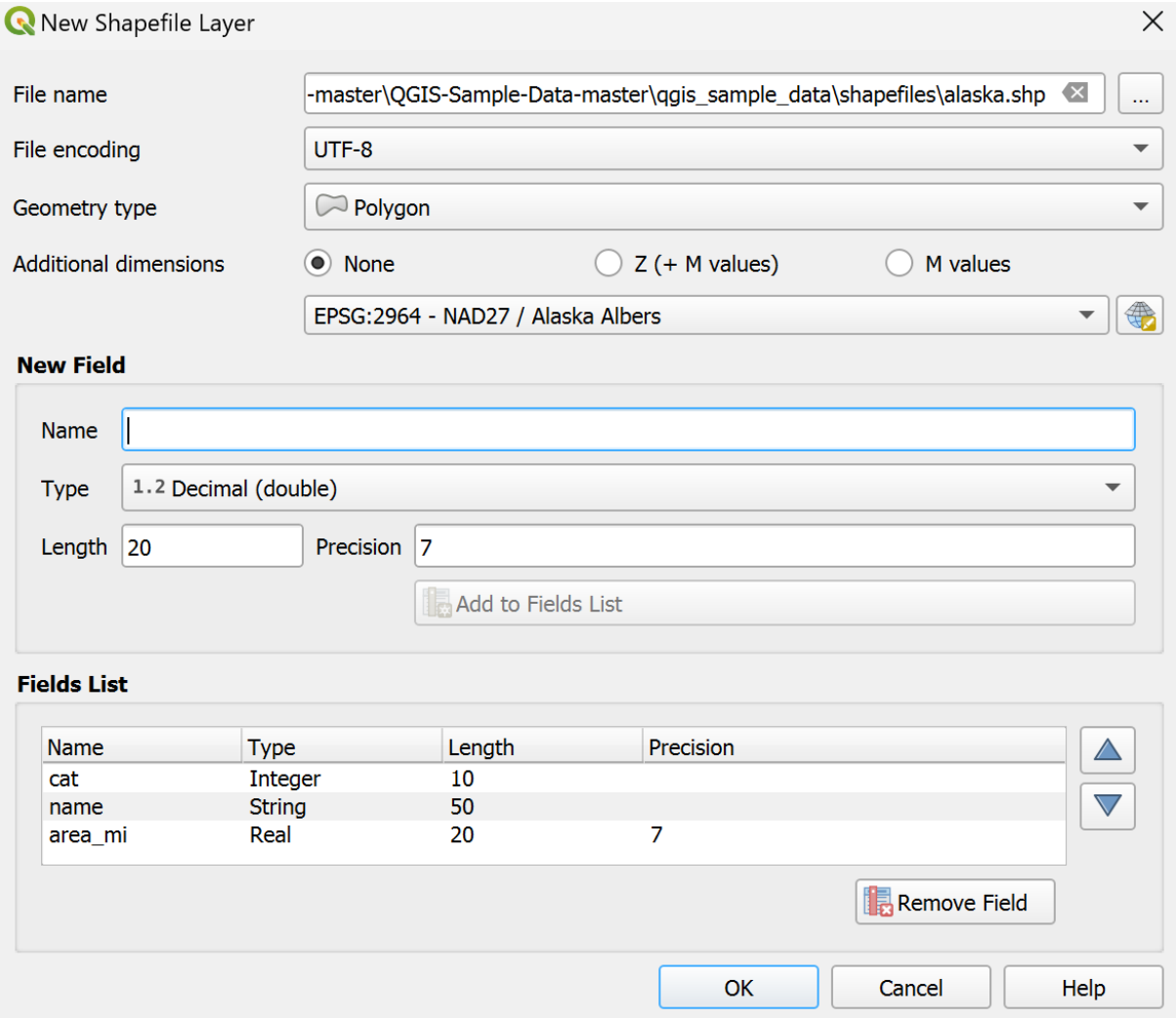


Fig. 11.25: Skapa en ny Shapefile-lagerdialog

För att lägga till fält i det lager du skapar:

1. Ange fältets *Namn*
2. Välj data *Typ*. Endast attributen *Decimal number*, *Whole number*, *Text data*, *Date* och *Boolean* stöds.
3. Beroende på valt dataformat anger du *Längd* och *Precision*.
4. Klicka på knappen *Add to Fields List* (Lägg till i listan över fält)
5. Upprepa stegen ovan för varje fält som du behöver lägga till
6. Du kan senare ändra fältordningen med hjälp av knapparna Flytta upp och Flytta ner
7. När du är nöjd med attributen klickar du på *OK*. QGIS lägger till det nya lagret i teckenförklaringen och du kan redigera det enligt beskrivningen i avsnitt *Digitalisering av ett befintligt lager*.

Som standard läggs en kolumn med första heltalet `id` till, men den kan tas bort.

Skapa ett nytt SpatiaLite-lager

För att skapa ett nytt SpatiaLite-lager trycker du på knappen *Nytt SpatiaLite-lager…* i menyn *Lager* ► *Skapa lager* ► eller i verktygsfältet *Datakällshanterare*. Dialogrutan *Nytt SpatiaLite-lager* kommer att visas som visas i Fig. 11.26.

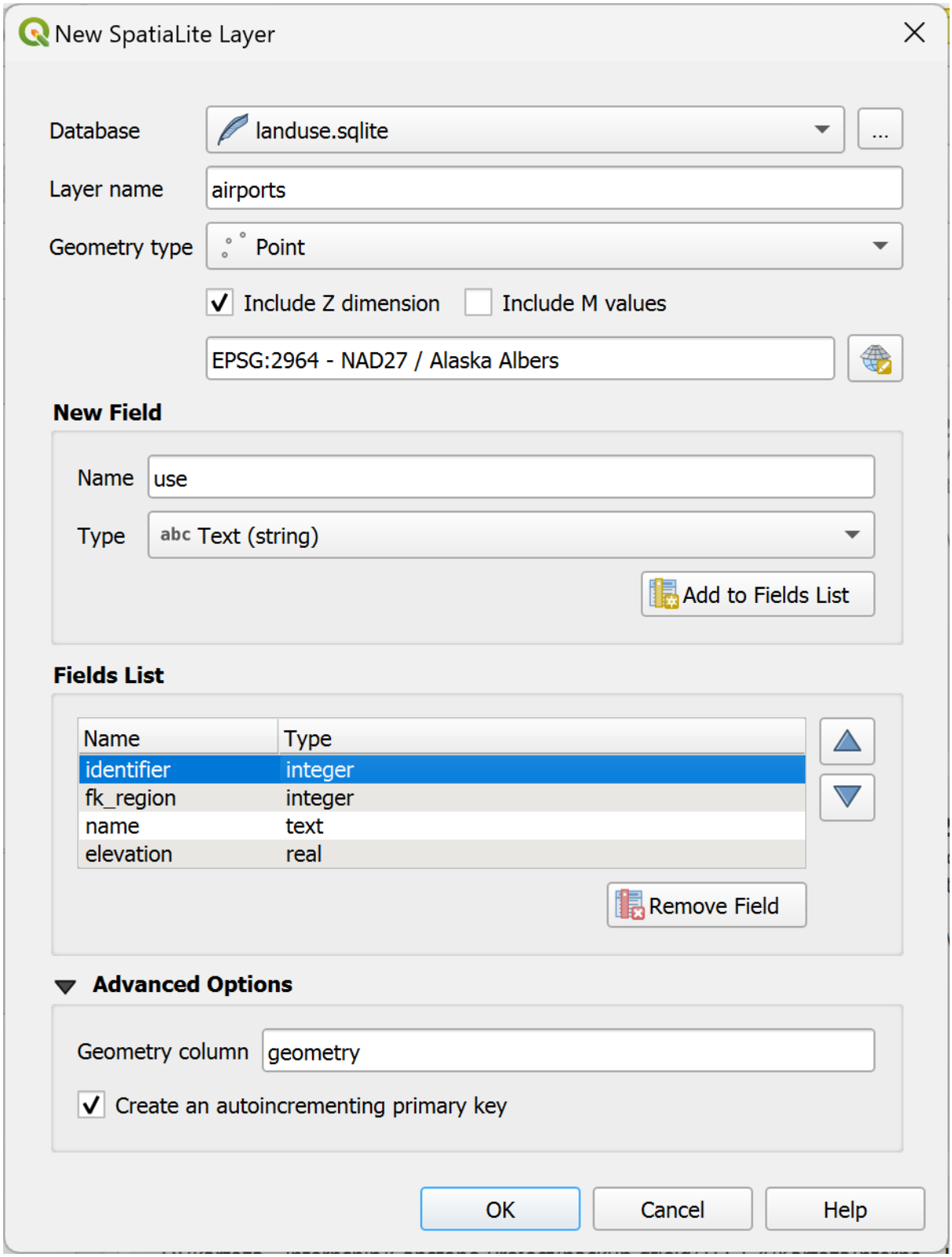


Fig. 11.26: Skapa en dialogruta för ett nytt SpatiaLite-lager

1. Det första steget är att ange databasfilens plats. Detta kan göras genom att trycka på …-knappen till höger om *Database*-fältet och välja en befintlig SpatiaLite-fil eller skapa en ny. QGIS kommer automatiskt att lägga till rätt tillägg till det namn du anger.
2. Ange ett namn (*Lagernamn*) för det nya lagret
3. Definiera *Geometri typ*. Om det inte är ett geometrilöst lager kan du ange om det ska *Inkludera Z-dimension* och/eller *Inkludera M-värden*.
4. Ange referenssystemet för koordinaterna med hjälp av knappen .

För att lägga till fält i det lager du skapar:

1. Ange fältets *Namn*
2. Välj datatyp *Type*. Följande typer stöds: *Textdata*, *Hela tal*, *Decimaltal*, *Datum* och *Datumtid*.
3. Klicka på knappen *Add to Fields List* (Lägg till i listan över fält)
4. Upprepa stegen ovan för varje fält som du behöver lägga till
5. Du kan senare ändra fältordningen med hjälp av knapparna Flytta upp och Flytta ner
6. När du är nöjd med attributen klickar du på *OK*. QGIS lägger till det nya lagret i teckenförklaringen och du kan redigera det enligt beskrivningen i avsnitt *Digitalisering av ett befintligt lager*.

Om du vill kan du markera *Create an autoincrementing primary key* under avsnittet *Advanced Options*. Du kan också byta namn på kolumnen *Geometry* (`geometry` som standard).

Ytterligare hantering av SpatiaLite-lager kan göras med *DB Manager*.

Skapa ett nytt Mesh-lager

För att skapa ett nytt Mesh-lager, tryck på knappen *Nytt nätlager…* i menyn *Lager* ► *Skapa lager* ► eller från verktygsfältet *Data Source Manager*. Dialogrutan *Nytt nätlager* kommer att visas som visas i Fig. 11.27.

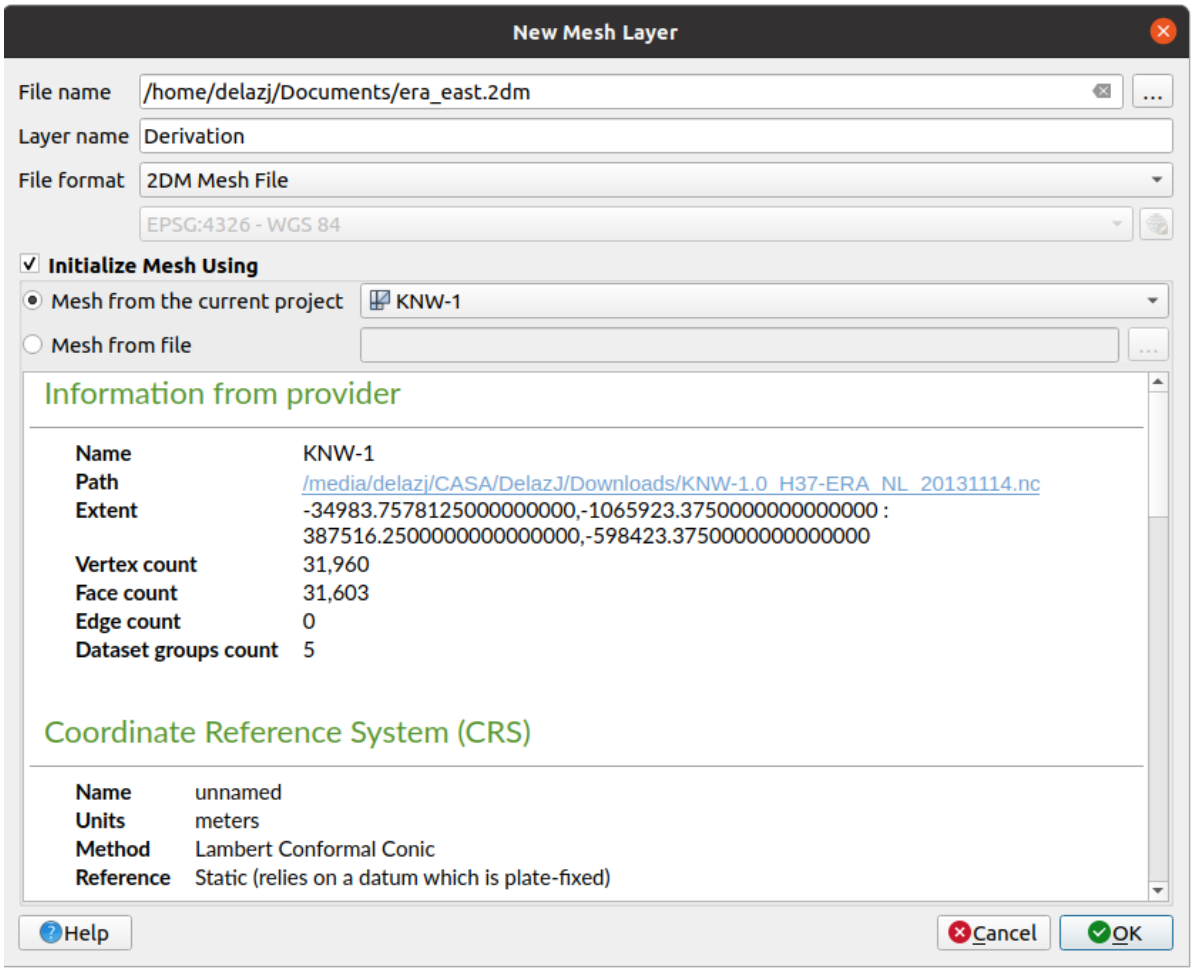


Fig. 11.27: Skapa en dialogruta för ett nytt mesh-lager

1. Det första steget är att ange platsen för mesh-filen. Detta kan göras genom att trycka på knappen … till höger om fältet *Filnamn* och välja en befintlig mesh-fil eller skapa en ny.
2. Ange ett namn (*Lagernamn*), dvs. det namn som lagret visas med i panelen *Lager*)
3. Välj *File format*: för närvarande stöds följande mesh-filformat: `2DM Mesh File (*.2dm)`, `Selafin File (*.slf)` och `UGRID (*.nc)`.
4. Ange *Koordinatreferenssystem* som ska tilldelas datasetet
5. Ovanstående steg genererar ett tomt lager som du sedan kan digitalisera vertikaler och lägga till datasetgrupper i. Det är dock också möjligt att initiera lagret med ett befintligt mesh-lager, dvs. fylla det nya lagret med vertikaler eller ytor från det andra lagret. För att göra detta:

    1. Check *Initialisera nät med hjälp av*
    2. och välj antingen en *Mesh från det aktuella projektet* eller *Mesh från en fil*. Information om den valda mesh-filen visas för kontroll.

    Observera att endast ramen för mesh-lagret överförs till det nya lagret; deras dataset kopieras inte.

Skapa ett nytt GPX-lager

Så här skapar du en ny GPX-fil:

1. Välj *Skapa lager* ► *Nytt GPX-lager…* från menyn *Lager*.
2. I dialogrutan väljer du var du vill spara den nya filen, namnger den och trycker på *Save*.
3. Tre nya lager läggs till i *Lagerpanel*:
   - ett punktlager för att digitalisera platser (`waypoints`) med fält som lagrar namn, höjd, kommentar, beskrivning, källa, webbadress och webbadressnamn
   - ett linjelager för att digitalisera sekvenser av platser som utgör en planerad rutt (`routes`) med fält som lagrar namn, symbol, nummer, kommentar, beskrivning, källa, url, url-namn
   - och ett linjelager för att spåra mottagarens rörelse över tiden (`spår`) med fält som lagrar namn, symbol, nummer, kommentar, beskrivning, källa, webbadress, webbadressnamn.
4. Du kan nu redigera vilket som helst av dem enligt beskrivningen i avsnitt *Digitalisering av ett befintligt lager*.

Skapa ett nytt temporärt skrapskikt

Temporary Scratch Layers är minneslager, vilket innebär att de inte sparas på disk och kommer att kasseras när QGIS stängs. De kan vara praktiska för att lagra funktioner som du behöver tillfälligt eller som mellanlager under geobearbetning.

För att skapa ett nytt Temporary Scratch-lager, välj posten *New Temporary Scratch Layer…* i menyn *Layer* ► *Create Layer* ► eller i verktygsfältet *Data Source Manager*. Dialogrutan *New Temporary Scratch Layer* kommer att visas som visas i Fig. 11.28. Och så vidare:

1. Ange *Lagernamn*
2. Välj *Geometri typ*. Här kan du skapa en:
   - lager av typen `No geometry`, används som enkel tabell,
   - `Point` eller `MultiPoint` lager,
   - `LineString/CompoundCurve` eller `MultiLineString/MultiCurve` lager,
   - `Polygon/CurvePolygon` eller `MultiPolygon/MultiSurface` lager.
3. För geometriska typer anger du datauppsättningens dimensioner: kontrollera om den ska *Include Z dimension* och/eller *Include M values*
4. Ange referenssystemet för koordinaterna med hjälp av knappen .
5. Lägg till fält i lagret. Observera att till skillnad från många andra format kan tillfälliga lager skapas utan några fält. Detta steg är alltså valfritt.
   1. Ange fältets *Namn*
   2. Välj data *Type*: *Text*, *Whole number*, *Decimal number*, *Boolean*, *Date*, *Time*, *Date & Time* och *Binary (BLOB)* stöds.
   3. Beroende på det valda dataformatet anger du *Längd* och *Precision*
   4. Klicka på knappen *Add to Fields List* (Lägg till i listan över fält)
   5. Upprepa stegen ovan för varje fält som du behöver lägga till
   6. Du kan senare ändra fältordningen med hjälp av knapparna Flytta upp och Flytta ner
6. När du är nöjd med inställningarna klickar du på *OK*. QGIS lägger till det nya lagret i panelen *Lager*, och du kan redigera det enligt beskrivningen i avsnitt *Digitalisering av ett befintligt lager*.

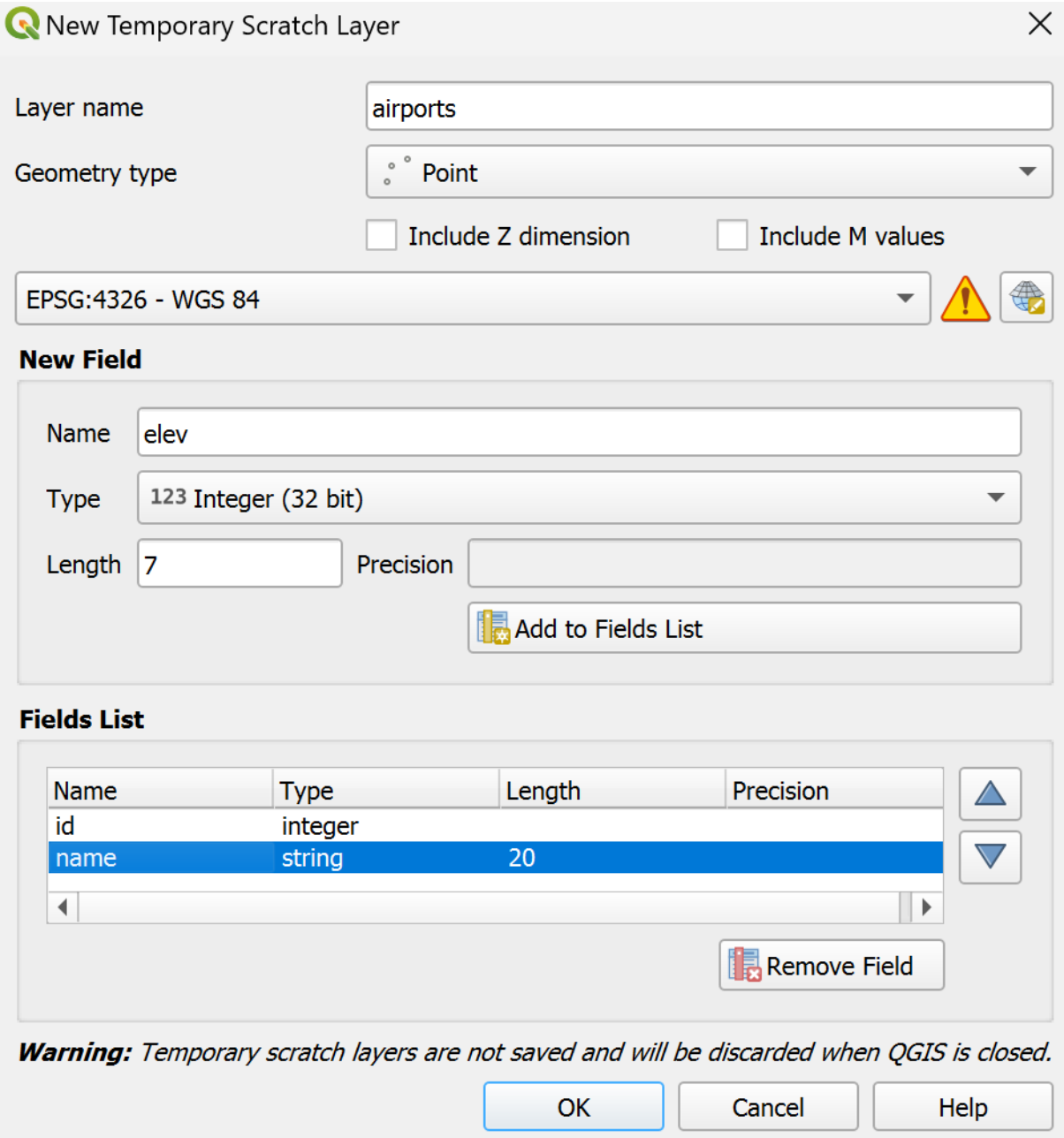


Fig. 11.28: Skapa en ny dialogruta för lagret Temporary Scratch

Du kan också skapa förfyllda tillfälliga scratch-lager med hjälp av t.ex. urklipp (se *Skapa nya lager från urklipp*) eller som ett resultat av en *Processing algorithm*.

Tips

Permanent lagring av ett minneslager på disk

För att undvika dataförlust när du avslutar ett projekt med tillfälliga scratch-lager kan du spara dessa lager i valfritt vektorformat som stöds av QGIS:

- genom att klicka på indikatorikonen bredvid lagret;
- genom att välja *Make permanent* i lagrets kontextuella meny;
- med hjälp av *Exportera* ► i snabbmenyn eller *Lager* ► *Spara som…* i menyn.

Alla dessa kommandon öppnar dialogrutan *Spara vektorlager som* som beskrivs i avsnittet *Skapa nya lager från ett befintligt lager* och den sparade filen ersätter den tillfälliga filen i panelen *Lager*.

11.2.2 Skapa nya lager från ett befintligt lager

Lager (raster, vektor och punktmoln) kan sparas i ett annat format och/eller omprojiceras till ett annat koordinatreferenssystem (CRS) med hjälp av menyn *Layer* ► *Save As…* eller genom att högerklicka på lagret i *Layers panel* och välja:

- *Export* ► *Spara som…* för raster- och punktmolnslager
- *Exportera* ► *Spara funktioner som…* eller *Exportera* ► *Save valda funktioner som…* för vektorlager.
- Dra och släpp lagret från lagerträdet till PostGIS-posten i *Browser Panel*. Observera att du måste ha en PostGIS-anslutning i *Browser Panel*.

Vanliga parametrar

Dialogrutan *Save Layer as…* visar flera parametrar för att ändra beteendet när lagret sparas. Några av de vanligaste parametrarna för raster och vektor är:

- *Filnamn*: Filens plats på hårddisken. Den kan hänvisa till utdatalagret eller till en behållare som lagrar lagret (t.ex. databasliknande format som GeoPackage, SpatiaLite eller Open Document Spreadsheets).
- *CRS*: kan ändras för att återföra data
- *Extent*: begränsar omfattningen av den indata som ska exporteras med hjälp av widgeten *extent_selector*
- *Lägg till sparad fil på kartan*: för att lägga till det nya lagret på duken

Vissa parametrar är dock specifika för vissa format:

Raster-specifika parametrar

Beroende på exportformat är det möjligt att vissa av dessa alternativ inte är tillgängliga:

- *Utmatningsläge* (det kan vara **raw data** eller **renderad bild**)
- *Format*: exporterar till alla rasterformat som GDAL kan skriva till, t.ex. GeoTiff, GeoPackage, MBTiles, Geospatial PDF, SAGA GIS Binary Grid, Intergraph Raster, ESRI .hdr Labelled…
- *Resolution*
- *Create Options*: använd avancerade alternativ (filkomprimering, blockstorlekar, kolorimetri …) när du genererar filer, antingen från *fördefinierade skapande profiler* relaterade till utdataformatet eller genom att ställa in varje parameter.
- *Pyramider* skapelse
- *VRT Tiles* ifall du valde att *Create VRT*
- *Inga datavärden*

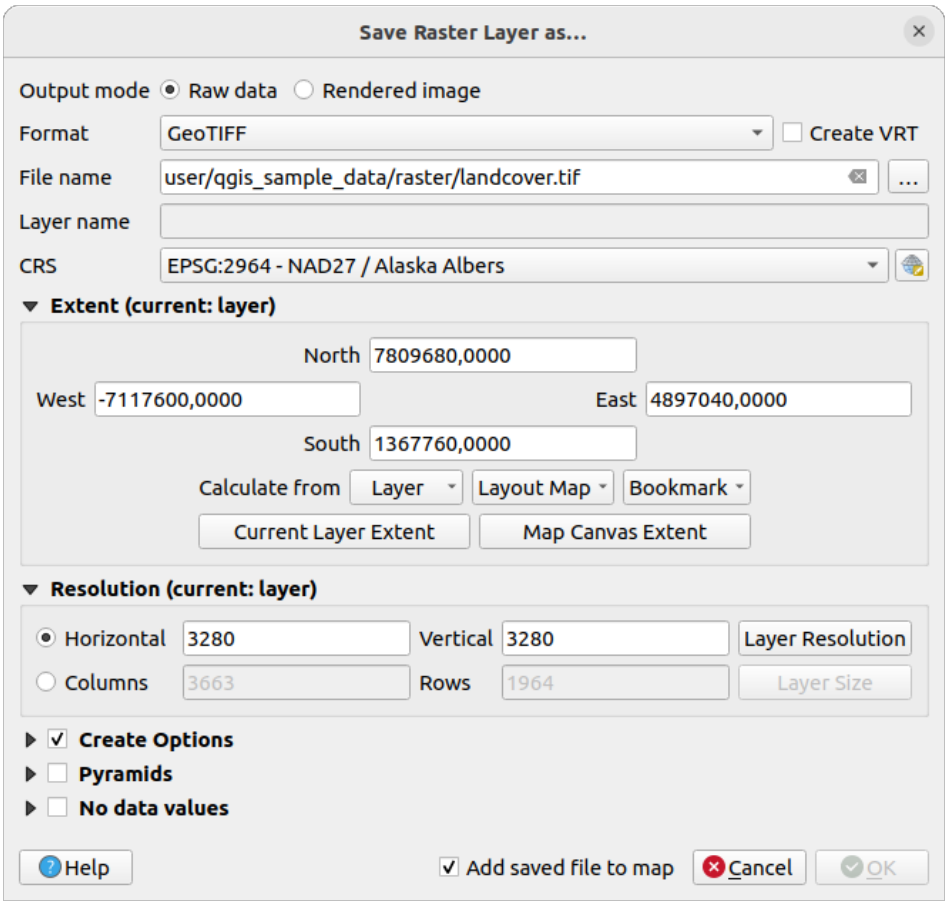


Fig. 11.29: Spara som ett nytt rasterlager

Vektorspecifika parametrar

Beroende på exportformat kan några av dessa alternativ vara tillgängliga:

- *Format*: exporterar till alla vektorformat som GDAL kan skriva till, t.ex. GeoPackage, GML, ESRI Shapefile, AutoCAD DXF, ESRI FileGDB, Mapinfo TAB eller MIF, SpatiaLite, CSV, KML, ODS, …
- *Lagernamn*: tillgänglig när *Filnamn* hänvisar till ett containerliknande format, denna post representerar utmatningslagret.
- *Kodning*
- *Spara endast utvalda funktioner*
- *Välj fält som ska exporteras och deras exportalternativ*: ger möjlighet att exportera fält med anpassade namn och *form widget*-inställningar:
  - Markera raderna under kolumnen *Name* för att välja fält som ska behållas i utdatalagret, eller tryck på knapparna *Markera allt* eller *Avmarkera allt*
  - Aktivera kryssrutan *Use aliases for exported name* för att fylla i kolumnen *Export name* med motsvarande fältalias eller återställa till det ursprungliga fältnamnet. Om du dubbelklickar på en cell redigeras även namnet.
  - Beroende på om widgetar för attributformulär används kan du *ersätta alla valda råa fältvärden med visade värden*. Om t.ex. en widget för ”värdekarta” används för ett fält kommer utdatalagret att innehålla beskrivningsvärdena i stället för de ursprungliga värdena. Ersättningen kan också göras fält för fält, i kolumnen *Replace with displayed values* (Ersätt med visade värden).
- *Persist layer metadata*: säkerställer att alla lager *metadata* som finns i källlagret kopieras och lagras:

  - i det nyskapade lagret, om utdata är i GeoPackage-format
  - som en `.qmd`-fil tillsammans med utdatalagret, för andra format. Observera att filbaserade format som stöder mer än en dataset (t.ex. SpatiaLite, DXF, …) kan ha ett oavsiktligt beteende.

- *Symbology export*: kan användas främst för DXF-export och för alla filformat som hanterar OGR-funktionsstilar (se anmärkning nedan) som DXF-, KML-, tab-filformat:
  - **Ingen symbologi**: standardstil för den applikation som läser data
  - **Feature symbology**: spara stil med OGR Feature Styles (se anmärkning nedan)
  - **Symbollagersymbologi**: spara med OGR Feature Styles (se anmärkning nedan) men exportera samma geometri flera gånger om det finns flera symbolskiktssymboler som används
  - Ett **Scale**-värde kan tillämpas på de senaste alternativen

Observera

OGR Feature Styles är ett sätt att lagra stil direkt i data som ett dolt attribut. Endast vissa format kan hantera den här typen av information. KML-, DXF- och TAB-filformat är sådana format. För avancerade detaljer kan du läsa dokumentet OGR Feature Styles specification.

- *Geometry*: du kan konfigurera geometrifunktionerna för utdatalagret
  - *geometry type*: behåller den ursprungliga geometrin för objekten när den är inställd på **Automatic**, annars tas den bort eller ersätts med valfri typ. Du kan lägga till en tom geometrikolumn i en attributtabell och ta bort geometrikolumnen i ett spatialt lager.
  - *Force multi-type*: tvingar fram skapandet av multigeometriska funktioner i lagret.
  - *Inkludera z-dimension* till geometrier.

Tips

Genom att åsidosätta lagrets geometrityp kan du t.ex. spara en geometrilös tabell (t.ex. `.csv`-fil) i en shapefil MED valfri geometrityp (punkt, linje, polygon), så att geometrier sedan kan läggas till manuellt i raderna med verktyget Add Part.

- *Datasource Options*, *Layer Options* eller *Custom Options* som gör att du kan konfigurera avancerade parametrar beroende på utdataformat. Vissa beskrivs i *Utforska dataformat och fält* men för fullständiga detaljer, se dokumentationen för drivrutinen GDAL. Varje filformat har sina egna anpassade parametrar, t.ex. för formatet `GeoJSON`, se dokumentationen för GDAL GeoJSON.

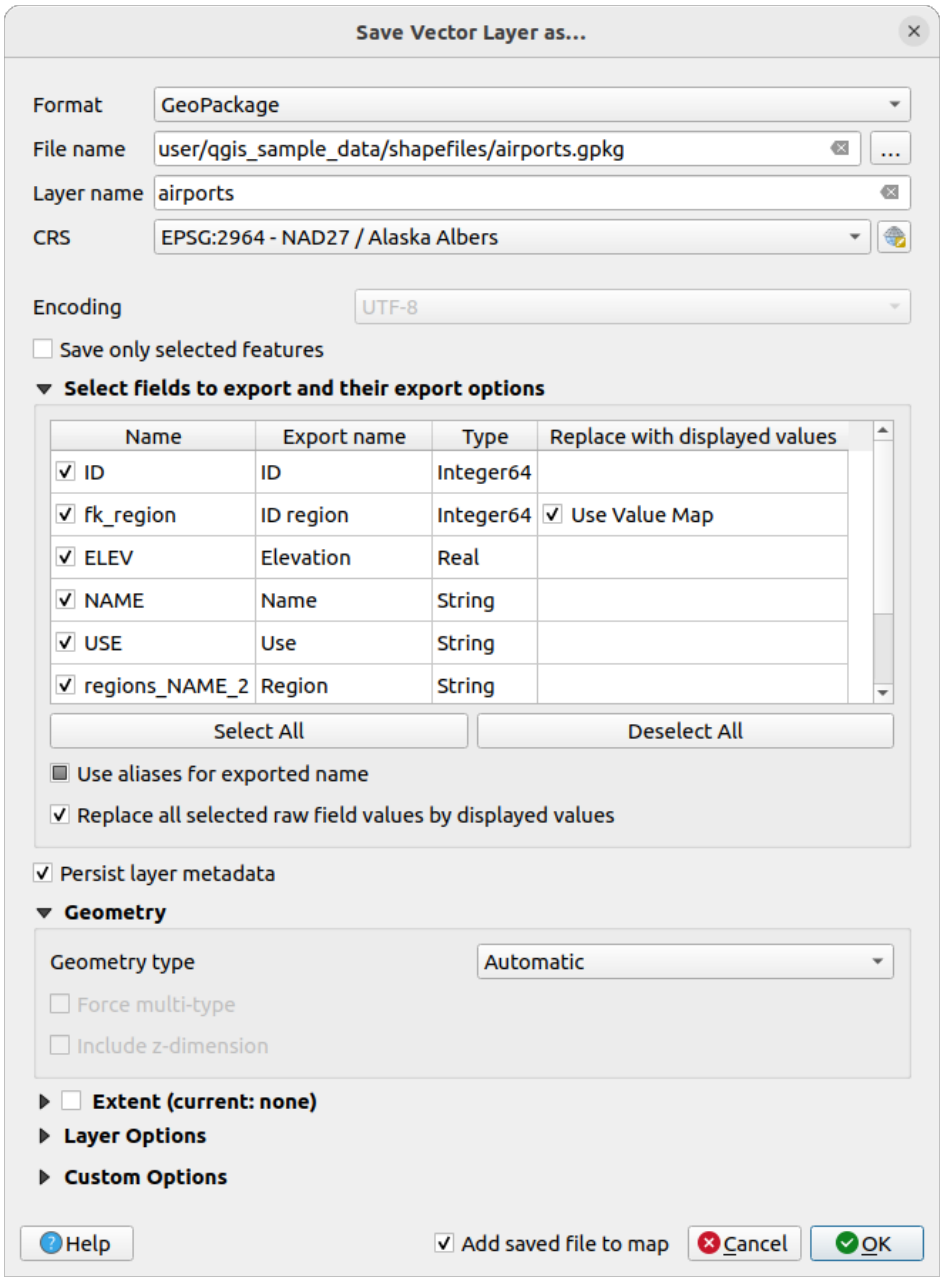


Fig. 11.30: Spara som ett nytt vektorlager

När du sparar ett vektorlager i en befintlig fil kan användaren, beroende på kapaciteten i utdataformatet (Geopackage, SpatiaLite, FileGDB …), bestämma om den ska:

- skriva över hela filen
- skriva över endast mållagret (lagernamnet är konfigurerbart)
- lägga till funktioner till det befintliga mållagret
- lägg till funktioner, lägg till nya fält om det finns några.

För format som ESRI Shapefile, MapInfo .tab är det också möjligt att lägga till funktioner.

Specifika parametrar för punktmoln

På samma sätt som raster- och vektorlager kan punktmolnslager sparas i ett annat format och/eller omprojiceras till ett annat koordinatreferenssystem (CRS). Detta gör att du kan exportera ett punktmolnslager till vektor- eller punktmolnsformat. Format som stöds för närvarande är: Temporary scratch (minneslager), GeoPackage, ESRI Shapefile, DXF och LAS/LAZ punktmoln. Förutom de vanliga parametrarna som anges ovan innehåller export av punktmolnslager följande alternativ:

- *Filtrera efter polygonlager*: Gör att du kan filtrera punktmolnsdata baserat på ett polygonlager.
- *Elevation Range*: Möjliggör filtrering av punktmolnsdata baserat på ett specificerat Z-intervall.
- *Begränsa antalet punkter*: Ger ett alternativ för att begränsa antalet punkter som exporteras från punktmolnslagret.

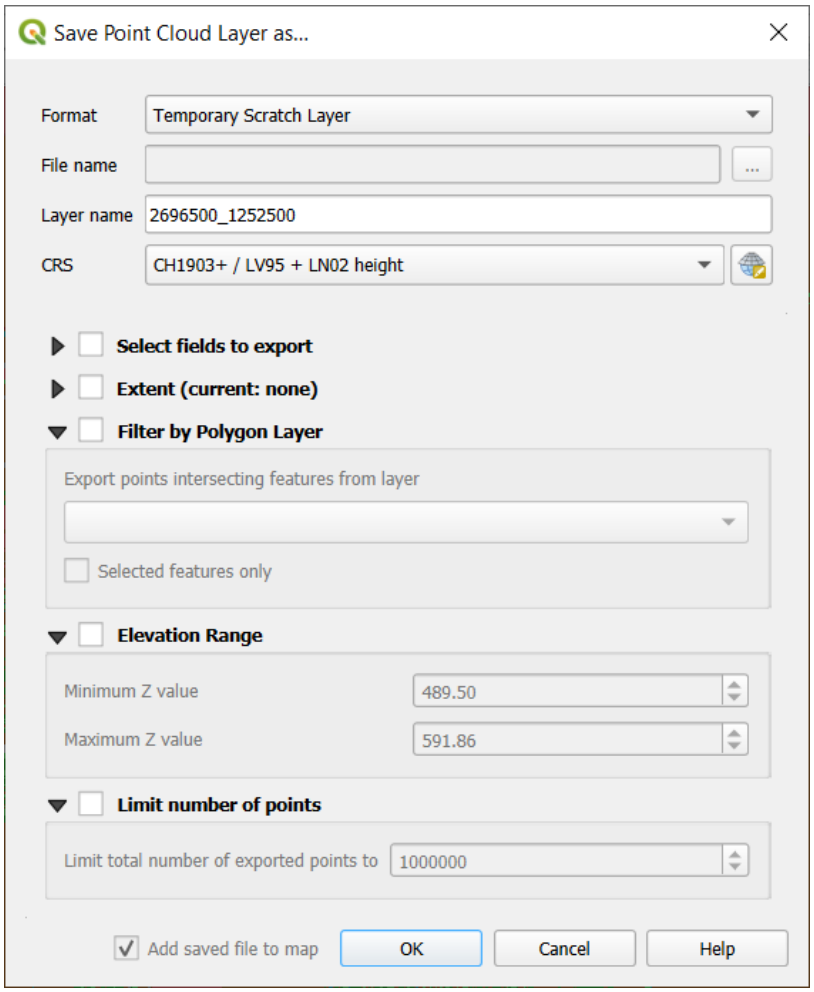


Fig. 11.31: Spara ett punktmolnslager som ett nytt lager

11.2.3 Skapa nya DXF-filer

Förutom dialogen *Save As…* som innehåller alternativ för att exportera ett enskilt lager till ett annat format, inklusive `*.DXF`, innehåller QGIS ett annat verktyg för att exportera flera lager som ett enda DXF-lager. Det finns tillgängligt i menyn *Project* ► *Import/Export* ► *Export Project to DXF…*.

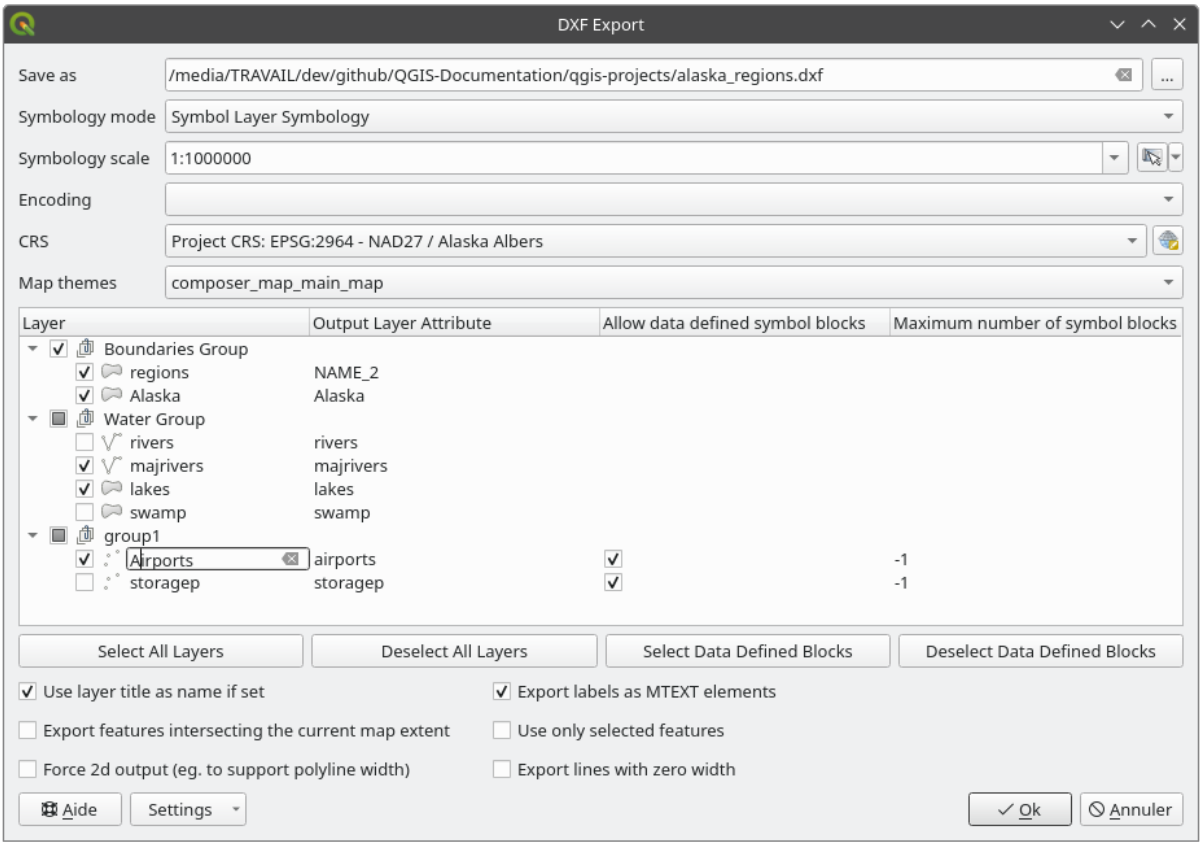


Fig. 11.32: Exportera ett projekt till DXF-dialogrutan

I dialogrutan *DXF Export*:

1. Ange destinationsfilen.
2. Välj symbologiläge och skala (se anmärkningen *OGR Feature Styles*), om tillämpligt.
3. Välj data *Encoding*.
4. Välj den *CRS* som ska tillämpas: de valda lagren kommer att reprojiceras till den angivna CRS.
5. Välj de lager som ska ingå i DXF-filerna antingen genom att markera dem i tabellwidgeten eller genom att automatiskt välja dem från ett befintligt *map theme*. Knapparna *Markera allt* och *Avmarkera allt* kan hjälpa till att snabbt ställa in vilka data som ska exporteras.

   För varje lager kan du:

   - Åsidosätt namnet på utdatalagret utan att ändra det ursprungliga projektlagret. Klicka på *Layer*-namnet i dialogrutan och skriv det utdatanamn som ska användas.
   - *Attribut för utmatningslager*: Välj om du vill exportera alla funktioner i ett enda DXF-lager eller förlita dig på ett fält vars värden används för att dela upp funktionerna i lager i DXF-utdata. I det senare fallet kommer varje lager att få sitt namn från motsvarande fältvärde.
   - *Tillåt datadefinierade symbolblock*:
   - *Maximalt antal symbolblock*: skapar symbolblock upp till den angivna gränsen och börjar med de block som innehåller flest referenser. De andra symbolerna skrivs som de är. `-1` betyder ingen begränsning.

Alternativt kan du också välja att:

- ☐ *Använd lagrets titel som namn om det är inställt* istället för själva lagernamnet: titeln hämtas från lagrets egenskaper *metadata* eller *server*;
- ☐ *Exportera funktioner som skär den aktuella kartutbredningen*;
- ☐ *Tvinga fram 2d-utdata (t.ex. för att stödja polylinjebredd)*;
- ☑ *Exportera etikett som MTEXT-element* eller TEXT-element;
- ☐ *Använd endast utvalda funktioner*;
- ☐ *Exportera linjer med noll bredd*: alla linjer exporteras med minimal bredd `0` (hårlinje) om den är aktiverad. Detta hjälper till att hålla linjerna minimala i filen oavsett zoomnivå, och kan vara praktiskt för att göra ytterligare CAD-redigering med den exporterade dxf, särskilt om det finns många funktioner bredvid varandra på kartan.

Observera

Prioriteringen för att definiera namnet på utdatalagret är följande:

1. Fältvärdet från *Output layer attribute*
2. Det åsidosatta namnet i *Layer*-kolumnen
3. Alternativet *Använd lagrets titel som namn om det är inställt*
4. Lagrets namn

Aktuella inställningar som definieras i dialogrutan *DXF Export* kan lagras i en XML-fil så att de kan återanvändas i andra sessioner. För detta har kombinationsrutan *Settings* två alternativ: *Load Settings from File…* och *Save Settings to File…*.

11.2.4 Skapa nya lager från urklipp

Funktioner som finns i urklippet kan klistras in i ett nytt lager. Detta gör du genom att välja några funktioner, kopiera dem till urklippet och sedan klistra in dem i ett nytt lager med *Edit* ► *Paste Features as* ► och välja:

- *Nytt vektorlager…*: dialogrutan *Spara vektorlager som…* visas (se *Skapa nya lager från ett befintligt lager* för parametrar)
- eller *Temporary Scratch Layer…*: du måste ange ett namn för lagret

Ett nytt lager, fyllt med valda funktioner och deras attribut, skapas (och läggs till i kartbilden).

Observera

Det är möjligt att skapa lager från urklipp med objekt som markerats och kopierats i QGIS samt objekt från ett annat program, så länge deras geometrier definieras med hjälp av välkänd text (WKT).

11.2.5 Skapa virtuella lager

Ett virtuellt lager är en speciell typ av vektorlager. Det gör att du kan definiera ett lager som resultatet av en SQL-fråga som involverar ett valfritt antal andra vektorlager som QGIS kan öppna. Virtuella lager innehåller inte data i sig utan kan ses som vyer.

För att skapa ett virtuellt lager öppnar du dialogrutan för skapande av virtuella lager:

- välja posten *Lägg till/redigera virtuellt lager* i menyn *Lager* ► *Lägg till lager* ►;
- aktivera fliken *Lägg till virtuellt lager* i dialogrutan *Datakällshanterare*;
- eller med hjälp av dialogrutan *DB Manager*.

I dialogrutan kan du ange en *Layer name* och en SQL *Query*. Frågan kan använda namnet (eller id) på laddade vektorlager som tabeller, samt deras fältnamn som kolumner.

Om du till exempel har ett lager som heter `airports` kan du skapa ett nytt virtuellt lager som heter `public_airports` med en SQL-fråga som:

```
SELECT *
FROM airports
WHERE USE = "Civilian/Public"
```

SQL-frågan kommer att köras, oavsett den underliggande leverantören av lagret `airports`, även om denna leverantör inte direkt stöder SQL-frågor.

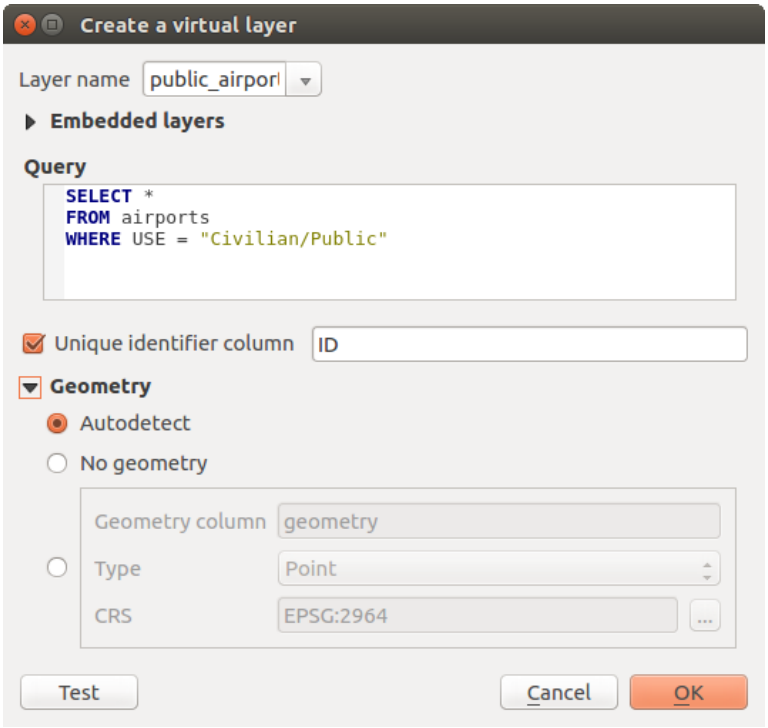


Fig. 11.33: Skapa dialog för virtuella lager

Joins och komplexa frågor kan också skapas, till exempel för att koppla samman flygplatser och landinformation:

```
SELECT airports.*, country.population
FROM airports
JOIN country
ON airports.country = country.name
```

Observera

Det är också möjligt att skapa virtuella lager med hjälp av SQL-fönstret i *DB Manager Plugin*.

Inbäddning av lager för användning i frågor

Förutom de vektorlager som finns tillgängliga i kartbilden kan användaren lägga till lager i listan *Embedded layers*, som kan användas i frågor utan att de behöver visas i kartbilden eller panelen Layers.

För att bädda in ett lager klickar du på *Lägg till* och anger *Lokalt namn*, *Datakälla*, *Kodning* och sökvägen till *Källa*.

Med knappen *Import* kan du lägga till lager i kartbilden i listan Inbäddade lager. Dessa lager kan sedan tas bort från panelen Lager utan att befintliga frågor bryts.

Frågespråk som stöds

Den underliggande motorn använder SQLite och SpatiaLite för att fungera.

Det innebär att du kan använda all SQL som din lokala installation av SQLite förstår.

Funktioner från SQLite och spatiala funktioner från SpatiaLite kan också användas i en virtuell lagerfråga. Om du t.ex. vill skapa ett punktlager av ett lager med enbart attribut kan du göra det med en fråga som liknar:

```
SELECT id, MakePoint(x, y, 4326) as geometry
FROM coordinates
```

Funktioner i QGIS-uttrycken kan också användas i en virtuell lagerfråga.

För att referera till geometrikolumnen i ett lager, använd namnet `geometry`.

I motsats till en ren SQL-fråga måste alla fält i en virtuell lagerfråga namnges. Glöm inte att använda nyckelordet `as` för att namnge dina kolumner om de är resultatet av en beräkning eller ett funktionsanrop.

Problem med prestanda

Med standardparametrar kommer den virtuella lagermotorn att göra sitt bästa för att upptäcka typen av de olika kolumnerna i frågan, inklusive typen av geometrikolumn om en sådan finns.

Detta görs genom att introspektera frågan när det är möjligt eller genom att hämta den första raden i frågan (LIMIT 1) som en sista utväg. Att hämta den första raden i resultatet bara för att skapa lagret kan vara oönskat av prestandaskäl.

Parametrar för dialogrutan för skapande:

- *Unique identifier column*: anger ett fält i frågan som representerar unika heltalsvärden som QGIS kan använda som radidentifierare. Som standard används ett automatiskt ökande heltalsvärde. Om du definierar en kolumn för unika identifierare går det snabbare att välja rader efter id.
- *Ingen geometri*: tvingar det virtuella lagret att ignorera alla geometriska fält. Det resulterande lagret är ett lager med enbart attribut.
- Geometry *Column*: anger namnet på geometrikolumnen.
- Geometry *Type*: anger geometrins typ.
- Geometry *CRS*: anger det virtuella lagrets referenssystem för koordinater.

Särskilda kommentarer

Motorn för det virtuella lagret försöker fastställa typen för varje kolumn i frågan. Om det misslyckas hämtas den första raden i frågan för att fastställa kolumntyperna.

Typen av en viss kolumn kan anges direkt i frågan med hjälp av några speciella kommentarer.

Syntaxen är följande: `/*:typ*/`. Den måste placeras precis efter namnet på en kolumn. `type` kan vara antingen `int` för heltal, `real` för flyttal eller `text`.

Till exempel:

```
SELECT id+1 as nid /*:int*/
FROM table
```

Geometrikolumnens typ och koordinatreferenssystem kan också ställas in med hjälp av särskilda kommentarer med följande syntax `/*:gtype:srid*/` där `gtype` är geometritypen (`point`, `linestring`, `polygon`, `multipoint`, `multilinestring` eller `multipolygon`) och `srid` ett heltal som representerar EPSG-koden för ett koordinatreferenssystem.

Användning av index

När du begär ett lager via ett virtuellt lager kommer källlagrets index att användas på följande sätt:

- om ett =-predikat används på lagrets primärnyckelkolumn, kommer den underliggande dataleverantören att tillfrågas om ett visst id (FilterFid)
- för alla andra predikat (>, <=, !=, etc.) eller på en kolumn utan primärnyckel, kommer en begäran byggd från ett uttryck att användas för att begära den underliggande vektordataleverantören. Det betyder att index kan användas på databasleverantörer om de finns.

Det finns en specifik syntax för att hantera spatiala predikat i förfrågningar och utlöser användningen av ett spatialt index: en dold kolumn med namnet `_search_frame_` finns för varje virtuellt lager. Den här kolumnen kan jämföras med en bounding box för att visa att den är lika. Ett exempel:

```
SELECT *
FROM vtab
WHERE _search_frame_=BuildMbr(-2.10,49.38,-1.3,49.99,4326)
```

Spatiala binära predikat som `ST_Intersects` snabbas upp avsevärt när de används tillsammans med denna spatiala indexsyntax.

11.3 Georeferens

Georeferensen är ett verktyg för att anpassa raster- eller vektorlager utan referens till kända koordinatsystem med hjälp av markkontrollpunkter (GCP). Det stöder export av transformerade raster (GeoTIFF) och vektorer (shapefiler, geopaket etc.) eller skrivning av tillhörande världsfiler (endast för raster). Det grundläggande tillvägagångssättet för georeferering av ett lager är att hitta punkter på det för vilka du kan bestämma exakta koordinater.

Funktioner

Ikon	Syfte	Ikon	Syfte
	Öppna raster		Öppna vektor
	Starta georeferering		
	Generera GDAL-skript		Ladda GCP-punkter
	Spara GCP-punkter som		Inställningar för omvandling
	Lägg till GCP-punkt		Ta bort GCP-punkt
	Flytta GCP-punkt		Panorera
	Zooma in		Zooma ut
	Zooma till lager		Zoom sista
	Zoom Nästa		Länka georeferensen till QGIS
	Länka QGIS till georeferensen		Full histogramsträckning
	Sträckning av lokalt histogram		

Tabell Georeferenser: Verktyg för georeferenser

11.3.1 Georefererande rasterlager

Som X- och Y-koordinater (DMS (dd mm ss.ss), DD (dd.dd) eller projicerade koordinater (mmmm.mm)), som motsvarar den valda punkten på bilden, kan två alternativa förfaranden användas:

- Rastret i sig ger ibland kors med koordinater som är ”skrivna” på bilden. I så fall kan du ange koordinaterna manuellt.
- Använda redan georefererade lager. Detta kan vara antingen vektor- eller rasterdata som innehåller samma objekt/funktioner som du har på den bild som du vill georeferera och med den projektion som du vill ha för din bild. I det här fallet kan du ange koordinaterna genom att klicka på referensdatasetet som laddats i QGIS kartkanvas.

Den vanliga proceduren för georeferering av en bild innebär att man väljer flera punkter på rastret, anger deras koordinater och väljer en relevant transformationstyp. Baserat på indataparametrarna och -data beräknar Georeferencer parametrarna för världsfilen. Ju fler koordinater du anger, desto bättre blir resultatet.

Det första steget är att starta QGIS och klicka på *Lager* ► *Georeferencer*, som visas i QGIS menyrad. Georeferencer-dialogrutan visas som i Fig. 11.34.

I det här exemplet använder vi ett topoark över South Dakota från SDGS. Det kan senare visualiseras tillsammans med data från GRASS `spearfish60`-platsen. Du kan ladda ner topoarket här: https://grass.osgeo.org/sampledata/spearfish_toposheet.tar.gz.

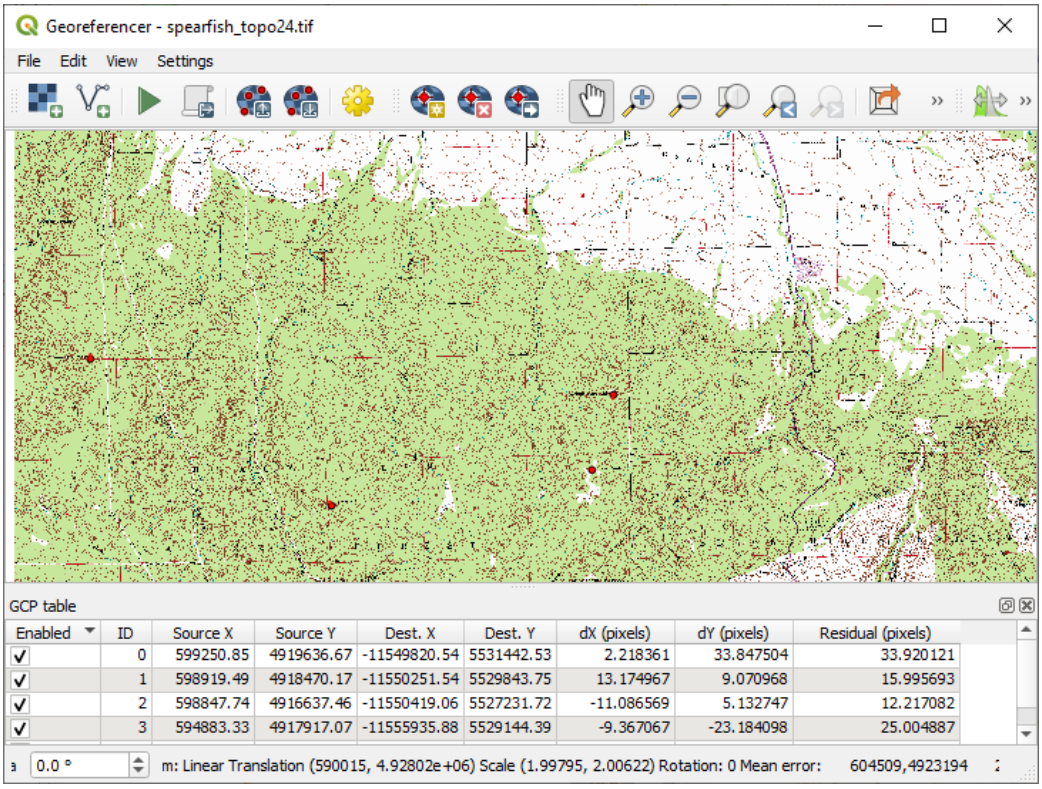


Fig. 11.34: Dialog för georeferenser

Inmatning av markkontrollpunkter (GCP)

1. För att börja georeferera ett oreferentierat raster måste vi ladda det med hjälp av knappen . Rastret kommer att visas i dialogrutans huvudarbetsområde. När rastret har laddats kan vi börja ange referenspunkter.

2. Använd knappen Lägg till GCP-punkt för att lägga till punkter i huvudarbetsområdet och ange deras koordinater (se figur Fig. 11.35). För denna procedur har du följande alternativ:

   - Klicka på en punkt i rasterbilden och ange X- och Y-koordinaterna manuellt, tillsammans med punktens CRS.
   - Klicka på en punkt i rasterbilden och välj knappen From map canvas för att lägga till X- och Y-koordinaterna med hjälp av en georefererad karta som redan är inläst i QGIS map canvas. CRS kommer att ställas in automatiskt.
   - När du anger GCP:er från huvudkartan har du möjlighet att dölja georeferensfönstret när du väljer punkter från huvudkartan. Om kryssrutan *Automatically hide georeferencer window* är markerad, efter att du har klickat på *From Map Canvas*, kommer huvudfönstret för georeferensen att döljas tills en punkt läggs till på kartbilden. Dialogrutan *Enter Map Coordinates* kommer att förbli öppen. Om rutan är avmarkerad förblir båda fönstren öppna medan du väljer en punkt på kartbilden. Detta alternativ gäller endast när georeferensfönstret inte är dockat i huvudgränssnittet.

3. Fortsätt att mata in punkter. Du bör ha minst fyra punkter, och ju fler koordinater du kan ange, desto bättre blir resultatet. Det finns ytterligare verktyg för att zooma och panorera arbetsområdet för att hitta en relevant uppsättning GCP-punkter.

Tips

> För att undvika att ständigt växla mellan knapparna Pan, Lägg till GCP-punkt och Flytta GCP-punkt kan du använda tangentbordets piltangenter för att flytta och mushjulet för att skala den georefererade kartan på ett bekvämt sätt.

4. När du har angett några punkter kan du använda knapparna Länk QGIS till Georeferencer och/eller Länk Georeferencer till QGIS som justerar kartutbredningen i QGIS-huvudfönstret till den aktuella vyn i Georeferencer och/eller vice versa.
5. Med verktyget kan du flytta GCP:erna både i Canvas och i georeferensfönstret om du behöver korrigera dem.

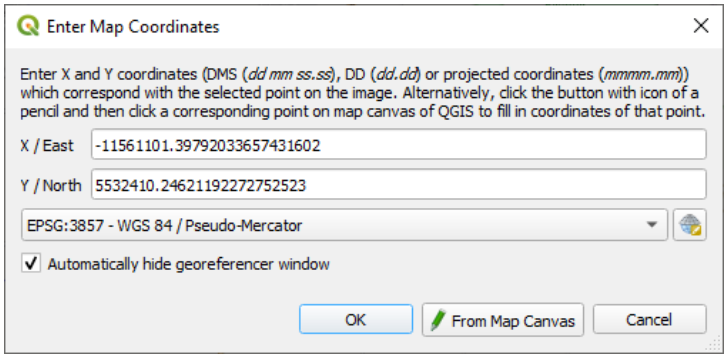


Fig. 11.35: Lägg till punkter i rasterbilden

De punkter som läggs till på kartan lagras i en separat textfil (`[filnamn].points`), vanligtvis tillsammans med rasterbilden. Detta gör att vi kan öppna georeferensen igen vid ett senare tillfälle och lägga till nya punkter eller ta bort befintliga för att optimera resultatet. Punktfilen innehåller värden av följande form: `mapX, mapY, pixelX, pixelY`. Du kan använda knapparna Ladda GCP-punkter och Spara GCP-punkter som för att hantera filerna.

Definiera transformationsinställningarna

När du har lagt till dina GCP:er i rasterbilden måste du definiera transformationsinställningarna för georeferensprocessen.

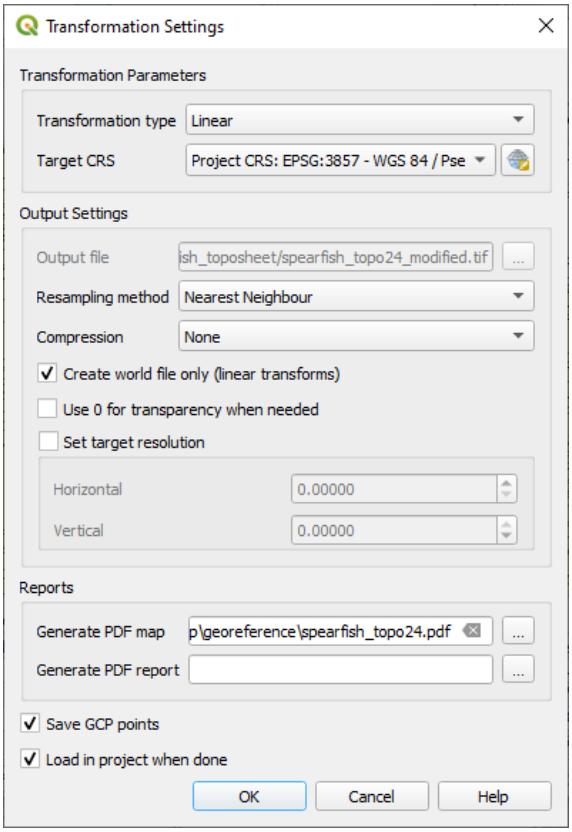


Fig. 11.36: Definiera inställningar för transformation av georeferensen

Tillgängliga omvandlingsalgoritmer

Det finns ett antal transformationsalgoritmer som är beroende av typen och kvaliteten på indata, vilken typ och mängd geometrisk distorsion du är villig att införa i slutresultatet och antalet markkontrollpunkter (GCP).

För närvarande finns följande *Transformationstyper* tillgängliga:

- Algoritmen **Linear** används för att skapa en world-fil och skiljer sig från de andra algoritmerna genom att den inte transformerar rasterpixlarna. Den tillåter positionering (translatering) av bilden och enhetlig skalning, men ingen rotation eller andra transformationer. Den är mest lämplig om din bild är en rasterkarta av god kvalitet, i ett känt CRS, men bara saknar georefereringsinformation. Minst 2 GCP:er behövs.
- Transformationen **Helmert** tillåter även rotation. Det är särskilt användbart om ditt raster är en lokal karta av god kvalitet eller en ortorektifierad flygbild, men inte är i linje med rutnätslagret i ditt CRS. Minst 2 GCP:er behövs.
- Algoritmen **Polynomial 1** möjliggör en mer allmän affin transformation, i synnerhet även en enhetlig skjuvning. Raka linjer förblir raka (d.v.s. kollinjära punkter förblir kollinjära) och parallella linjer förblir parallella. Detta är särskilt användbart för georeferering av datakartogram, som kan ha ritats (eller data samlats in) med olika markpixelstorlekar i olika riktningar. Minst 3 GCP:er krävs.
- **Polynomial**-algoritmerna 2-3 använder mer allmänna 2:a eller 3:e gradens polynom i stället för bara affin transformation. Detta gör att de kan ta hänsyn till krökning eller annan systematisk förvrängning av bilden, till exempel fotograferade kartor med böjda kanter. Minst 6 (respektive 10) GCP:er krävs. Vinklar och lokal skala bevaras inte eller behandlas inte enhetligt över hela bilden. I synnerhet kan raka linjer bli krökta och det kan uppstå betydande distorsion vid kanterna eller långt från eventuella GCP:er till följd av att de dataanpassade polynomen extrapoleras för långt.
- Algoritmen **Projective** generaliserar Polynomial 1 på ett annat sätt och tillåter transformationer som representerar en central projektion mellan två icke-parallella plan, bilden och kartduken. Raka linjer förblir raka, men parallelliteten bevaras inte och skalan över bilden varierar i enlighet med perspektivförändringen.

Denna transformationstyp är mest användbar för georeferenser av vinklade fotografier (snarare än platta skanningar) av kartor av god kvalitet eller sneda flygbilder. Minst 4 GCP:er krävs.

- Slutligen ”gummipläterar” **Thin Plate Spline** (TPS)-algoritmen rastret med hjälp av flera lokala polynom för att matcha de angivna GCP:erna, med minimerad övergripande ytkrökning. Områden bort från GCP flyttas runt i utdata för att tillgodose GCP-matchningen, men kommer annars att vara minimalt lokalt deformerade. TPS är mest användbart för georeferenser av skadade, deformerade eller på annat sätt något felaktiga kartor eller dåligt ortorektifierade flygbilder. Det är också användbart för ungefärlig georeferering och implicit omprojicering av kartor med okänd projektionstyp eller parametrar, men där ett regelbundet rutnät eller en tät uppsättning ad hoc-GCP:er kan matchas med ett referenskartlager. Tekniskt sett krävs minst 10 GCP:er, men vanligtvis fler för att lyckas.

I alla algoritmer utom TPS kommer parametrarna att anpassas så att det totala restfelet minimeras om fler än de minsta GCP:erna anges. Detta är till hjälp för att minimera effekterna av registreringsfel, dvs. små felaktigheter i pekarklick eller inskrivna koordinater, eller andra små lokala bilddeformationer. Om det inte finns andra GCP:er som kan kompensera kan sådana fel eller deformationer leda till betydande förvrängningar, särskilt nära den georefererade bildens kanter. Om fler än de minsta GCP:erna anges kommer de dock bara att matcha ungefär i utdata. Däremot kommer TPS att exakt matcha alla angivna GCP:er, men kan införa betydande deformationer mellan närliggande GCP:er med registreringsfel.

Definiera resamplingsmetoden

Vilken typ av omsampling du väljer beror sannolikt på dina indata och det slutliga målet med övningen. Om du inte vill ändra statistiken för rastret (annat än vad som följer av ojämn geometrisk skalning om du använder andra transformationer än Linear, Helmert eller Polynomial 1), kanske du vill välja ”Nearest neighbour”. Däremot kommer t.ex. ”kubisk resampling” vanligtvis att generera ett visuellt jämnare resultat.

Det är möjligt att välja mellan fem olika omprovningsmetoder:

1. Närmaste granne
2. Bilineär (2x2-kärna)
3. Kubisk (4x4-kärna)
4. Kubisk B-Spline (4x4 kernel)
5. Lanczos (6x6-kärna)

Definiera inställningarna för omvandling

Det finns flera alternativ som måste definieras för det georefererade utdatarastret.

- Kryssrutan *Create world file* är endast tillgänglig om du väljer att använda den linjära transformationstypen, eftersom detta innebär att rasterbilden faktiskt inte kommer att transformeras. I detta fall är fältet *Output raster* inte aktiverat, eftersom endast en ny världsfil kommer att skapas.
- För alla andra transformationstyper måste du definiera en *Output raster*. Som standard kommer en ny fil ([filnamn]_modifierad) att skapas i samma mapp tillsammans med den ursprungliga rasterbilden.
- Som ett nästa steg måste du definiera *Target CRS* (Coordinate Reference System) för det georefererade rastret (se *Arbeta med projektioner*).
- Om du vill kan du **genererar en pdf-karta** och även **en pdf-rapport**. Rapporten innehåller information om de använda transformationsparametrarna, en bild av residualerna och en lista med alla GCP:er och deras RMS-fel.
- Dessutom kan du aktivera kryssrutan *Set Target Resolution* och definiera pixelupplösningen för utmatningsrastret. Standardvärdet för horisontell och vertikal upplösning är 1.
- Checkboxen *Use 0 for transparency when needed* kan aktiveras om pixlar med värdet 0 ska visualiseras som transparenta. I vårt exempel på toposheet skulle alla vita områden vara transparenta.
- Kryssrutan *Spara GCP-punkter* lagrar GCP-punkter i en fil bredvid utdatarastret.

- Slutligen, *Load in project when done* laddar utmatningsrastret automatiskt in i QGIS-kartan när omvandlingen är klar.

Genomföra omvandlingen

När alla GCP:er har samlats in och alla transformationsinställningar har definierats trycker du bara på knappen Start georeferencing för att skapa det nya georefererade rastret.

11.3.2 Vektorlager för georeferenser

Georeferering av vektorlager fungerar på samma sätt som georeferering av rasterlager, men i stället för att matcha bildpixlar matchar du vektorgeometrier (punkter, linjer eller polygoner) med kända spatiala referenser.

Standardförfarandet börjar på samma sätt som för rastergeoreferenser: öppna QGIS och lägg till ett lager i kartbilden som ska användas som referens. Detta kan vara ett georefererat raster- eller vektorlager eller ett WMS-lager.

Öppna dialogrutan Georeferencer från *Layer* ► *Georeferencer*.

Börja georeferera genom att följa dessa steg (i det här exemplet använder vi en orefererad `alaska.shp`):

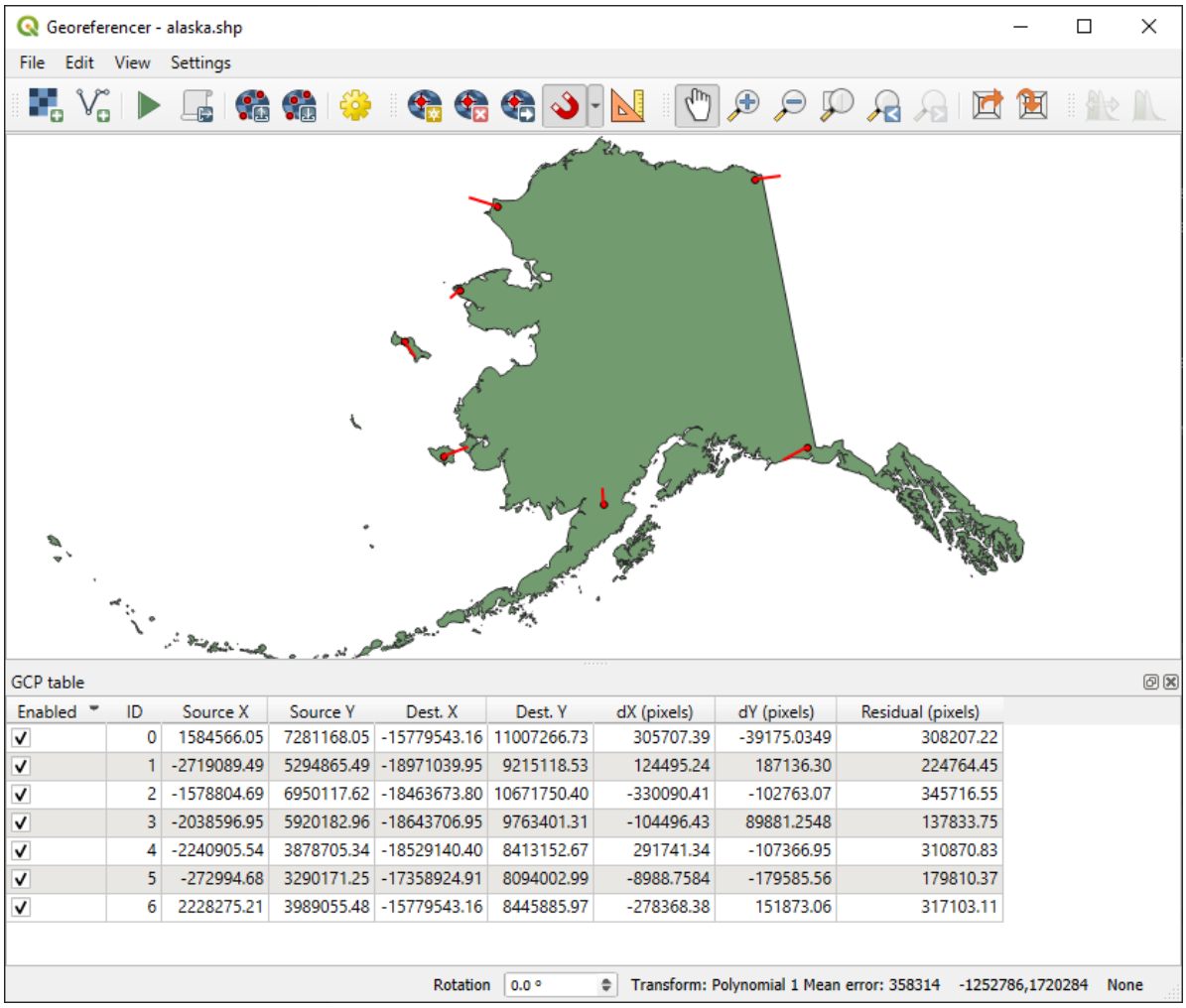


Fig. 11.37: Dialog för vektorbaserad georeferens

1. Ladda det oreferentierade vektorlagret med hjälp av knappen . Vektorlagret kommer att visas i dialogrutans huvudarbetsområde.

2. Använd knappen Lägg till GCP-punkt för att lägga till en punkt i arbetsområdet. Ange dess koordinater manuellt och ställ in CRS, eller klicka på knappen From map canvas för att hämta koordinaterna från ett georefererat lager i QGIS huvudkarta. I så fall kommer CRS att ställas in automatiskt.
3. Definiera transformationsinställningarna:
   - Välj *Transformationstyp*. Transformationsalgoritmerna är desamma som för georeferering av raster. Se *Definiera transformationsinställningarna* för mer information.
   - Definiera *Target CRS* (Coordinate Reference System) för den georefererade vektorn (se *Arbeta med projektioner*).
   - Ange format och sökväg för utdatafilen (t.ex. GeoPackage, Shapefile). Som standard skapas en ny fil med suffixet `_modified` i samma mapp som den ursprungliga vektorfilen.
   - Alternativt kan du aktivera **Generera PDF-karta** och **Generera PDF-rapport**. Rapporten innehåller transformationsparametrar, GCP-residualer och en sammanfattning av RMS-fel.
   - Aktivera *Save GCP Points* för att lagra GCP:er i en fil tillsammans med det utgående vektorlagret.
   - Aktivera *Load in project when done* för att lägga till resultatet direkt i kartbilden.
4. Klicka på Starta georeferering för att köra transformationen och generera det georefererade vektorlagret.

11.3.3 Visa och anpassa lagrets egenskaper

Genom att klicka på alternativet *Källegenskaper* i menyn *Inställningar* öppnas *Raster Layer properties* eller *Vector Layer properties* beroende på vilken typ av lager du georefererar.

11.3.4 Konfigurera georeferensen

Du kan anpassa beteendet för georeferensen i *Inställningar* ► *Konfigurera georeferensen* (eller använd kortkommandot `Ctrl+P`).

- Under *Point Tip* kan du använda kryssrutorna för att växla mellan att visa GCP-ID och X/Y-koordinater i både Georeferencer-fönstret och huvudkartan.
- *Residual Units* styr om residualenheterna ska anges i pixlar eller kartenheter
- *PDF Report* låter dig ställa in marginalstorlek i mm för rapport export
- *PDF Map* låter dig välja pappersstorlek för exporten av kartan
- Slutligen kan du aktivera kryssrutan *Visa Georeferensfönstret dockat*. Detta dockar Georeferensfönstret i QGIS huvudfönster i stället för att visa det som ett separat fönster som kan minimeras.

11.4 Utforska dataformat och fält

11.4.1 Rasterdata

GIS-rasterdata är matriser av diskreta celler som representerar egenskaper/fenomen på, över eller under jordytan. Varje cell i rasterrutnätet har samma storlek och cellerna är vanligtvis rektangulära (i QGIS är de alltid rektangulära). Typiska rasterdatauppsättningar inkluderar fjärranalysdata, t.ex. flygfotografier eller satellitbilder och modellerade data, t.ex. höjd eller temperatur.

Till skillnad från vektordata har rasterdata vanligtvis inte en associerad databaspost för varje cell. De geokodas med hjälp av pixelupplösning och X/Y-koordinaten för en hörnpixel i rasterlagret. Detta gör att QGIS kan positionera data korrekt på kartbilden.

GeoPackage-formatet är praktiskt för lagring av rasterdata när du arbetar med QGIS. Det populära och kraftfulla GeoTiff-formatet är ett bra alternativ.

QGIS använder georeferensinformation i rasterlagret (t.ex. GeoTiff) eller en associerad *världsfil* för att visa data på rätt sätt.

11.4.2 Vektordata

Många av de funktioner och verktyg som finns i QGIS fungerar på samma sätt oavsett vektordatakälla. Men på grund av skillnaderna i formatspecifikationer (GeoPackage, ESRI Shapefile, MapInfo och MicroStation filformat, AutoCAD DXF, PostGIS, SpatiaLite, Oracle Spatial, MS SQL Server, SAP HANA Spatial databaser och många fler) kan QGIS hantera vissa av deras egenskaper på olika sätt. Stöd ges av GDAL-vektordrivrutinerna. I detta avsnitt beskrivs hur man arbetar med dessa detaljer.

Observera

QGIS stöder funktionstyperna (multi)punkt, (multi)linje, (multi)polygon, CircularString, CompoundCurve, CurvePolygon, MultiCurve, MultiSurface, alla med valfria Z- och/eller M-värden.

Du bör också notera att vissa drivrutiner inte stöder vissa av dessa funktionstyper, t.ex. funktionstyperna CircularString, CompoundCurve, CurvePolygon, MultiCurve, MultiSurface. QGIS kommer att konvertera dem.

GeoPackage

Formatet GeoPackage (GPKG) är plattformsoberoende och implementeras som en SQLite-databasbehållare och kan användas för att lagra både vektor- och rasterdata. Formatet definierades av Open Geospatial Consortium (OGC) och publicerades 2014.

GeoPackage kan användas för att lagra följande i en SQLite-databas:

- **vektor** funktioner
- **plattmatrisuppsättningar av bildmaterial** och **raster**-kartor
- attribut (icke spatiala data)
- utökningarnas

Sedan QGIS version 3.8 kan GeoPackage även lagra QGIS-projekt. GeoPackage-lager kan ha JSON-fält.

GeoPackage är standardformatet för vektordata i QGIS.

ESRI Shapefile-format

ESRI Shapefile-formatet är fortfarande ett av de mest använda vektorfilformaten, även om det har vissa begränsningar jämfört med till exempel GeoPackage och SpatiaLite.

En dataset i ESRI Shapefile-format består av flera filer. Följande tre krävs:

1. `.shp`-fil som innehåller geometrierna för objekten
2. `.dbf`-fil som innehåller attributen i dBase-format
3. `.shx` indexfil

En dataset i ESRI Shapefile-format kan också innehålla en fil med suffixet `.prj`, som innehåller projektionsinformation. Det är mycket användbart att ha en projektionsfil, men det är inte obligatoriskt. En datauppsättning i Shapefile-format kan innehålla ytterligare filer. För ytterligare information, se ESRI:s tekniska specifikation <https://www.esri.com/content/dam/esrisites/sitecore-archive/Files/Pdfs/library/whitepapers/pdfs/shapefile.pdf>`_.

GDAL har stöd för läsning och skrivning av komprimerade ESRI Shapefile-format (`shz` och `shp.zip`).

Förbättrad prestanda för dataset i ESRI Shapefile-format

Om du vill förbättra ritningsprestandan för en dataset i ESRI Shapefile-format kan du skapa ett spatialt index. Ett spatialt index förbättrar hastigheten för både zoomning och panorering. Spatiala index som används av QGIS har filnamnstillägget `.qix`.

Använd dessa steg för att skapa indexet:

1. Ladda en dataset i ESRI Shapefile-format (se *Bläddrarpanelen*)
2. Öppna dialogrutan *Lageregenskaper* genom att dubbelklicka på lagrets namn i teckenförklaringen eller genom att högerklicka och välja *Properties…* från snabbmenyn
3. På fliken *Source* klickar du på knappen *Create Spatial Index*

Problem med att ladda en .prj-fil

Om du laddar en dataset i ESRI Shapefile-format med en `.prj`-fil och QGIS inte kan läsa koordinatreferenssystemet från den filen, måste du definiera rätt projektion manuellt på fliken *Lageregenskaper* ► *Källa* i lagret genom att klicka på knappen Välj CRS. Detta beror på att `.prj`-filer ofta inte innehåller de fullständiga projektionsparametrar som används i QGIS och som anges i dialogrutan *CRS*.

Av samma anledning skapas två olika projektionsfiler om du skapar en ny dataset i ESRI Shapefile-format med QGIS: en `.prj`-fil med begränsade projektionsparametrar, kompatibel med ESRI-programvara, och en `.qpj`-fil som innehåller alla parametrar för CRS. När QGIS hittar en `.qpj`-fil kommer den att användas istället för `.prj`.

Avgränsade textfiler

Avgränsade textfiler är mycket vanliga och används ofta på grund av sin enkelhet och läsbarhet - data kan visas och redigeras i en vanlig textredigerare. En avgränsad textfil är en tabell med kolumner som avgränsas av ett definierat tecken och rader som avgränsas av radbrytningar. Den första raden innehåller vanligtvis kolumnnamnen. En vanlig typ av avgränsad textfil är en CSV-fil (Comma Separated Values), där kolumnerna är åtskilda med kommatecken. Avgränsade textfiler kan också innehålla positionsinformation (se *Lagring av geometriinformation i avgränsade textfiler*).

QGIS tillåter dig att ladda en avgränsad textfil som ett lager eller en vanlig tabell (se *Bläddrarpanelen* eller *Importera en avgränsad textfil*). Kontrollera först att filen uppfyller följande krav:

1. Filen måste ha en avgränsad rubrikrad med fältnamn. Detta måste vara den första raden i datan (helst den första raden i textfilen).
2. Om geometri ska vara aktiverad måste filen innehålla fält som definierar geometrin. Dessa fält kan ha vilket namn som helst.
3. Fälten för X- och Y-koordinater (om geometrin definieras med koordinater) måste anges som siffror. Koordinatsystemet är inte viktigt.
4. Om du har en CSV-fil med kolumner som inte är strängar kan du ha en tillhörande CSVT-fil (se avsnitt *Använda CSVT-fil för att styra fältformatering*).

Höjdpunktsdatafilen `elevp.csv` i QGIS exempeldataset (se avsnitt *Nedladdning av exempeldata*) är ett exempel på en giltig textfil:

```
X;Y;ELEV
-300120;7689960;13
-654360;7562040;52
1640;7512840;3
[...]
```

Några saker att notera om textfilen:

1. I textfilen i exemplet används `;` (semikolon) som avgränsare (vilket tecken som helst kan användas för att avgränsa fälten).
2. Den första raden är rubrikraden. Den innehåller fälten `X`, `Y` och `ELEV`.

3. Inga citattecken (") används för att avgränsa textfält
4. X-koordinaterna finns i fältet `X`
5. Y-koordinaterna finns i fältet `Y`

Lagring av geometriinformation i avgränsade textfiler

Avgränsade textfiler kan innehålla geometriinformation i två huvudformer:

- Som koordinater i separata kolumner (t.ex. `Xcol`, `Ycol`...), för punktgeometridata;
- Som välkänd text (WKT) representation av geometri i en enda kolumn, för alla geometrityper.

Funktioner med krökta geometrier (CircularString, CurvePolygon och CompoundCurve) stöds. Här är några exempel på geometrityper i en avgränsad textfil med geometrier kodade som 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))
```

Avgränsade textfiler stöder också Z- och M-koordinater i geometrier:

```
LINESTRINGZ(10.0 20.0 30.0, 11.0 21.0 31.0, 11.0 22.0 30.0)
```

Använda CSVT-fil för att styra fältformatering

Vid inläsning av CSV-filer antar GDAL-drivrutinen att alla fält är strängar (dvs. text) om inget annat anges. Du kan skapa en CSVT-fil för att tala om för GDAL (och QGIS) vilken datatyp de olika kolumnerna har:

Typ	Namn	Exempel
Heltal	Heltal	4
Boolean	Integer(Boolean)	sant
Decimaltal	Verklig	3.456
Datum	Datum (ÅÅÅÅ-MM-DD)	2016-07-28
Tid	Tid (HH:MM:SS+nn)	18:33:12+00
Datum & Tid	DatumTid (ÅÅÅÅ-MM-DD HH:MM:SS+nn)	2016-07-28 18:33:12+00
CoordX	CoordX	8.8249
Samordnare	Samordnare	47.2274
Punkt(X)	Punkt(X)	8.8249
Punkt(Y)	Punkt(Y)	47.2274
WKT	WKT	PUNKT(15 20)

CSVT-filen är en vanlig textfil på **EN rad** med datatyperna inom citationstecken och åtskilda med kommatecken, t.ex.:

```
"Integer","Real","String"
```

Du kan även ange bredd och precision för varje kolumn, t.ex.:

```
"Integer(6)","Real(5.5)","String(22)"
```

Den här filen sparas i samma mapp som filen `.csv`, med samma namn, men med `.csvt` som tillägg.

Du kan hitta mer information på GDAL CSV Driver.

Tips

Detektera fälttyper

Istället för att använda en CSVT-fil för att ange datatyperna ger QGIS möjlighet att automatiskt upptäcka fälttyperna och att ändra de antagna fälttyperna.

PostGIS-lager

PostGIS-lager lagras i en PostgreSQL-databas. Fördelarna med PostGIS är spatial indexering, filtrering och frågefunktioner. Med PostGIS fungerar vektorfunktioner som att välja och identifiera mer exakt än de gör med GDAL-lager i QGIS.

Tips

PostGIS-lager

Normalt identifieras ett PostGIS-lager av en post i tabellen geometry_columns. QGIS kan ladda lager som inte har någon post i tabellen geometry_columns. Detta inkluderar både tabeller och vyer. Se din PostgreSQL-manual för information om hur du skapar vyer.

Detta avsnitt innehåller några detaljer om hur QGIS får tillgång till PostgreSQL-lager. För det mesta bör QGIS helt enkelt förse dig med en lista över databastabeller som kan laddas, och den kommer att ladda dem på begäran. Men om du har problem med att läsa in en PostgreSQL-tabell i QGIS kan informationen nedan hjälpa dig att förstå QGIS-meddelanden och ge dig anvisningar för att ändra PostgreSQL-tabell- eller vydefinitionen så att QGIS kan läsa in den.

Observera

En PostgreSQL-databas kan också lagra QGIS-projekt.

Primär nyckel

QGIS kräver att PostgreSQL-lager innehåller en kolumn som kan användas som en unik nyckel för lagret. För tabeller betyder detta vanligtvis att tabellen behöver en primärnyckel eller en kolumn med en unik begränsning på den. I QGIS måste denna kolumn vara av typen int4 (ett heltal med storleken 4 byte). Alternativt kan kolumnen ctid användas som primärnyckel. Om en tabell saknar dessa poster kommer kolumnen oid att användas istället. Prestanda kommer att förbättras om kolumnen är indexerad (observera att primära nycklar automatiskt indexeras i PostgreSQL).

QGIS erbjuder en kryssruta **Select at id** som är aktiverad som standard. Det här alternativet hämtar id:n utan attribut, vilket är snabbare i de flesta fall.

Visa

Om PostgreSQL-lagret är en vy finns samma krav, men vyer har inte alltid primära nycklar eller kolumner med unika begränsningar på dem. Du måste definiera ett primärnyckelfält (måste vara ett heltal) i QGIS-dialogrutan innan du kan ladda vyn. Om det inte finns någon lämplig kolumn i vyn kommer QGIS inte att ladda lagret. Om detta inträffar är lösningen att ändra vyn så att den innehåller en lämplig kolumn (en typ av heltal och antingen en primärnyckel eller med en unik begränsning, helst indexerad).

När det gäller tabellen är en kryssruta **Select at id** aktiverad som standard (se ovan för innebörden av kryssrutan). Det kan vara vettigt att inaktivera det här alternativet när du använder dyra vyer.

Observera

PostgreSQL främmande tabell

PostgreSQL främmande tabeller stöds inte uttryckligen av PostgreSQL-leverantören och kommer att hanteras som en vy.

Säkerhetskopiering av QGIS layer_style-tabell och databas

Om du vill göra en säkerhetskopia av din PostGIS-databas med hjälp av kommandona `pg_dump` och `pg_restore`, och standardlagerstilarna som sparats av QGIS inte återställs efteråt, måste du ställa in XML-alternativet till `DOCUMENT` före återställningskommandot:

1. Gör en PLAIN backup av tabellen `layer_style`
2. Öppna filen i en textredigerare
3. Ändra raden `SET xmloption = content;` till `SET XML OPTION DOCUMENT;`
4. Spara till fil…
5. Använd psql för att återställa tabellen i den nya databasen

Filtrera databasens sida

QGIS gör det möjligt att filtrera funktioner redan på serversidan. Markera *Inställningar* ► *Alternativ* ► *Datakällor* ► *Utför uttryck på serversidan om möjligt* för att göra det. Endast uttryck som stöds kommer att skickas till databasen. Uttryck som använder operatorer eller funktioner som inte stöds kommer att falla tillbaka till lokal utvärdering.

Stöd för PostgreSQL-datatyper

Datatyper som stöds av PostgreSQL-leverantören inkluderar: heltal, flottör, boolean, binärt objekt, varchar, geometri, tidsstämpel, array, hstore och json.

Importera data till PostgreSQL

Data kan importeras till PostgreSQL / PostGIS med hjälp av flera verktyg, inklusive DB Manager-plugin och kommandoradsverktygen shp2pgsql och ogr2ogr.

DB Manager

QGIS levereras med ett kärntillägg som heter DB Manager. Den kan användas för att ladda data och har stöd för scheman. Se avsnitt *DB Manager Plugin* för mer information.

shp2pgsql

PostGIS innehåller ett verktyg som heter **shp2pgsql**, som kan användas för att importera datamängder i Shapefile-format till en PostGIS-aktiverad databas. Om du till exempel vill importera en dataset i Shapefile-format med namnet `lakes.shp` till en PostgreSQL-databas med namnet `gis_data`, använder du följande kommando:

```
shp2pgsql -s 2964 lakes.shp lakes_new | psql gis_data
```

Detta skapar ett nytt lager med namnet `lakes_new` i databasen `gis_data`. Det nya lagret kommer att ha en spatial referensidentifierare (SRID) på 2964. Se avsnitt *Arbeta med projektioner* för mer information om spatiala referenssystem och projektioner.

Tips

Exportera dataset från PostGIS

Det finns också ett verktyg för att exportera PostGIS-dataset till Shapefile-format: **pgsql2shp**. Det levereras med din PostGIS-distribution.

ogr2ogr

Förutom **shp2pgsql** och **DB Manager** finns det ytterligare ett verktyg för att mata in geografiska data i PostGIS: **ogr2ogr**. Det är en del av din GDAL-installation.

Gör så här för att importera en dataset i Shapefile-format till PostGIS:

```
ogr2ogr -f "PostgreSQL" PG:"dbname=postgis host=myhost.de user=postgres
password=topsecret" alaska.shp
```

Detta importerar datasetet i Shapefile-format `alaska.shp` till PostGIS-databasen *postgis* med användaren *postgres* och lösenordet *topsecret* på värdservern *myhost.de*.

Observera att GDAL måste byggas med PostgreSQL för att stödja PostGIS. Du kan verifiera detta genom att skriva (i):

```
ogrinfo --formats | grep -i post
```

Om du föredrar att använda PostgreSQL: s **COPY** -kommando istället för standard **INSERT INTO** -metoden kan du exportera följande miljövariabel (åtminstone tillgänglig på och):

```
export PG_USE_COPY=YES
```

ogr2ogr skapar inte spatiala index som **shp2pgsl** gör. Du måste skapa dem manuellt med hjälp av det vanliga SQL-kommandot **CREATE INDEX** efteråt, som ett extra steg (enligt beskrivningen i nästa avsnitt *Förbättrad prestanda*).

Förbättrad prestanda

Att hämta funktioner från en PostgreSQL-databas kan vara tidskrävande, särskilt över ett nätverk. Du kan förbättra ritningsprestandan för PostgreSQL-lager genom att se till att ett PostGIS-spatialt index finns på varje lager i databasen. PostGIS stöder skapandet av ett GiST-index (Generalized Search Tree) för att påskynda spatial sökning (GiST-indexinformation hämtas från PostGIS-dokumentationen som finns på https://postgis.net).

Tips

Du kan använda DBManager för att skapa ett index för ditt lager. Du bör först markera lagret och klicka på *Table* ► *Edit table*, gå till *Indexes*-fliken och klicka på *Add Spatial Index*.

Syntaxen för att skapa ett GiST-index är:

```
CREATE INDEX [indexname] ON [tablename]
  USING GIST ( [geometryfield] GIST_GEOMETRY_OPS );
```

Observera att det kan ta lång tid att skapa indexet för stora tabeller. När indexet har skapats bör du utföra en `VACUUM ANALYZE`. Se PostGIS-dokumentationen (POSTGIS-PROJECT i *Litteratur- och webbreferenser*) för mer information.

I följande exempel skapas ett 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-lager

Om du vill spara ett vektorlager i SpatiaLite-format kan du göra det genom att följa instruktionerna i *Skapa nya lager från ett befintligt lager*. Du väljer `SpatiaLite` som *Format* och anger både *Filnamn* och *Lagernamn*.

Du kan också välja `SQLite` som format och sedan lägga till `SPATIALITE=YES` i fältet *Custom Options* ► *Data source*`. Detta talar om för GDAL att skapa en SpatiaLite databas. Se även https://gdal.org/en/latest/drivers/vector/sqlite.html.

QGIS stöder också redigerbara vyer i SpatiaLite. För SpatiaLite-datahantering kan du också använda kärntillägget *DB Manager*.

Om du vill skapa ett nytt SpatiaLite-lager, se avsnitt *Skapa ett nytt SpatiaLite-lager*.

GeoJSON-specifika parametrar

När *exporterar lager* till GeoJSON finns det några specifika *Lageralternativ* tillgängliga. Dessa alternativ kommer från GDAL som är ansvarig för skrivningen av filen:

- *COORDINATE_PRECISION* det maximala antalet siffror efter decimalavgränsaren som ska skrivas in i koordinater. Standardvärdet är 15 (obs: för Lat Lon-koordinater anses 6 räcka). Trunkering kommer att ske för att ta bort efterföljande nollor.
- *RFC7946* som standard kommer GeoJSON 2008 att användas. Om inställningen är YES kommer den uppdaterade standarden RFC 7946 att användas. Standard är NO (alltså GeoJSON 2008). Se https://gdal.org/en/latest/drivers/vector/geojson.html#rfc-7946-write-support för de viktigaste skillnaderna, i korthet: endast EPSG:4326 är tillåtet, andra crs:er kommer att transformeras, polygoner kommer att skrivas så att de följer högerregeln för orientering, värdena för en ”bbox”-array är [west, south, east, north], inte [minx, miny, maxx, maxy]. Vissa namn på tilläggsmedlemmar är förbjudna i FeatureCollection-, Feature- och Geometry-objekt, standardkoordinatprecisionen är 7 decimaler
- *WRITE_BBOX* inställd på YES för att inkludera geometriernas avgränsningsbox på objekt- och objektsamlingsnivå

Förutom GeoJSON finns det också ett alternativ att exportera till ”GeoJSON - Newline Delimited” (se https://gdal.org/en/latest/drivers/vector/geojsonseq.html). I stället för en FeatureCollection med Features kan du strömma en typ (förmodligen bara Features) sekventiellt separerade med nya linjer.

GeoJSON - Newline Delimited har också några specifika Layer-alternativ tillgängliga:

- *COORDINATE_PRECISION* se ovan (samma som för GeoJSON)
- *RS* om poster ska inledas med tecknet RS=0x1E. Skillnaden är hur funktionerna separeras: endast med en ny rad (LF) (Newline Delimited JSON, geojsonl) eller genom att även föregå en postseparator (RS) (ger GeoJSON Text Sequences, geojsons). Standardvärdet är NO. Filer får tillägget `.json` om tillägget inte anges.

SAP HANA Spatiala lager

Detta avsnitt innehåller några detaljer om hur QGIS får tillgång till SAP HANA-lager. För det mesta bör QGIS helt enkelt förse dig med en lista över databastabeller och vyer som kan laddas, och den kommer att ladda dem på begäran. Men om du har problem med att ladda en SAP HANA-tabell eller -vy i QGIS kan informationen nedan hjälpa dig att förstå grundorsaken och hjälpa till att lösa problemet.

Identifiering av funktioner

Om du vill använda alla QGIS funktioner för funktionsredigering måste QGIS kunna identifiera varje funktion i ett lager på ett otvetydigt sätt. Internt använder QGIS ett 64-bitars signerat heltal för att identifiera funktioner, medan det negativa intervallet är reserverat för speciella ändamål.

Därför kräver SAP HANA-leverantören en unik nyckel som kan mappas till ett positivt 64-bitars heltal för att fullt ut stödja QGIS funktioner för redigering av funktioner. Om det inte är möjligt att skapa en sådan mappning kan du fortfarande visa funktionerna, men redigeringen kanske inte fungerar.

Lägga till tabeller

När du lägger till en tabell som ett lager använder SAP HANA-providern tabellens primärnyckel för att mappa den till ett unikt funktions-ID. För att få fullt stöd för funktionsredigering måste du därför ha en primärnyckel till din tabelldefinition.

SAP HANA-leverantören stöder primärnycklar med flera kolumner, men om du vill få bästa möjliga prestanda bör din primärnyckel vara en enda kolumn av typen `INTEGER`.

Lägga till vyer

När du lägger till en vy som ett lager kan SAP HANA-leverantören inte automatiskt identifiera kolumner som entydigt identifierar en funktion. Dessutom är vissa vyer skrivskyddade och kan inte redigeras.

För att få fullt stöd för funktionsredigering måste vyn vara uppdaterbar (kontrollera kolumnen `IS_READ_ONLY` i systemvyn `SYS.VIEWS` för vyn i fråga) och du måste manuellt förse QGIS med en eller flera kolumner som identifierar en funktion. Kolumnerna kan anges genom att använda *Layer* ► *Add Layer* ► *Add SAP HANA Spatial Layer* och sedan välja kolumnerna i kolumnen *Feature id*. För bästa prestanda bör värdet för *Feature id* vara en enda `INTEGER`-kolumn.

11.4.3 Lager som korsar 180° longitud

Många GIS-paket packar inte in lager med ett geografiskt referenssystem (lat/lon) som korsar 180 graders longitudlinjen. Om vi öppnar ett sådant lager i QGIS kan vi därför se två vitt skilda platser, som borde visas nära varandra. I Fig. 11.38, den lilla punkten längst till vänster på kartduken (Chatham Islands) borde vara inom rutnätet, till höger om Nya Zeelands huvudöar.

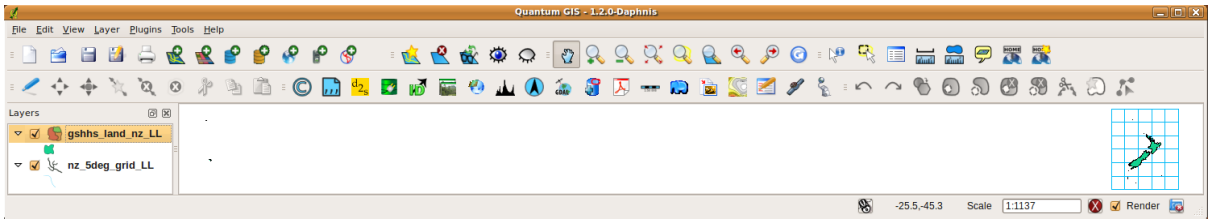


Fig. 11.38: Karta i lat/lon som korsar 180° longitudlinjen

Lösning i PostGIS

En lösning är att omvandla longitudvärdena med hjälp av PostGIS och funktionen ST_ShiftLongitude. Denna funktion läser varje punkt/vertex i varje komponent i varje funktion i en geometri och flyttar dess longitudkoordinat från -180..0° till 180..360° och vice versa om den ligger mellan dessa intervall. Denna funktion är symmetrisk så resultatet är en 0..360° representation av en -180..180° data och en -180..180° representation av en 0..360° data.

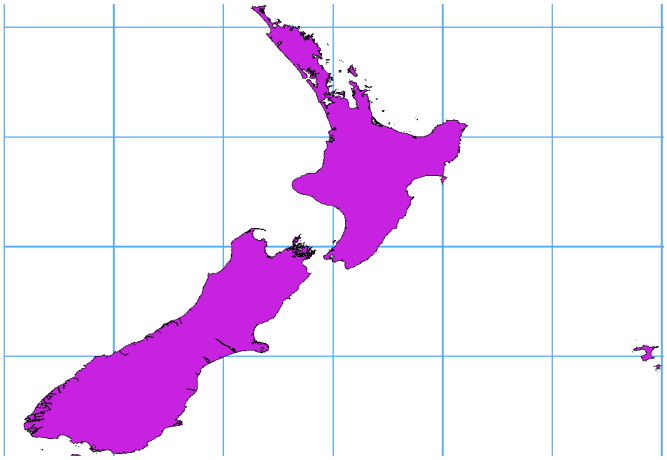


Fig. 11.39: Övergång till 180° longitud med hjälp av funktionen **ST_ShiftLongitude**

1. Importera data till PostGIS (*Importera data till PostgreSQL*) med hjälp av t.ex. DB Manager-tillägget.
2. Använd PostGIS kommandoradsgränssnitt för att ge följande kommando:

```
-- In this example, "TABLE" is the actual name of your PostGIS table
update TABLE set geom=ST_ShiftLongitude(geom);
```

3. Om allt gick bra bör du få en bekräftelse på antalet funktioner som uppdaterades. Sedan kan du ladda kartan och se skillnaden (*Figure_vector_crossing_map*).

KAPITEL 12

Arbeta med vektordata

12.1 Dialogrutan Egenskaper för vektor

Dialogrutan *Lageregenskaper* för ett vektorlager innehåller allmänna inställningar för att hantera utseendet på lagrets funktioner i kartan (symbologi, märkning, diagram), interaktion med musen (åtgärder, karttips, formulärdesign). Här finns också information om lagret.

För att komma till dialogrutan *Lageregenskaper*:

- Dubbelklicka på lagret i panelen *Lager* eller högerklicka och välj *Egenskaper…* från popup-menyn;
- Gå till menyn *Layer* ► *Lageregenskaper…* när lagret är markerat.

Dialogrutan vector *Lageregenskaper* innehåller följande avsnitt:

Information	*Källa*	*Simbologi*[1]
Labels[1]	*Mask*[1]	*3D-vy*[1]
Diagram	*Fält*	*Attributformulär*
Sammanfogningar	*Extra lagringsutrymme*	*Åtgärder*
Visa	*Rendering*	*Temporal*
Variabler	*Elevation*	*Metadata*
Beroenden	*Förklaring*	*QGIS Server*
Digitalisering	*Externa tillägg*[2] flikar	

[1] Finns även i *Layer stylingpanel*

[2] *Externa tillägg* som du installerar kan lägga till flikar i den här dialogen. Dessa presenteras inte i detta dokument. Hänvisa till deras dokumentation.

Tips

** Dela hela eller delar av egenskaperna för lagrets stilar**

Med menyn *Style* längst ner i dialogrutan kan du importera eller exportera dessa eller delar av dessa egenskaper från/till flera olika destinationer (fil, urklipp, databas). Se *Hantera anpassade stilar*.

Observera

Eftersom egenskaper (symbologi, etikett, åtgärder, standardvärden, formulär …) för inbäddade lager (se *Inbäddning av lager från externa projekt*) hämtas från den ursprungliga projektfilen och för att undvika ändringar som kan bryta detta beteende, görs dialogen för lageregenskaper otillgänglig för dessa lager.

12.1.1 Informationsegenskaper

Fliken *Information* är skrivskyddad och utgör en intressant plats för att snabbt hämta sammanfattad information och metadata om det aktuella lagret. Tillhandahållen information är:

- allmänna uppgifter som namn i projektet, källsökväg, lista över hjälpfiler, senaste spartid och storlek, den använda leverantören
- anpassade egenskaper, som används för att lagra ytterligare information om lagret i det aktiva projektet. Standardanpassade egenskaper kan inkludera *layer notes*, *förklaring widgets*, *layer variables*, *form properties*… Fler anpassade egenskaper kan skapas och hanteras med hjälp av PyQGIS, särskilt genom metoden setCustomProperty().
- baserat på lagrets leverantör: lagringsformat, geometrityp, kodning av datakälla, utsträckning, antal funktioner …
- det koordinatsatta referenssystemet: namn, enheter, metod, noggrannhet, referens (dvs. om det är statiskt eller dynamiskt)
- plockad från *fyllda metadata*: access, extents, länkar, kontakter, historia…
- och relaterad till dess geometri (spatial utsträckning, CRS …) eller dess attribut (antal fält, egenskaper hos varje …).

12.1.2 Källegenskaper

Använd den här fliken för att definiera allmänna inställningar för vektorlagret.

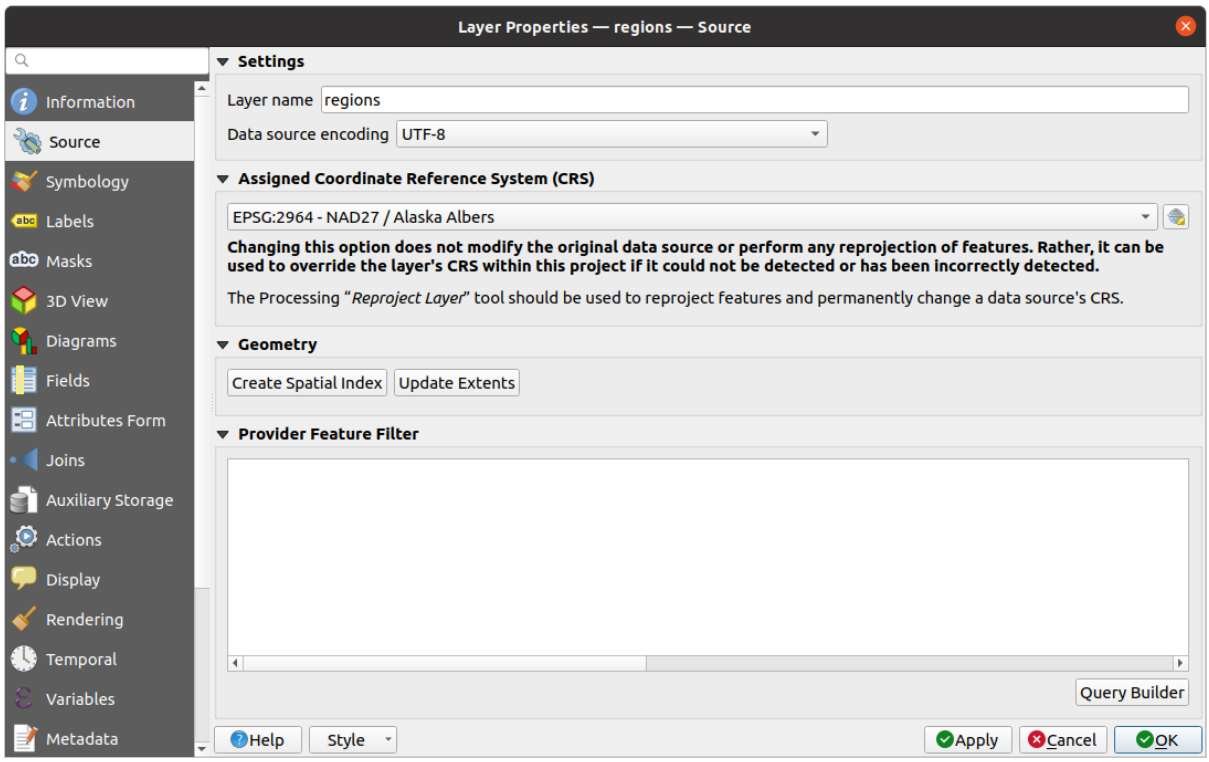


Fig. 12.1: Fliken Källa i dialogen Egenskaper för vectorlager

Inställningar

- Ange ett *Layernamn* som skiljer sig från lagrets filnamn och som kommer att användas för att identifiera lagret i projektet (i *Layers Panel*, med uttryck, i utskriftslayoutens förklaring, …)
- Beroende på dataformat, välj *Datakällans kodning* om den inte upptäcks korrekt av QGIS.

Koordinatreferenssystem och geometri

- Visar lagrets *Tilldelade koordinatreferenssystem (CRS)*. Du kan ändra lagrets CRS genom att välja ett nyligen använt CRS i listrutan eller genom att klicka på Välj CRS-knappen (se *Väljare för referenssystem för koordinater*). Använd denna process endast om det CRS som tillämpas på lagret är fel eller om inget CRS har tillämpats. Om du vill omprojicera dina data till ett annat CRS, använd hellre algoritmer för omprojektering av lager från Processing eller *Spara det till ett annat lager*.
- *Create spatial index* (endast för format som stöds av OGR).
- *Uppdatera information om utbredning* för ett lager.

Frågeverktyg

Dialogrutan Query Builder nås via knappen *Query Builder* längst ned på fliken *Source* i dialogrutan Lageregenskaper, under gruppen *Provider feature filter*.

Query Builder tillhandahåller ett gränssnitt som gör att du kan definiera en delmängd av funktionerna i lagret med hjälp av en SQL-liknande WHERE-klausul och visa resultatet i huvudfönstret. Så länge som frågan är aktiv är endast de funktioner som motsvarar resultatet tillgängliga i projektet.

Du kan använda ett eller flera lagerattribut för att definiera filtret i `Query Builder`. Användningen av mer än ett attribut visas i Fig. 12.2. I exemplet kombinerar filtret attributen

- `toa` (fält `DateTime`: `cast("toa" som tecken) > '2017-05-17'` och `cast("toa" som tecken) < '2019-12-24T18:00:00'`),
- `namn` (`String` fält: `"name" > 'S'`) och
- `FID` (`Integer` fält: `FID > 10`)

med hjälp av operatörerna AND, OR och NOT samt parenteser. Denna syntax (inklusive DateTime-formatet för fältet `toa`) fungerar för GeoPackage-dataset.

Filtret görs på dataleverantörens (OGR, PostgreSQL, MS SQL Server …) nivå. Så syntaxen beror på dataleverantören (DateTime stöds till exempel inte för ESRI Shapefile-formatet). Det fullständiga uttrycket:

```
cast("toa" as character) > '2017-05-17' AND
cast("toa" as character) < '2019-12-24T18:00:00' AND
NOT ("name" > 'S' OR FID > 10)
```

Du kan också öppna dialogrutan *Query Builder* med alternativet *Filter…* från menyn *Layer* eller den kontextuella menyn för lagret. Avsnitten *Fields*, *Values* och *Operators* i dialogen hjälper dig att konstruera den SQL-liknande fråga som visas i rutan *Provider specific filter expression*.

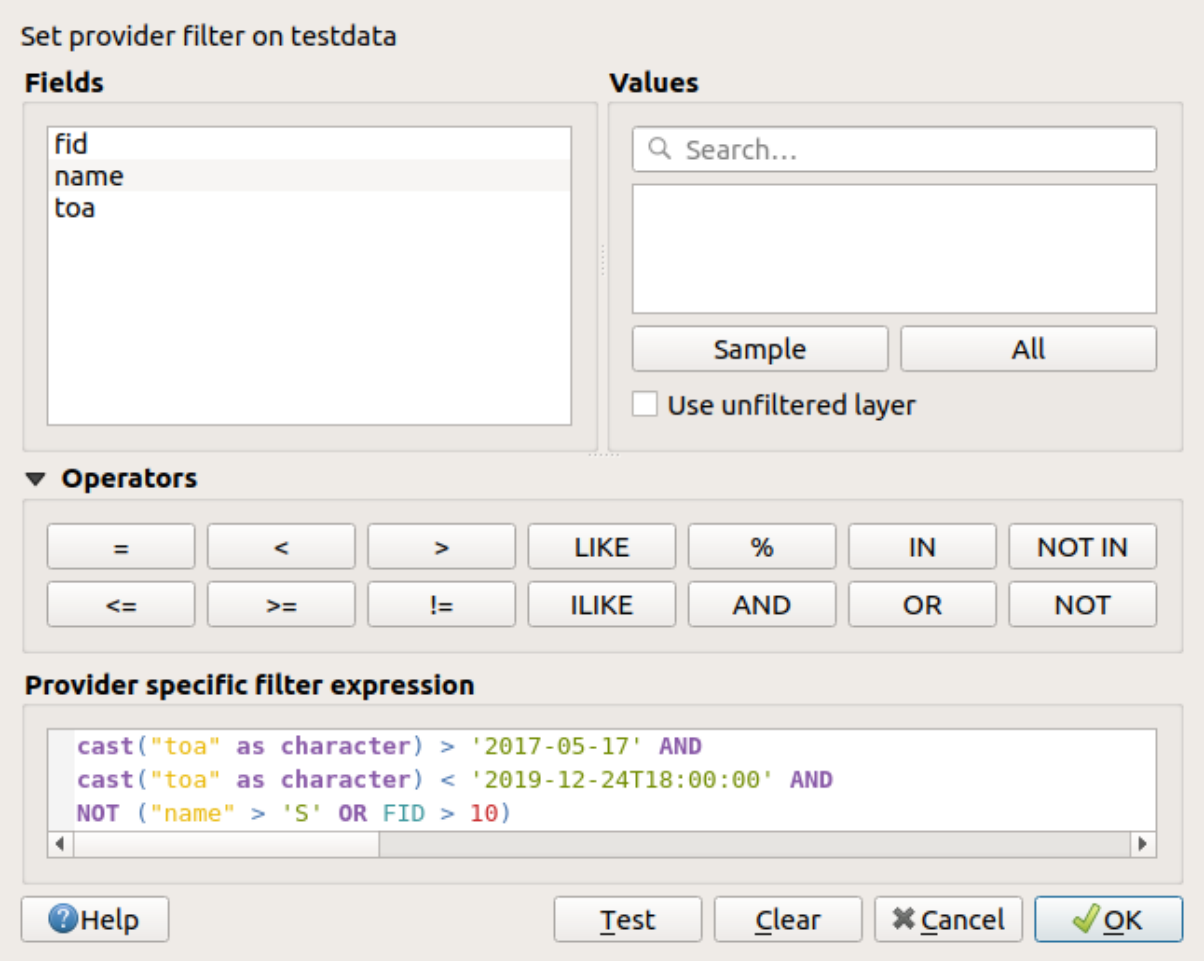


Fig. 12.2: Frågeverktyg

Listan **Fields** innehåller alla fält i lagret. För att lägga till en attributkolumn i fältet SQL WHERE clause, dubbelklicka på dess namn eller skriv in det i SQL-rutan.

I ramen **Values** visas värdena för det fält som för tillfället är markerat. Om du vill lista alla unika värden i ett fält klickar du på knappen *All*. Om du istället vill lista de 25 första unika värdena i kolumnen klickar du på knappen *Sample*. Om du vill lägga till ett värde i fältet SQL WHERE clause dubbelklickar du på dess namn i listan Values. Du kan använda sökrutan högst upp i Values-rutan för att enkelt bläddra igenom och hitta attributvärden i listan.

Avsnittet **Operatorer** innehåller alla användbara operatorer. Om du vill lägga till en operator i fältet SQL WHERE clause klickar du på lämplig knapp. Relationsoperatorer (= , > , …), strängjämförelseoperator (`LIKE`) och logiska operatorer (`AND`, `OR`, …) finns tillgängliga.

Knappen *Test* hjälper dig att kontrollera din fråga och visar en meddelanderuta med antalet funktioner som uppfyller den aktuella frågan. Använd knappen *Clear* för att radera SQL-frågan och återställa lagret till dess ursprungliga tillstånd (dvs. ladda alla funktioner fullt ut). Det är möjligt att *Save…* spara frågan som en `.QQF`-fil eller *Load…* ladda frågan från en fil till dialogrutan.

När ett filter tillämpas behandlar QGIS den resulterande delmängden som om den vore hela lagret. Om du till exempel har använt filtret ovan för ”Borough” (`"TYPE_2" = 'Borough'`) kan du inte visa, fråga, spara eller redigera `Anchorage`, eftersom det är en ”Municipality” och därför inte ingår i delmängden.

Tips

Filtrerade lager visas i panelen Lager

I panelen *Lager* listas det filtrerade lagret med en ikon Filter bredvid som anger vilken fråga som används när musen hålls över knappen. Dubbelklicka på ikonen för att öppna dialogrutan *Query Builder* för redigering. Detta kan också göras via menyn *Lager* ► *Filter…*.

12.1.3 Egenskaper för symbologi

Fliken *Symbology* ger dig ett omfattande verktyg för att rendera och symbolisera dina vektordata. Du kan använda verktyg som är gemensamma för alla vektordata, samt speciella symboliseringsverktyg som utformats för olika typer av vektordata. Alla typer har dock följande dialogstruktur: i den övre delen finns en widget som hjälper dig att förbereda klassificeringen och den symbol som ska användas för funktioner, och längst ned finns widgeten *Rendering av lager*.

Tips

Växla snabbt mellan olika lagerrepresentationer

Med hjälp av menyn *Styles* ► *Add* längst ner i dialogrutan *Lageregenskaper* kan du spara så många stilar som behövs. En stil är en kombination av alla egenskaper hos ett lager (t.ex. symbologi, märkning, diagram, fältformulär, åtgärder …) som du vill ha. Sedan är det bara att växla mellan stilarna från lagrets snabbmeny i *Layers Panel* för att automatiskt få olika representationer av dina data.

Tips

Exportera vektorsymbologi

Du har möjlighet att exportera vektorsymbologi från QGIS till Google *.kml, *.dxf och MapInfo *.tab-filer. Öppna bara höger musmeny för lagret och klicka på *Save As…* för att ange namnet på utdatafilen och dess format. I dialogrutan kan du använda menyn *Symbology export* för att spara symbologin antingen som *Feature symbology* ► eller som *Symbol layer symbology* ►. Om du har använt symbollager rekommenderas att du använder den andra inställningen.

Funktioner för rendering

Renderingsprogrammet är ansvarigt för att rita en funktion tillsammans med rätt symbol. Oavsett typ av lagergeometri finns det fyra vanliga typer av renderare: enkel symbol, kategoriserad, graderad och regelbaserad. För punktlager finns renderare för punktförskjutning, punktkluster och värmekarta tillgängliga, medan polygonlager också kan renderas med renderare för sammanslagna funktioner, inverterade polygoner och 2,5 D.

Det finns ingen kontinuerlig färgåtergivning, eftersom det i själva verket bara är ett specialfall av den graderade återgivningen. De kategoriserade och graderade renderarna kan skapas genom att ange en symbol och en färgramp - de ställer in färgerna för symbolerna på lämpligt sätt. För varje datatyp (punkter, linjer och polygoner) finns det olika typer av vektorsymbolskikt. Beroende på vilken renderingstyp som valts innehåller dialogrutan olika ytterligare avsnitt.

Observera

Om du ändrar renderingstyp när du ställer in stilen för ett vektorlager kommer de inställningar som du gjorde för symbolen att bibehållas. Tänk på att denna procedur endast fungerar för en ändring. Om du upprepar ändringen av renderingstyp kommer inställningarna för symbolen att gå förlorade.

Renderare för en enda symbol

Renderingsprogrammet *Single Symbol* används för att rendera alla funktioner i lagret med hjälp av en enda användardefinierad symbol. Se *Symbolväljaren* för ytterligare information om symbolrepresentation.

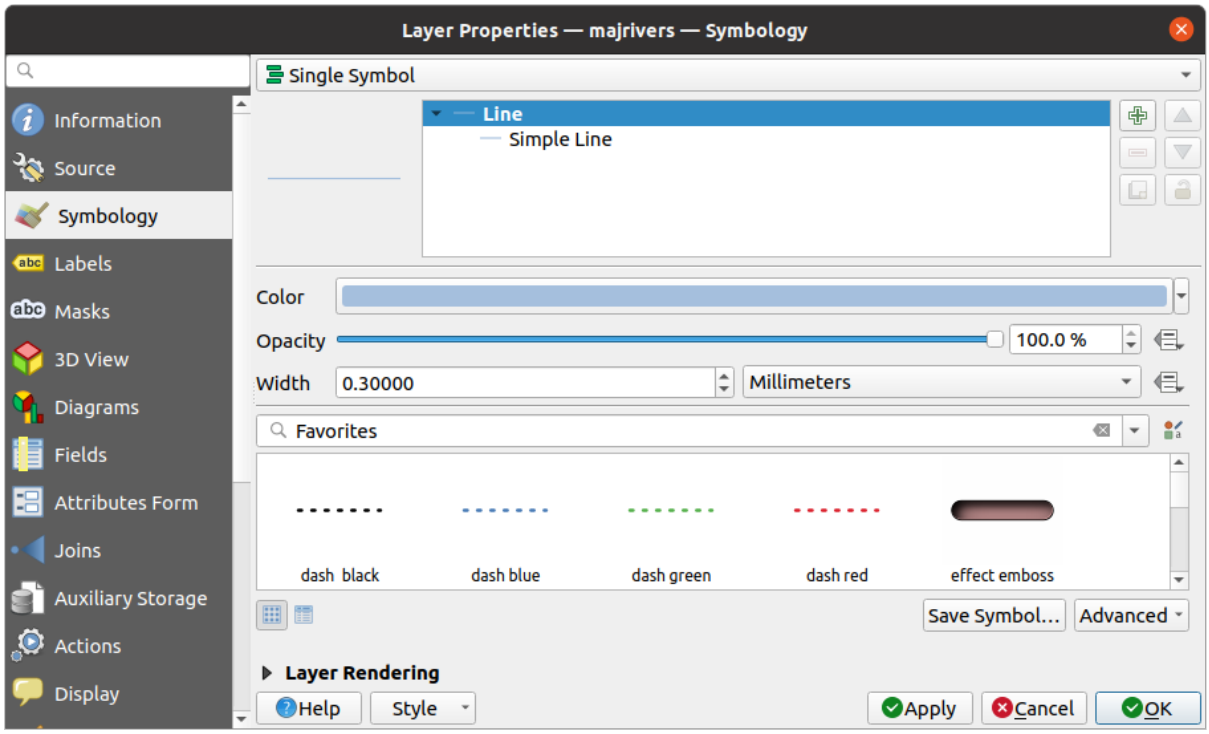


Fig. 12.3: Egenskaper för linjer med en enda symbol

No Symbols Renderer

Renderaren *Inga symboler* är ett specialfall av renderaren Single Symbol eftersom det tillämpar samma rendering på alla funktioner. Med den här renderingen ritas ingen symbol för funktioner, men etiketter, diagram och andra delar som inte är symboler visas ändå.

Det går fortfarande att göra urval på lagret i Canvas och valda funktioner återges med en standardsymbol. Funktioner som håller på att redigeras visas också.

Detta är tänkt som en praktisk genväg för lager som du bara vill visa etiketter eller diagram för, och du behöver inte rendera symboler med helt transparent fyllning/kant för att uppnå detta.

Kategoriserad Renderare

Renderingsprogrammet *Categorized* används för att rendera funktionerna i ett lager med hjälp av en användardefinierad symbol vars aspekt återspeglar de diskreta värdena i ett fält eller ett uttryck.

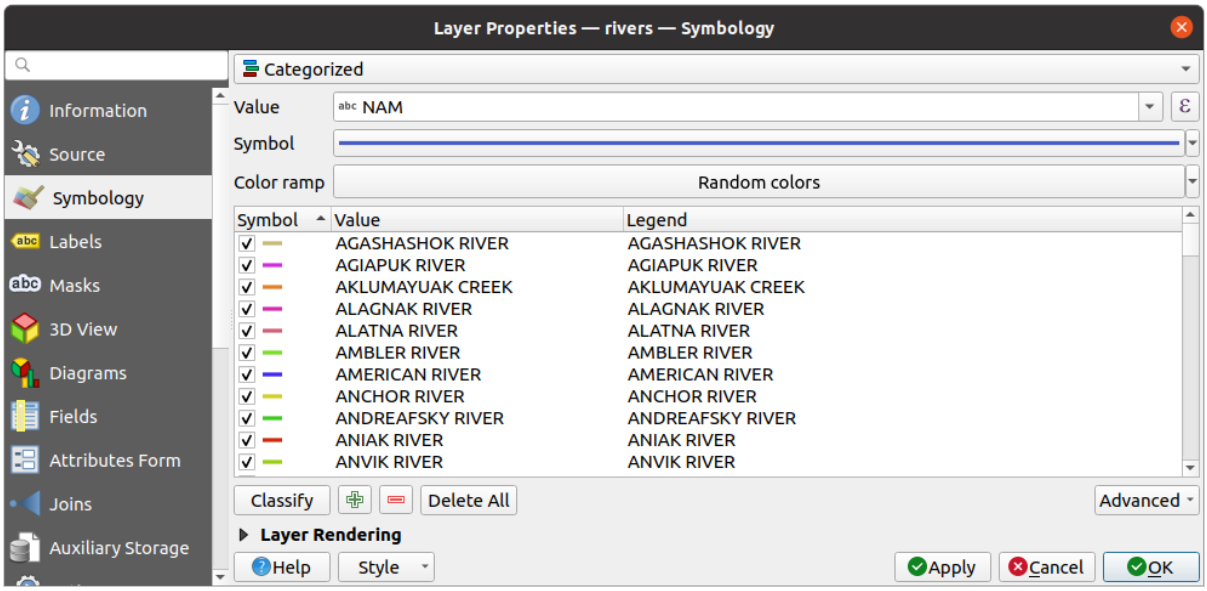


Fig. 12.4: Kategoriserade Symboliseringsalternativ

För att använda kategoriserad symbologi för ett lager:

1. Välj *Value* för klassificeringen: det kan vara ett befintligt fält eller ett *expression* som du kan skriva in i rutan eller skapa med hjälp av den tillhörande ε -knappen. Genom att använda uttryck för kategorisering behöver du inte skapa ett fält endast för symbologisyften (t.ex. om dina klassificeringskriterier härleds från ett eller flera attribut).

   Det uttryck som används för att klassificera egenskaper kan vara av vilken typ som helst, t.ex:

   - vara en jämförelse. I detta fall returnerar QGIS värdena `1` (**True**) och `0` (**False**). Några exempel:

     ```
     myfield >= 100
     $id = @atlas_featureid
     myfield % 2 = 0
     within( $geometry, @atlas_geometry )
     ```

   - kombinera olika områden:

     ```
     concat( field_1, ' ', field_2 )
     ```

   - vara en beräkning på fält:

     ```
     myfield % 2
     year( myfield )
     field_1 + field_2
     substr( field_1, -3 )
     ```

   - användas för att omvandla linjära värden till diskreta klasser, t.ex:

     ```
     CASE WHEN x > 1000 THEN 'Big' ELSE 'Small' END
     ```

   - kombinera flera diskreta värden till en enda kategori, t.ex:

     ```
     CASE
     WHEN building IN ('residence', 'mobile home') THEN 'residential'
     WHEN building IN ('commercial', 'industrial') THEN 'Commercial and␣
      ↪Industrial'
     END
     ```

> **Tips**
>
> Du kan använda alla typer av uttryck för att kategorisera funktioner, men för vissa komplexa uttryck kan det vara enklare att använda *rule-based rendering*.

2. Konfigurera *Symbol*, som kommer att användas som bassymbol för alla klasser;
3. Ange *Color ramp*, dvs. det färgområde från vilket den färg som tillämpas på varje symbol väljs.

   Förutom de vanliga alternativen i *color ramp widget* kan du tillämpa en *Random Color Ramp* på kategorierna. Du kan klicka på *Shuffle Random Colors* för att återskapa en ny uppsättning slumpmässiga färger om du inte är nöjd.
4. Klicka sedan på knappen *Classify* för att skapa klasser utifrån de distinkta värdena i det angivna fältet eller uttrycket.
5. *Tillämpa* ändringarna om *live update* inte används och varje objekt på kartbilden kommer att återges med symbolen för sin klass.

   Som standard lägger QGIS till en *all other values*-klass till listan. Den här klassen är tom i början, men används som standardklass för alla objekt som inte ingår i de andra klasserna (t.ex. när du skapar objekt med nya värden för klassificeringsfältet/uttrycket).

Ytterligare justeringar kan göras av standardklassificeringen:

- Du kan Lägg till nya kategorier, Ta bort valda kategorier eller *Ta bort alla* av dem.
- En klass kan inaktiveras genom att avmarkera kryssrutan till vänster om klassnamnet; motsvarande funktioner döljs på kartan.
- Dra och släpp raderna för att ändra ordningen på klasserna
- Om du vill ändra symbolen, värdet eller teckenförklaringen för en klass dubbelklickar du på objektet.

Om du högerklickar på ett eller flera valda objekt visas en kontextuell meny för att:

- *Copy Symbol* och *Paste Symbol*, ett bekvämt sätt att använda objektets representation på andra
- *Färgändring…* av den/de valda symbolen/symbolerna
- *Ändra opacitet…* för den eller de valda symbolerna
- *Ändra utmatningsenhet…* för den eller de valda symbolerna
- *Ändra bredd…* för den eller de valda linjesymbolerna
- *Ändra storlek…* för den eller de valda punktsymbolerna
- *Ändringsvinkel…* för den eller de valda punktsymbolerna
- *Sammanfoga kategorier*: Grupperar flera valda kategorier till en enda. Detta gör det enklare att utforma lager med ett stort antal kategorier, där det kan vara möjligt att gruppera många olika kategorier till en mindre och mer hanterbar uppsättning kategorier som gäller för flera värden.

> **Tips**
>
> Eftersom den symbol som behålls för de sammanslagna kategorierna är den som tillhör den högst upp i listan, kan det vara bra att flytta den kategori vars symbol du vill återanvända till toppen innan du slår samman kategorierna.

- *Unmerge Categories* som tidigare har slagits samman

De skapade klasserna visas också i en trädhierarki i panelen *Layers*. Dubbelklicka på en post i kartlegenden för att redigera den tilldelade symbolen. Högerklicka så får du upp några *fler alternativ*.

Menyn *Advanced* ger tillgång till alternativ för att snabba upp klassificeringen eller finjustera återgivningen av symbolerna:

- *Matcha med sparade symboler*: Med hjälp av *symbolbiblioteket*, tilldelar varje kategori en symbol vars namn representerar kategorins klassificeringsvärde
- *Matcha med symboler från filen…*: Förutsatt en fil med symboler, tilldelar varje kategori en symbol vars namn representerar klassificeringsvärdet för kategorin
- *Symbolnivåer…* för att definiera ordningen för återgivning av symboler.

Graderad rendering

Renderingsprogrammet *Graduated* används för att rendera alla funktioner från ett lager med hjälp av en användardefinierad symbol vars färg eller storlek återspeglar tilldelningen av en vald funktions attribut till en klass.

Precis som med Categorized Renderer kan du med Graduated Renderer definiera rotation och storleksskala från angivna kolumner.

På samma sätt som med Categorized Renderer kan du också välja:

- Klassificeringens *Värde*: det kan vara ett befintligt fält eller ett *uttryck* som du kan skriva in i rutan eller skapa med hjälp av den tillhörande -knappen. Genom att använda uttryck för klassificering behöver du inte skapa ett fält endast för symbologiändamål (t.ex. om dina klassificeringskriterier härleds från ett eller flera attribut).
- Symbolen (med hjälp av dialogrutan Symbolväljare)
- Format för teckenförklaring och precision
- Den metod som ska användas för att ändra symbolen: färg eller storlek
- Färgerna (med hjälp av färgramplistan) om färgmetoden är vald
- Storleken (med hjälp av storleksdomänen och dess enhet)

Då kan du använda fliken Histogram som visar ett interaktivt histogram av värdena från det tilldelade fältet eller uttrycket. Klassbrytningar kan flyttas eller läggas till med hjälp av histogramwidgeten.

Observera

Du kan använda panelen Statistical Summary för att få mer information om ditt vektorlager. Se *Panel för statistisk sammanfattning*.

På fliken Classes kan du ange antalet klasser och även sättet att klassificera funktioner inom klasserna (med hjälp av listan Mode). De tillgängliga lägena är:

- Lika antal (kvantipel): varje klass har samma antal element (tanken med en boxplot).
- Lika intervall: varje klass kommer att ha samma storlek (t.ex. med värdena 1 till 16 och fyra klasser kommer varje klass att ha en storlek på fyra).
- Fast intervall: varje klass har ett fast intervall av värden (t.ex. med värdena 1-16 och en intervallstorlek på 4 blir klasserna 1-4, 5-8, 9-12 och 13-16).
- Logaritmisk skala: lämplig för data med ett brett värdeintervall. Smala klasser för låga värden och breda klasser för stora värden (t.ex. för decimaltal med intervallet [0..100] och två klasser, kommer den första klassen att vara från 0 till 10 och den andra klassen från 10 till 100).
- Natural Breaks (Jenks): variansen inom varje klass minimeras medan variansen mellan klasserna maximeras.
- Pretty Breaks: beräknar en sekvens av cirka n+1 lika stora nice-värden som täcker intervallet för värdena i x. Värdena väljs så att de är 1, 2 eller 5 gånger en potens av 10. (baserat på pretty från R:s statistiska miljö https://www.rdocumentation.org/packages/base/topics/pretty).

- Standardavvikelse: Klasser byggs upp beroende på värdenas standardavvikelse.

Listboxen i mitten av fliken *Symbology* visar klasserna tillsammans med deras intervall, etiketter och symboler som kommer att återges.

Klicka på knappen **Classify** för att skapa klasser med hjälp av det valda läget. Varje klass kan inaktiveras genom att avmarkera kryssrutan till vänster om klassens namn.

Om du vill ändra symbol, värde och/eller etikett för klassen dubbelklickar du bara på det objekt som du vill ändra.

Om du högerklickar på ett eller flera valda objekt visas en kontextuell meny för att:

- *Copy Symbol* och *Paste Symbol*, ett bekvämt sätt att använda objektets representation på andra
- *Färgändring…* av den/de valda symbolen/symbolerna
- *Ändra opacitet…* för den eller de valda symbolerna
- *Ändra utmatningsenhet…* för den eller de valda symbolerna
- *Ändra bredd…* för den eller de valda linjesymbolerna
- *Ändra storlek…* för den eller de valda punktsymbolerna
- *Ändringsvinkel…* för den eller de valda punktsymbolerna

Exemplet i Fig. 12.5 visar den graderade renderingsdialogen för major_rivers-lagret i QGIS-datasetet.

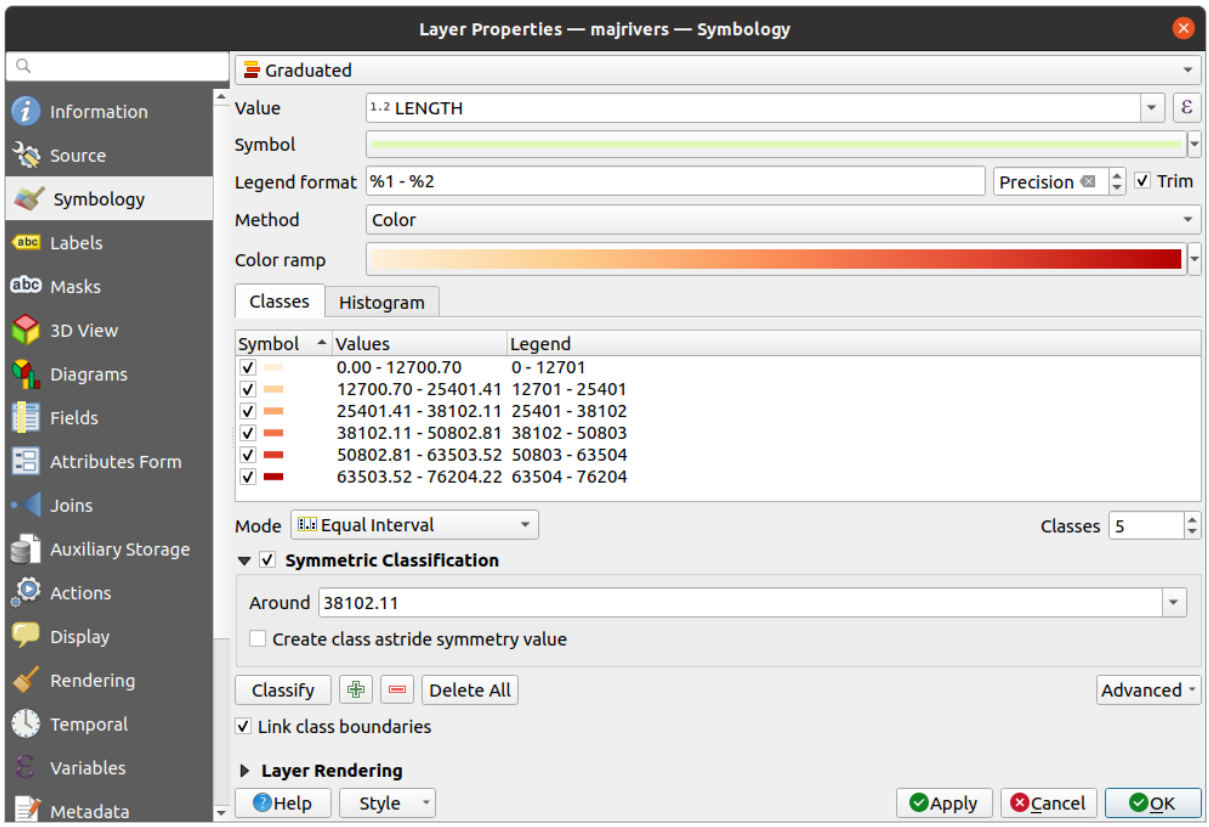


Fig. 12.5: Graderade alternativ för symbolisering

De skapade klasserna visas också i en trädhierarki i panelen *Layers*. Dubbelklicka på en post i kartlegenden för att redigera den tilldelade symbolen. Högerklicka så får du upp några *fler alternativ*.

Proportionell symbol och multivariat analys

Proportional Symbol och Multivariate Analysis är inte renderingstyper som är tillgängliga från rullgardinslistan Symbology rendering. Men med alternativen *data-defined override* som tillämpas över något av de tidigare renderingsalternativen, kan du i QGIS visa dina punkt- och linjedata med sådan representation.

Skapa proportionell symbol

För att tillämpa en proportionell rendering:

1. Applicera först *single symbol renderer* på lagret.
2. Ställ sedan in symbolen så att den gäller för funktionerna.
3. Markera objektet på den övre nivån i symbolträdet och använd Data-defined override *knappen* bredvid alternativet *Size* (för punktlager) eller *Width* (för linjelager).
4. Välj ett fält eller ange ett uttryck, och för varje funktion kommer QGIS att tillämpa utdatavärdet på egenskapen och proportionellt ändra storlek på symbolen i kartbilden.

   Om det behövs kan du använda alternativet *Size assistant…* i menyn för att tillämpa en transformation (exponentiell, flannery…) på omskalningen av symbolstorleken (se *Använda det datadefinierade assistentgränssnittet* för mer information).

Du kan välja att visa proportionella symboler i *Layers-panelen* och *print layout förklaring item*: vik upp rullgardinsmenyn *Advanced* längst ned i huvuddialogen på fliken *Symbology* och välj **Data-defined size förklaring…** för att konfigurera legendobjekten (se *Datadefinierad storleksförklaring* för mer information).

Skapande av multivariat analys

En multivariat analys hjälper dig att utvärdera förhållandet mellan två eller flera variabler, t.ex. kan den ena representeras av en färgramp medan den andra representeras av en storlek.

Det enklaste sättet att skapa multivariat analys i QGIS är att:

1. Applicera först en kategoriserad eller graderad rendering på ett lager och använd samma typ av symbol för alla klasser.
2. Applicera sedan en proportionell symbologi på klasserna:
   1. Klicka på knappen *Change* ovanför klassificeringsramen: du får upp dialogrutan *Symbolväljaren*.
   2. Skala om storleken eller bredden på symbollagret med hjälp av widgeten *data defined override* enligt ovan.

Liksom den proportionella symbolen kan den skalade symbologin läggas till i lagerträdet ovanpå de kategoriserade eller graderade klassymbolerna med hjälp av funktionen *data defined size förklaring*. Och båda representationerna är också tillgängliga i objektet för utskriftslayoutlegenden.

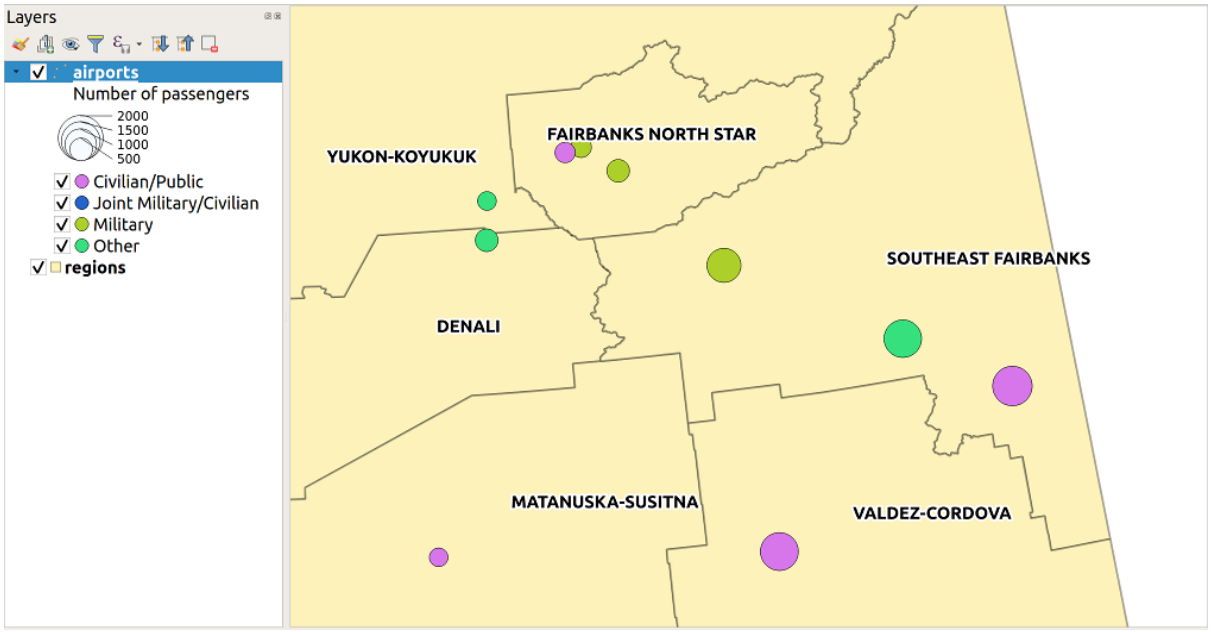


Fig. 12.6: Multivariat exempel med skalad storleksförklaring

Regelbaserad rendering

Regler är QGIS *uttryck* som används för att särskilja objekt enligt deras attribut eller egenskaper för att tillämpa specifika renderingsinställningar på dem. Regler kan vara nästlade, och objekt tillhör en klass om de tillhör alla de övre nästlingsnivåerna.

Renderingsprogrammet *Rule-based* är alltså utformat för att rendera alla funktioner från ett lager med hjälp av symboler vars aspekt återspeglar tilldelningen av en vald funktion till en finkornig klass.

Så här skapar du en regel:

1. Aktivera en befintlig rad genom att dubbelklicka på den (som standard lägger QGIS till en symbol utan regel när renderingsläget är aktiverat) eller klicka på knappen Redigera regel eller Lägg till regel.
2. I dialogrutan *Edit Rule* som öppnas kan du definiera en etikett som hjälper dig att identifiera varje regel. Det är denna etikett som kommer att visas i *Layers Panel* och även i teckenförklaringen för utskriftskompositören.
3. Ange manuellt ett uttryck i textrutan bredvid alternativet *Filter* eller tryck på knappen ε bredvid för att öppna dialogrutan för skapande av uttrycksträngar.
4. Använd de medföljande funktionerna och lagerattributen för att bygga ett *uttryck* för att filtrera de funktioner som du vill hämta. Tryck på knappen *Test* för att kontrollera resultatet av sökningen.
5. Du kan ange en längre etikett för att komplettera regelbeskrivningen.
6. Du kan använda alternativet *Scale Range* för att ange skalor som regeln ska tillämpas på.
7. Konfigurera slutligen *-symbolen så att den använder* för dessa funktioner.
8. Tryck sedan på *OK*.

En ny rad med en sammanfattning av regeln läggs till i dialogen Lageregenskaper. Du kan skapa så många regler som behövs genom att följa stegen ovan eller kopiera och klistra in en befintlig regel. Dra och släpp reglerna ovanpå varandra för att nesta dem och förfina de övre regelegenskaperna i underklasser.

Den regelbaserade renderingen kan kombineras med kategoriserade eller graderade renderingar. Om du väljer en regel kan du organisera dess funktioner i underklasser med hjälp av rullgardinsmenyn *Refine selected rules*. Förfinade klasser

visas som underobjekt till regeln, i en trädhierarki och precis som deras överordnade kan du ställa in symbologin och regeln för varje klass. Automatiserad regelförfining kan baseras på:

- **scales**: givet en lista med skalor skapar detta alternativ en uppsättning klasser som de olika användardefinierade skalområdena gäller för. Varje ny skalbaserad klass kan ha sin egen symbologi och sitt eget definitionsuttryck. Detta kan t.ex. vara ett bekvämt sätt att visa samma funktioner med olika symboler i olika skalor, eller att endast visa en uppsättning funktioner beroende på skalan (t.ex. lokala flygplatser i stor skala jämfört med internationella flygplatser i liten skala).
- **categories**: tillämpar en *categorized renderer* på de funktioner som faller inom den valda regeln.
- eller **ranges**: tillämpar en *graduerad rendering* på de funktioner som faller inom den valda regeln.

Förfinade klasser visas som underobjekt till regeln, i en trädhierarki och precis som ovan kan du ställa in symbologin för varje klass. Symbolerna för de nästlade reglerna staplas ovanpå varandra, så var försiktig när du väljer dem. Det är också möjligt att avmarkera *Symbols* i dialogrutan *Redigera regel* för att undvika att en viss symbol visas i stapeln.

I dialogrutan *Redigera regel* kan du undvika att skriva alla regler och använda alternativet *Else* för att fånga upp alla funktioner som inte matchar någon av de andra reglerna på samma nivå. Detta kan också uppnås genom att skriva `Else` i kolumnen *Regel* i dialogrutan *Lageregenskaper* ► *Symbology* ► *Rule-based*.

Om du högerklickar på ett eller flera valda objekt visas en kontextuell meny för att:

- *Copy* och *Paste*, ett smidigt sätt att skapa nya objekt baserat på befintliga objekt
- *Copy Symbol* och *Paste Symbol*, ett bekvämt sätt att använda objektets representation på andra
- *Färgändring…* av den/de valda symbolen/symbolerna
- *Ändra opacitet…* för den eller de valda symbolerna
- *Ändra utmatningsenhet…* för den eller de valda symbolerna
- *Ändra bredd…* för den eller de valda linjesymbolerna
- *Ändra storlek…* för den eller de valda punktsymbolerna
- *Ändringsvinkel…* för den eller de valda punktsymbolerna
- *Refine Current Rule*: öppnar en undermeny som gör det möjligt att förfina den aktuella regeln med **skalor**, **kategorier** eller **intervall**. Samma sak som att välja *motsvarande meny* längst ner i dialogrutan.

Om du avmarkerar en rad i dialogen för regelbaserad rendering döljs funktionerna för den specifika regeln och de nästlade reglerna i kartbilden.

De skapade reglerna visas också i en trädhierarki i kartans teckenförklaring. Dubbelklicka på en post i kartlegenden för att redigera den tilldelade symbolen. Högerklicka så får du upp några *fler alternativ*.

Exemplet i Fig. 12.7 visar den regelbaserade renderingsdialogen för flodlagret i QGIS exempeldataset.

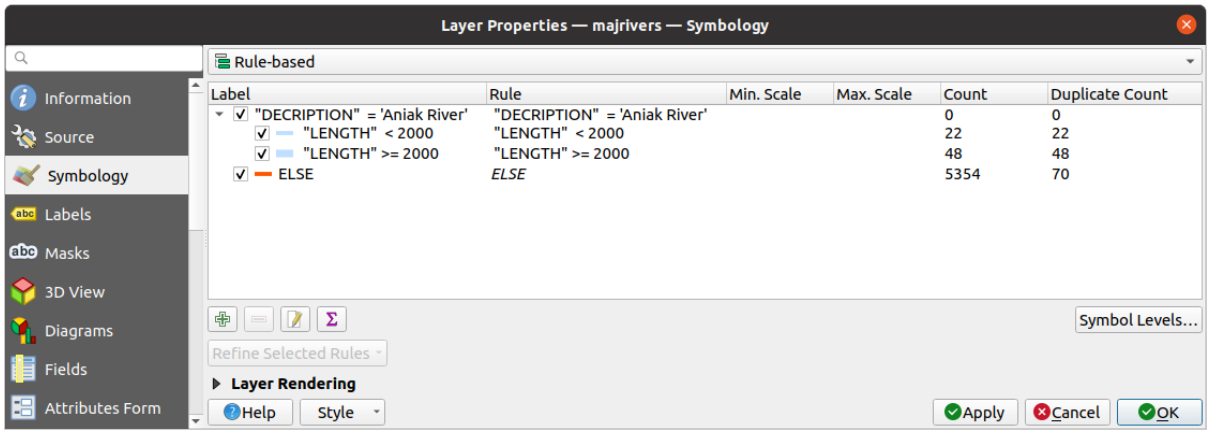


Fig. 12.7: Regelbaserade symboliseringsalternativ

Punktförskjutning Renderare

Renderingsprogrammet *Point Displacement* tar punktfunktionerna som faller inom en given avståndstolerans från varandra och placerar deras symboler runt deras barycenter, enligt olika placeringsmetoder. Detta kan vara ett bekvämt sätt att visualisera alla funktioner i ett punktlager, även om de har samma plats (t.ex. bekvämligheter i en byggnad).

För att konfigurera en renderare för punktförskjutning måste du göra följande:

1. Ställ in *Center symbol*: hur den virtuella punkten i mitten ska se ut
2. Välj *Renderer*-typ: hur du vill klassificera funktioner i lagret (enstaka, kategoriserade, regelbaserade …)
3. Tryck på knappen *Renderer Settings…* för att konfigurera funktionernas symbologi enligt den valda renderingsprogrammet
4. Ange toleransen för *Distance* inom vilken nära objekt anses överlappa varandra och sedan förflyttas över samma virtuella punkt. Stöder vanliga symbolenheter.
5. Konfigurera *Placeringsmetoder*:
   - **Ring**: placerar alla funktioner på en cirkel vars radie beror på antalet funktioner som ska visas.
   - **Koncentriska ringar**: använder en uppsättning koncentriska cirklar för att visa funktionerna.
   - **Grid**: genererar ett regelbundet rutnät med en punktsymbol vid varje korsning.
6. Förskjutna symboler placeras på *Displacement lines*. Det minsta avståndet mellan förskjutningslinjerna beror på renderingsprogrammet för punktsymboler, men du kan ändå anpassa vissa inställningar, t.ex. *Stroke width*, *Stroke color* och *Size adjustment* (t.ex. för att öka avståndet mellan de renderade punkterna).
7. Använd gruppalternativen *Labels* för att utföra punktmärkning: etiketterna placeras nära den förskjutna symbolen och inte vid den verkliga positionen för objektet.
   1. Välj *Label-attribut*: ett fält i lagret som ska användas för märkning
   2. Ange *Label font* egenskaper och storlek
   3. Välj en *Etikettens färg*
   4. Ange en *Label distance factor*: för varje punktfunktion, förskjuter etiketten från symbolens centrum proportionellt mot symbolens diagonala storlek.
   5. Aktivera *Use scale dependent labeling* om du vill visa etiketter endast i skalor som är större än en viss *Minimum map scale*.

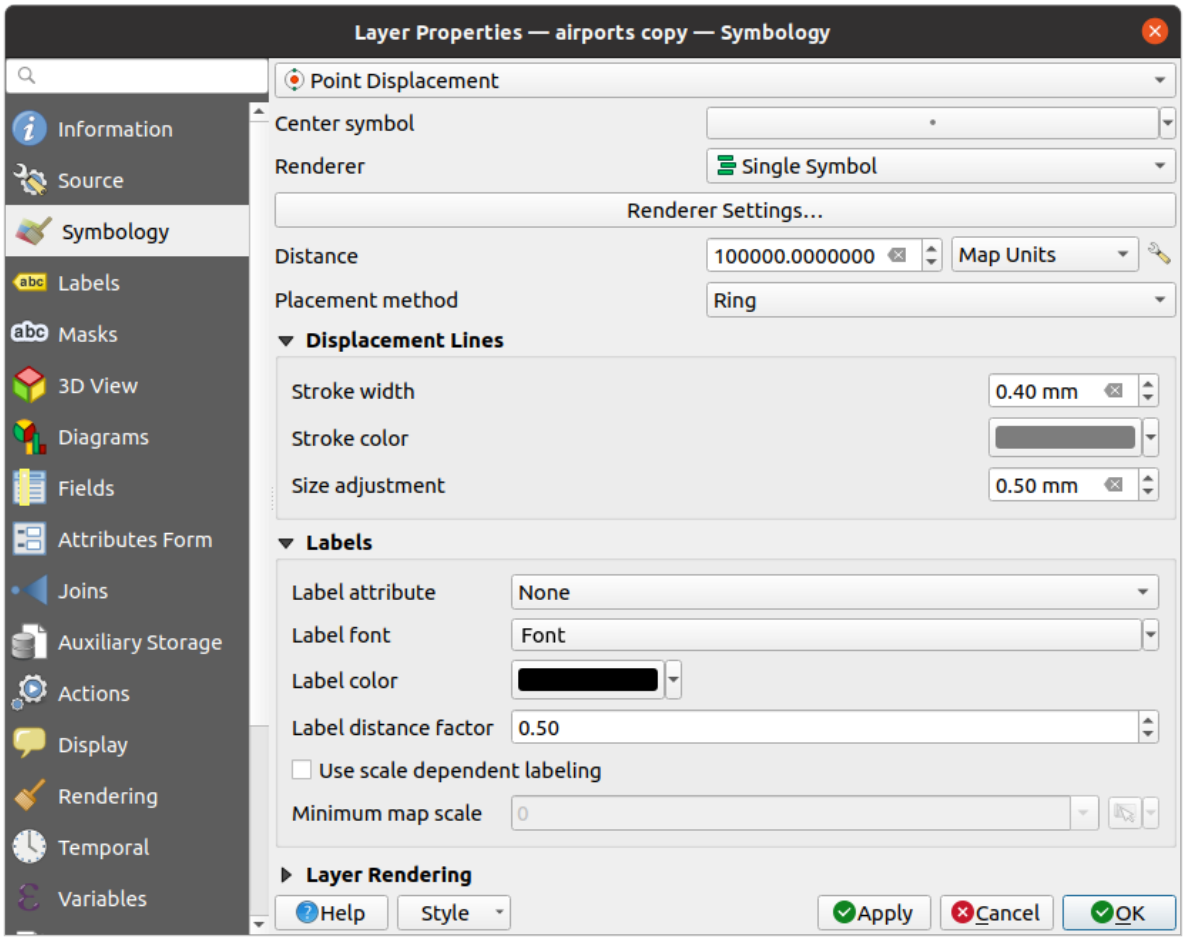


Fig. 12.8: Dialog för punktförskjutning

Observera

Renderingsprogrammet Point Displacement ändrar inte objektets geometri, vilket innebär att punkterna inte flyttas från sin position. De ligger fortfarande kvar på sin ursprungliga plats. Förändringarna är endast visuella, för renderingsändamål. Använd istället Processing *Förskjutning av punkter*-algoritmen om du vill skapa förskjutna funktioner.

Rendering av punktkluster

Till skillnad från renderingsprogrammet *Point Displacement* som förstorar placeringen av de närmaste eller överlagrade punktfunktionerna, grupperar renderingsprogrammet *Point Cluster* närliggande punkter till en enda renderad markörsymbol. Punkter som ligger inom ett angivet avstånd från varandra slås samman till en enda symbol. Aggregering av punkter görs baserat på den närmaste gruppen som bildas, snarare än att bara tilldela dem den första gruppen inom sökavståndet.

Från huvuddialogrutan kan du:

1. Ställ in symbolen för att representera punktklustret i *Cluster symbol*; standardrenderingen visar antalet aggregerade funktioner tack vare `@cluster_size` *variabeln* på Font-markörens symbollager.
2. Välj typ av *Renderer*, dvs. hur du vill klassificera funktioner i lagret (enstaka, kategoriserade, regelbaserade …)

3. Tryck på knappen *Renderingsinställningar…* för att konfigurera objektens symbologi som vanligt. Observera att denna symbologi endast är synlig på objekt som inte är klustrade, eftersom *Cluster symbol* annars används. När alla punktobjekt i ett kluster tillhör samma renderingsklass, och därmed skulle få samma färg, representerar den färgen klustrets variabel `@cluster_color`.
4. Anger det maximala *Distance* som ska beaktas vid klustring av funktioner. Stöder vanliga symbolenheter.

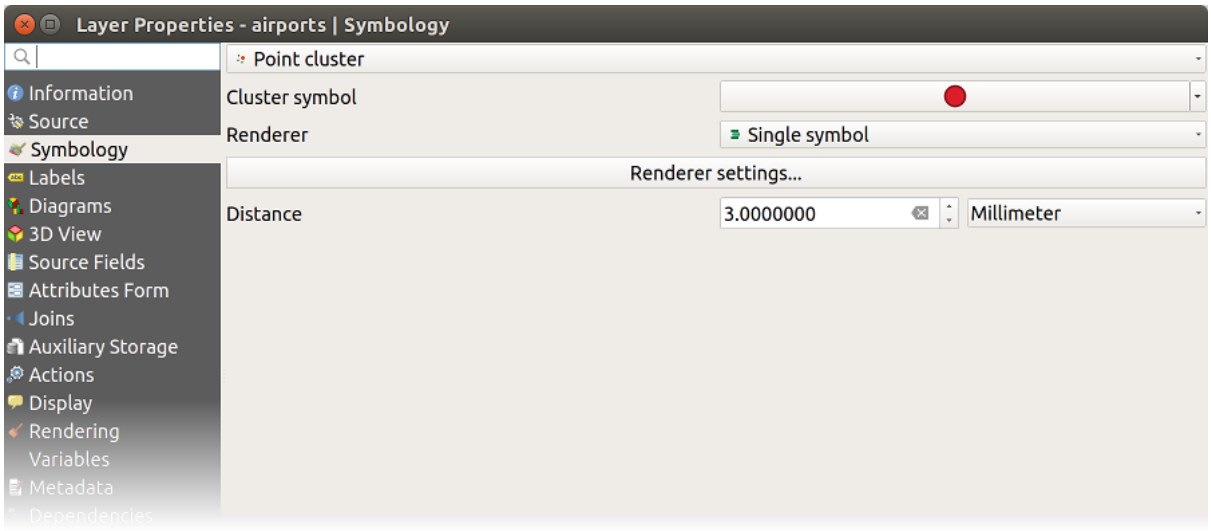


Fig. 12.9: Dialog för punktkluster

Observera

Renderingsprogrammet Point Cluster ändrar inte objektets geometri, vilket innebär att punkterna inte flyttas från sin position. De ligger fortfarande kvar på sin ursprungliga plats. Förändringarna är endast visuella, för renderingsändamål. Använd istället algoritmen Processing *K-means klustring* eller *DBSCAN-klustring* om du vill skapa klusterbaserade funktioner.

Rendering av sammanslagna funktioner

Med renderingsprogrammet *Merged Features* kan områdes- och linjefunktioner ”lösas upp” till ett enda objekt före rendering för att säkerställa att komplexa symboler eller överlappande funktioner representeras av en enhetlig och sammanhängande kartografisk symbol.

Rendering av inverterad polygon

Renderingsprogrammet *Inverted Polygon* gör det möjligt för användaren att definiera en symbol som ska fyllas i utanför lagrets polygoner. Som ovan kan du välja underrenderingar, nämligen Enkel symbol, Graderad, Kategoriserad, Regelbaserad eller 2,5D-rendering.

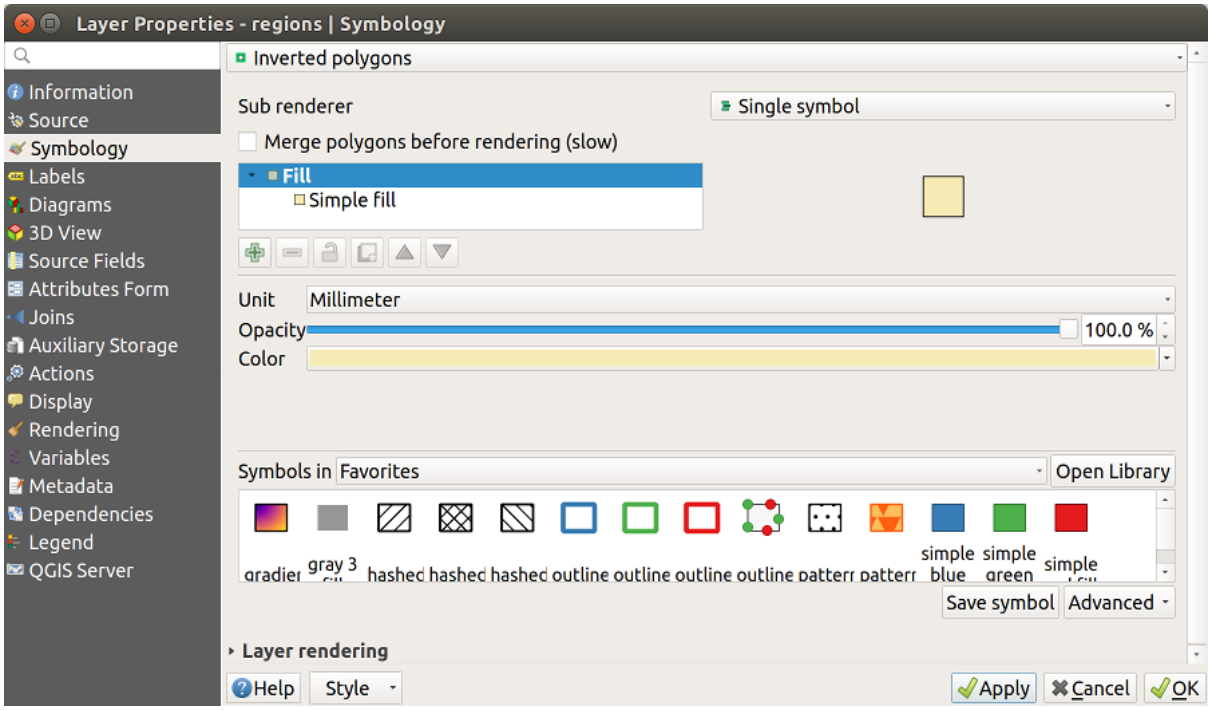


Fig. 12.10: Dialogrutan Inverterad polygon

Rendering av värmekarta

Med renderaren *Heatmap* kan du skapa dynamiska värmekartor i realtid för (fler)punktsskikt. Du kan ange värmekartans *Radius* i millimeter, punkter, pixlar, kartenheter eller tum, välja och redigera en *Color ramp* för värmekartans stil och använda ett reglage för att välja en avvägning mellan renderingshastighet och kvalitet. Du kan också definiera en *Maximum value*-gräns och *Weight points by* med hjälp av ett fält eller ett uttryck.

Använd Data defined override för att dynamiskt styra *Radius* och *Maximum value* baserat på attributen för dina data. Radien för en värmekartspunkt kan t.ex. bestämmas av dess befolkningsattribut, eller så kan det maximala värdet baseras på ett tidsintervall.

När du lägger till eller tar bort en funktion uppdaterar värmekartans renderare värmekartans stil automatiskt. *Color ramp* visas som en legendstapel och i *Legend settings* kan du ställa in *Labels* för värdena *Maximum* och *Minimum*. Du kan också ändra orientering och riktning för förklaringen i *Layout*.

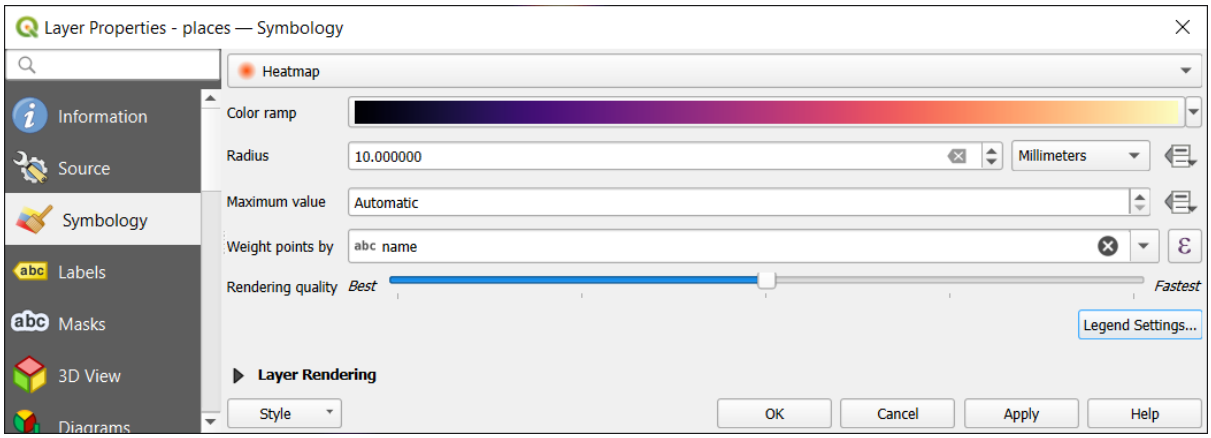


Fig. 12.11: Dialog för värmekarta

2.5D-rendering

Med hjälp av renderingsprogrammet *2.5D* kan du skapa en 2,5D-effekt på lagrets funktioner. Du börjar med att välja ett *Height*-värde (i kartenheter). För det kan du använda ett fast värde, ett av lagrets fält eller ett uttryck. Du måste också välja en *Angle* (i grader) för att återskapa betraktarens position (0 grader betyder väst och växer i takt med klockan). Använd avancerade konfigurationsalternativ för att ställa in *Roof Color* och *Wall Color*. Om du vill simulera solstrålning på väggarna ska du se till att markera alternativet *Shade walls based on aspect*. Du kan också simulera en skugga genom att ange *Color* och *Size* (i kartenheter).

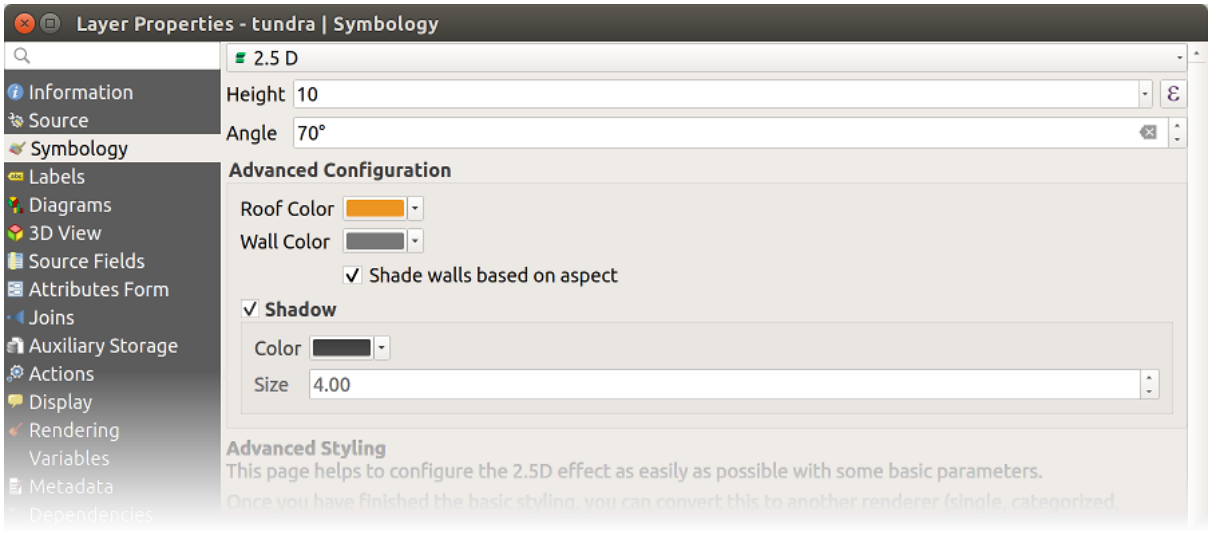


Fig. 12.12: 2.5D-dialog

Tips

Använda 2.5D-effekten med andra renderingsprogram

När du har ställt in den grundläggande stilen för 2,5D-renderaren kan du konvertera den till en annan renderare (enkel, kategoriserad, graderad). 2.5D-effekterna behålls och alla andra renderingsspecifika alternativ finns tillgängliga för finjustering (på så sätt kan du t.ex. ha kategoriserade symboler med en snygg 2.5D-representation eller lägga till lite extra styling till dina 2.5D-symboler). För att se till att skuggan och själva ”byggnaden” inte stör andra funktioner i närheten kan du behöva aktivera symbolenivåer (*Avancerat* ► *Symbolenivåer…*). Värdena för höjd och vinkel i 2,5D sparas i lagrets variabler, så att du kan redigera dem i efterhand på fliken Variabler i lagrets egenskapsdialog.

Inbäddad rendering

Renderingsprogrammet *Embedded Symbols* gör det möjligt att visa den ”inbyggda” symbologin för en angiven datakälla. Detta är oftast fallet med `KML`- och `TAB`-datauppsättningar som har fördefinierad symbologi.

Rendering av lager

På fliken Symbology kan du också ställa in några alternativ som alltid påverkar alla funktioner i lagret:

- *Opacity* : Med det här verktyget kan du göra det underliggande lagret i kartbilden synligt. Använd skjutreglaget för att anpassa synligheten för ditt vektorlager efter dina behov. Du kan också göra en exakt definition av den procentuella synligheten i menyn bredvid skjutreglaget.
- *Blending mode* på nivåerna *Layer* och *Feature*: Med dessa verktyg kan du uppnå speciella renderingseffekter som du kanske tidigare bara kände till från grafikprogram. Pixlarna i de överliggande och underliggande lagren blandas med hjälp av de inställningar som beskrivs i *Blandningslägen*.
- Applicera *paint effects* på alla lagerfunktioner med knappen *Draw Effects*.
- *Control feature rendering order* låter dig, med hjälp av attribut för funktioner, definiera den z-ordning i vilken de ska återges. Aktivera kryssrutan och klicka på knappen bredvid. Du får då upp dialogrutan *Define Order* där du:

  1. Välj ett fält eller skapa ett uttryck som du vill använda på lagerfunktionerna.
  2. Ange i vilken ordning de hämtade funktionerna ska sorteras, dvs. om du väljer **Ascending**-ordning återges funktionerna med lägre värde under dem med högre värde.
  3. Definiera när funktioner som returnerar NULL-värde ska återges: **först** (nederst) eller **sidst** (överst).
  4. Upprepa ovanstående steg så många gånger som du vill använda regler.

  Den första regeln tillämpas på alla funktioner i lagret och z-ordnar dem enligt deras returnerade värde. Sedan, inom varje grupp av funktioner med samma värde (inklusive de med NULL-värde) och därmed samma z-nivå, tillämpas nästa regel för att sortera dem. Och så vidare…

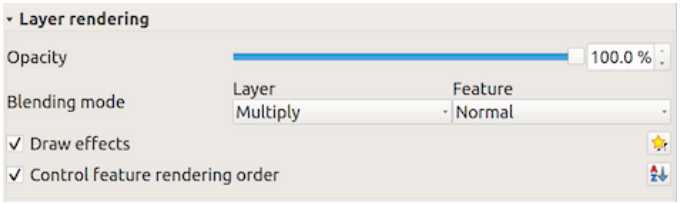


Fig. 12.13: Alternativ för rendering av lager

Andra inställningar

Symbolnivåer

För renderingsprogram som tillåter staplade symbollager (endast heatmap gör inte det) finns det ett alternativ för att styra renderingsordningen för varje symbols nivåer.

För de flesta renderingsmaskinerna kan du komma åt alternativet Symbols levels genom att klicka på knappen *Advanced* under listan med sparade symboler och välja *Symbol levels*. För *Regelbaserad rendering* är alternativet direkt tillgängligt via knappen *Symbols Levels…*, medan samma knapp finns i dialogrutan *Rendering settings* för renderingsprogrammet *Punktförskjutning Renderare*.

För att aktivera symbolnivåer, markera kryssrutan *Enable symbol levels*. Varje rad visar ett litet urval av den kombinerade symbolen, dess etikett och det individuella symbollagret uppdelat i kolumner med ett nummer bredvid.

Siffrorna representerar den nivå i renderingsordningen som symbollagret kommer att ritas i. Nivåer med lägre värden ritas först och ligger längst ned, medan nivåer med högre värden ritas sist och ligger ovanpå de andra.

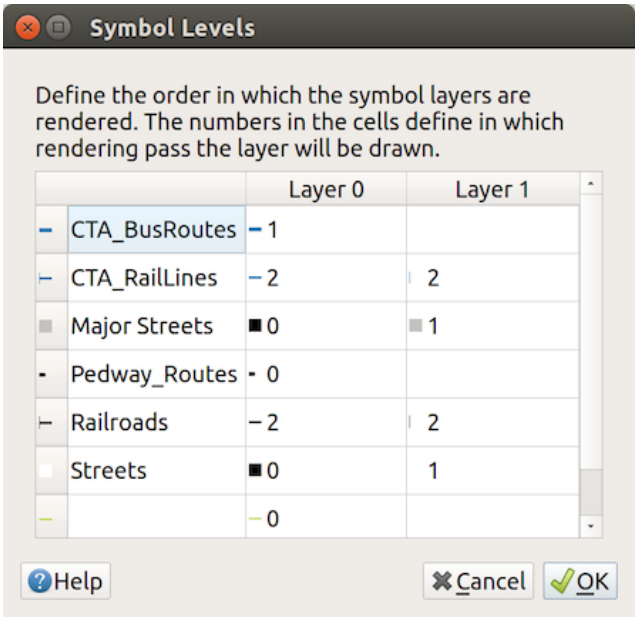


Fig. 12.14: Dialog för symbolnivåer

Observera

Om symbolnivåerna är avaktiverade kommer de fullständiga symbolerna att ritas i enlighet med deras respektive funktionsordning. Överlappande symboler kommer helt enkelt att döljas av andra symboler nedanför. Dessutom kommer liknande symboler inte att ”smälta samman” med varandra.

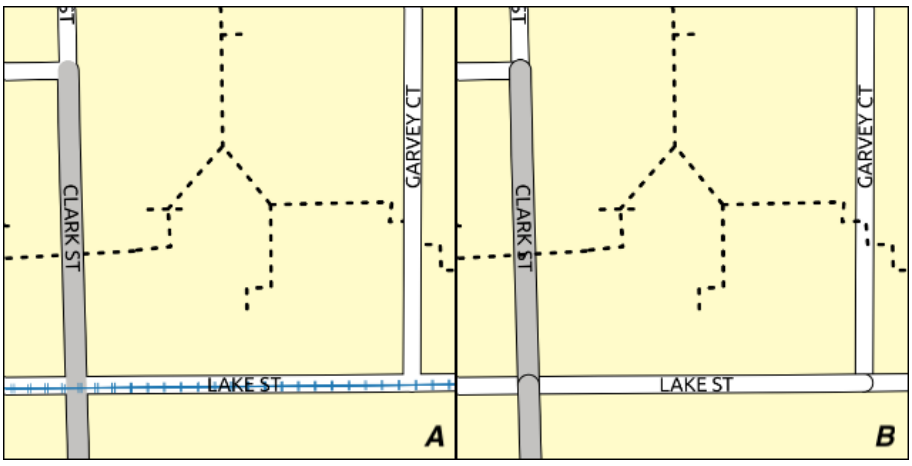


Fig. 12.15: Skillnad mellan aktiverade (A) och inaktiverade (B) symbolnivåer

Datadefinierad storleksförklaring

När ett lager återges med *proportionell symbol eller multivariat återgivning* eller när ett *diagram med skalad storlek* tillämpas på lagret, kan du tillåta visning av de skalade symbolerna i både *Layers-panelen* och *print layout förklaring*.

För att aktivera dialogrutan *Data-defined Size Legend* för att rendera symbologi, välj alternativet eponym i knappen *Advanced* under listan med sparade symboler. För diagram finns alternativet tillgängligt under fliken *Legend*. Dialogrutan innehåller följande alternativ för att:

- välj typ av teckenförklaring: *Legend not enabled*, *Separerade legendobjekt* och *Kollapsad förklaring*. För det sistnämnda alternativet kan du välja om legendobjekten ska vara inriktade på **Botten** eller på **Centrum**;
- förhandsgranska *symbolen som ska användas* för legendrepresentation;
- infoga titeln i teckenförklaringen;
- ändra storlek på de klasser som ska användas: som standard ger QGIS dig en förklaring med fem klasser (baserat på naturliga vackra avbrott) men du kan använda din egen klassificering med alternativet *Manual size classes*. Använd knapparna och för att ställa in dina egna klassvärden och etiketter.
- För kollapsad förklaring är det möjligt att:
  - *Align symbols* i mitten eller längst ner
  - konfigurera den horisontella ledaren *Line symbol* från symbolen till motsvarande legendtext.

En förhandsgranskning av förklaringen visas i dialogrutans högra panel och uppdateras när du ställer in parametrarna.

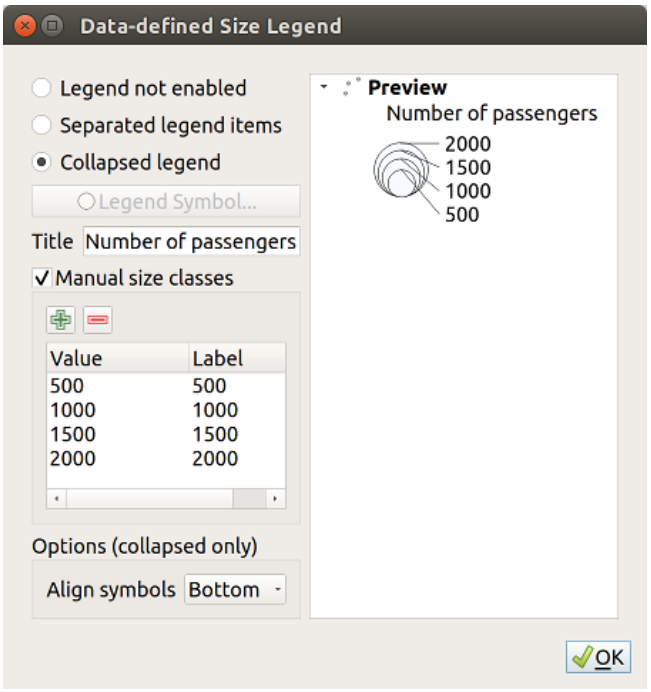


Fig. 12.16: Inställning av storlek skalad förklaring

Observera

För närvarande kan den datadefinierade storlekslegenden för lagersymbologi endast tillämpas på punktlager som använder enkel, kategoriserad eller graderad symbologi.

Inställningar för animering

För att låta en symbol bli en *animerad symbol* kan du använda panelen *Animation settings*. I den här panelen kan du aktivera animering för symbolen och ställa in en specifik bildfrekvens för symbolens omritning.

1. Börja med att gå till den översta symbolnivån och välj *Avancerad*-menyn längst ned till höger i dialogrutan
2. Hitta alternativet *Animationsinställningar*
3. Markera *Is Animated* för att aktivera animering för symbolen
4. Konfigurera *Frame rate*, d.v.s. hur snabbt animationen ska spelas upp
5. Du kan nu använda variabeln `@symbol_frame` i alla egenskaper som definieras av subsymboldata för att animera den egenskapen.

Om du t.ex. ställer in symbolens rotation till det datadefinierade uttrycket `@symbol_frame % 360` kommer symbolen att rotera över tid, med en rotationshastighet som bestäms av symbolens bildfrekvens:

Dra effekter

För att förbättra renderingen av lager och undvika (eller åtminstone minska) användningen av annan programvara för slutlig rendering av kartor, erbjuder QGIS en annan kraftfull funktion: alternativen *Draw Effects*, som lägger till färgeffekter för att anpassa visualiseringen av vektorlager.

Alternativet finns tillgängligt i dialogrutan *Lageregenskaper* ► *Symbologi*, under gruppen *Lagerrendering* (gäller hela lagret) eller i *symbol layer properties* (gäller motsvarande funktioner). Du kan kombinera båda användningarna.

Färgeffekter kan aktiveras genom att markera alternativet *Rita effekter* och klicka på knappen Customize effects. Då öppnas dialogrutan *Effektegenskaper* (se Fig. 12.17). Följande effekttyper, med anpassade alternativ finns tillgängliga:

- **Källa**: Ritar objektets ursprungliga stil enligt konfigurationen av lagrets egenskaper. Stilens *Opacity* kan justeras liksom *Blend mode* och *Draw mode*. Dessa är gemensamma egenskaper för alla typer av effekter.

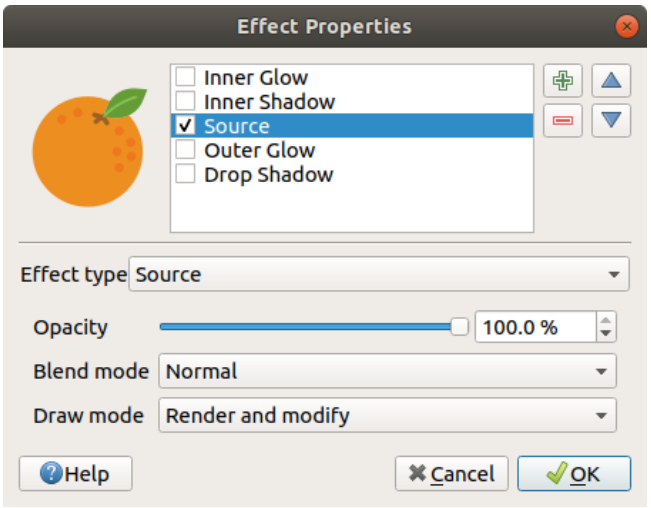


Fig. 12.17: Rita effekter: Dialogrutan Källa

- **Oskärpa**: Lägger till en oskärpeeffekt på vektorlagret. De anpassade alternativ som du kan ändra är *Blur type* (*Stack blur (fast)* eller *Gaussian blur (quality)*) och *Blur strength*.

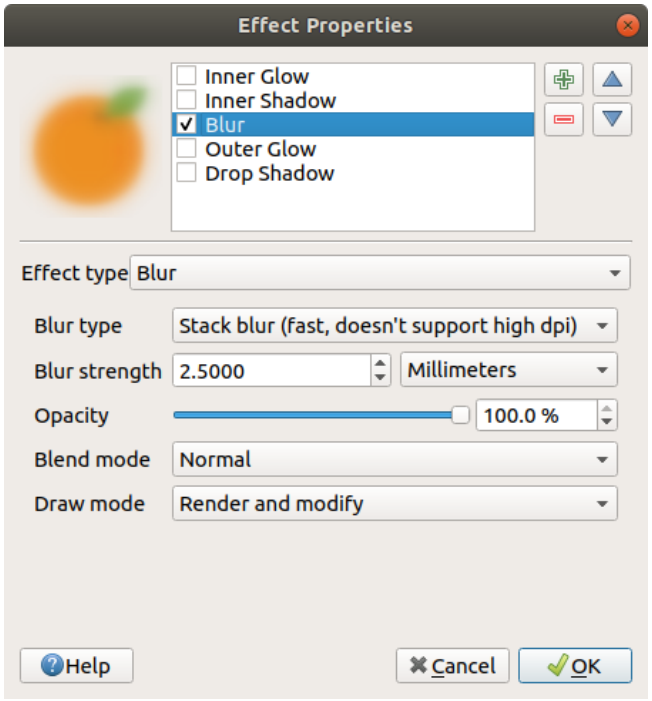


Fig. 12.18: Rita effekter: Dialogrutan Oskärpa

- **Colorise**: Denna effekt kan användas för att skapa en version av stilen med en enda nyans. Basen kommer alltid att vara en gråskaleversion av symbolen och du kan:
  - Använd *Gråskala* för att välja hur den ska skapas: alternativen är ”By lightness”, ”By luminosity”, ”By average” och ”Off”.
  - Om *Colorise* är markerad kommer det att vara möjligt att blanda en annan färg och välja hur stark den ska vara.
  - Styr nivåerna *Brightness*, *Contrast* och *Saturation* för den resulterande symbolen.

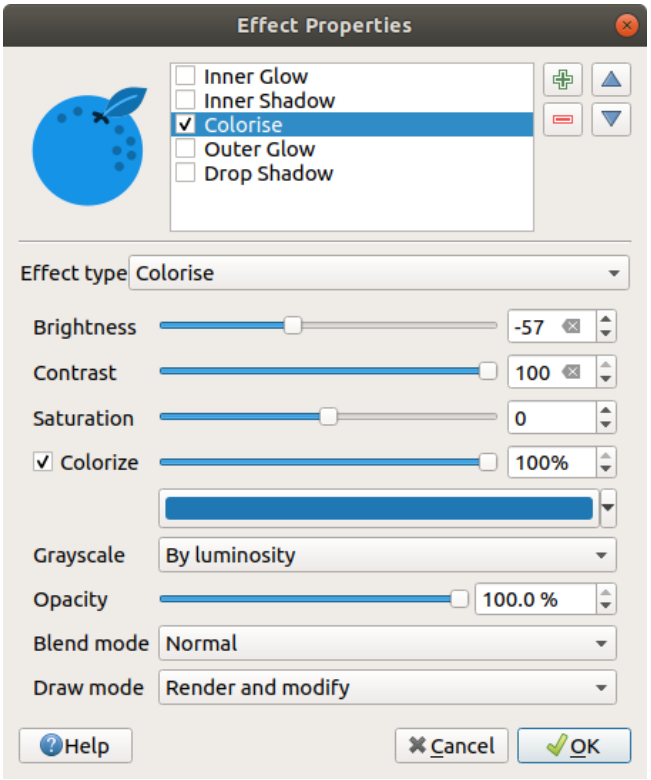


Fig. 12.19: Rita effekter: Dialogrutan Colorize

- **Släppskugga**: Med den här effekten läggs en skugga på objektet, vilket ser ut som att lägga till en extra dimension. Denna effekt kan anpassas genom att ändra vinkeln och avståndet för *Förskjutning*, vilket avgör vart skuggan flyttas och närheten till källobjektet. *Släppskugga* har också möjlighet att ändra *Oskärperadie* och *Färg* för effekten.

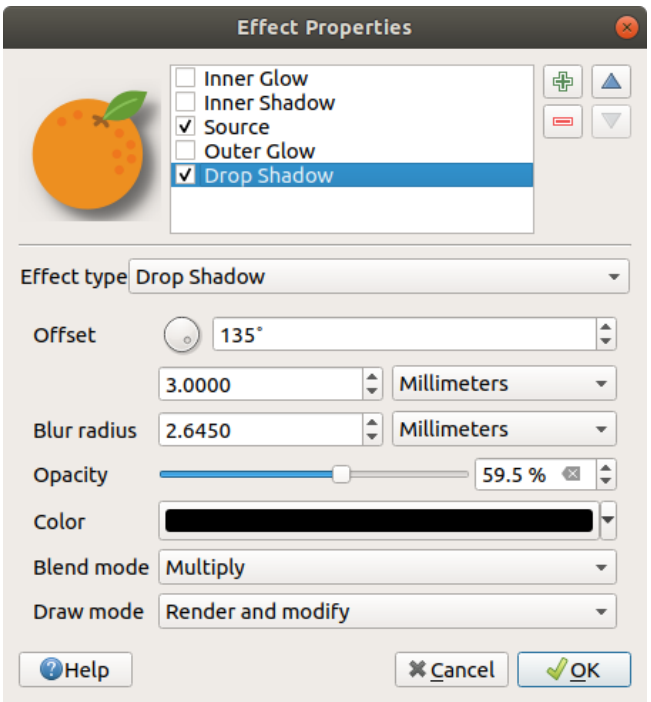


Fig. 12.20: Rita effekter: Dialogrutan Drop Shadow

- **Innerskugga**: Denna effekt liknar *Drop Shadow*-effekten, men den lägger till skuggeffekten på insidan av funktionens kanter. De tillgängliga alternativen för anpassning är desamma som för effekten *Drop Shadow*.

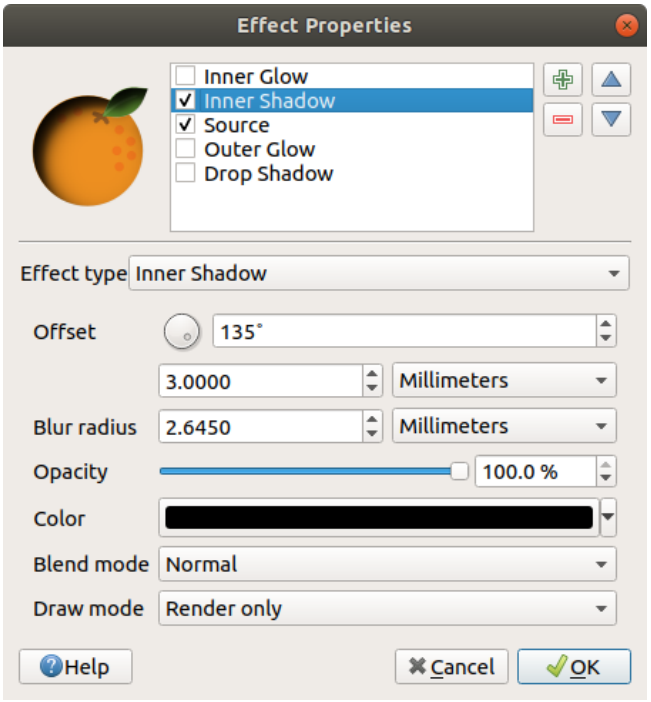


Fig. 12.21: Rita effekter: Dialog för inre skugga

- **Inner Glow**: Lägger till en glödande effekt inuti funktionen. Denna effekt kan anpassas genom att justera *Spread* (bredden) på glöden eller *Blur radius*. Den sistnämnda anger avståndet från objektets kant där du vill att oskärpa ska uppstå. Dessutom finns det alternativ för att anpassa färgen på glöden med hjälp av en *Single color* eller en *Color ramp*.

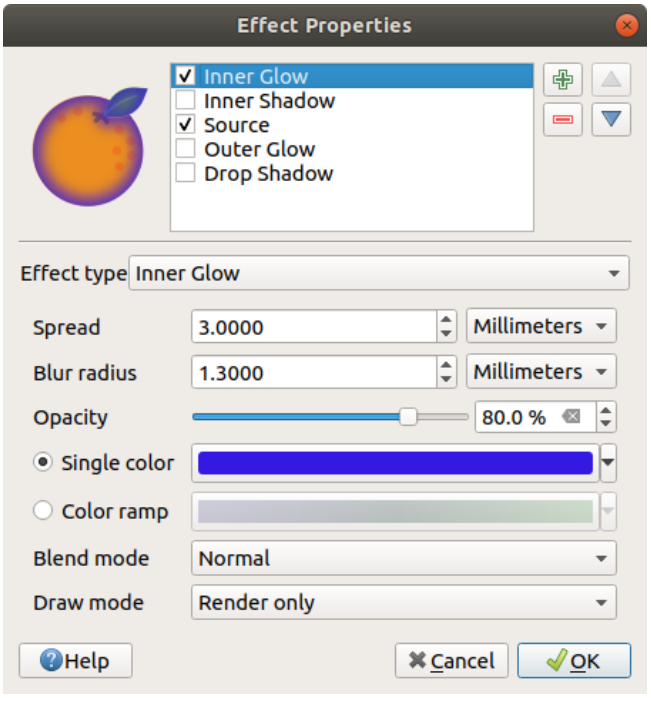


Fig. 12.22: Rita effekter: Dialog Inre glöd

- **Outer Glow**: Denna effekt liknar *Inner Glow*-effekten, men den lägger till glödande effekt på utsidan av funktionens kanter. De tillgängliga alternativen för anpassning är desamma som för *Inner Glow*-effekten.

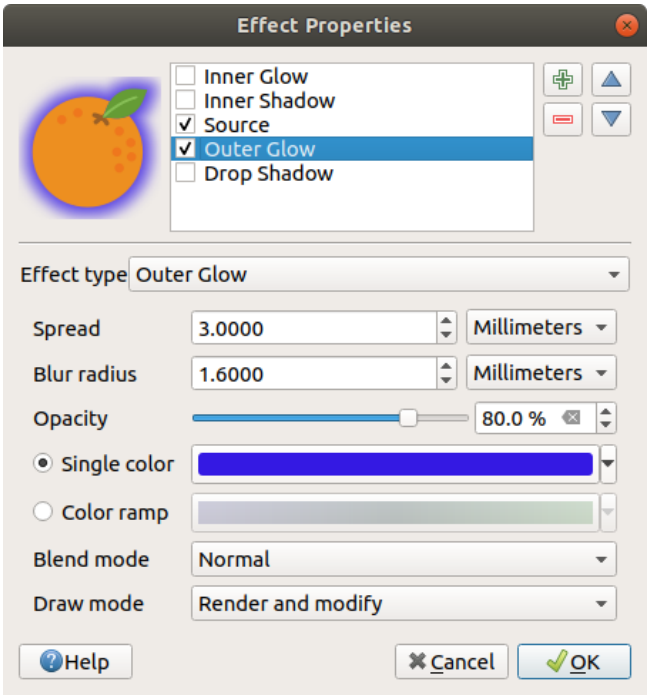


Fig. 12.23: Rita effekter: Dialogrutan Yttre glöd

- **Transform**: Tillför möjligheten att transformera symbolens form. De första alternativen som är tillgängliga för anpassning är *Reflect horizontal* och *Reflect vertical*, som faktiskt skapar en reflektion på de horisontella och/eller vertikala axlarna. De andra alternativen är:
  - *Shear X,Y*: Snedställer objektet längs X- och/eller Y-axeln.
  - *Skala X,Y*: Förstorar eller förminskar objektet längs X- och/eller Y-axeln med den angivna procentsatsen.
  - *Rotation*: Vrider objektet runt dess mittpunkt.
  - och *Translate X,Y* ändrar objektets position baserat på ett avstånd som anges på X- och/eller Y-axeln.

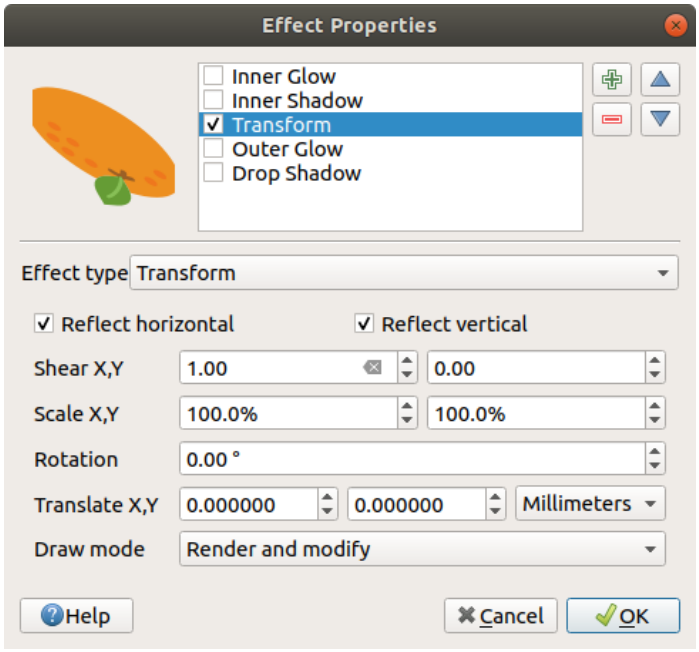


Fig. 12.24: Rita effekter: Dialogrutan Transformera

En eller flera effekttyper kan användas samtidigt. Du (av)aktiverar en effekt med hjälp av dess kryssruta i effektlistan. Du kan ändra den valda effekttypen med hjälp av alternativet *Effekttyp*. Du kan ändra ordning på effekterna med hjälp av knapparna Flytta upp och Flytta ner, och även lägga till/ta bort effekter med hjälp av knapparna Lägg till ny effekt och Ta bort effekt.

Det finns några gemensamma alternativ för alla typer av ritteffekter. Alternativen *Opacity* och *Blend mode* fungerar på samma sätt som de som beskrivs i *Rendering av lager* och kan användas i alla ritteffekter utom transform.

Det finns också ett alternativ (*Draw mode*) för varje effekt, och du kan välja om symbolen ska renderas och/eller modifieras enligt vissa regler:

- Effekter renderas från topp till botten.
- *Render only*-läget innebär att effekten kommer att vara synlig.
- *Modifier only*-läget innebär att effekten inte kommer att synas men att de ändringar som den tillämpar kommer att överföras till nästa effekt (den som ligger omedelbart under).
- Läget *Render and Modify* gör effekten synlig och skickar eventuella ändringar till nästa effekt. Om effekten är högst upp i effektlistan eller om effekten omedelbart ovanför inte är i modifieringsläge, kommer den att använda den ursprungliga källsymbolen från lageregenskaperna (på samma sätt som källa).

12.1.4 Etiketter Egenskaper

Egenskapen *Labels* ger dig alla nödvändiga och lämpliga funktioner för att konfigurera smart märkning på vektorlager. Den här dialogrutan kan också nås från panelen *Layer Styling* eller med knappen Layer Labeling Options i verktygsfältet **Labels**.

Längst upp i dialogrutan har du:

- en kombinationsruta för att välja lämplig märkningsmetod för det aktiva lagret
- knappen Konfigurera projekt labeling rules: hjälper dig att styra interaktioner mellan etiketter och funktioner över lagren i projektet. Mer information finns på *Konfigurera regler för projektmärkning*.

- knappen Automatiserade placeringsinställningar (gäller alla lager): konfigurera allmänna egenskaper för etikettplacering och konfliktlösning. Mer information finns på *Inställning av den automatiska placeringsmotorn*.

Det första steget är att välja märkningsmetod från rullgardinsmenyn. Tillgängliga metoder är:

- *No labels*: standardvärdet, visar inga etiketter från lagret
- *Enkela etiketter*: Visa etiketter på kartan med hjälp av ett enda attribut eller ett uttryck
- *Regelbaserad märkning*
- och *Blocking*: gör det möjligt att ställa in ett lager som bara ett hinder för andra lagers etiketter utan att rendera några egna etiketter.

I nästa steg antar vi att du väljer alternativet *Enkela etiketter*, vilket öppnar följande dialogruta.

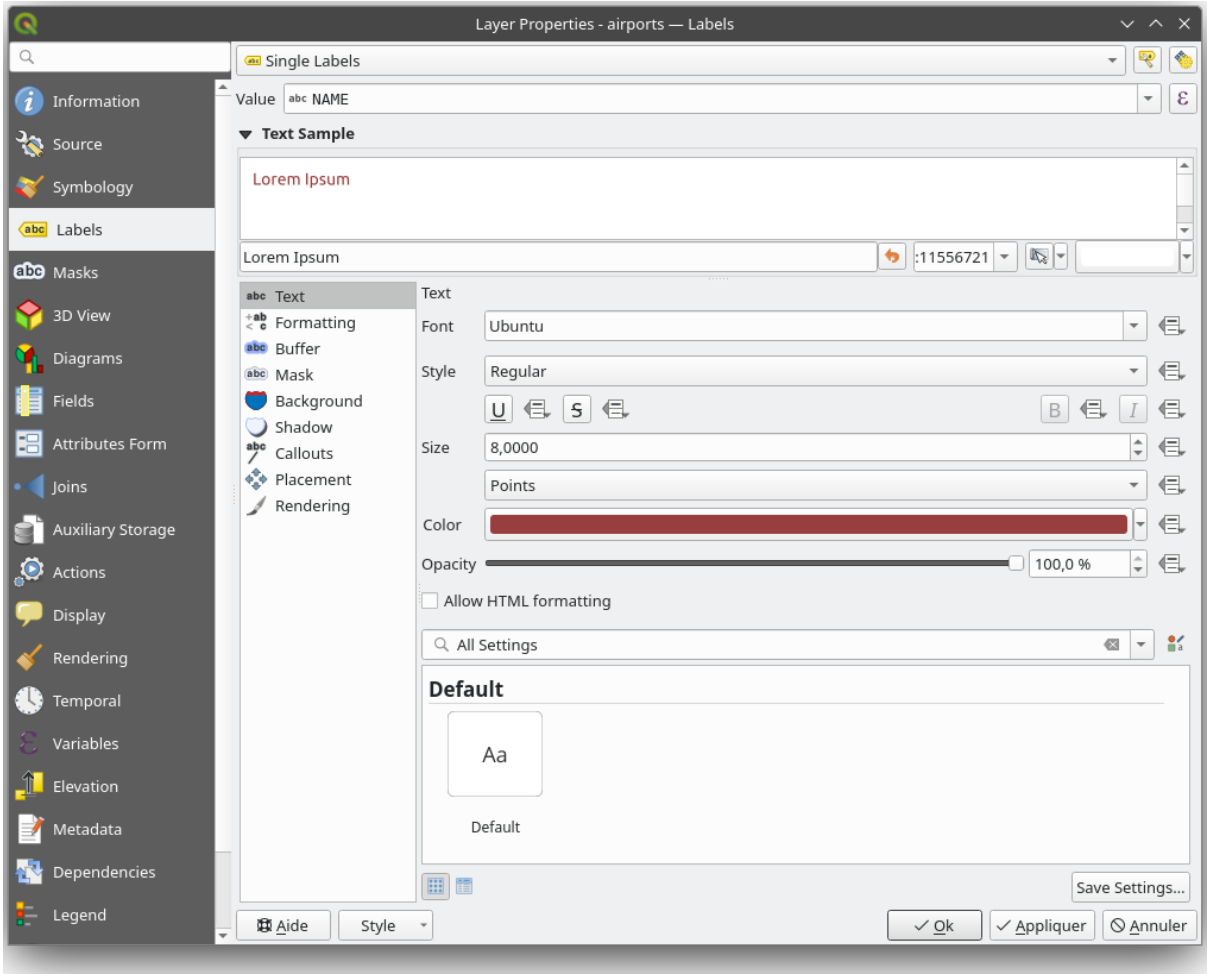


Fig. 12.25: Inställningar för lageretikettering - Enstaka etiketter

Längst upp i dialogrutan finns en rullgardinslista med *Value*. Du kan välja en attributkolumn som ska användas för märkning. Som standard används *display field*. Klicka på ε om du vill definiera etiketter baserade på uttryck - se *Definiera etiketter baserat på uttryck*.

Observera

Etiketter med deras formatering kan visas som poster i legenderna om detta är aktiverat på fliken *Legend*.

Nedan visas alternativ för att anpassa etiketterna, under olika flikar:

- abc *Text*
- *Formatering*
- abc *Buffert*
- abc *Mask*
- *Bakgrund*
- *Skugga*
- abc *Callouts*
- *Placement*
- *Rendering*

Beskrivning av hur man ställer in varje egenskap finns på *Ställa in en etikett*.

Konfigurera regler för projektmärkning

Genom att trycka på knappen Configure project labeling rules bredvid listrutan labeling method kan du skapa regler som styr hur etiketter från ett lager kan interagera med etiketter eller funktioner från ett annat lager.

1. Tryck på knappen Add rule och välj en av regeltyperna i rullgardinsmenyn:
   - *Prevent labels overlapping features*: förhindrar att etiketter placeras så att de överlappar funktioner från ett annat lager.
   - *Pull labels towards features*: förhindrar att etiketter placeras för långt från funktioner från ett annat lager. Det maximala avståndet kan ställas in i den enhet som du väljer.
   - *Push labels away from features*: förhindrar att etiketter placeras för nära funktioner från ett annat lager. Minimiavståndet kan ställas in i valfri enhet, liksom *regelns prioritet* (de högst prioriterade reglerna är viktigare att respektera i händelse av en konflikt om etikettplacering).
   - *Push labels away from other labels*: förhindrar att etiketter placeras för nära etiketter från ett annat lager.

   Uppmärksamma

   Eftersom de tre sista kräver en build baserad på GEOS >= 3.10, kanske de inte finns tillgängliga i din QGIS-installation beroende på vilken underliggande GEOS-version som används.

2. Fyll i egenskaperna efter eget tycke och smak; du kan ge regeln ett mer meningsfullt namn.
3. Tryck på *OK*.
4. Lägg till så många regler som behövs.
5. Tryck vid behov på Edit rule för att ändra den valda regeln eller på Remove rule för att ta bort den från projektet.

De inställda reglerna är tillgängliga från alla lager *Labels* egenskapsflik, genom att trycka på knappen Configure project labeling rules. Du kan tillfälligt aktivera eller inaktivera någon av dem genom att använda kryssrutan bredvid namnet. Håll muspekaren över en regel för att förhandsgranska dess detaljer.

Inställning av den automatiska placeringsmotorn

Du kan använda inställningarna för automatiserad placering för att konfigurera ett automatiserat beteende för etiketterna på projektnivå. I det övre högra hörnet på fliken *Labels* klickar du på knappen Automatiserade placeringsinställningar (gäller alla lager), vilket öppnar en dialogruta med följande alternativ:

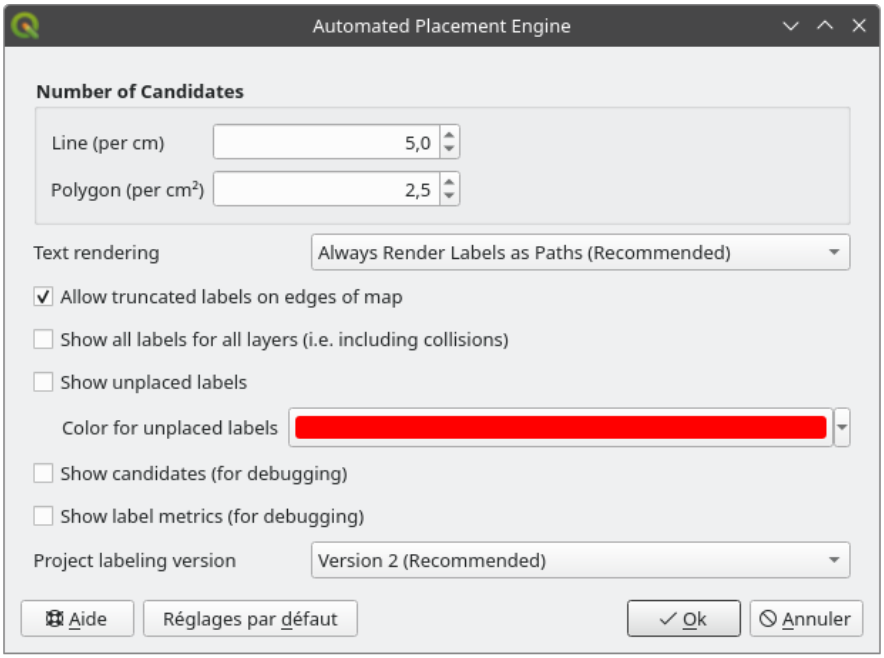


Fig. 12.26: Etiketternas automatiserade placeringsmotor

- *Number of candidates*: beräknar och tilldelar linje- och polygonelement antalet möjliga etikettplaceringar baserat på deras storlek. Ju längre eller bredare en funktion är, desto fler kandidater har den, och dess etiketter kan placeras bättre med mindre risk för kollision.
- *Textrendering*: anger standardvärdet för widgetar för etikettrendering när *exportera en kartkanvas* eller *en layout* till PDF eller SVG. Om *Alltid rendera etiketter som text* är valt kan etiketter redigeras i externa program (t.ex. Inkscape) som vanlig text. MEN bieffekten är att renderingskvaliteten försämras och att det uppstår problem med renderingen när vissa textinställningar, t.ex. buffertar, används. Det är därför *Alltid rendera etiketter som sökvägar (rekommenderas)* som exporterar etiketter som konturer men garanterar fullständig kompatibilitet med alla tillgängliga formateringsalternativ, rekommenderas. Med alternativet *Föredra rendering av etiketter som text* renderas etiketter som textobjekt, såvida inte detta resulterar i renderingsartefakter eller dålig kvalitet (beroende på inställningarna för textformat).

> **Observera**
>
> Vid rendering av etiketter som text till en vektorbaserad enhet (t.ex. PDF eller SVG) måste man se till att de nödvändiga teckensnitten är tillgängliga för användarna när de öppnar de skapade filerna, annars kommer standardfallback-teckensnitt att användas för att visa utdata istället. (Även om PDF-export KAN automatiskt bädda in vissa teckensnitt när det är möjligt, beroende på användarens plattform).

- *Allow truncated labels on edges of map*: styr om etiketter som delvis faller utanför kartans utbredning ska återges. Om det är markerat visas dessa etiketter (när det inte finns något sätt att placera dem helt inom det synliga området). Om den inte är markerad kommer delvis synliga etiketter att hoppas över. Observera att den här inställningen inte har någon effekt på hur etiketter visas i *layout map item*.
- *Visa alla etiketter för alla lager (dvs. inklusive kolliderande objekt)*. Observera att detta alternativ även kan ställas in per lager (se *Fliken Rendering*)

- *Visa ej placerade etiketter*: gör det möjligt att avgöra om några viktiga etiketter saknas på kartorna (t.ex. på grund av överlappningar eller andra begränsningar). De visas med en anpassningsbar färg.
- *Visa kandidater (för felsökning)*: styr om rutor ska ritas på kartan som visar alla kandidater som genererats för etikettplacering. Som etiketten säger är det endast användbart för felsökning och för att testa vilken effekt olika inställningar för märkning har. Detta kan vara praktiskt för en bättre manuell placering med verktyg från *verktygsfältet för etiketter*.
- *Visa etikettmätvärden (för felsökning)*: visar etikettens textgränser i rött och baslinjerna i blått
- *Projektets version av etikettering*: QGIS stöder två olika versioner av automatisk placering av etiketter:
  - *Version 1*: det gamla systemet (används av QGIS version 3.10 och tidigare, och när du öppnar projekt som skapats i dessa versioner i QGIS 3.12 eller senare). I version 1 är prioriteringarna för etiketter och hinder endast ”grova riktlinjer”, och det är möjligt att en etikett med låg prioritet placeras över ett hinder med hög prioritet i den här versionen. Det kan därför vara svårt att uppnå önskat märkningsresultat med den här versionen och den rekommenderas därför endast för kompatibilitet med äldre projekt.
  - *Version 2 (rekommenderas)*: detta är standardsystemet i nya projekt som skapas i QGIS 3.12 eller senare. I version 2 har logiken för när etiketter får överlappa *obstacles* omarbetats. Den nyare logiken förbjuder alla etiketter från att överlappa alla hinder med en större hindervikt jämfört med etikettens prioritet. Som ett resultat ger den här versionen mycket mer förutsägbara och mer lättförståeliga märkningsresultat.

Regelbaserad märkning

Med regelbaserad märkning kan flera märkningskonfigurationer definieras och tillämpas selektivt på grundval av uttrycksfilter och skalintervall, som i *Regelbaserad rendering*.

Så här skapar du en regel:

1. Välj alternativet **Regelbaserad märkning** i huvudrullgardinsmenyn på fliken *Etiketter*
2. Klicka på knappen Lägg till regel längst ned i dialogrutan.
3. Fyll den nya dialogen med:
   - *Description*: en text som används för att identifiera regeln i fliken *Labels* och som en *label förklaring entry* i utskriftslayoutens förklaring
   - *Filter*: ett uttryck för att välja de funktioner som etikettinställningarna ska tillämpas på
   - Om det redan finns regler kan alternativet *Else* användas för att välja alla funktioner som inte matchar något filter i reglerna i samma grupp.
4. Du kan ange ett *skalintervall* inom vilket etikettregeln ska tillämpas.
5. De alternativ som finns tillgängliga i gruppboxen *Etiketter* är de vanliga *etikettinställningar*. Konfigurera dem och tryck på *OK*.

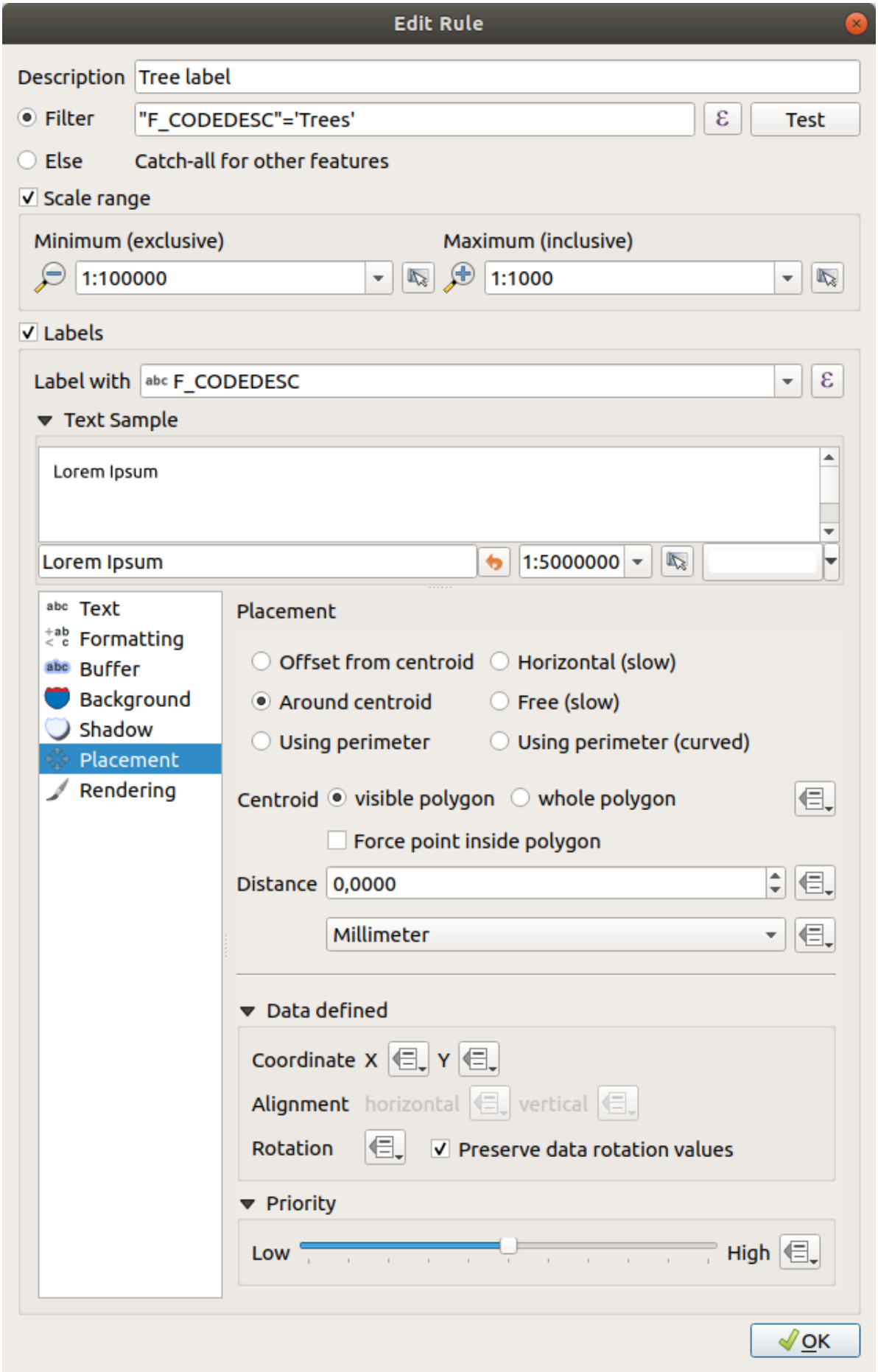


Fig. 12.27: Rule inställningar

En sammanfattning av befintliga regler visas i huvuddialogen (se Fig. 12.28). Du kan lägga till flera regler, ordna om eller sammanfoga dem med dra-och-släpp. Du kan också ta bort dem med knappen eller redigera dem med knappen eller genom att dubbelklicka.

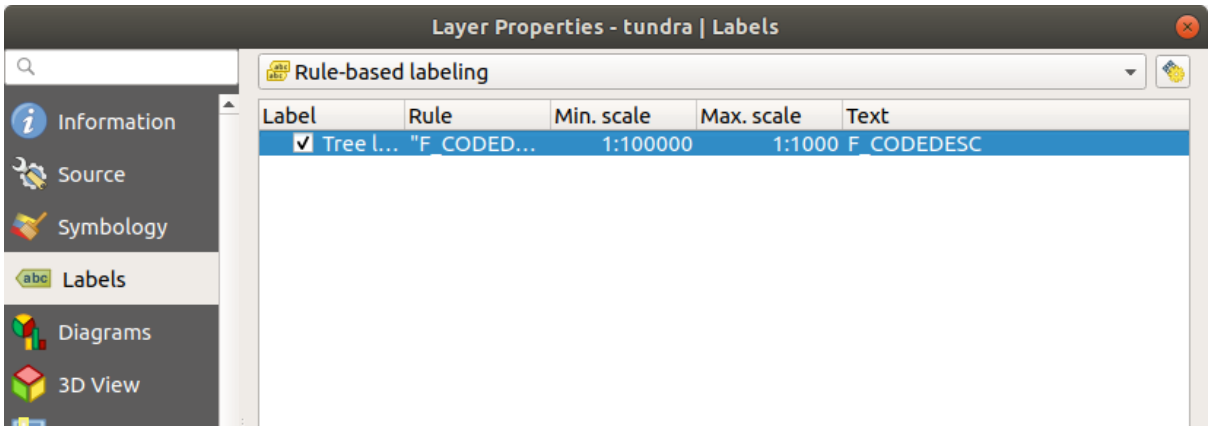


Fig. 12.28: Regelbaserad panel för märkning

Definiera etiketter baserat på uttryck

Oavsett om du väljer enkel eller regelbaserad märkningstyp kan du använda uttryck för att märka objekt i QGIS.

Om du använder metoden *Single labels* klickar du på knappen nära listrutan *Value* i fliken *Labels* i dialogrutan Egenskaper.

I Fig. 12.29 ser du ett exempeluttryck för att märka lagret alaska trees med trädtyp och area, baserat på fältet ”VEGDESC”, lite beskrivande text och funktionen `$area` i kombination med `format_number()` för att få det att se snyggare ut.

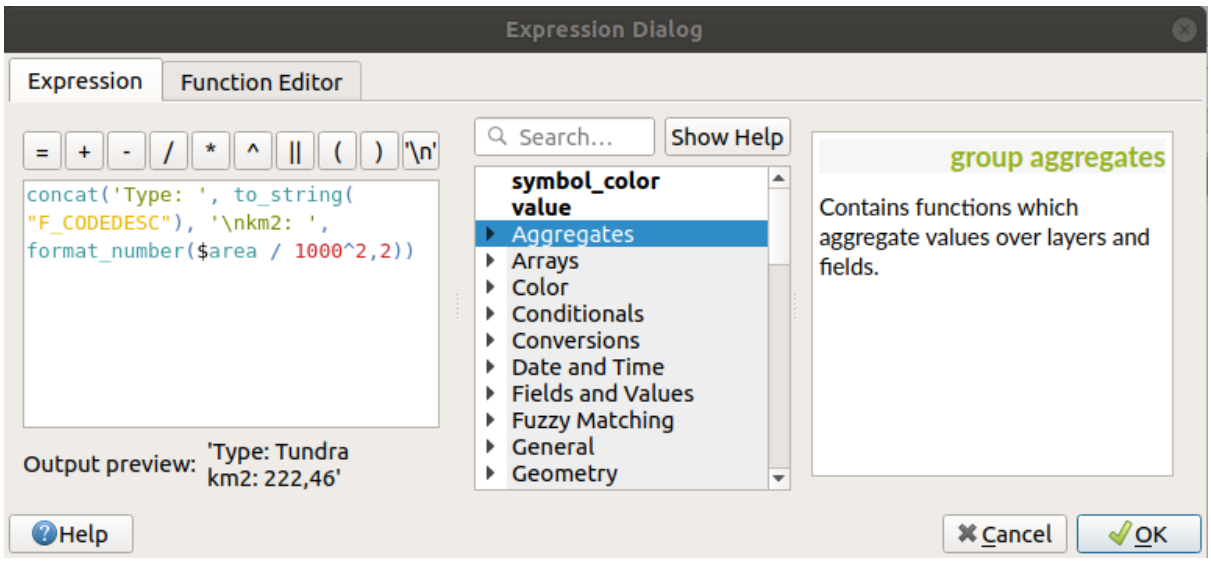


Fig. 12.29: Använda uttryck för märkning

Expressionsbaserad märkning är lätt att arbeta med. Allt du behöver ta hand om är det:

- Du kan behöva kombinera alla element (strängar, fält och funktioner) med en strängkonkateneringsfunktion som `concat`, `+` eller `||`. Tänk på att i vissa situationer (när null eller numeriska värden är inblandade) kommer

inte alla dessa verktyg att passa dina behov.

- Strängar skrivs med ”enkla citattecken”.
- Fält skrivs inom ”dubbla citattecken” eller utan citattecken.

Låt oss ta en titt på några exempel:

1. Etikett baserad på två fält ’namn’ och ’plats’ med ett kommatecken som separator:

```
"name" || ', ' || "place"
```

Returnerar:

```
John Smith, Paris
```

2. Etikett baserad på två fält ’namn’ och ’plats’ med andra texter:

```
'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")
```

Returnerar:

```
My name is John Smith and I live in Paris
```

3. Etikett baserad på två fält ”namn” och ”plats” med andra texter som kombinerar olika konkateneringsfunktioner:

```
concat('My name is ', name, ' and I live in ' || place)
```

Returnerar:

```
My name is John Smith and I live in Paris
```

Eller, om fältet ”place” är NULL, returneras:

```
My name is John Smith
```

4. Etikett med flera rader baserad på två fält ”namn” och ”plats” med en beskrivande text:

```
concat('My name is ', "name", '\n' , 'I live in ' , "place")
```

Returnerar:

```
My name is John Smith
I live in Paris
```

5. Etikett baserad på ett fält och funktionen $area för att visa platsens namn och dess avrundade ytstorlek i en konverterad enhet:

```
'The area of ' || "place" || ' has a size of '
|| round($area/10000) || ' ha'
```

Returnerar:

```
The area of Paris has a size of 10500 ha
```

6. Skapa ett CASE ELSE-villkor. Om befolkningsvärdet i fältet *population* är <= 50000 är det en stad, annars är det en tätort:

```
concat('This place is a ',
CASE WHEN "population" <= 50000 THEN 'town' ELSE 'city' END)
```

Returnerar:

```
This place is a town
```

7. Visningsnamn för städerna och ingen etikett för de andra funktionerna (för kontexten ”stad”, se exemplet ovan):

```
CASE WHEN "population" > 50000 THEN "NAME" END
```

Returnerar:

```
Paris
```

Som du kan se i uttrycksbyggaren har du hundratals funktioner tillgängliga för att skapa enkla och mycket komplexa uttryck för att märka dina data i QGIS. Se kapitlet *Uttryck* för mer information och exempel på uttryck.

Använda datadefinierad åsidosättning för märkning

Med funktionen Data defined override åsidosätts inställningarna för märkningen av poster i attributtabellen eller uttryck som baseras på dem. Denna funktion kan användas för att ställa in värden för de flesta av de märkningsalternativ som beskrivs ovan.

Låt oss till exempel använda Alaska QGIS exempeldataset och märka `airports`-lagret med deras namn, baserat på deras militära `USE`, dvs. om flygplatsen är tillgänglig för :

- militärfolk, visa den då i grå färg, storlek 8;
- andra, visas sedan i blå färg, storlek 10.

För att göra detta, efter att du har aktiverat märkningen på lagrets `NAME`-fält (se *Ställa in en etikett*):

1. Aktivera fliken *Text*.
2. Klicka på ikonen bredvid egenskapen *Size*.
3. Välj *Edit…* och skriv:

```
CASE
  WHEN "USE" like '%Military%' THEN 8 -- because compatible values are
↪'Military'
                                    -- and 'Joint Military/Civilian'
  ELSE 10
END
```

4. Tryck på *OK* för att validera. Dialogrutan stängs och knappen blir , vilket betyder att en regel körs.
5. Klicka sedan på knappen bredvid färgegenskapen, skriv in uttrycket nedan och validera:

```
CASE
  WHEN "USE" like '%Military%' THEN '150, 150, 150'
  ELSE '0, 0, 255'
END
```

På samma sätt kan du anpassa alla andra egenskaper hos etiketten på det sätt du vill. Se mer information om widgeten Data-define override beskrivning och hantering i avsnittet *Inställning av datadefinierad åsidosättning*.

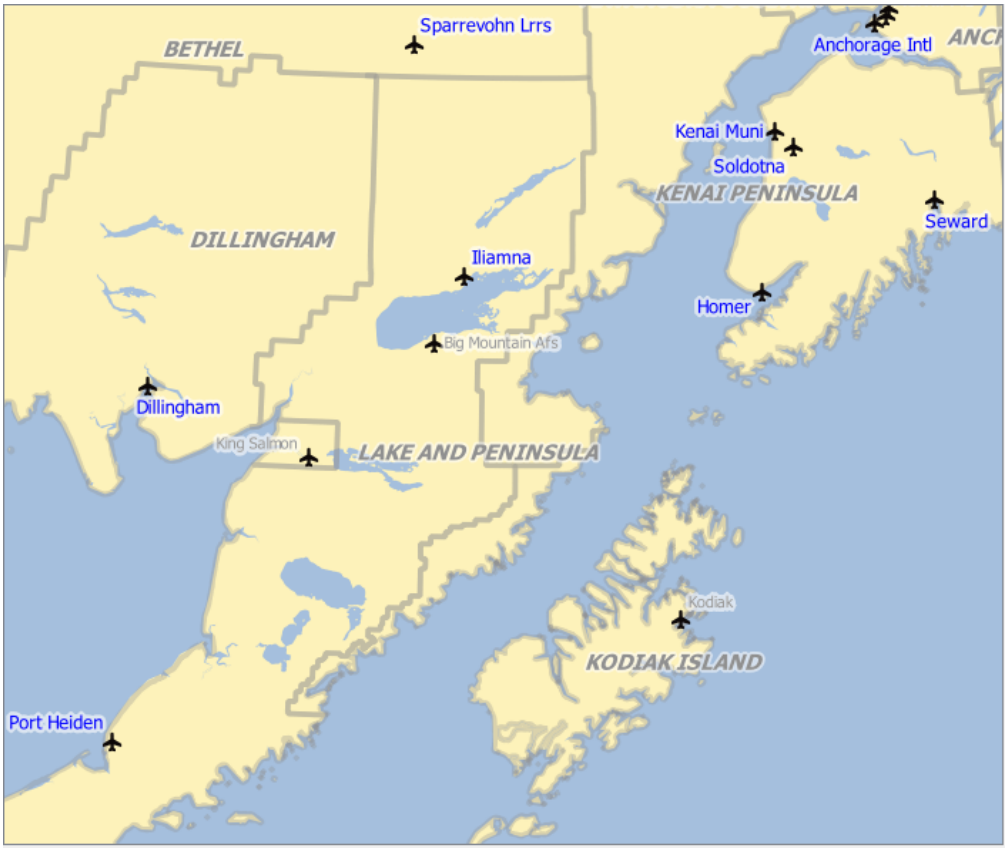


Fig. 12.30: Flygplatsernas etiketter formateras baserat på deras attribut

Tips

Använd den datadefinierade åsidosättningen för att märka varje del av flerdelade funktioner

Det finns ett alternativ för att ställa in märkningen för flerdelade funktioner oberoende av dina etikettegenskaper. Välj *Rendering*, `Feature options`, gå till knappen Data-define override bredvid kryssrutan *Label every part of multipart-features* och definiera etiketterna enligt beskrivningen i *Inställning av datadefinierad åsidosättning*.

Verktygsfältet för etiketter

Verktygsfältet *Label Toolbar* tillhandahåller några verktyg för att manipulera egenskaperna *label* (inklusive deras *callout*) eller *diagram*:

Fig. 12.31: Verktygsfältet Etikett

- Highlight Pinned Labels, Diagrams and Callouts. Om objektets vektorlager är redigerbart är markeringen grön, annars är den blå.
- Växla visning av oplacerade etiketter: Gör det möjligt att avgöra om några viktiga etiketter saknas på kartorna (t.ex. på grund av överlappningar eller andra begränsningar). De visas med en anpassningsbar färg (se *Inställning av den automatiska placeringsmotorn*).

- Pin/Unpin Etiketter och diagram. Genom att klicka eller dra i ett område fäster du överlagrade objekt. Om du klickar på eller drar ett område med `Shift`, tas objekten bort. Slutligen kan du också klicka på eller dra ett område med `Ctrl` för att växla deras fastsättningsstatus.
- Visa/dölj etiketter och diagram. Om du klickar på objekten, eller klickar och drar ett område med `Shift`, döljs de. När ett objekt är dolt behöver du bara klicka på funktionen för att återställa dess synlighet. Om du drar ett område kommer alla objekt i området att återställas.
- Flytta en etikett, ett diagram eller en bildtext: klicka för att markera objektet och klicka för att flytta det till önskad plats. De nya koordinaterna lagras i *auxiliary fields*. Om du väljer objektet med det här verktyget och trycker på `Delete` raderas det lagrade positionsvärdet.
- Rotera en etikett. Klicka för att välja etiketten och klicka igen för att tillämpa önskad rotation. På samma sätt lagras den nya vinkeln i ett extra fält. Om du markerar en etikett med detta verktyg och trycker på `Delete` raderas rotationsvärdet för denna etikett.
- Ändra etikettegenskaper. Den öppnar en dialogruta för att ändra de klickade etikettegenskaperna; det kan vara själva etiketten, dess koordinater, vinkel, typsnitt, storlek, multilinjejustering … så länge som denna egenskap har mappats till ett fält. Här kan du ställa in alternativet till *Label every part of a feature*.

Varning

Etikettverktyg skriver över aktuella fältvärden

Om du använder verktygsfältet *Label* för att anpassa märkningen skrivs faktiskt det nya värdet för egenskapen in i det mappade fältet. Var därför försiktig så att du inte oavsiktligt ersätter data som du kan behöva senare!

Observera

Mekanismen *Egenskaper för hjälplagring* kan användas för att anpassa märkningen (position osv.) utan att ändra den underliggande datakällan.

Anpassa etiketterna från kartbilden

I kombination med *Label Toolbar* hjälper den datadefinierade åsidosättningsinställningen dig att manipulera etiketter i kartbilden (flytta, redigera, rotera). Vi beskriver nu ett exempel med den datadefinierade åsidosättningsfunktionen för funktionen Move Label, Diagram or Callout (se Fig. 12.32).

1. Importera `lakes.shp` från QGIS exempeldataset.
2. Dubbelklicka på lagret för att öppna Lageregenskaper. Klicka på *Labels* och *Placement*. Välj *Offset from centroid*.
3. Leta efter posterna *Data defined*. Klicka på ikonen för att definiera fälttypen för *Coordinate*. Välj `xlabel` för X och `ylabel` för Y. Ikonerna är nu gulmarkerade.

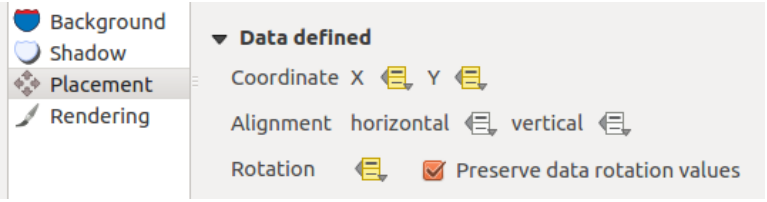


Fig. 12.32: Märkning av vektorpolygonlager med datadefinierad åsidosättning

4. Zooma in på en sjö.
5. Gör lagret redigerbart med hjälp av knappen Växla redigering.
6. Gå till verktygsfältet Etikett och klicka på ikonen . Nu kan du flytta etiketten manuellt till en annan position (se Fig. 12.33). Den nya positionen för etiketten sparas i kolumnerna `xlabel` och `ylabel` i attributtabellen.
7. Det är också möjligt att lägga till en linje som förbinder varje sjö med dess flyttade etikett med hjälp av:
   - etikettens *callout-egenskap*
   - eller *geometri generator symbol lager* med uttrycket nedan:

```
make_line( centroid( $geometry ), make_point( "xlabel", "ylabel" ) )
```

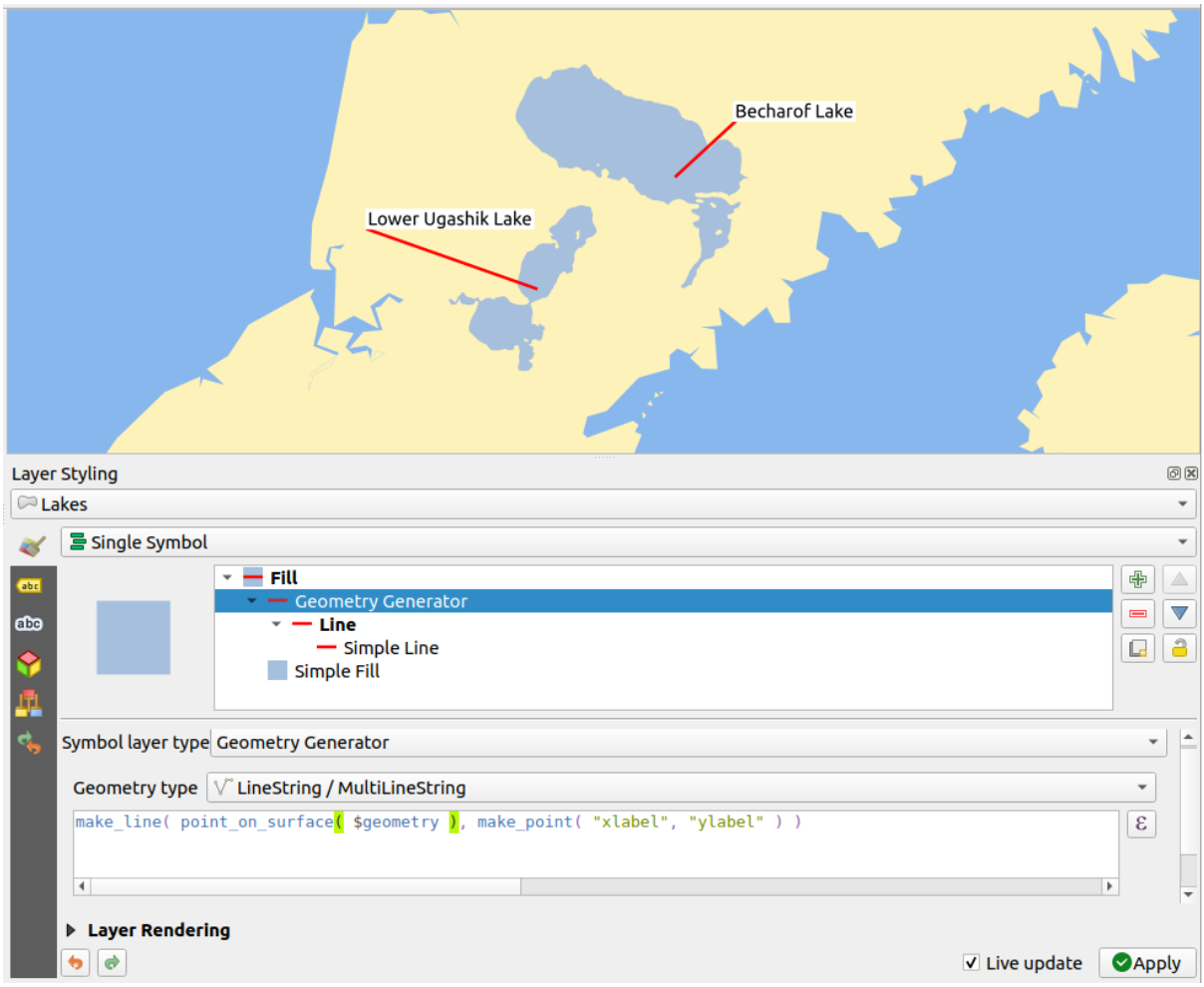


Fig. 12.33: Flyttade etiketter

Observera

Mekanismen *Egenskaper för hjälplagring* kan användas med datadefinierade egenskaper utan att ha en redigerbar datakälla.

12.1.5 Diagram Egenskaper

På fliken *Diagram* kan du lägga till en grafisk överlagring till ett vektorlager (se Fig. 12.34). Denna dialogruta kan också nås från panelen *Layer Styling* eller med knappen Layer Diagram Options i verktygsfältet **Etiketter**.

Den nuvarande kärnimplementeringen av diagram ger stöd för:

- *Inga diagram*: standardvärdet med inget diagram som visas över funktionerna;
- *Pie chart*, ett cirkelformat statistiskt diagram uppdelat i skivor för att illustrera numeriska proportioner. Båglängden för varje skiva är proportionell mot den kvantitet den representerar;
- *Text diagram*, en horisontellt delad cirkel som visar statistiska värden inuti;
- *Histogram*, staplar med varierande färger för varje attribut som är placerade bredvid varandra;
- *Stacked bars*, staplar staplar med olika färger för varje attribut ovanpå varandra vertikalt eller horisontellt;
- *Stacked diagram*, staplar diagram av samma eller varierande typ bredvid varandra, vertikalt eller horisontellt. Mer information finns i *Stacked Diagrams*.

I det övre högra hörnet på fliken *Diagram ger knappen \autoPlacementSettings\ :sup:`Inställningar för automatisk placering (gäller alla lager)* möjlighet att styra diagrammet *labels placering* på kartbilden.

Tips

Skifta snabbt mellan olika typer av diagram

Med tanke på att inställningarna är nästan gemensamma för de olika diagramtyperna kan du när du utformar ditt diagram enkelt ändra diagramtyp och kontrollera vilken som är mer lämplig för dina data utan förlust.

För varje typ av diagram är egenskaperna indelade i flera flikar:

- *Attribut*
- *Rendering*
- *Size*
- *Placering*
- *Alternativ*
- *Förklaring*

Attribut

Attribut definierar vilka variabler som ska visas i diagrammet. Använd add item-knappen för att välja önskade fält i panelen ”Assigned Attributes”. Genererade attribut med *Uttryck* kan också användas.

Du kan flytta upp och ner på alla rader genom att klicka och dra och sortera hur attributen ska visas. Du kan också ändra etiketten i kolumnen ”Legend” eller attributfärgen genom att dubbelklicka på objektet.

Denna etikett är den standardtext som visas i teckenförklaringen i utskriftslayouten eller i lagerträdet.

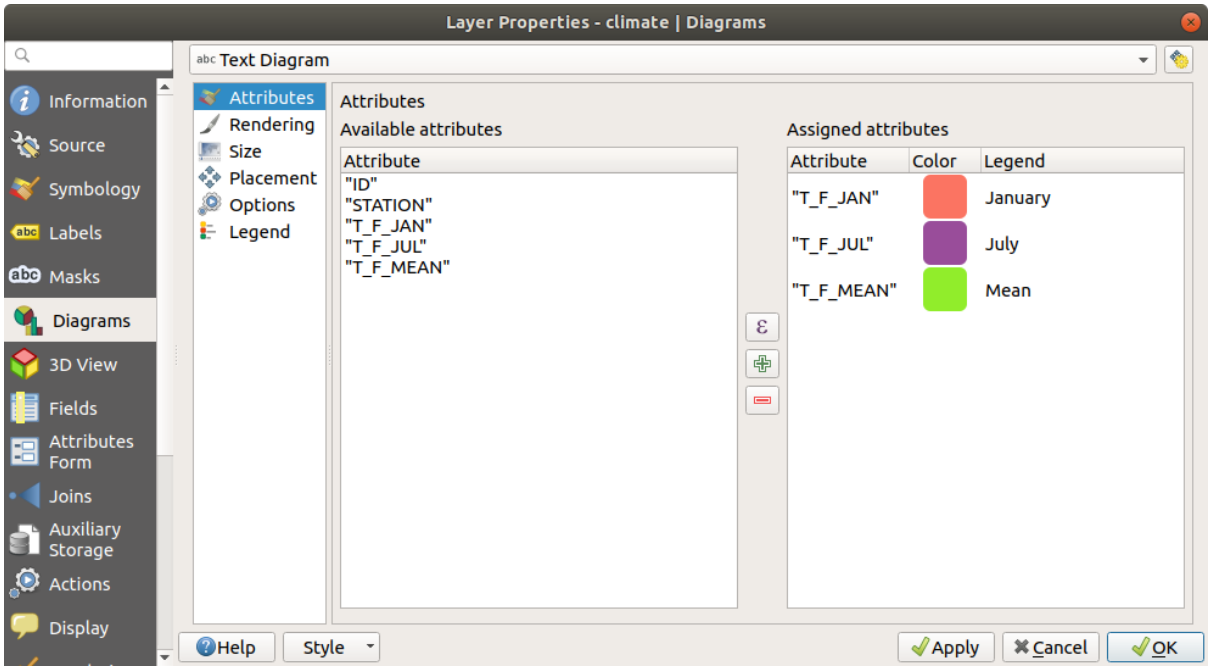


Fig. 12.34: Egenskaper för diagram - fliken Attribut

Rendering

Rendering definierar hur diagrammet ser ut. Här finns allmänna inställningar som inte påverkar de statistiska värdena, t.ex:

- grafikens opacitet, dess konturbredd och färg;
- beroende på typ av diagram:
  - för histogram och staplade staplar, stapelns bredd och avståndet mellan staplarna. Du kanske vill ställa in avståndet till `0` för staplade staplar. Dessutom kan *Axis line symbol* göras synlig på kartans canvas och anpassas med hjälp av *line symbol properties*.
  - för textdiagram, cirkelns bakgrundsfärg och den *font* som används för texter;
  - för cirkeldiagram, *Startvinkel* för den första skivan och deras *Riktning* (medurs eller inte).
- användningen av *målningseffekter* på grafiken.

På den här fliken kan du också hantera och finjustera diagrammets synlighet med olika alternativ:

- *Diagram z-index*: styr hur diagram ritas ovanpå varandra och ovanpå etiketter. Ett diagram med ett högt index ritas över andra diagram och etiketter;
- *Visa alla diagram*: visar alla diagram även om de överlappar varandra;
- *Visa diagram*: gör att endast specifika diagram kan visas;
- *Always Show*: väljer ut specifika diagram som alltid ska återges, även om de överlappar andra diagram eller kartetiketter;
- ställa in *Skalaberoende synlighet*;

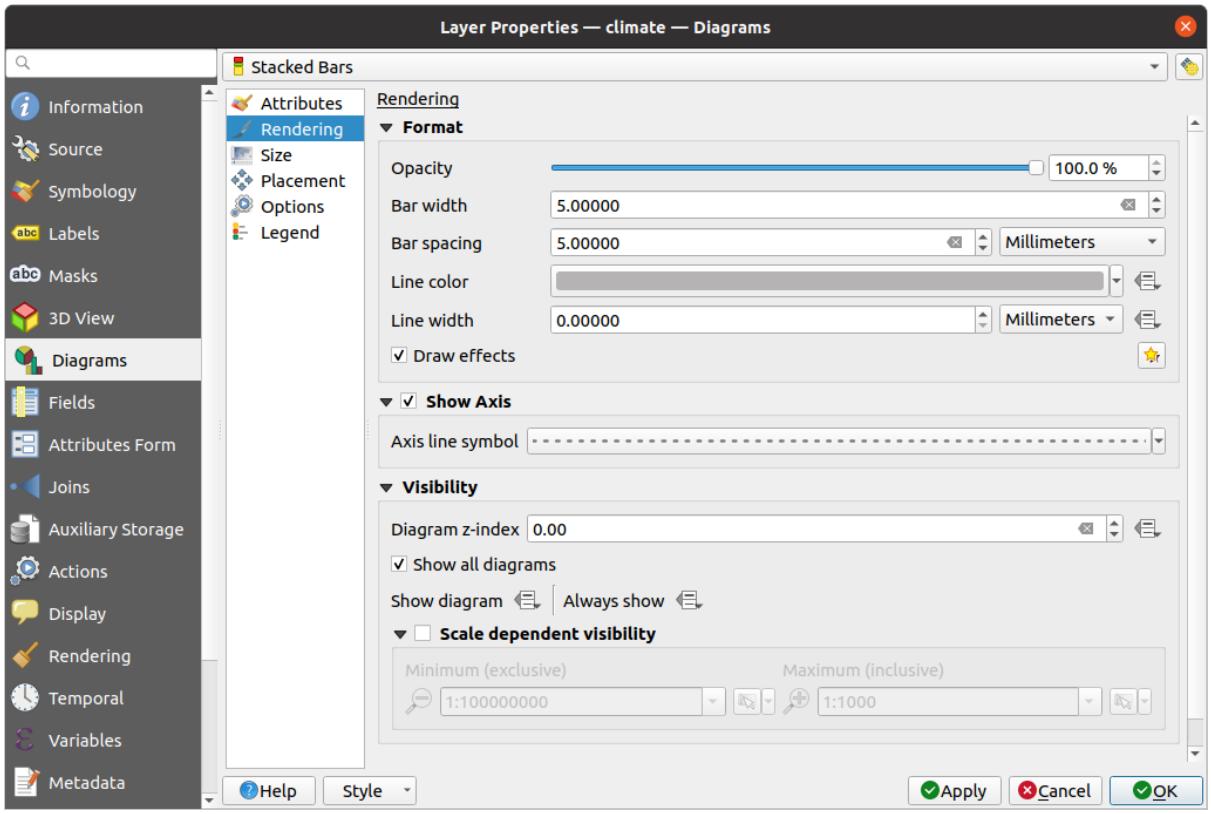


Fig. 12.35: Egenskaper för diagram - fliken Rendering

Storlek

Size är huvudfliken för att ställa in hur den valda statistiken ska representeras. Diagramstorleken *units* kan vara 'Millimeter', 'Punkter', 'Pixlar', 'Kartenheter' eller 'Tum'. Du kan använda:

- *Fixed size*, en unik storlek för att representera grafiken av alla funktioner (inte tillgänglig för histogram)
- eller *Skalerad storlek*, baserat på ett uttryck som använder lagerattribut:
  1. I *Attribute*, välj ett fält eller skapa ett uttryck
  2. Tryck på *Find* för att returnera attributets *Maximum value* eller ange ett eget värde i widgeten.
  3. För histogram och staplade staplar anger du ett värde för *Bar length* som används för att representera attributens *Maximum value*. För varje funktion kommer stapellängden sedan att skalas linjärt för att behålla denna matchning.
  4. För cirkeldiagram och textdiagram anger du ett värde för *Storlek*, som används för att representera attributens *Maximum value*. För varje egenskap kommer cirkelytan eller diametern att skalas linjärt för att hålla denna matchning (från `0`). En *Minimum size* kan dock ställas in för små diagram.

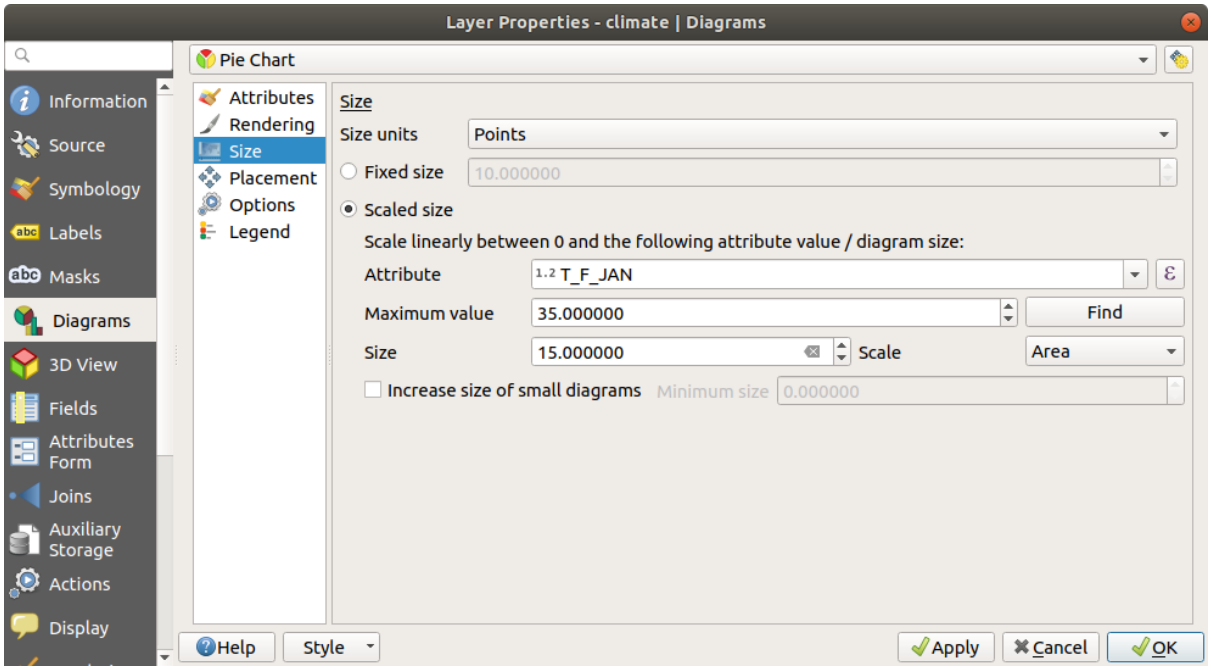


Fig. 12.36: Egenskaper för diagram - fliken Storlek

Placering

Placement definierar diagrammets position. Beroende på lagrets geometrityp finns det olika alternativ för placeringen (mer information finns i *Placement*):

- *Around point* eller *Over point* för punktgeometri. Den förstnämnda variabeln kräver en radie att följa.
- *Around line* eller *Over line* för linjegeometri. Precis som för punktobjekt kräver den första variabeln ett avstånd att respektera och du kan ange diagrammets placering i förhållande till objektet (’ovanför’, ’på’ och/eller ’under’ linjen) Det är möjligt att välja flera alternativ samtidigt. I så fall kommer QGIS att leta efter den optimala positionen för diagrammet. Kom ihåg att du också kan använda linjeorienteringen för diagrammets position.
- *Around centroid* (på ett bestämt *Distance*), *Over centroid*, *Using perimeter* och *Inside polygon* är alternativen för polygonfunktioner.

Gruppen *Coordinate* ger direkt kontroll över diagrammets placering, funktion för funktion, genom att använda deras attribut eller ett uttryck för att ange koordinaterna *X* och *Y*. Informationen kan också fyllas i med hjälp av verktyget *Move labels and diagrams*.

I avsnittet *Priority* kan du definiera placeringsprioriteten för varje diagram, d.v.s. om det finns olika diagram eller etiketter som är kandidater för samma plats, kommer objektet med den högsta prioriteten att visas och de andra kan utelämnas.

Avråd diagram och etiketter från att täcka funktioner definierar funktioner som ska användas som *hinder*, dvs. QGIS kommer att försöka att inte placera diagram eller etiketter över dessa funktioner. Prioritetsrangordningen används sedan för att utvärdera om ett diagram kan utelämnas på grund av en högre viktad hinderfunktion.

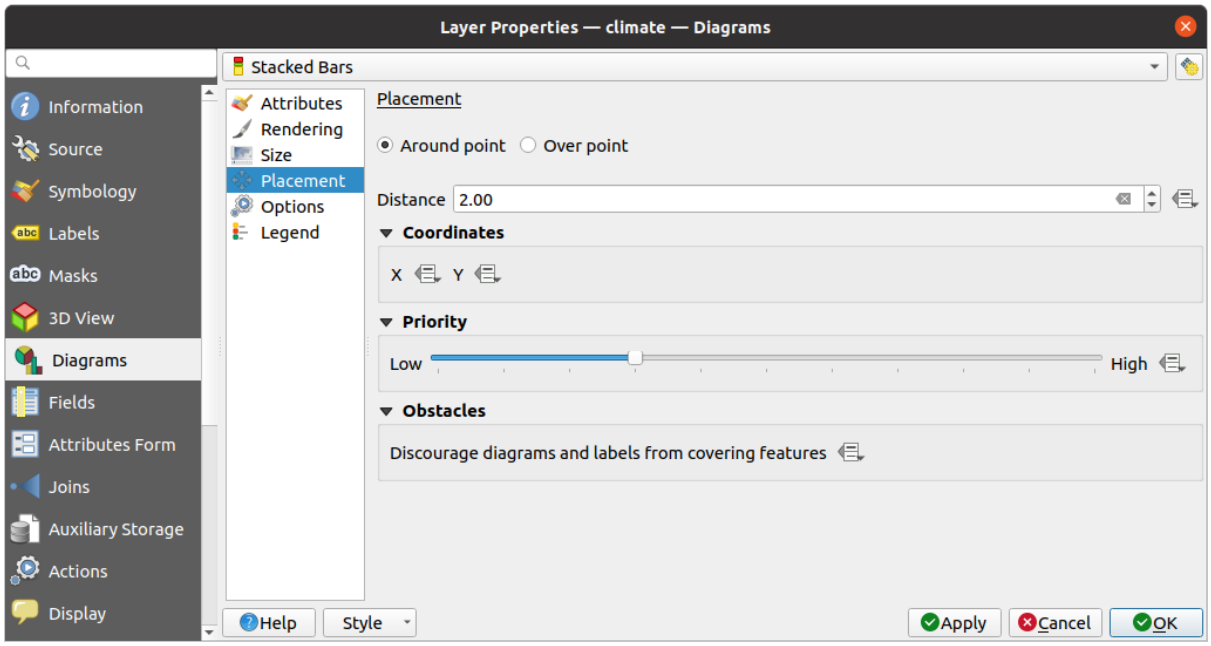


Fig. 12.37: Dialog för vektoregenskaper med diagramegenskaper, fliken Placering

Alternativ

Fliken *Options* har inställningar för histogram och staplade staplar. Du kan välja om *Bar orientation* ska vara *Up*, *Down*, *Right* eller *Left*, för horisontella och vertikala diagram.

Historia

På fliken *Legend* kan du välja att visa objekt i diagrammet i *Layers-panelen* och i *print layout förklaring*, bredvid lagersymbologin:

- markera *Visa legendposter för diagramattribut* för att i legenderna visa egenskaperna `Color` och `Legend`, som tidigare tilldelats på fliken *Attributes*;
- och, när en *skalad storlek* används för diagrammen, tryck på knappen *Legend Entries for Diagram Size…* för att konfigurera diagramsymbolens aspekt i legenderna. Detta öppnar dialogrutan *Data-defined Size Legend* vars alternativ beskrivs i *Datadefinierad storleksförklaring*.

När den är inställd visas diagrammets teckenförklaring (attribut med färg och diagramstorlek) även i teckenförklaringen för utskriftslayouten, bredvid lagersymbologin.

Staplade diagram

Med staplade diagram kan användare skapa komplexa diagram som befolkningspyramider, där två underdiagram, nämligen histogram, ligger sida vid sida och visas horisontellt.

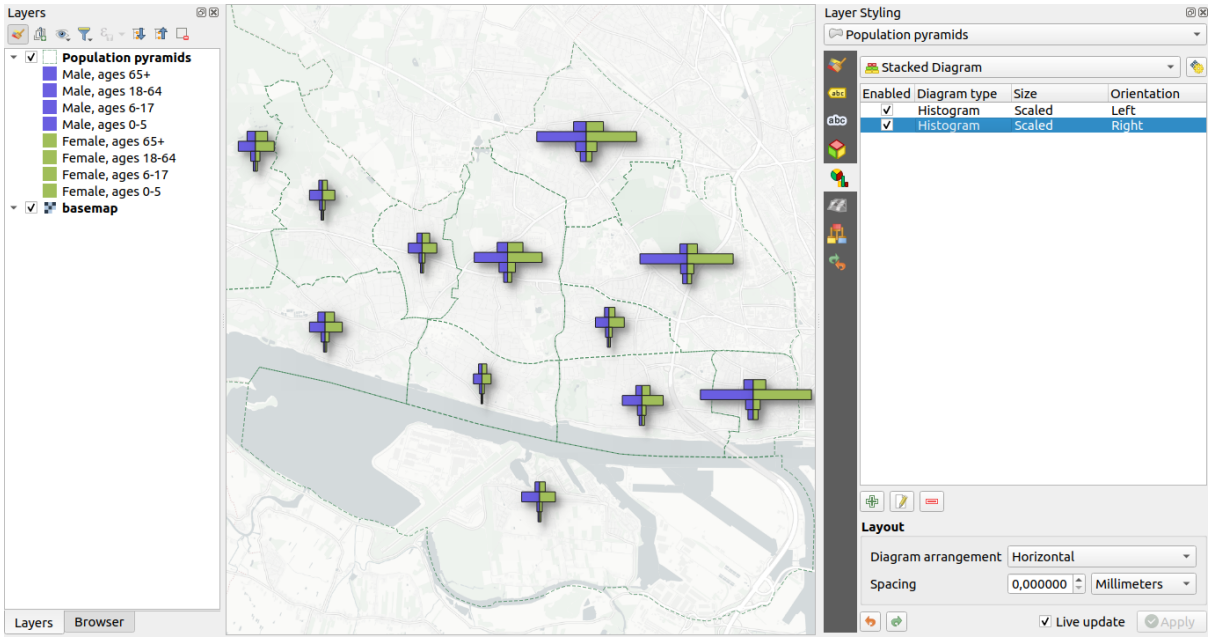


Fig. 12.38: Befolkningspyramider byggda för varje lagerfunktion

Multitemporala diagram kan också konstrueras som staplade diagram. Antalet underdiagram samt avståndet mellan dem kan konfigureras.

Dessutom kan underdiagram ha olika typer (t.ex. ett cirkeldiagram tillsammans med ett histogram) och ha sina egna oberoende inställningar som *Attribut*, *Rendering*, *Storlek*, *Alternativ* och *Legend*.

inställningarna för *Placement* i ett staplat diagram, liksom vissa synlighetsinställningar (som finns på fliken *Rendering*), bestäms av placerings- och synlighetsinställningarna för det första underdiagrammet i stapeln.

Slutligen ges subdiagrammens ordning av posternas ordning i stapeldiagrammets lista. Det första underdiagrammet visas till vänster i ett horisontellt staplat diagram, eller i den övre delen av ett vertikalt.

12.1.6 Masker Egenskaper

Fliken *Masker* hjälper dig att konfigurera det aktuella lagrets symboler så att de överlagras med andra symbollager eller etiketter, från vilket lager som helst. Detta är tänkt att förbättra läsbarheten för symboler och etiketter vars färger ligger nära varandra och kan vara svåra att tyda när de överlappar varandra; det lägger till en anpassad och transparent mask runt objekten för att ”dölja” delar av symbollagren i det aktuella lagret.

Om du vill använda masker på det aktiva lagret måste du först aktivera antingen *mask symbol layers* eller *mask labels* i projektet. Sedan, från fliken *Masks*, kontrollera:

- *Masked symbol layers*: listar i en trädstruktur alla symbollager i det aktuella lagret. Där kan du välja det symbolskiktsobjekt som du vill transparent ”klippa ut” när de överlappar de valda maskkällorna
- fliken *Mask sources*: lista alla maskeringsetiketter och maskeringssymbolskikt som definierats i projektet. Välj de objekt som ska generera masken över de valda maskerade symbollagren

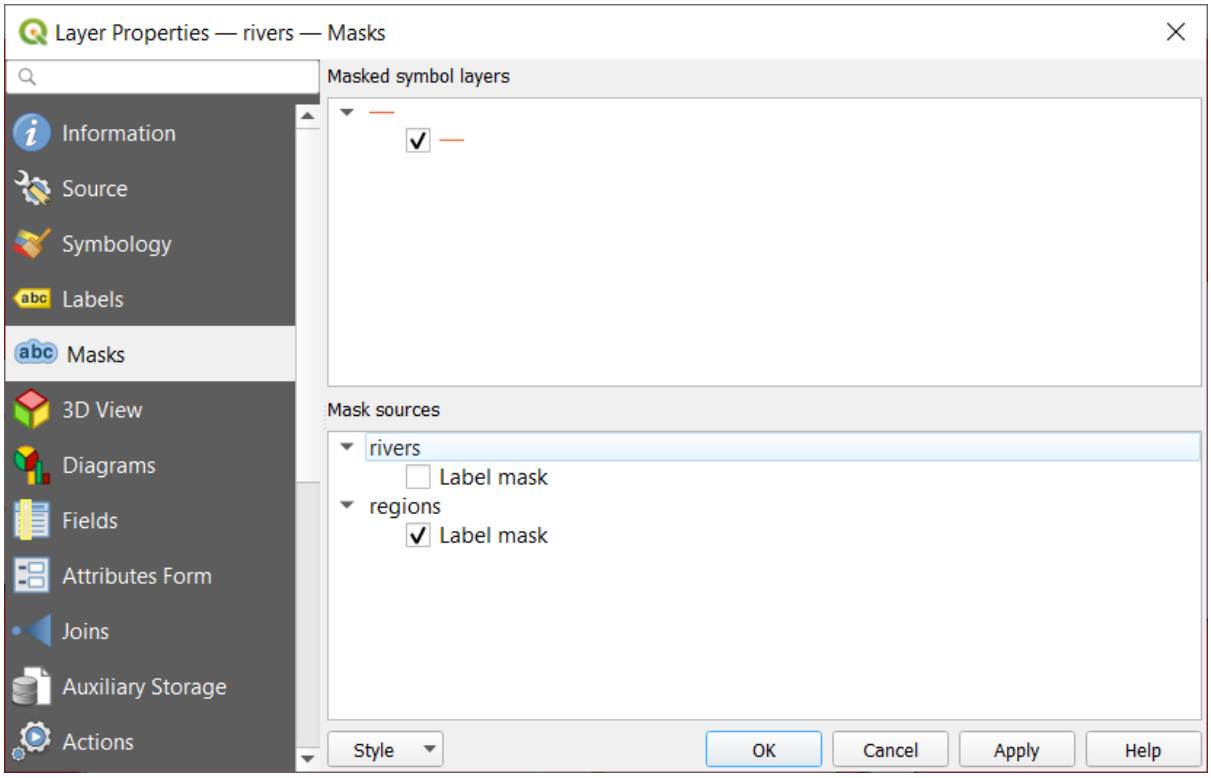


Fig. 12.39: Egenskaper för lager - fliken Masker

12.1.7 egenskaper för 3D-vy

Fliken *3D View* ger inställningar för vektorlager som ska visas i verktyget *3D Map view*.

Om du vill visa ett lager i 3D väljer du antingen i kombinationsrutan högst upp på fliken:

- *Enkel symbol*: objekten återges med hjälp av en gemensam 3D-symbol vars egenskaper kan vara *datadefinierade* eller inte. Läs detaljer om *inställning av en 3D-symbol* för varje lagergeometrityp.
- *Regelbaserad*: flera symbolkonfigurationer kan definieras och tillämpas selektivt baserat på uttrycksfilter och skalområde. Mer information om hur man gör finns på *Regelbaserad rendering*.

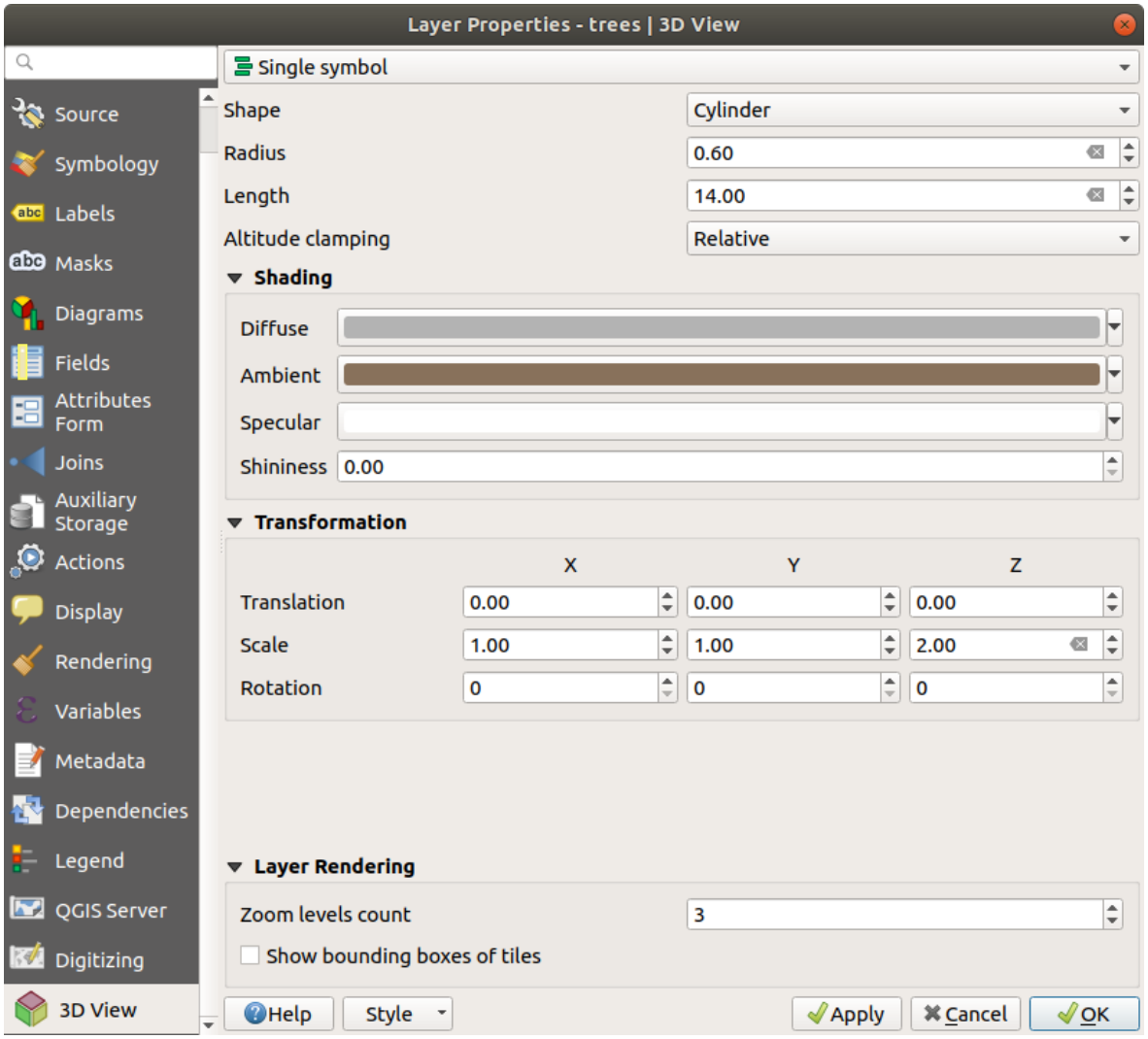


Fig. 12.40: 3D-egenskaper för ett punktlager

Uppmärksamma

Företräda *Elevation* fliken för inställningar för symbolhöjd och terräng

Egenskapernas höjd och höjdrelaterade egenskaper (*Altitude clamping*, *Altitude binding*, *Extrusion* eller *Height*) i fliken *3D View* ärver sina standardvärden från lagrets *Elevationsegenskaper* och bör helst ställas in från fliken *Elevation*.

För bättre prestanda laddas data från vektorlager i bakgrunden med hjälp av multithreading och återges i rutor vars storlek kan styras från avsnittet *Layer rendering* på fliken:

- *Zoom levels count*: bestämmer hur djupt kvadratträdet ska vara. Till exempel innebär en zoomnivå att det kommer att finnas en enda kakelplatta för hela lagret. Tre zoomnivåer innebär att det kommer att finnas 16 brickor på bladnivån (varje extra zoomnivå multiplicerar det med 4). Standardvärdet är `3` och det maximala värdet är `8`.
- *Visa begränsningsrutor för brickor*: särskilt användbart om det finns problem med att brickor inte visas när de borde.

12.1.8 Fältegenskaper

Fliken *Fields* innehåller information om fält som är relaterade till lagret och hjälper dig att organisera dem.

Lagret kan göras *redigerbart* med hjälp av Växla redigeringsläge. För närvarande kan du ändra dess struktur med hjälp av knapparna Nytt fält och Ta bort fält.

När du skapar New field är alternativet *Comment* endast tillgängligt för datakällor som tillåter redigering av kommentarer (se *Databasposter* för mer information). Du kan också ange alias i dialogrutan *Add Field* för OGR-format som stöds (GeoPackage och ESRI File Geodatabase).

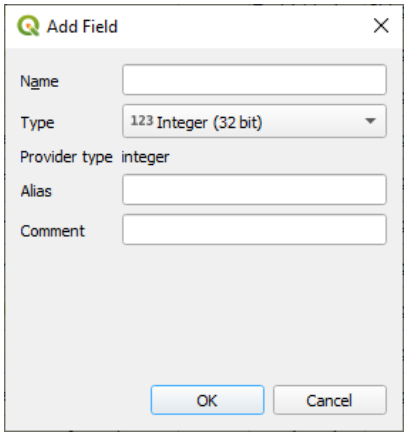


Fig. 12.41: Dialogrutan Lägg till fält

Du kan också byta namn på fält genom att dubbelklicka på dess namn. Detta stöds endast för dataleverantörer som PostgreSQL, Oracle, Memory-lager och vissa GDAL-lager beroende på GDAL-versionen.

Om det anges i den underliggande datakällan eller i *forms properties* visas även fältets alias. Ett alias är ett fältnamn som är läsbart för människor och som du kan använda i feature-formuläret eller attributtabellen. Alias sparas i projektfilen.

Förutom de fält som ingår i datasetet, *virtuella fält* och *Extra lagringsutrymme*, listar fliken *Fält* även fält från alla *joined layers*. Beroende på fältets ursprung får det en annan bakgrundsfärg.

För varje listat fält visas även skrivskyddade egenskaper som *Type*, *Type name*, *Length* och *Precision*.

Beroende på dataleverantör kan du koppla en kommentar till ett fält, t.ex. när det skapas. Den här informationen hämtas och visas i kolumnen *Comment* och visas senare när du håller muspekaren över fältetiketten i ett funktionsformulär.

I kolumnen *Configuration* kan du ställa in hur fältet ska bete sig under vissa omständigheter:

- `Inte sökbart`: markera det här alternativet om du inte vill att det här fältet ska efterfrågas av *search locator bar*
- ”Exponera inte via WMS”: markera det här alternativet om du inte vill visa det här fältet om lagret serveras som WMS från QGIS-servern
- ”Exponera inte via WFS”: markera det här alternativet om du inte vill visa det här fältet om lagret visas som WFS från QGIS-servern

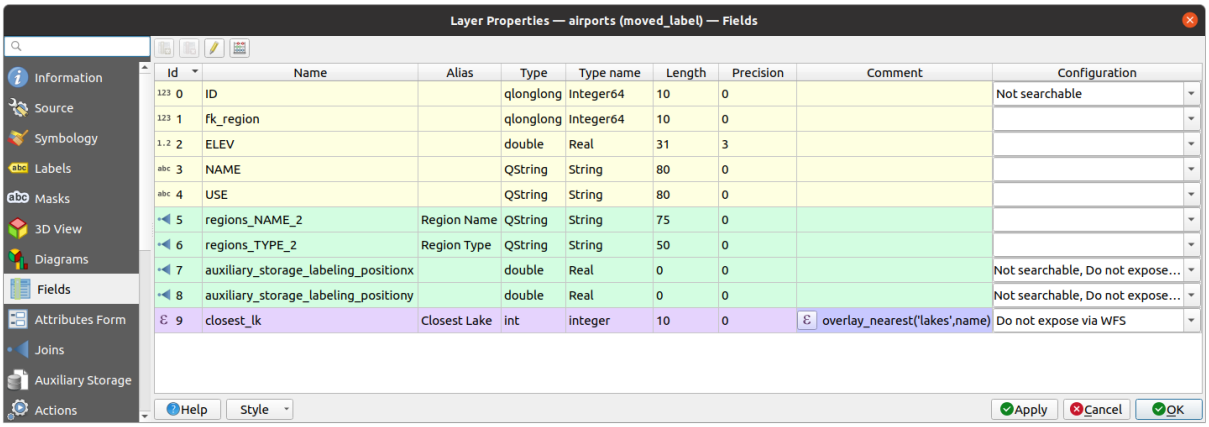


Fig. 12.42: Fliken Egenskaper för fält

12.1.9 Attribut Formulär Egenskaper

Fliken *Attributes Form* hjälper dig att ställa in det formulär som ska visas när du skapar nya funktioner, redigerar eller frågar efter befintliga. Detta påverkar dem i både *tabell- och formulärvyer*. Du kan definiera:

- utseendet och beteendet (etikett, synlighet, widget, begränsningar …) för varje fält i det aktuella lagret, de *joinade lagren* och de *relationella lagren*;
- placeringen av aktiverade *actions*;
- formulärets struktur (anpassad eller autogenererad);
- extra HTML-, QML- och textwidgets för att förbättra formuläret;
- extra logik i Python för att hantera interaktion med formuläret eller fältwidgets.

Längst upp till höger i dialogrutan kan du ställa in om formuläret ska öppnas som standard när du skapar nya funktioner. Detta kan konfigureras per lager eller globalt med alternativet *Suppress attribute form pop-up after feature creation* i menyn *Inställningar* ► *Alternativ* ► *Digitizing*.

Anpassa ett formulär för dina data

Som standard, när du klickar på en funktion med verktyget Identify Features eller växlar attributtabellen till läget *formulärsvy*, visar QGIS ett grundläggande formulär med fördefinierade widgetar (i allmänhet snurrlådor och textlådor — varje fält representeras på en dedikerad rad av dess etikett bredvid widgeten). Om *relations* har angetts för lagret visas fält från de refererande lagren i en inbäddad ram längst ned i formuläret, enligt samma grundstruktur.

Den här renderingen är resultatet av standardvärdet `Autogenerate` för inställningen *Attribute editor layout* i fliken *Lageregenskaper* ► *Attributes Form*. Denna egenskap har tre olika värden:

- `Autogenerate`: behåller den grundläggande strukturen ”en rad - ett fält” för formuläret men gör det möjligt att anpassa varje motsvarande widget.
- `Drag-and-drop designer`: Förutom anpassning av widgetar kan formulärstrukturen göras mer komplex, t.ex. med widgetar inbäddade i grupper och flikar.
- `Provide ui file`: gör det möjligt att använda en Qt-designerfil, och därmed en potentiellt mer komplex och fullt utrustad mall, som funktionsformulär.

Det autogenererade formuläret

När alternativet `Autogenerera` är aktiverat visar panelen *Tillgängliga widgets* listor över fält (från det aktuella lagret, länk- och relationsskikten) och de aktiverade åtgärder som skulle visas i formuläret. Välj ett fält så kan du konfigurera dess utseende och beteende i den högra panelen:

- lägga till *anpassad etikett och automatiserade kontroller* till fältet;
- ange en *partikulär widget* som ska användas.

I det här redigeringsläget kan endast egenskaperna för åtgärderna visas. Dessutom visas de via en rullgardinsmeny *Actions* i funktionsformuläret och enligt egenskapen *Show in attribute table* i andra sammanhang.

Dra och släpp-designern

Med dra-och-släpp-designern kan du skapa ett formulär med flera behållare (flikar eller grupper) för att presentera attributfälten eller andra widgetar som inte är direkt kopplade till ett visst fält (som HTML/QML-widgetar eller *åtgärder* som definieras för lagret), vilket visas till exempel i Fig. 12.43.

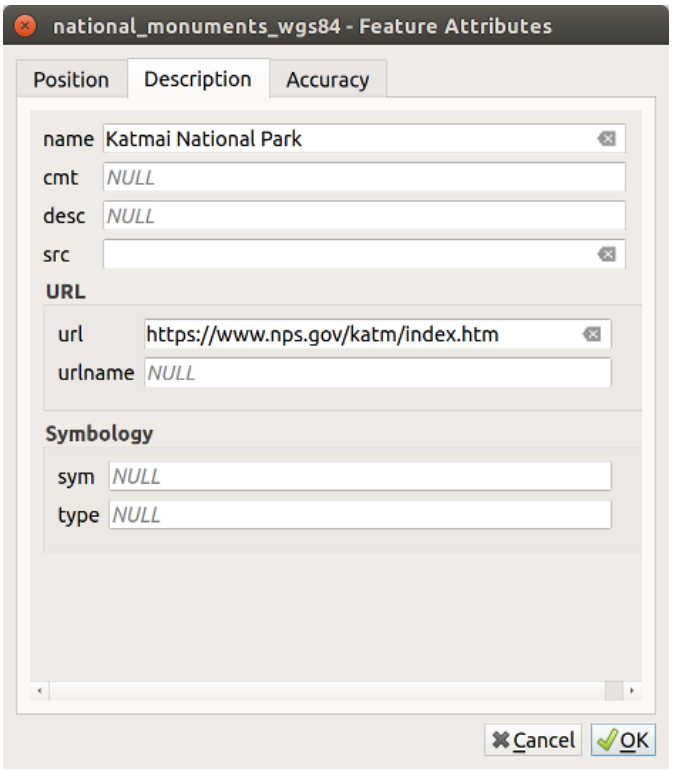


Fig. 12.43: Inbyggt formulär med flikar och namngivna grupper

1. Välj `Drag and drop designer` från *Select attribute layout editor* comboboxen. Detta aktiverar panelen *Form Layout* bredvid panelen *Available widgets*, som fylls med befintliga fält. Det valda fältet visar sina *properties* i en tredje panel.

2. Markera fält som du inte vill använda i panelen *Form Layout* och tryck på knappen för att ta bort dem. Du kan också växla urvalet med knappen Invert selection.

3. Dra och släpp fält från den första panelen till *Form Layout* för att lägga till dem igen. Samma fält kan läggas till flera gånger.

4. Dra och släpp fält i panelen *Form Layout* för att ändra deras position.

5. Lägg till behållare för att associera fält som tillhör samma kategori och strukturera formuläret bättre.

1. Det första steget är att använda ikonen Lägg till en ny flik eller grupp i formulärlayouten. Fält och andra grupper kommer att visas i den.
2. Ställ sedan in egenskaperna för behållaren, t.ex:
   - *Label*: den titel som ska användas för behållaren
   - *Container Type*: det kan vara en *Tab*, *Group box in container* (en hopfällbar gruppbox inuti en flik eller en annan grupp) eller en *Row* (en behållartyp som låter dig ordna dina widgets i en horisontell rad och automatiskt bestämmer antalet kolumner baserat på antalet widgets),
   - *Within*: med den här valfria funktionen kan du välja en befintlig behållare i vilken den nya behållaren (*Group box in container* eller *Row*) ska bäddas in.
   - och *Antal kolumner* som de inbäddade fälten ska fördelas över

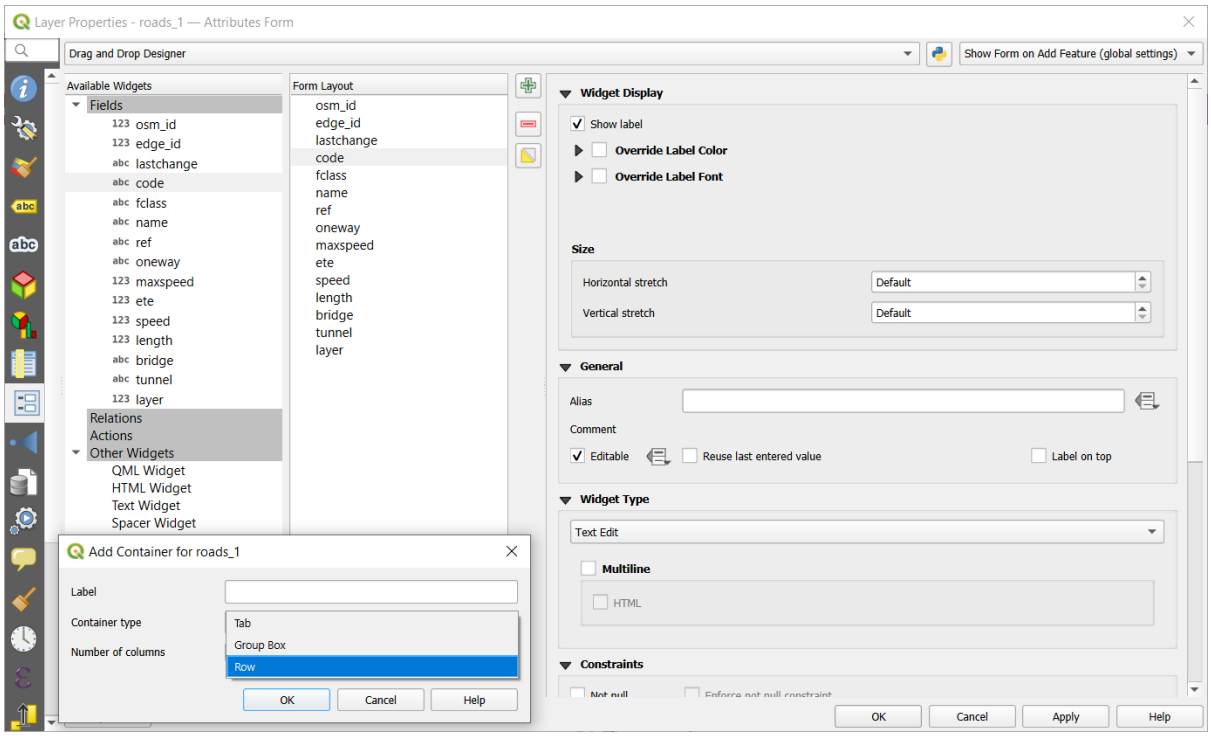


Fig. 12.44: Dialog för att skapa behållare med layouten **Attributredigerare**

Dessa och andra egenskaper kan senare uppdateras genom att välja objektet och i den tredje panelen:

- dölja eller visa behållarens etikett
- byta namn på behållaren
- ställa in antalet kolumner
- ange ett uttryck för att styra behållarens synlighet. Uttrycket kommer att utvärderas varje gång värdena i formuläret ändras och fliken eller grupprutan visas/döljs i enlighet med detta
- *Visa som gruppruta*: konverterar en flik till en gruppruta på översta nivån och vice versa
- om det gäller en gruppruta, ange om den vid formulärröppningen ska visas som *Collapsed* för alla funktioner, eller bara för dem som matchar ett uttryck (*Control collapsed by expression*).
- konfigurera behållarens *Style*-utseende, med anpassade *Bakgrundsfärg*-, etikettfärg- och teckensnittsegenskaper

3. Du kan skapa och bädda in så många behållare du vill genom att trycka på ikonen Lägg till en ny flik eller grupp i formulärlayouten.

6. Nästa steg är att tilldela de relevanta fälten till varje container, genom att dra och släppa. Gruppboxar och flikar kan också flyttas på samma sätt.
7. *Anpassa widgeten* för de fält som används
8. Om lagret är involverat i en *en eller många till många relation*, dra och släpp relationsnamnet från panelen *Available Widgets* till panelen *Form Layout*. Det associerade formuläret för lagerattributet bäddas in på vald plats i det aktuella lagrets formulär. När det gäller de andra objekten väljer du relationsetiketten för att konfigurera vissa egenskaper:
   - dölja eller visa relationsetiketten
   - visa länkknappen
   - visa knappen för att ta bort länken
9. Om lagret har en eller flera *actions* aktiverade för `Lager` eller `Funktion`-omfång, kommer åtgärderna att listas under *Åtgärder* och du kan dra och släppa dem som med de andra fälten. Den associerade åtgärden kommer att bäddas in på den valda platsen i det aktuella lagrets formulär.
10. Anpassa formuläret ytterligare genom att lägga till en eller flera widgets från *Other Widgets* (se *Andra widgets*)
11. Använd dialogrutan för lagrets egenskaper
12. Öppna ett formulär för funktionsattribut (t.ex. med hjälp av verktyget Identify features) så visas det nya formuläret.

Andra widgets

Drag and drop-designern erbjuder ett antal widgetar som inte är kopplade till ett visst fält. De kan användas för att förbättra utseendet på formuläret eller för att visa dynamiskt beräknade värden.

- *HTML Widget*: bäddar in en HTML-sida, HTML-källan kan innehålla resultatet av dynamiskt beräknade uttryck.

  HTML-widgetar kan t.ex. användas för att visa bilder som lagrats som BLOB i ett fält (låt oss kalla det ”foto”):

  1. I läget `Drag-and-drop designer` lägger du till en *HTML Widget* i din *Form Layout*.
  2. Dubbelklicka på *HTML Widget* för att konfigurera den.
  3. Ändra standard *Title* eller dölj den.
  4. Tryck på knappen och ange följande QGIS-uttryck:

     ```
     '<img src= "' || to_base64("photo") || '">'
     ```

     Se till att du ersätter *photo* med ditt eget BLOB-fältnamn. Uttrycket ovan skapar en sträng med HTML-bildtagg i vilken BLOB-filen är kodad.
  5. Använd dialogrutan och tryck sedan på knappen .
  6. QGIS tillämpar automatiskt HTML-formatering och funktioner för att utvärdera ditt uttryck, vilket resulterar i följande kod:

     ```
     <script>document.write(expression.evaluate("'<img src=\"data:image/png;
     ↪base64,' || to_base64(\"photo\") || '\">'"));</script>
     ```

     En förhandsgranskning av din bild visas till höger.
- *QML Widget*: bäddar in ett Qt QML-dokument som visar grafiska element i attributform. Förutom den anpassade *Title* för den tillagda widgeten och om den ska visas eller inte, kan du välja bland fördefinierade *QML code*-element:
  - *Fri text…*: låter dig skriva från grunden eller klistra in en befintlig kod

- *Rectangle*: ger en minimal kod för att visa en rektangel
- *Bar chart*: ger en minimal kod för visning av ett stapeldiagram
- *Pie chart*: ger en minimal kod för visning av ett cirkeldiagram

Du kan utöka koden med QML-syntax, använda lagerfält eller QGIS-uttryck som beräknas dynamiskt.

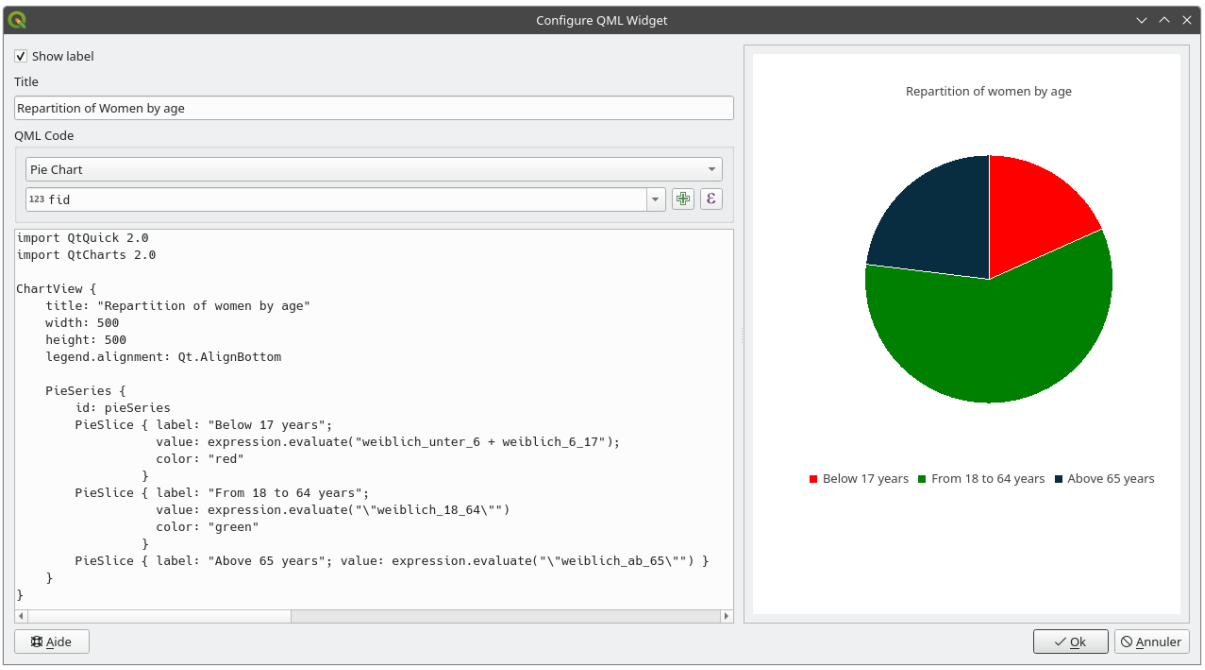


Fig. 12.45: Ställa in en QML-graf så att den visas i attributform

- *Text Widget*: visar en textwidget som stöder grundläggande HTML-markering och som kan innehålla resultatet av dynamiskt beräknade uttryck.
- *Spacer Widget*: infogar en tom transparent rektangel som ökar det vertikala avståndet mellan två widgetar.

Tips

Visa dynamiskt innehåll

De widgetar som nämns ovan (utom *Spacer Widget*) stöder uttryck som kan användas för att visa innehåll som ändras dynamiskt när ett annat fält i formuläret ändras. Detta kan uppnås genom att använda funktionen `current_value('field_name')` i uttrycket för att undersöka värdet på ett annat fält.

Använda anpassad ui-fil

Med alternativet `Provide ui-file` kan du använda komplexa dialogrutor som skapats med Qt-Designer. Genom att använda en UI-fil får du stor frihet när du skapar en dialogruta. Observera att för att kunna länka de grafiska objekten (textruta, kombinationsruta…) till lagrets fält måste du ge dem samma namn.

Använd *Edit UI* för att ange sökvägen till den fil som ska användas.

UI-filer kan också finnas på en fjärrserver. I det här fallet anger du webbadressen till formuläret i stället för filsökvägen i *Edit UI*.

Du hittar några exempel i lektionen Creating a new form i QGIS-training-manual-index-reference. För mer avancerad information, se https://woostuff.wordpress.com/2011/09/05/qgis-tips-custom-feature-forms-with-python-logic/.

Förbättra ditt formulär med anpassade funktioner

QGIS-formulär kan ha en Python-funktion som anropas när dialogrutan öppnas. Använd denna funktion för att lägga till extra logik i dina dialogrutor. Formulärkoden kan specificeras på tre olika sätt:

- `load from the environment`: använd en funktion, t.ex. i `startup.py` eller från ett installerat tillägg
- `load from an external file`: en filväljare låter dig välja en Python-fil från ditt filsystem eller ange en URL för en fjärrfil.
- `Provide code in this dialog`: en Python-editor visas där du direkt kan skriva in den funktion som ska användas.

I samtliga fall måste du ange namnet på den funktion som ska anropas (`open` i exemplet nedan).

Ett exempel är (i modulen MyForms.py):

```
def open(dialog,layer,feature):
    geom = feature.geometry()
    control = dialog.findChild(QWidget,"My line edit")
```

Referens i Python Init Funktion så här: `öppna`

Konfigurera fältets beteende

Huvuddelen av fliken *Attributes Form* hjälper dig att ställa in vilken typ av widget som används för att fylla i eller visa värden i fältet, i attributtabellen eller funktionsformuläret: du kan definiera hur användaren interagerar med varje fält och de värden eller intervall av värden som får läggas till i varje fält.

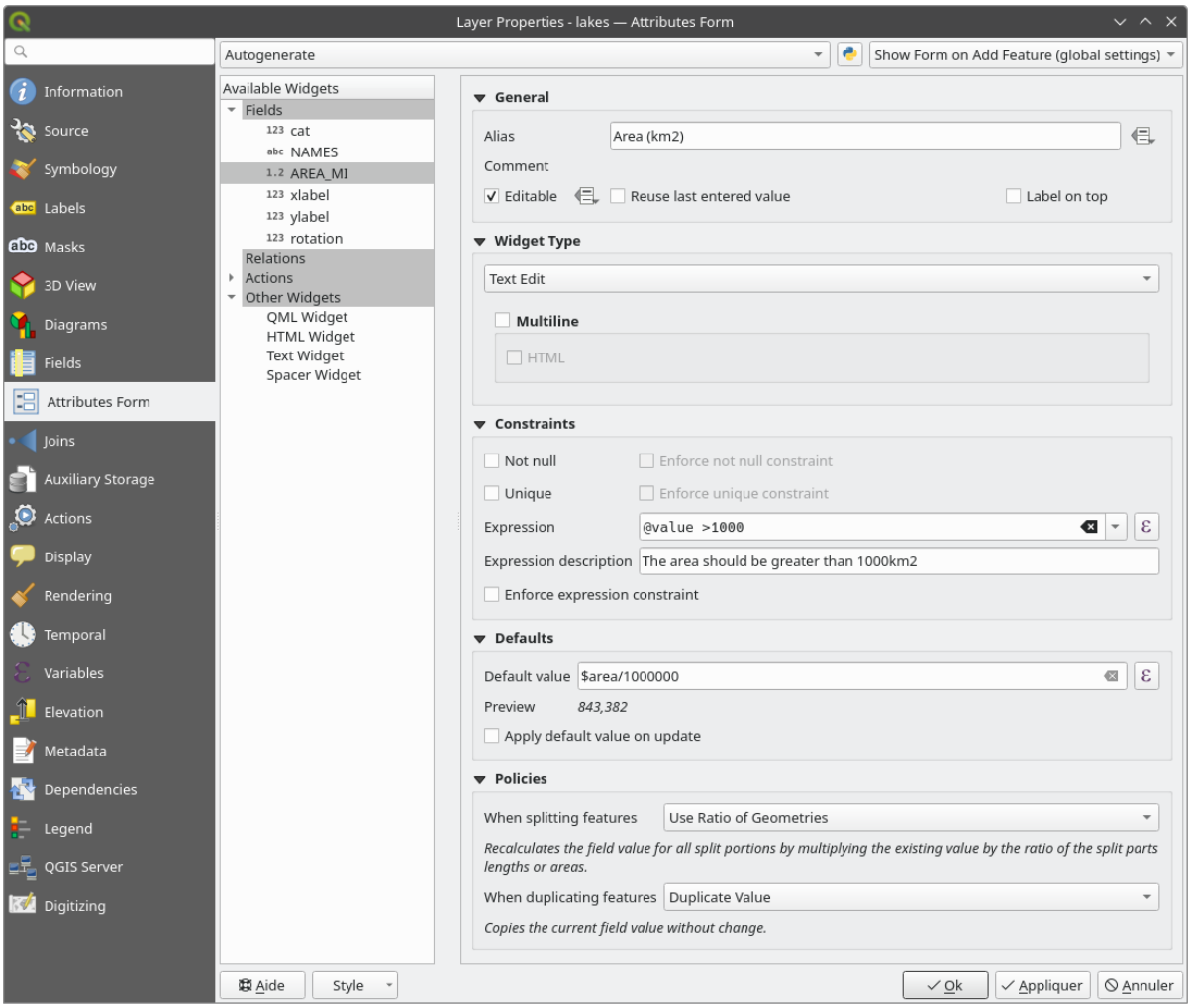


Fig. 12.46: Dialog för att välja en redigeringswidget för en attributkolumn

Allmänna inställningar

Oavsett vilken typ av widget som används för fältet finns det några gemensamma egenskaper som du kan ställa in för att styra om och hur ett fält kan redigeras.

Widgetvisning

Den här gruppen är endast tillgänglig för designerläget *Drag and drop* och hjälper dig att konfigurera utseendet på den widget som tilldelats fältet:

- *Visa etikett*: anger om fältnamnet ska visas i formuläret
- *Overriv etikettfärg*: tillämpar en specifik färg på fältets etikett
- *Override label font*: tillämpar specifika teckensnittsegenskaper (fet, kursiv, understruken, genomstruken eller teckensnittsfamilj) på fältets etikett
- *Size*: gör det möjligt att styra hur widgetar relativt sett ska ändra storlek när man ändrar storlek på ett attributformulär.
  - *Horizontal stretch*: anger ett högre horisontellt värde för widgetar som behöver mer horisontellt utrymme.
  - *Vertical stretch*: bestämmer hur widgetar ändras vertikalt när formuläret ändras i storlek.

Allmäna inställningar

- *Alias*: ett mänskligt läsbart namn att använda för fält. Aliaset kommer att visas i funktionsformuläret, attributtabellen eller i panelen *Identify results*. Det kan också användas som ersättning för fältnamn i *expression builder*, vilket underlättar förståelsen och granskningen av uttryck. Alias sparas i projektfilen.
- *Comment*: visar fältets kommentar som visas på fliken *Fields*, i skrivskyddat tillstånd. Denna information visas som verktygstips när man håller muspekaren över fältetiketten i ett funktionsformulär.
- *Redigerbar*: avmarkera det här alternativet för att ställa in fältet som skrivskyddat (inte manuellt modifierbart) även när lagret är i redigeringsläge. Observera att om du markerar den här inställningen åsidosätts inte någon redigeringsbegränsning från leverantören. Det här alternativet kan styras av en datadefinierad egenskap med hjälp av knappen Data-defined override.
- *Återanvänd det senast angivna värdet*: kommer ihåg det senast angivna värdet i det här fältet och använder det som standard för nästa funktion som redigeras i lagret.
- *Label on top*: placerar fältnamnet ovanför eller bredvid widgeten i funktionsformuläret.

Begränsningar

Du kan begränsa det värde som ska infogas i fältet. Denna begränsning kan vara:

- *Not null*: kräver att användaren anger ett värde;
- *Unique*: garanterar att det infogade värdet är unikt i hela fältet;
- baserat på ett anpassat *uttryck*: t.ex. `not regexp_match(col0,'[^A-Za-z]')` kommer att säkerställa att värdet på fältet *col0* endast har bokstäver i alfabetet. En kort beskrivning kan läggas till för att hjälpa dig att komma ihåg begränsningen.

När ett värde läggs till eller redigeras i ett fält skickas det till de befintliga begränsningarna och:

- om det uppfyller alla krav visas en grön bock bredvid fältet i formuläret;
- om det inte uppfyller alla krav färgas fältet i gult eller orange och ett motsvarande kors visas bredvid widgeten. Du kan hålla muspekaren över korset för att påminna dig om vilka begränsningar som gäller för fältet och fixa värdet:
  - Ett gult kryss visas när den icke uppfyllda begränsningen är en icke tvingande begränsning (`mjuk begränsning`) och det inte hindrar dig från att spara ändringarna med ”fel” värden;
  - Ett orange kors kan inte ignoreras och gör att du inte kan spara dina ändringar förrän de uppfyller begränsningarna. Det visas när alternativet *Tvinga fram begränsning* är markerat (`hård begränsning`).

Standardvärden

- *Standardvärde*: för nya funktioner, fyller automatiskt i fältet som standard med ett fördefinierat värde eller ett *uttrycksbaserat värde*. Det kan du till exempel göra:
  - använd `$x`, `$length`, `$area` för att automatiskt fylla i ett fält med objektets X-koordinat, längd, area eller annan geometrisk information när det skapas;
  - öka ett fält med 1 för varje ny funktion med hjälp av `maximum("field")+1`;
  - spara datumet för skapandet av funktionen med hjälp av `now()`;
  - använda *variabler* i uttryck, vilket gör det enklare att t.ex. infoga operatörsnamnet (`@user_full_name`), sökvägen till projektfilen (`@project_path`), …

En förhandsgranskning av det resulterande standardvärdet visas längst ned i widgeten.

> **Observera**
>
> Alternativet `Default value` känner inte till värdena i något annat fält i den funktion som skapas, så det går inte att använda ett uttryck som kombinerar något av dessa värden, dvs. ett uttryck som `concat(field1, field2)` kanske inte fungerar.

- *Apply default value on update*: när ett funktionsattribut eller en geometri ändras beräknas standardvärdet på nytt. Detta kan vara praktiskt för att spara värden som den senaste användaren som ändrar data, senaste gången den ändrades …

Policys

med *Policies* kan du bestämma hur värden ska tilldelas fältet när *Splitting features* eller *Duplicating features* används:

- *Duplicera värden*: Behåller det befintliga värdet för fältet för de nya funktionerna.
- *Use Default Value*: Återställer fältet genom att räkna om dess *standardvärde*. Om det inte finns någon klausul för standardvärde behålls det befintliga värdet för de nya funktionerna.
- *Remove Value*: Återställer fältet till ett oinställt tillstånd.
- *Use Ratio Geometries*: Omräknar fältvärdet för alla delade delar genom att multiplicera det befintliga värdet med förhållandet mellan de delade delarnas längder eller ytor.

Redigera widgetar

Baserat på fälttypen bestämmer QGIS automatiskt och tilldelar en standardwidgetyp till fältet. Du kan sedan ersätta widgeten med en annan widget som är kompatibel med fälttypen. De tillgängliga widgetarna är:

- **Binär (BLOB)**: Denna funktion är endast tillgänglig för binära fält och visar som standard en etikett med storleken på den inbäddade datan, om den inte är tom. En rullgardinsmeny bredvid etiketten gör det möjligt att:
  - *Embed file*, ersätta eller fylla fältet
  - *Rensa innehåll*, tar bort alla data i fältet
  - *Spara innehåll till fil*, exporterar data som en fil på hårddisken

  Det är också möjligt att förhandsgranska den inbäddade binära filen i fältet, om den kombineras i en drag-och-släpp-form med t.ex. en *QML- eller HTML-widget*.
- **Kryssruta**: Visar en kryssruta vars tillstånd definierar det värde som ska infogas.
- **Klassificering**: Endast tillgängligt när en *categorized symbology* tillämpas på lagret, visar en kombinationsruta med klassernas värden.
- **Färg**: Visar en *color widget* som gör det möjligt att välja en färg; färgvärdet lagras som en html-notation i attributtabellen.
- **Datum/Tid**: Visar ett radfält som kan öppna en kalenderwidget för att ange ett datum, en tid eller båda. Kolumntypen måste vara text. Du kan välja ett anpassat format, popup-kalender etc.
- **Uppräkning**: Öppnar en kombinationsruta med fördefinierade värden hämtade från databasen. Detta stöds för närvarande endast av PostgreSQL-leverantören för fält av typen `enum`.
- **Bilaga**: Använder en ”Öppna fil”-dialog för att lagra filsökvägen i ett relativt eller absolut läge. Den kan användas för att visa dokumentsökvägen som en hyperlänk eller för att återge dokumentet i en särskild widget i formuläret. Dokumenttyper som stöds är bild, webbsida, ljud och video, och de filformat som stöds beror på operativsystemet. Du kan också konfigurera ett *externt lagringssystem* för att hämta/lagra resurser.

> **Tips**
>
> **Relativ sökväg i widget för bifogade filer**
>
> Om den sökväg som väljs med filbläddraren ligger i samma katalog som projektfilen `.qgs` eller lägre, konverteras sökvägarna till relativa sökvägar. Detta ökar portabiliteten för ett `.qgs`-projekt med multimediainformation.

- **Dold**: En kolumn med dolda attribut är osynlig. Användaren kan inte se innehållet i den.
- **Nyckel/värde**: Visar en tabell med två kolumner för att lagra uppsättningar av nyckel / värdepar i ett enda fält. Detta stöds för närvarande av PostgreSQL-leverantören för fält av typen `hstore`.
- **JSON-visning**: Visar JSON-data i en syntaxmarkerad textredigering eller i trädvy. Den här widgeten är för närvarande skrivskyddad. Flera alternativ finns tillgängliga för att ändra hur data visas. ”Standardvy” anger om widgeten ska visas i text- eller trädläge. ”Format JSON” har tre alternativ som endast är relaterade till trädvyn:
  - Indragen: Visa data i en mänskligt läsbar form med nya rader och fyra mellanslagstecken för indragning.
  - Kompakt: Visa data i en optimerad sträng med en radstorlek utan nya rader eller mellanslag.
  - Inaktiverad: Visa data så som de kommer från leverantören.
- **Lista**: Visar en tabell med en kolumn för att lägga till olika värden inom ett enda fält. Detta stöds för närvarande av PostgreSQL-leverantören för fält av typen `` array``.
- **Intervall**: Gör att du kan ställa in numeriska värden från ett specifikt intervall. Redigeringswidgeten kan vara antingen ett skjutreglage eller en snurrbox.

> **Observera**
>
> Vissa lager, t.ex. GeoPackage eller ESRI File Geodatabase, med fördefinierade **range Field Domains** kommer automatiskt att identifieras av QGIS och tilldelas en **Range**-widget för de relevanta fälten. Widgeten kommer att fyllas i med de minimi- och maximivärden som anges i domänen.

- **Relationsreferens**: Detta är den standardwidget som tilldelas referensfältet (dvs. den främmande nyckeln i det underordnade lagret) när en *relation* anges. Den ger direkt åtkomst till den överordnade funktionens formulär, som i sin tur innehåller listan och formuläret för dess underordnade funktioner. Antalet poster i widgeten kan begränsas för effektivitetens skull, och om limit inte anges kommer alla poster att laddas.
- **Text Edit** (standard): Detta öppnar ett textredigeringsfält som gör det möjligt att använda enkel text eller flera rader. Om du väljer flera rader kan du också välja html-innehåll.
- **Unika värden**: Du kan välja ett av de värden som redan används i attributtabellen. Om ”Editable” är aktiverat visas en rad med stöd för autokomplettering, annars används en kombinationsruta.
- **Uuid-generator**: Genererar ett skrivskyddat UUID-fält (Universally Unique Identifiers) om det är tomt.
- **Värdekarta**: En kombinationsruta med fördefinierade objekt. Värdet lagras i attributet och beskrivningen visas i kombinationsrutan. Du kan definiera värden manuellt eller ladda dem från ett lager eller en CSV-fil.

> **Observera**
>
> Vissa lager, t.ex. GeoPackage eller ESRI File Geodatabase, med fördefinierade **kodade fältdomäner** kommer automatiskt att identifieras av QGIS och tilldelas en **Value Map**-widget för de relevanta fälten.

- **Värderelation**: Erbjuder värden från en relaterad tabell i en kombinationsruta. Du kan välja lager, nyckelkolumn och värdekolumn. Flera alternativ finns tillgängliga för att ändra standardbeteendena: tillåt nollvärde, ordning efter värde, tillåt flera val och användning av automatisk ifyllning. Formulären kommer att visa antingen en rullgardinslista eller ett radredigeringsfält när kryssrutan för ifyllning är aktiverad.

Om ett lager som lagras i PostgreSQL, GeoPackage eller SpatiaLite är konfigurerat för att använda en widget för värderelation, men det nödvändiga lagret inte redan är inläst i projektet, kommer QGIS automatiskt att söka efter lagret i samma databas/anslutning.

12.1.10 Skarvar Egenskaper

Med fliken *Joins* kan du associera funktioner i det aktuella lagret med funktioner från ett annat laddat vektorlager (eller tabell). Kopplingen baseras på ett attribut som delas av lagren, i ett ett-till-ett-förhållande. Mer information om sammanfogningar finns i *Sammanfogning av funktioner mellan två lager*.

12.1.11 Egenskaper för hjälplagring

Det vanliga sättet att anpassa styling och märkning är att använda datadefinierade egenskaper enligt beskrivningen i *Inställning av datadefinierad åsidosättning*. Det kanske dock inte är möjligt om de underliggande data är skrivskyddade. Dessutom kan konfigurationen av dessa datadefinierade egenskaper vara mycket tidskrävande eller inte önskvärd! Om du t.ex. vill använda kartverktygen i *Verktygsfältet för etiketter* fullt ut måste du lägga till och konfigurera mer än 20 fält i din ursprungliga datakälla (X- och Y-positioner, rotationsvinkel, typsnitt, färg och så vidare).

Mekanismen Extra lagringsutrymme erbjuder en lösning på dessa begränsningar och besvärliga konfigurationer. Auxiliary Fields är ett omständligt sätt att automatiskt hantera och lagra dessa datadefinierade egenskaper (etiketter, diagram, symbologi…) i en SQLite-databas tack vare redigerbara joins. Detta gör att du kan lagra egenskaper för lager som inte är redigerbara.

En flik finns tillgänglig i dialogen för egenskaper för vektorlager för att hantera extra lagring:

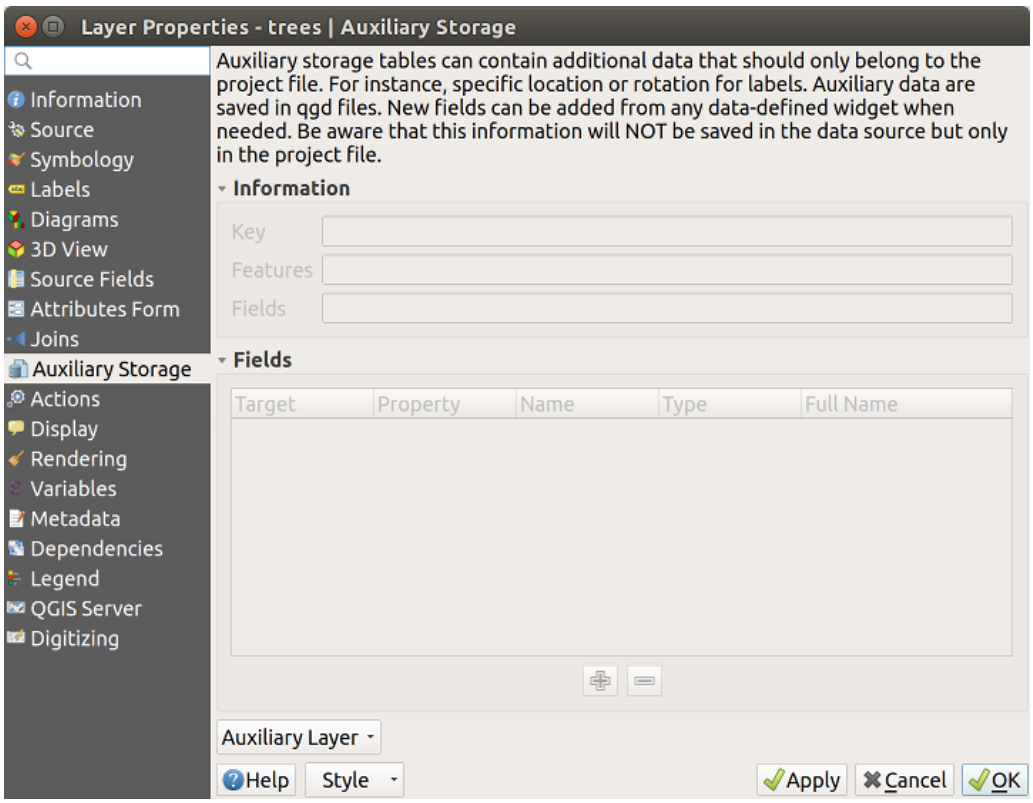


Fig. 12.47: Extra förvaringsflik

Labeling

Med tanke på att datakällan kan anpassas tack vare datadefinierade egenskaper utan att vara redigerbar, är kartverktygen för märkning som beskrivs i *Verktygsfältet för etiketter* alltid tillgängliga så snart märkningen är aktiverad.

I själva verket behöver det extra lagringssystemet ett extra lager för att lagra dessa egenskaper i en SQLite-databas (se *Databas för extra lagring*). Skapandeprocessen körs första gången du klickar på kartan medan ett kartverktyg för märkning är aktiverat. Därefter visas ett fönster där du kan välja den primärnyckel som ska användas för sammanfogning (för att säkerställa att objekten identifieras på ett unikt sätt):

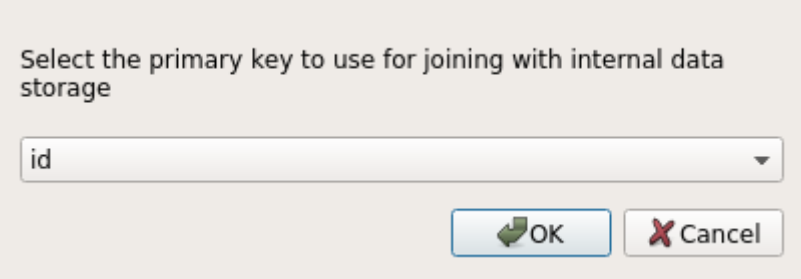


Fig. 12.48: Dialog för skapande av extra lager

Så snart ett extra lager har konfigurerats för den aktuella datakällan kan du hämta information om det på fliken:

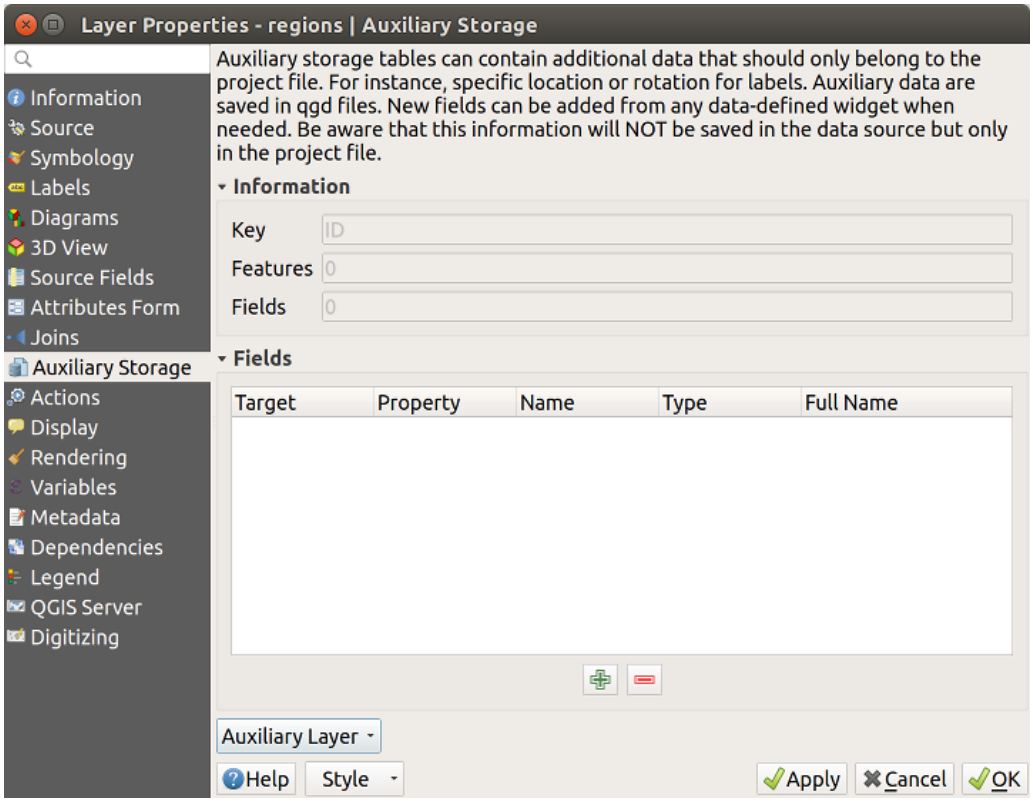


Fig. 12.49: Nyckel för extra lager

Hjälplagret har nu dessa egenskaper:

- den primära nyckeln är `ID`,
- det finns `0` funktioner som använder ett hjälpfält,

- finns det `0` hjälpfält.

Nu när hjälplagret har skapats kan du redigera lagrets etiketter. Klicka på en etikett medan kartverktyget Change Label är aktiverat, så kan du uppdatera stylingegenskaper som storlekar, färger och så vidare. Motsvarande datadefinierade egenskaper skapas och kan hämtas:

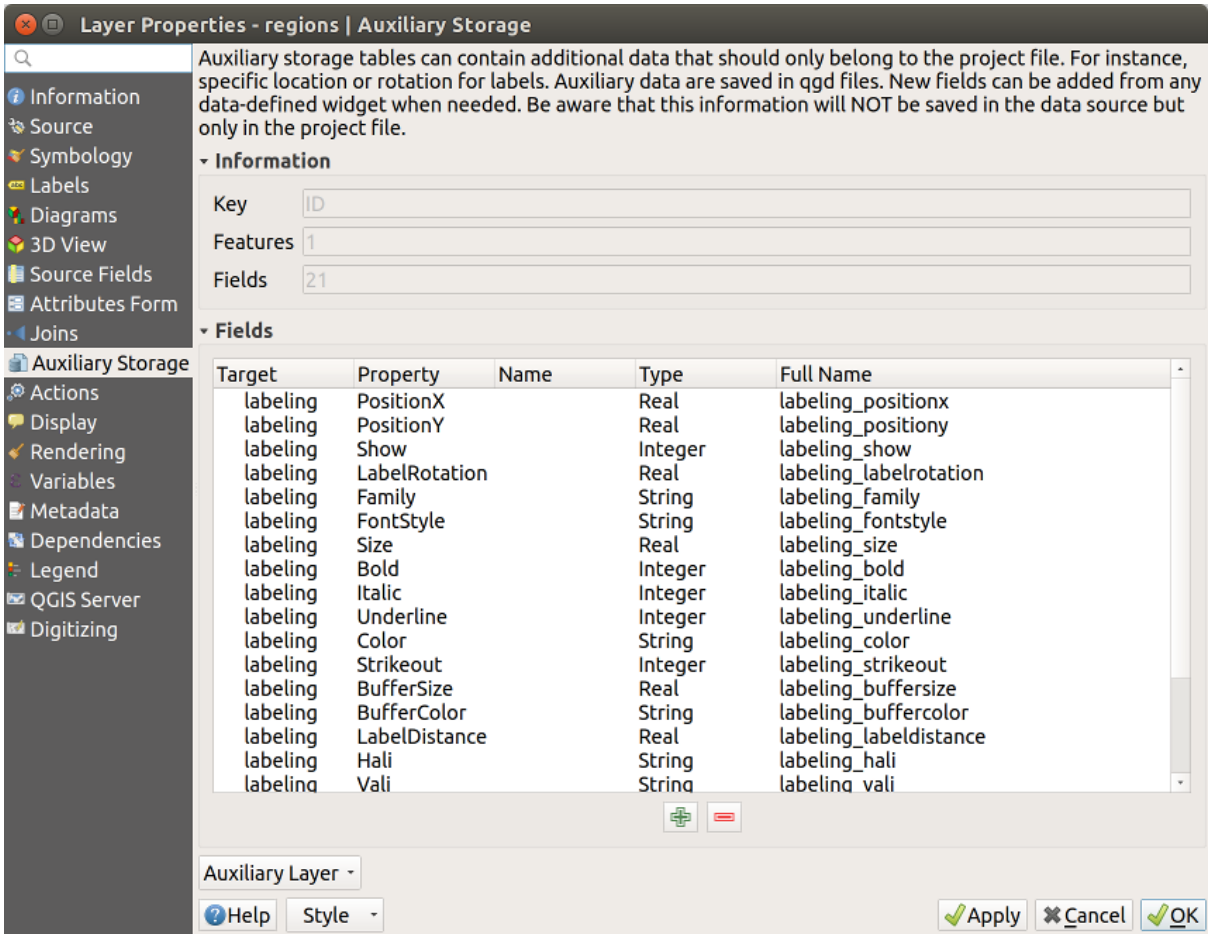


Fig. 12.50: Hjälpfältar

Som du kan se i figuren ovan skapas och konfigureras `21` fält automatiskt för märkning. Till exempel är hjälpfältstypen `FontStyle` en `String` och har namnet `labeling_fontstyle` i den underliggande SQLite-databasen. Det finns också `1` funktion som för närvarande använder dessa hjälpfält.

Observera att ikonen visas på egenskapsfliken *Labels*, vilket indikerar att de datadefinierade åsidosättningsalternativen är korrekt inställda:

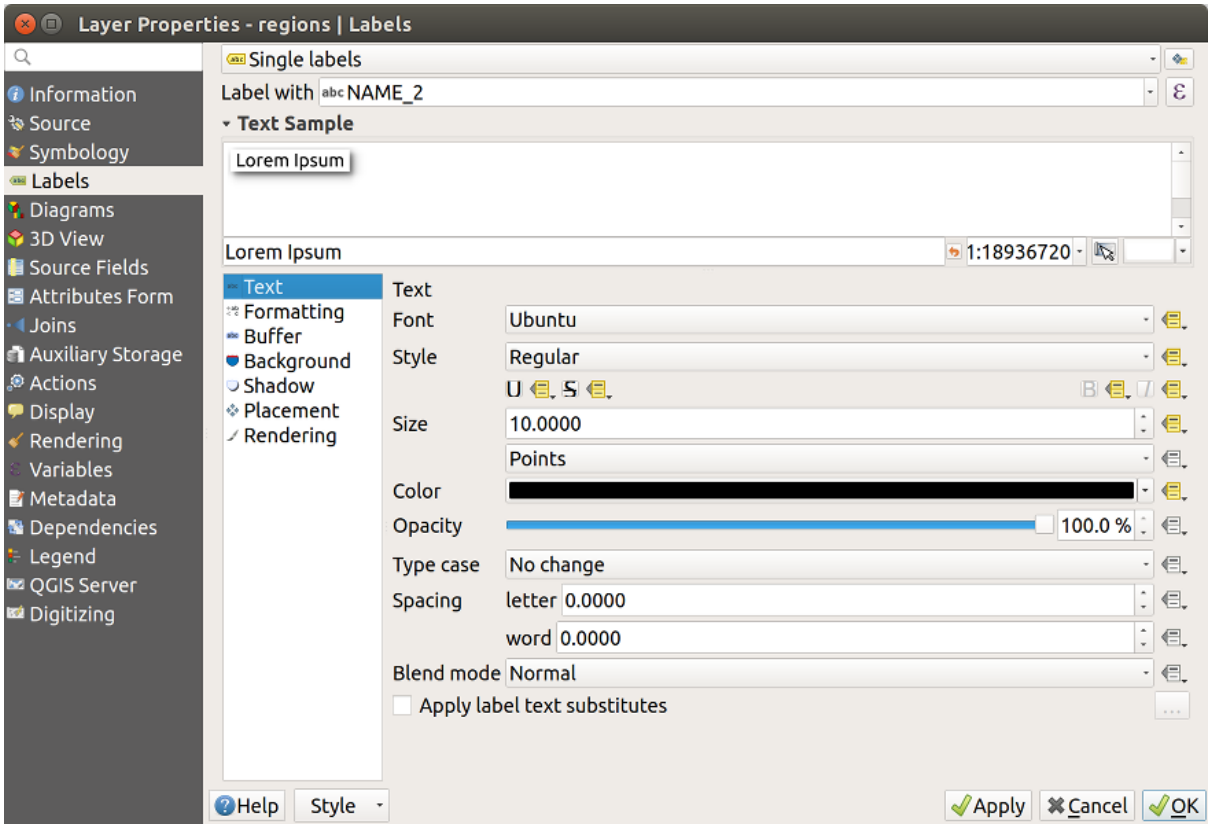


Fig. 12.51: Datadefinierade egenskaper skapas automatiskt

Annars finns det ett annat sätt att skapa ett hjälpfält för en specifik egenskap tack vare knappen Datadefinierad åsidosättning. Genom att klicka på *Lagra data i projektet* skapas automatiskt ett hjälpfält för fältet *Opacity*. Om du klickar på den här knappen och hjälplagret inte har skapats ännu visas först ett fönster (Fig. 12.48) för att välja den primärnyckel som ska användas för sammanfogning.

Symbologi

Liksom den metod som beskrivs ovan för att anpassa etiketter kan hjälpfält också användas för att stilisera symboler och diagram. Detta gör du genom att klicka på Data-defined override och välja *Store data in the project* för en specifik egenskap. Till exempel fältet *Fyll color*:

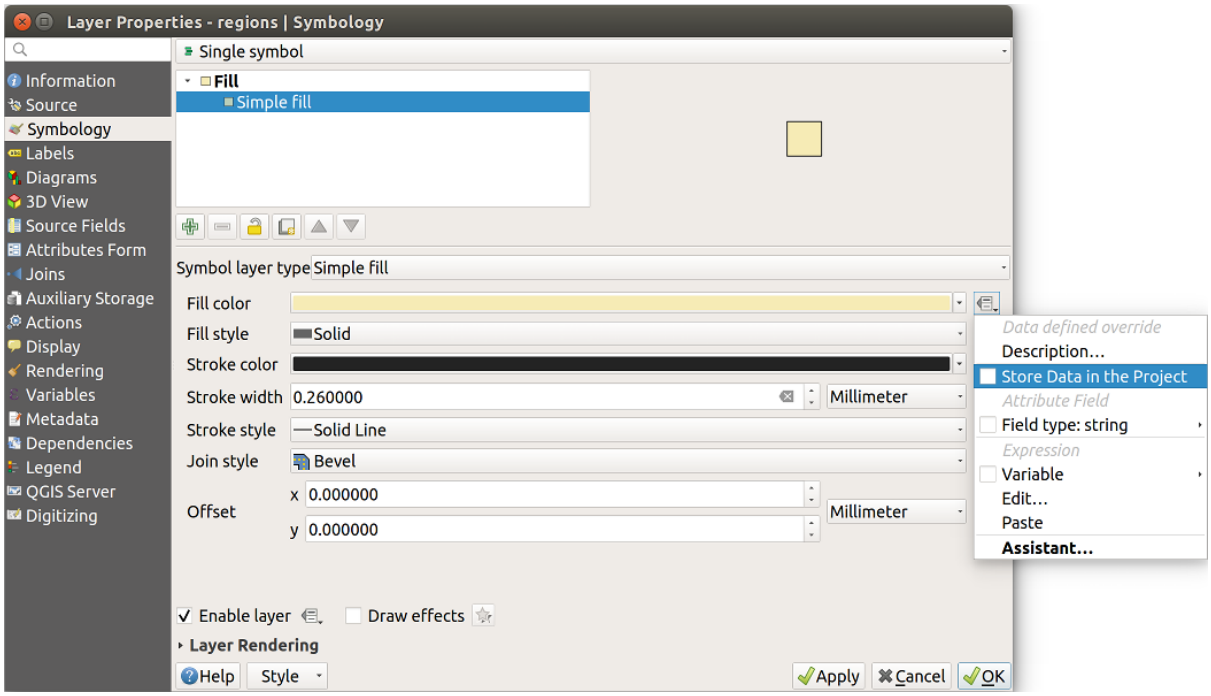


Fig. 12.52: Meny för datadefinierad egenskap för symbol

Det finns olika attribut för varje symbol (t.ex. fyllnadsstil, fyllnadsfärg, streckfärg osv.), så varje hjälpfält som representerar ett attribut behöver ett unikt namn för att undvika konflikter. När du har valt *Lagra data i projektet* öppnas ett fönster som visar fältets *Typ* och uppmanar dig att ange ett unikt namn för hjälpfältet. När du t.ex. skapar ett hjälpfält av typen *Fyll color* öppnas följande fönster:

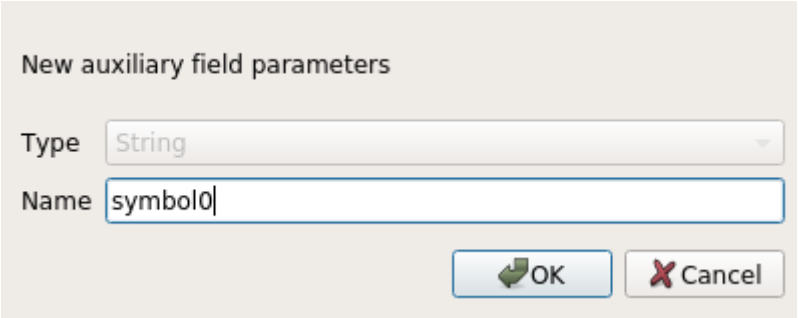


Fig. 12.53: Namn på hjälpfältet för en symbol

När hjälpfältet har skapats kan det hämtas på fliken Auxiliary Storage:

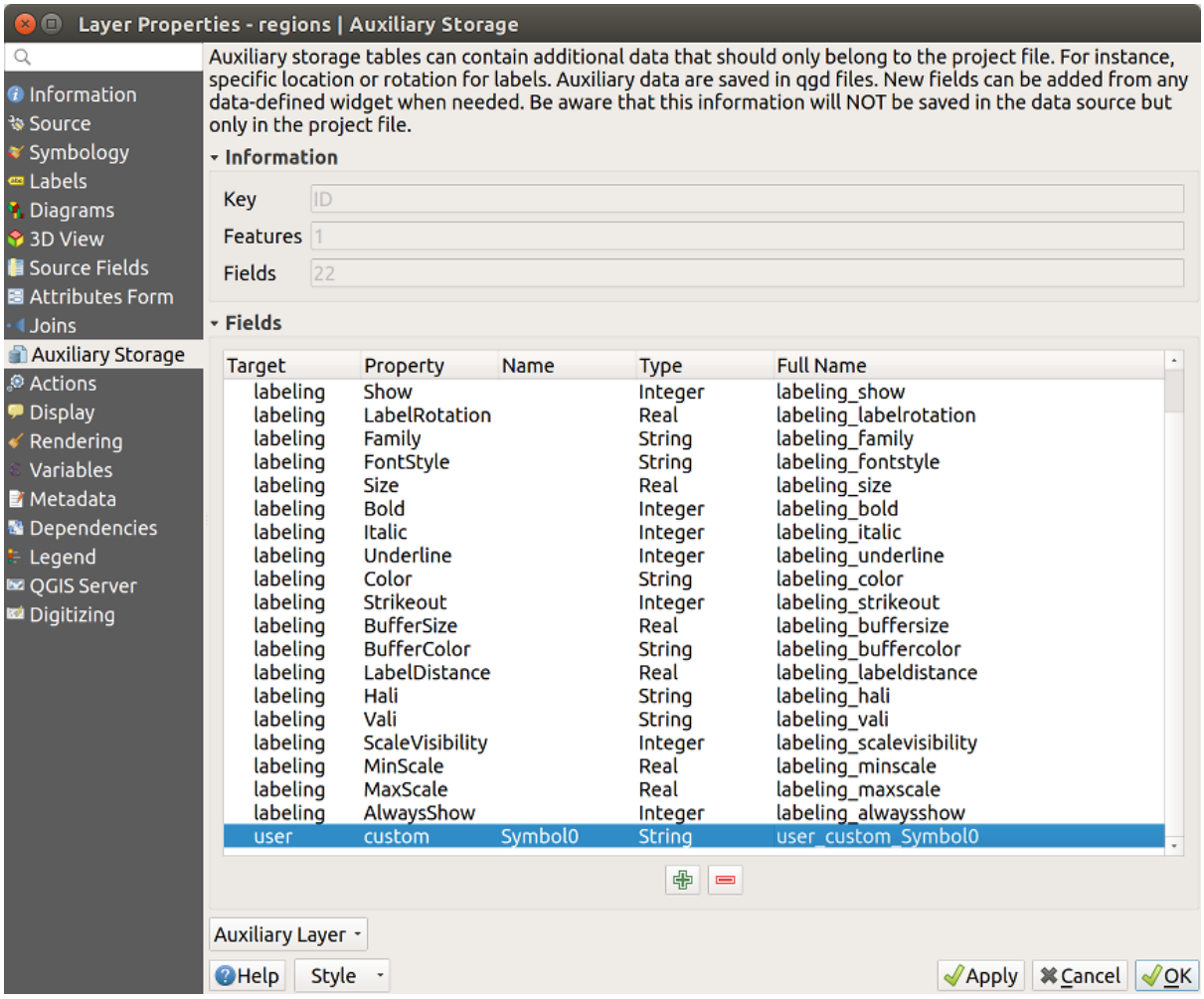


Fig. 12.54: Symbol för hjälpfält

Attributtabell och widgets

Hjälpfält kan redigeras med hjälp av *attributtabell*. Alla hjälpfält är dock inte synliga i attributtabellen från början.

Extra fält som representerar attribut för ett lagers symbologi, märkning, utseende eller diagram visas automatiskt i attributtabellen. Undantaget är attribut som kan modifieras med hjälp av *Label Toolbar* som är dolda som standard. Hjälpfält som representerar en `Färg` har en widget **Färg** inställd som standard, annars är hjälpfälten inställda som standard på widgeten **Textredigering**.

Hjälpfält som representerar attribut som kan ändras med hjälp av verktygsfältet *Label* är **Dolda** i attributtabellen som standard. Du gör ett fält synligt genom att öppna fliken *Attribute Form properties* och ändra värdet för ett hjälpfält *Widget Type* från **Hidden** till ett annat relevant värde. Ändra t.ex. **auxiliary_storage_labeling_size** till **Text Edit** eller ändra **auxiliary_storage_labeling_color** till widgeten **Color**. Dessa fält kommer nu att vara synliga i attributtabellen.

Hjälpfälten i attributtabellen kommer att se ut som på följande bild:

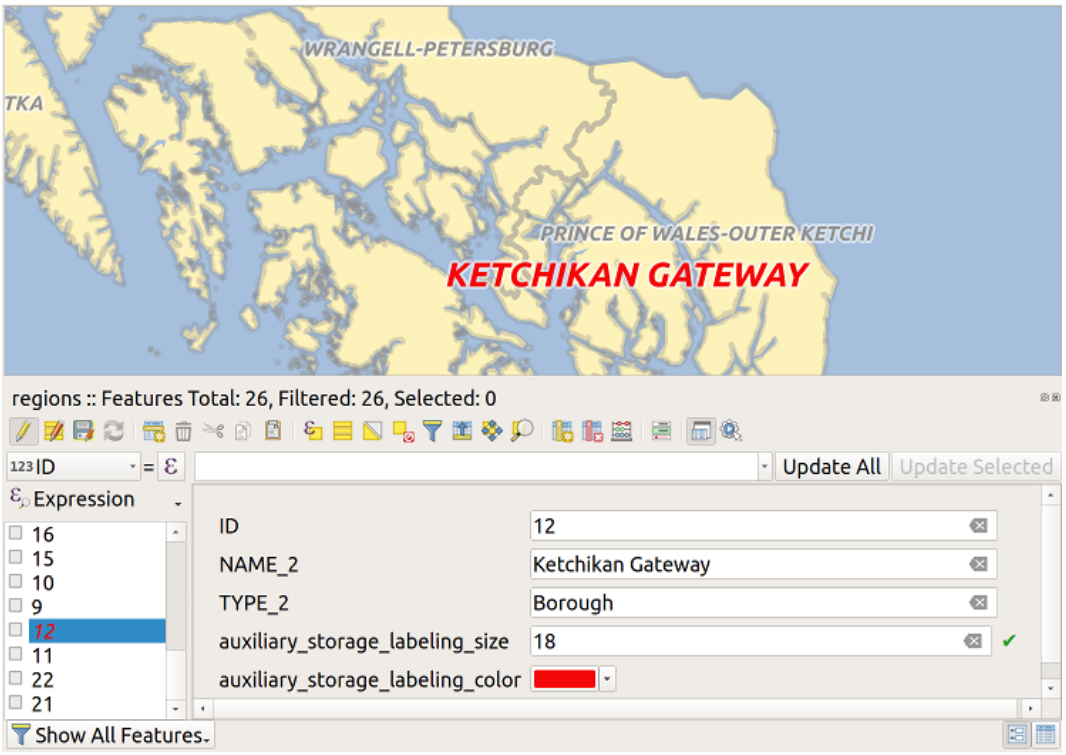


Fig. 12.55: Formulär med hjälpfält

Hantering

I menyn *Auxiliary Layer* kan du hantera hjälpfälten:

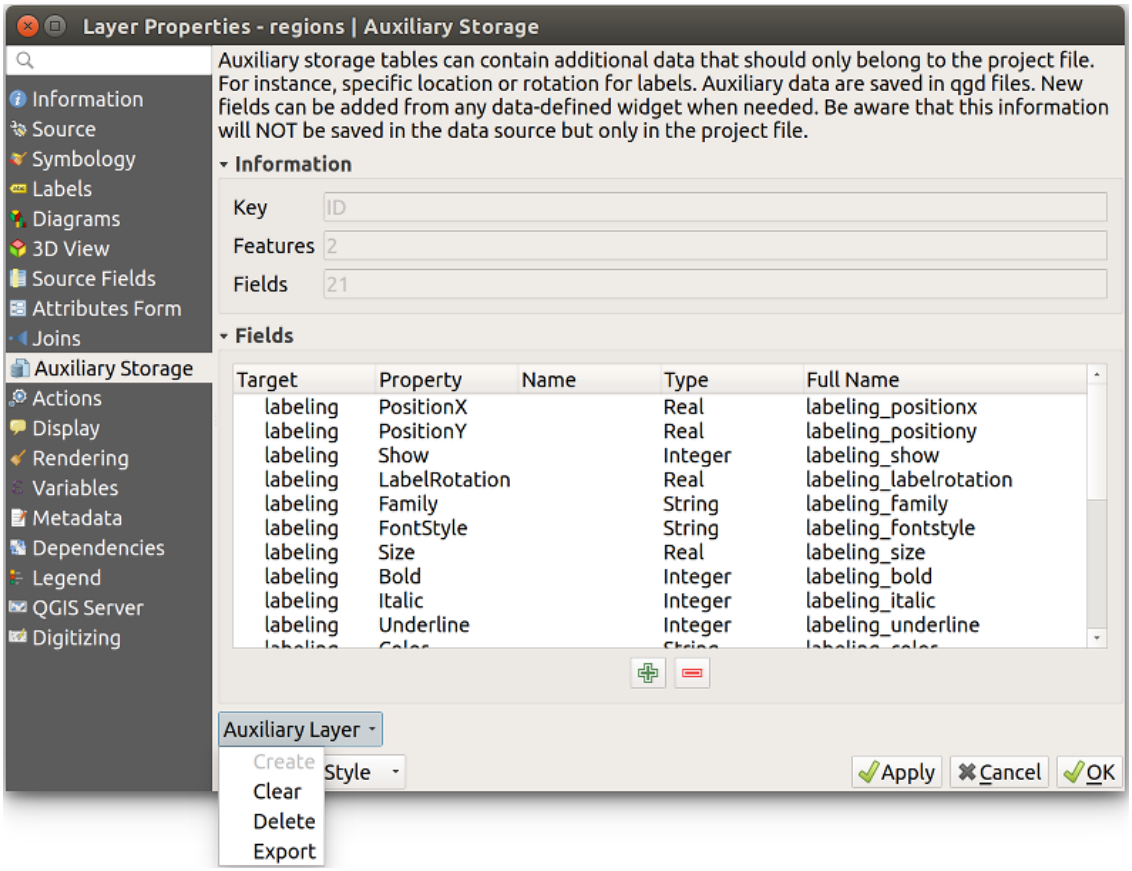


Fig. 12.56: Hantering av extra lager

Det första alternativet *Create* är inaktiverat i det här fallet eftersom hjälplagret redan är skapat. Men om du gör ett nytt arbete kan du använda den här åtgärden för att skapa ett extra lager. Som förklaras i *Labeling* behövs då en primärnyckel.

Åtgärden *Clear* gör det möjligt att behålla alla hjälpfält, men ta bort deras innehåll. På så sätt kommer antalet funktioner som använder dessa fält att sjunka till `0`.

Åtgärden *Delete* tar bort hjälplagret helt och hållet. Med andra ord raderas motsvarande tabell från den underliggande SQLite-databasen och anpassningen av egenskaperna går förlorad.

Slutligen gör åtgärden *Exportera* det möjligt att spara hjälplagret som en *nytt vektorlager*. Observera att geometrier inte lagras i hjälplagret. I det här fallet exporteras dock geometrierna även från den ursprungliga datakällan.

Databas för extra lagring

När du sparar ditt projekt med formatet `.qgs` sparas SQLite-databasen som används för extra lagring på samma plats men med tillägget `.qgd`.

För enkelhetens skull kan ett arkiv användas istället tack vare formatet `.qgz`. I detta fall är både filerna `.qgd` och `.qgs` inbäddade i arkivet.

12.1.12 Åtgärder Egenskaper

Fliken *Åtgärder* ger möjlighet att utföra en åtgärd baserad på attributen för en funktion. Detta kan användas för att utföra ett antal olika åtgärder, t.ex. köra ett program med argument som bygger på attributen för en funktion eller skicka parametrar till ett webbrapporteringsverktyg.

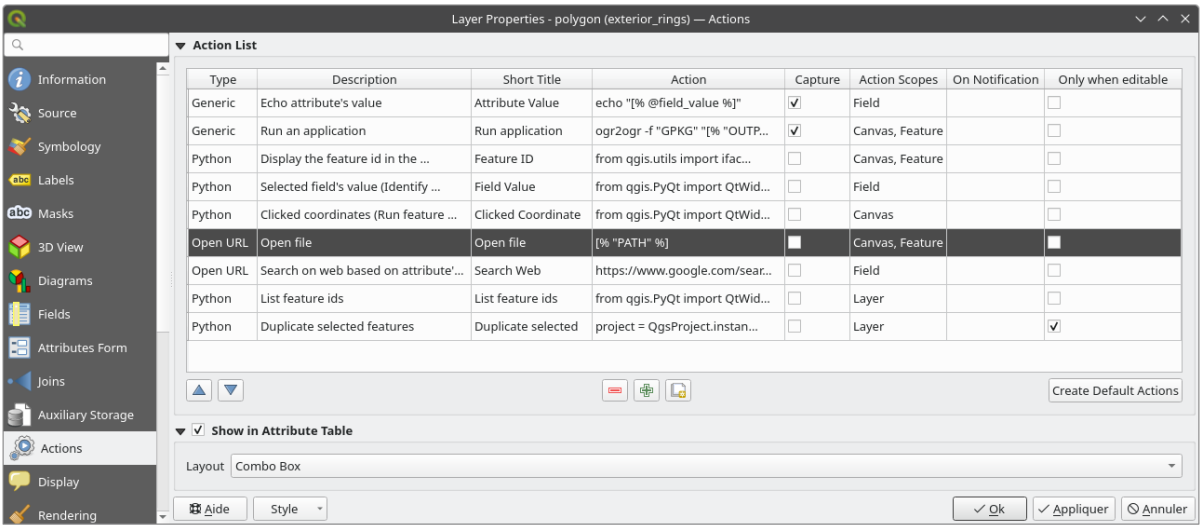


Fig. 12.57: Översikt över åtgärdsdialog med några exempel på åtgärder

Actions är användbara när du ofta vill köra en extern applikation eller visa en webbsida baserat på ett eller flera värden i ditt vektorlager. De finns av olika typer och kan användas så här:

- åtgärderna *Generic*, *macOS*, *Windows* och *Unix* startar en extern process.
- *Python* åtgärder exekverar ett Python-uttryck.
- *Generic* och *Python* åtgärder är synliga överallt.
- åtgärderna *macOS*, *Windows* och *Unix* är endast synliga på respektive plattform (du kan t.ex. definiera tre ”Edit”-åtgärder för att öppna en editor och användarna kan endast se och utföra en ”Edit”-åtgärd för sin plattform för att köra editorn).
- *Öppna URL*: Använder en HTTP GET-begäran för att öppna en angiven URL.
- *Sänd URL (urlencoded eller JSON)*: Samma som åtgärden *Open URL* men med en HTTP POST-begäran. Data skickas till en URL med ”application/x-www-form-urlencoded” eller ”application/json” om filen är en giltig JSON.

  Ett exempel på åtgärdskallelse kan vara:

  ```
  http://localhost:8000?/[% url_encode(map('file', 'index.php')) %]
  ```

- *Sänd URL (multipart)*: Samma som åtgärden *Open URL* men med en HTTP POST-begäran. Data skickas till en URL med hjälp av ”multipart/form-data”.

Det finns flera exempel i dialogrutan. Du kan ladda dem genom att klicka på *Create Default Actions*. Om du vill redigera något av exemplen dubbelklickar du på dess rad. Ett exempel är att utföra en sökning baserad på ett attributvärde. Detta koncept används i följande diskussion.

Med *Show in Attribute Table* kan du visa de markerade funktionsomfattade åtgärderna i attributtabellens dialogruta, antingen som *Combo Box* eller som *Separate Buttons* (se *Konfigurera kolumnerna*).

Definiera åtgärder

För att definiera en attributåtgärd öppnar du dialogrutan vector *Lageregenskaper* och klickar på fliken *Åtgärder*. På fliken *Actions* klickar du på Lägg till en ny åtgärd för att öppna dialogrutan *Redigera åtgärd*.

Välj åtgärden *Type* och ange ett beskrivande namn för åtgärden. Själva åtgärden måste innehålla namnet på den applikation som kommer att köras när åtgärden anropas. Du kan lägga till ett eller flera attributfältsvärden som argument till applikationen. När åtgärden anropas kommer alla teckenuppsättningar som börjar med `%` följt av namnet på ett fält att ersättas med värdet för det fältet. Specialtecknen `%%` ersätts med värdet för det fält som valdes från tabellen för identifieringsresultat eller attribut (se *Använda åtgärder*). Dubbla citationstecken kan användas för att gruppera text till ett enda argument för programmet, skriptet eller kommandot. Dubbla citattecken ignoreras om de föregås av ett backslash.

Åtgärder kan anropa en enda process med argument, så booleska operatorer (t.ex. `&`, `&&`, `;`, `|`) fungerar inte. I UNIX-liknande operativsystem kan flera kommandon köras via `bash -c`.

Med *Action Scopes* kan du definiera var åtgärden ska vara tillgänglig. Du har följande valmöjligheter:

1. *Field*: åtgärden är tillgänglig när du högerklickar i cellen i attributtabellen, i funktionsformuläret och i standardåtgärdsknappen i huvudverktygsfältet.
2. *Feature*: åtgärden är tillgänglig när du högerklickar i cellen i attributtabellen.
3. *Canvas*: åtgärd är tillgänglig i huvudåtgärdsknappen i verktygsfältet.
4. *Form*: Åtgärden är endast tillgänglig i ett funktionsformulär som utformats med hjälp av läget *drag-and-drop*.
5. *Layer*: åtgärd är tillgänglig via åtgärdsknappen i attributtabellens verktygsfält. Tänk på att den här typen av åtgärd omfattar hela lagret och inte de enskilda funktionerna.

Om du har fältnamn som är delsträngar av andra fältnamn (t.ex. `col1` och `col10`), bör du ange detta genom att omge fältnamnet (och tecknet *% c) med hakparenteser (t.ex. `[%col10]`*). Detta förhindrar att fältnamnet `%col10` misstas för fältnamnet `%col1` med ett `0` i slutet. Parenteserna kommer att tas bort av QGIS när fältets värde ersätts. Om du vill att det ersatta fältet ska omges av hakparenteser använder du en andra uppsättning så här: `[[%col10]]`.

Med hjälp av verktyget *Identify Features* kan du öppna dialogrutan *Identify Results*. Den innehåller ett *(Derived)*-objekt som innehåller information som är relevant för lagertypen. Värdena i det här objektet kan nås på samma sätt som de andra fälten genom att det härledda fältnamnet föregås av `(Derived).`. Ett punktlager har t.ex. fälten `X` och `Y`, och värdena i dessa fält kan användas i åtgärden med `%(Derived).X` och `%(Derived).Y`. De härledda attributen är endast tillgängliga från dialogrutan *Identify Results*, inte från dialogrutan *Attribute Table*.

Två exempel på åtgärder visas nedan:

- `konqueror https://www.google.com/search?q=%nam`
- `konqueror https://www.google.com/search?q=%%`

I det första exemplet anropas webbläsaren konqueror och får en URL att öppna. URL:en utför en Google-sökning på värdet i fältet `nam` från vårt vektorlager. Observera att den applikation eller det skript som anropas av åtgärden måste finnas i sökvägen, eller så måste du ange hela sökvägen. För att vara säker kan vi skriva om det första exemplet som: `/opt/kde3/bin/konqueror https://www.google.com/search?q=%nam`. Detta kommer att säkerställa att konqueror-applikationen kommer att köras när åtgärden åberopas.

I det andra exemplet används funktionen %% notation, som inte är beroende av ett visst fält för sitt värde. När åtgärden aktiveras ersätts %% med värdet för det valda fältet i tabellen för identifieringsresultat eller attribut.

Duplicering av åtgärder

I QGIS kan du duplicera befintliga åtgärder. För att duplicera en attributåtgärd öppnar du dialogen vector *Lageregenskaper* och klickar på fliken *Actions*. På fliken *Actions* klickar du på Duplicera en åtgärd för att öppna dialogrutan *Duplicera åtgärd*. Du måste ha valt minst en befintlig åtgärd för att kunna skapa en kopia.

Gör eventuella ändringar i den dialogruta som visas. Se *Definiera åtgärder* för ytterligare information. När du är klar trycker du på *OK* för att skapa en kopia av åtgärden med alla ändringar du gjort. Om du inte har redigerat beskrivningen, eller om du har ändrat den så att den är identisk med beskrivningen av någon annan befintlig åtgärd, kommer ”_1” att läggas till i slutet av beskrivningen.

Använda åtgärder

QGIS erbjuder många sätt att utföra åtgärder som du har aktiverat på ett lager. Beroende på deras inställningar kan de vara tillgängliga:

- i rullgardinsmenyn på knappen Run Feature Action från verktygsfältet *Attributes* eller dialogrutan *Attribute table*;
- när du högerklickar på en funktion med verktyget Identify Features (se *Identifiering av funktioner* för mer information);
- från panelen *Identify Results*, under avsnittet *Actions*;
- som objekt i kolumnen *Actions* i dialogrutan *Attributtabell*.

Om du anropar en åtgärd som använder notationen `%%` högerklickar du på det fältvärde i dialogrutan *Identify Results* eller dialogrutan *Attribute Table* som du vill skicka till programmet eller skriptet.

Här är ett annat exempel som hämtar data från ett vektorlager och infogar dem i en fil med hjälp av bash och kommandot `echo` (så det fungerar bara på eller kanske). Lagret i fråga har fält för ett artnamn `taxon_name`, latitud `lat` och longitud `long`. Vi skulle vilja kunna göra ett spatialt urval av lokaler och exportera dessa fältvärden till en textfil för den valda posten (visas i gult i QGIS-kartområdet). Här är åtgärden för att uppnå detta:

```
bash -c "echo \"%taxon_name %lat %long\" >> /tmp/species_localities.txt"
```

Efter att ha valt några platser och kört åtgärden på var och en av dem, kommer utdatafilen att visa något liknande detta:

```
Acacia mearnsii -34.0800000000 150.0800000000
Acacia mearnsii -34.9000000000 150.1200000000
Acacia mearnsii -35.2200000000 149.9300000000
Acacia mearnsii -32.2700000000 150.4100000000
```

Som en övning kan vi skapa en åtgärd som gör en Google-sökning på lagret `lakes`. Först måste vi bestämma den URL som krävs för att utföra en sökning på ett nyckelord. Detta görs enkelt genom att bara gå till Google och göra en enkel sökning och sedan ta tag i URL: en från adressfältet i din webbläsare. Från denna lilla ansträngning ser vi att formatet är https://www.google.com/search?q=QGIS, där `QGIS` är söktermen. Med denna information kan vi gå vidare:

1. Kontrollera att lagret `lakes` är laddat.
2. Öppna dialogrutan *Lageregenskaper* genom att dubbelklicka på lagret i teckenförklaringen eller högerklicka och välj *Egenskaper* från popup-menyn.
3. Klicka på fliken *Actions*.
4. Klicka på Lägg till en ny åtgärd.
5. Välj åtgärdstypen *Open URL*,
6. Ange ett namn för åtgärden, till exempel `Google Search`.
7. Dessutom kan du lägga till ett *Kort namn* eller till och med en *Icon*.

8. Välj *Action Scopes*. Se *Definiera åtgärder* för ytterligare information. Lämna standardinställningarna för detta exempel.
9. För åtgärden lägger du till den URL som används för att göra en Google-sökning, upp till men inte inklusive sökordet: `https://www.google.com/search?q=`
10. Texten i fältet *Action* ska nu se ut så här:

```
https://www.google.com/search?q=
```

11. Klicka på rullgardinsmenyn som innehåller fältnamnen för lagret `lakes`. Den finns precis till vänster om knappen *Insert*.
12. Välj *NAMES* i rullgardinsmenyn och klicka på *Insert*.
13. Din åtgärdstext ser nu ut så här:

```
https://www.google.com/search?q=[%NAMES%]
```

14. För att slutföra och lägga till åtgärden klickar du på knappen *OK*.

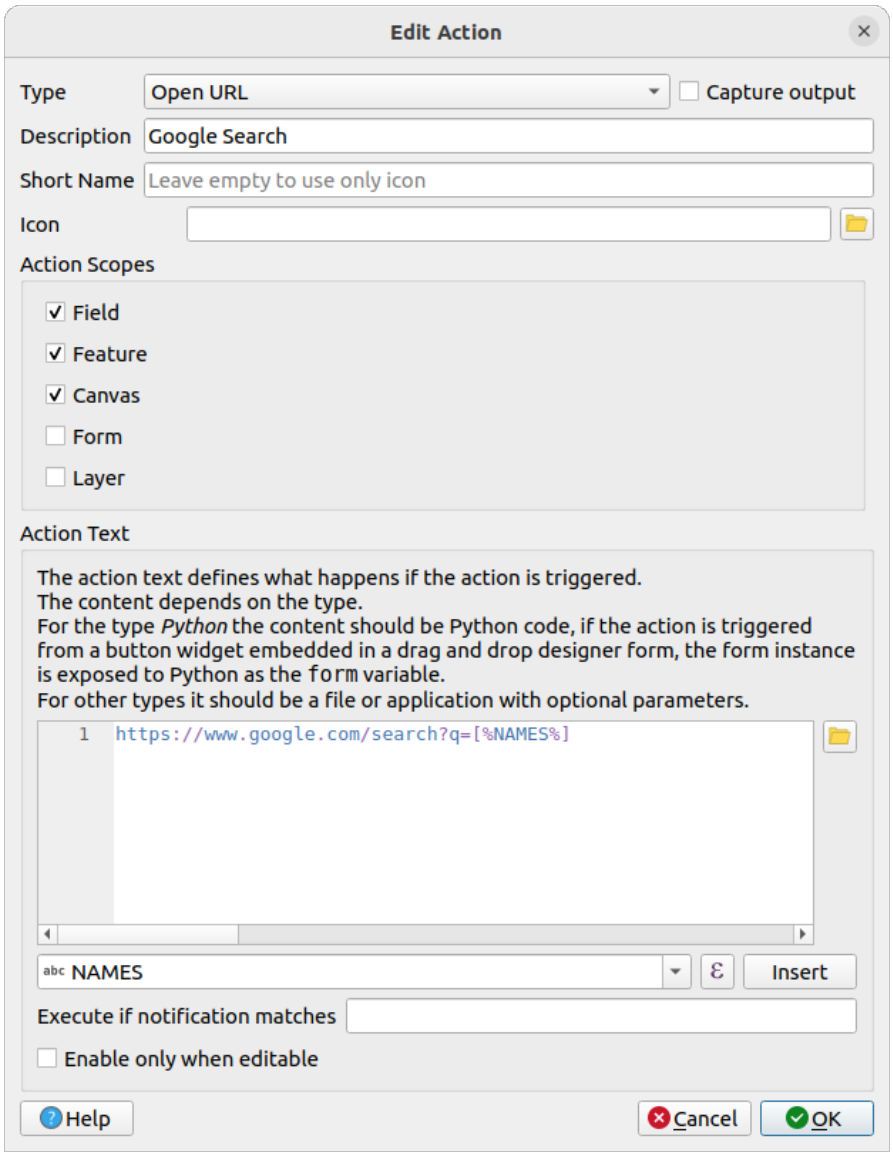


Fig. 12.58: Dialogrutan Redigera åtgärd konfigurerad med exemplet

Detta avslutar åtgärden och den är klar att användas.

Stäng dialogrutan *Lageregenskaper* och zooma in till ett område av intresse. Se till att lagret `lakes` är aktivt och identifiera en sjö. I resultatrutan ser du nu att vår åtgärd är synlig:

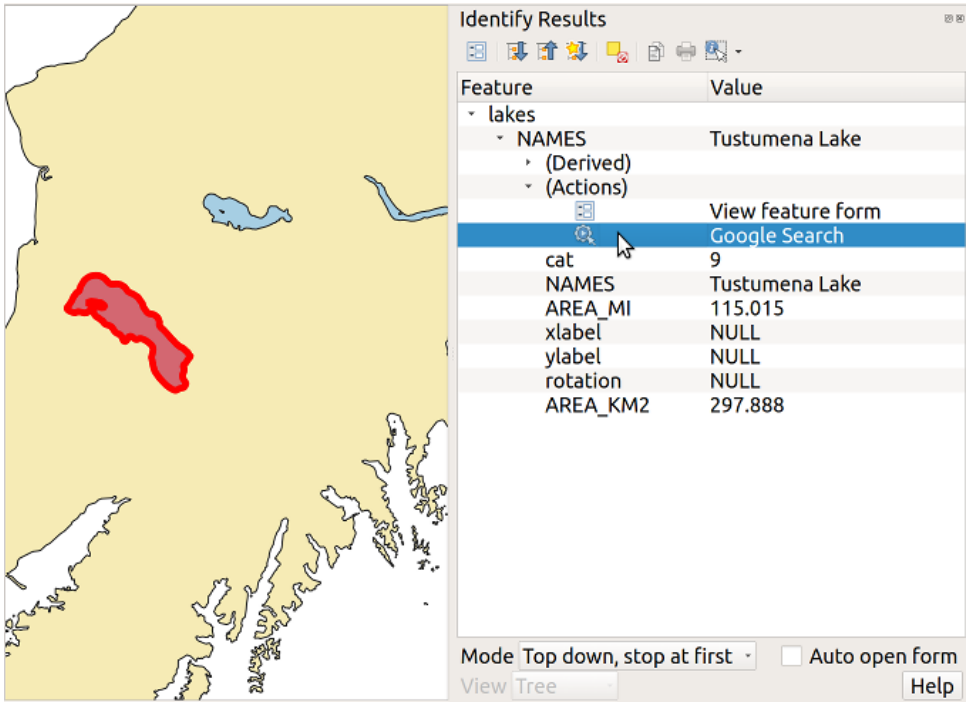


Fig. 12.59: Välj funktion och välj åtgärd

När vi klickar på åtgärden öppnas vår standardwebbläsare och vi navigerar till URL:en https://www.google.com/search?q=Tustumena. Det är också möjligt att lägga till ytterligare attributfält till åtgärden. Därför kan du lägga till en + i slutet av åtgärdstexten, välja ett annat fält och klicka på *Insert Field*. I det här exemplet finns det helt enkelt inget annat fält som skulle vara meningsfullt att söka efter.

Du kan definiera flera åtgärder för ett lager, och varje åtgärd visas i dialogrutan *Identify Results*.

Du kan också starta åtgärder från attributtabellen genom att markera en rad, högerklicka och sedan välja åtgärd från popup-menyn.

Det finns många olika användningsområden för åtgärder. Om du till exempel har ett punktlager som innehåller platser för bilder eller foton tillsammans med ett filnamn, kan du skapa en åtgärd för att starta en visare för att visa bilden. Du kan också använda åtgärder för att starta webbaserade rapporter för ett attributfält eller en kombination av fält genom att ange dem på samma sätt som i exemplet med Google-sökningen.

Vi kan också göra mer komplexa exempel, t.ex. med hjälp av **Python**-åtgärder.

När vi skapar en åtgärd för att öppna en fil med ett externt program kan vi vanligtvis använda absoluta sökvägar eller så småningom relativa sökvägar. I det andra fallet är sökvägen relativ till platsen för det externa programmets körbara fil. Men hur är det om vi behöver använda relativa sökvägar, i förhållande till det valda lagret (ett filbaserat lager, som Shapefile eller SpatiaLite)? Följande kod kommer att göra tricket:

```
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)

(fortsättning från föregående sida)

```
import subprocess
subprocess.Popen( [command, image ] )
```

Vi måste bara komma ihåg att åtgärden är av typen *Python* och att variablerna *command* och *imagerelpath* måste ändras för att passa våra behov.

Men hur är det om den relativa sökvägen måste vara relativ till den (sparade) projektfilen? Koden för Python-åtgärden skulle vara:

```
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 ] )
```

Ett annat exempel på en Python-handling är den som gör att vi kan lägga till nya lager i projektet. Till exempel kommer följande exempel att lägga till en vektor respektive ett raster i projektet. Namnen på de filer som ska läggas till i projektet och de namn som ska ges till lagren är datadrivna (*filnamn* och *lagernamn* är kolumnnamn i tabellen med attribut för vektorn där åtgärden skapades):

```
qgis.utils.iface.addVectorLayer('/yourpath/[% "filename" %].shp',
  '[% "layername" %]', 'ogr')
```

För att lägga till ett raster (en TIF-bild i det här exemplet) blir det:

```
qgis.utils.iface.addRasterLayer('/yourpath/[% "filename" %].tif',
  '[% "layername" %]')
```

12.1.13 Egenskaper för display

Fliken *Display* hjälper dig att konfigurera fält som ska användas för att identifiera funktioner:

- *Visningsnamn*: baserat på ett fält eller ett *uttryck*. Som standard sätts detta till det första fältet i lagret om inget fält med komponenten `<name>` finns. Detta används som:
  - etiketten som visas ovanpå funktionsinformationen i resultaten *Identifiera verktyg*
  - fältet som används i *locator bar* när du letar efter funktioner i alla lager
  - identifieraren för funktionen i attributtabellen *formulärvy*
  - objektets identifierare när kartan eller layouten exporteras till ett lagerformat, t.ex. Geospatial PDF
  - information om karttips, dvs. det meddelande som visas i kartbilden när man håller muspekaren över en funktion i det aktiva lagret med ikonen :sup:`Show Map Tips` intryckt. Gäller när *Enable Map Tips* är aktiv och ingen *HTML Map Tip* är inställd.
- *Enable Map Tips* styr om karttips ska visas för lagret
- *HTML Map Tip* ger en komplex och fullständig HTML-textredigerare för karttips, som blandar QGIS-uttryck och html-stilar och -taggar (multilinjer, teckensnitt, bilder, hyperlänkar, tabeller, …). Du kan kontrollera resultatet av ditt kodprov i ramen *Preview* (även praktiskt för förhandsgranskning av *Display name*). Dessutom kan du välja och redigera befintliga uttryck med hjälp av knappen *Insert/Edit Expression*.

Observera

Förstå beteendet hos knappen *Insert/Edit Expression* (infoga/redigera uttryck)

Om du markerar text inom ett uttryck (mellan ”[%” och ”%]”), eller om ingen text är markerad men markören befinner sig inom ett uttryck, markeras hela uttrycket automatiskt för redigering. Om markören eller en markerad text befinner sig utanför ett uttryck öppnas dialogrutan med markeringen.

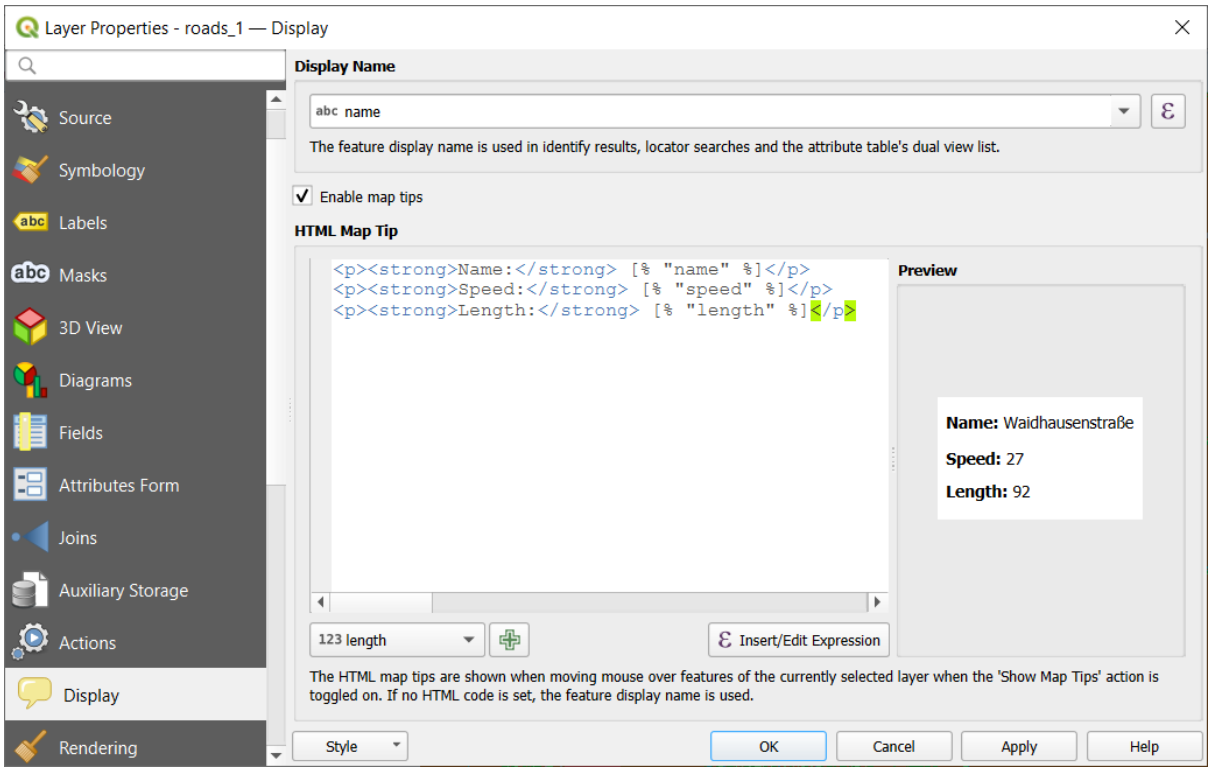


Fig. 12.60: HTML-kod för karttips

För att visa karttips:

1. Välj menyalternativet *Visa* ► *Visa karttips* eller klicka på ikonen Visa karttips i *Attributes Toolbar*.
2. Se till att det lager du vill använda är aktivt och att egenskapen *Enable Map Tips* är markerad.
3. Flytta muspekaren över en funktion så visas motsvarande information där.

Map tip är en funktion som gäller för flera lager, vilket innebär att när den har aktiverats förblir den aktiverad och gäller för alla lager i projektet som är aktiverade med Map tip tills den stängs av.

Fig. 12.61: Kartspets gjord med HTML-kod

12.1.14 Rendering av egenskaper

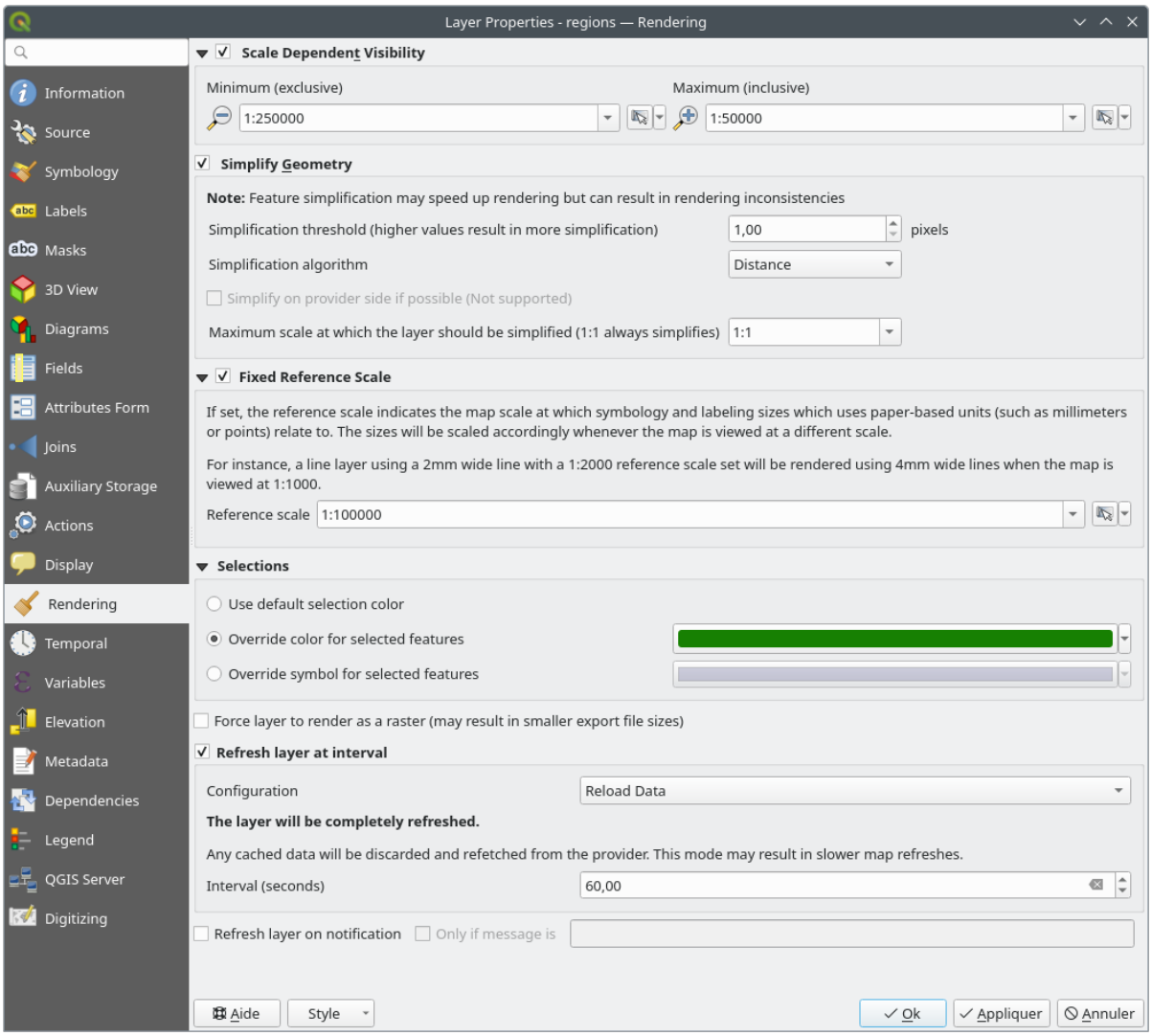


Fig. 12.62: Dialogrutan Egenskaper för lagerrendering

Fliken *Rendering* erbjuder följande egenskaper:

- Under *Skalaberoende synlighet* kan du ställa in skalorna *Maximum (inklusive)* och *Minimum (exklusive)* och definiera ett skalområde där objekten ska vara synliga. Utanför detta intervall är de dolda. Knappen Set to current canvas scale hjälper dig att använda den aktuella kartans canvasskala som gräns för synlighetsintervallet. Se *Väljare för synlighetsskala* för mer information.

  > **Observera**
  >
  > Du kan också aktivera skalberoende synlighet på ett lager från panelen *Layers*: högerklicka på lagret och välj *Set Layer Scale Visibility* i snabbmenyn.

- QGIS erbjuder stöd för generalisering av funktioner i realtid. Detta kan förbättra renderingstiderna när man ritar många komplexa funktioner i små skalor. Den här funktionen kan aktiveras eller inaktiveras i lagerinställningarna med hjälp av alternativet *Simplify geometry*. Det finns också en global inställning som aktiverar generalisering som standard för nytillkomna lager (se *global simplification* för mer information).

> **Observera**
>
> Generalisering av funktioner kan i vissa fall ge upphov till artefakter i renderingsresultatet. Det kan t.ex. handla om sprickor mellan polygoner och felaktig rendering vid användning av offsetbaserade symbollager.

- Den okontrollerade *Fixed reference scale* anger den kartskala som symbologi- och märkningsstorlekar som använder pappersbaserade enheter (t.ex. millimeter eller punkter) relaterar till. Storlekarna kommer att skalas i enlighet med detta när kartan visas i en annan skala.

  Till exempel kommer ett linjelager som använder en 2 mm bred linje med inställningen 1:2000 *Reference scale* att återges med 4 mm breda linjer när kartan visas i 1:1000.

- Med gruppen *Selections* kan du styra om en viss färg eller symbol ska användas i stället för standardvärdena (*Projektegenskaper* ► *Allmänt* ► *Markeringsfärg*) för ett visst lager. Detta är användbart för att förbättra synligheten för valda objekt med viss symbologi:

  - *Använd standardfärg för urval*
  - *Overstyra färg för valda funktioner*: t.ex. om lagret använder en gul färg som standard och den gula standardmarkeringen inte är synlig.
  - *Overstyra symbol för valda funktioner*: t.ex. om ett linjelager använder en tunn symbol, och färgläggning av linjerna kanske inte gör dem tillräckligt synliga, kan det hjälpa att åsidosätta symbolen med en tjockare linje. Om lagret använder rastersymboler eller gradientfyllningar/linjer/shapeburst med färgrampsymboler tillämpas inte standardfärgen för urval alls; det kan vara till hjälp att kunna ange en specifik enklare symbol som ska användas för utvalda objekt i lagret.

- Rendering av extremt detaljerade lager (t.ex. polygonlager med ett stort antal noder) kan leda till att layoutexporten i PDF/SVG-format blir enorm eftersom alla noder ingår i den exporterade filen. Detta kan också göra att den resulterande filen blir mycket långsam att arbeta med/öppna i andra program.

  Om du markerar ☑ *Force layer to render as raster* tvingas dessa lager att rasteriseras så att de exporterade filerna inte behöver inkludera alla noder som finns i dessa lager och renderingen blir därför snabbare.

  Du kan också göra det genom att tvinga layouten att exporteras som raster, men det är en allt-eller-inget-lösning eftersom rasteriseringen tillämpas på alla lager. Alternativt kan du förlita dig på geometriförenkling i *layout export settings*.

- ☐ *Refresh layer at interval*: styr om och hur ofta ett lager kan uppdateras. Tillgängliga alternativ för *Konfiguration* är:

  - *Reload data*: lagret kommer att uppdateras helt och hållet. Eventuella cachelagrade data kasseras och hämtas på nytt från leverantören. Detta läge kan resultera i långsammare uppdateringar av kartan.
  - *Redraw layer only*: detta läge är användbart för animering eller när lagrets stil uppdateras med jämna mellanrum. Uppdateringar av Canvas skjuts upp för att undvika att uppdatera flera gånger om mer än ett lager har ett inställt intervall för automatisk uppdatering.

  Det är också möjligt att ställa in *Intervall (sekunder)* mellan på varandra följande uppfriskningar.

- Beroende på dataleverantören (t.ex. PostgreSQL) kan meddelanden skickas till QGIS när ändringar tillämpas på datakällan, utanför QGIS. Använd alternativet ☑ *Friskställ lagret vid meddelande* för att utlösa en uppdatering. Du kan också begränsa skiktuppdateringen till en specifik meddelandeuppsättning i textrutan ☑ *Only if message is*.

12.1.15 Temporala egenskaper

Fliken *Temporal* innehåller alternativ för att styra återgivningen av lagret över tid. Sådan dynamisk rendering kräver att *temporal navigation* är aktiverad över kartans canvas.

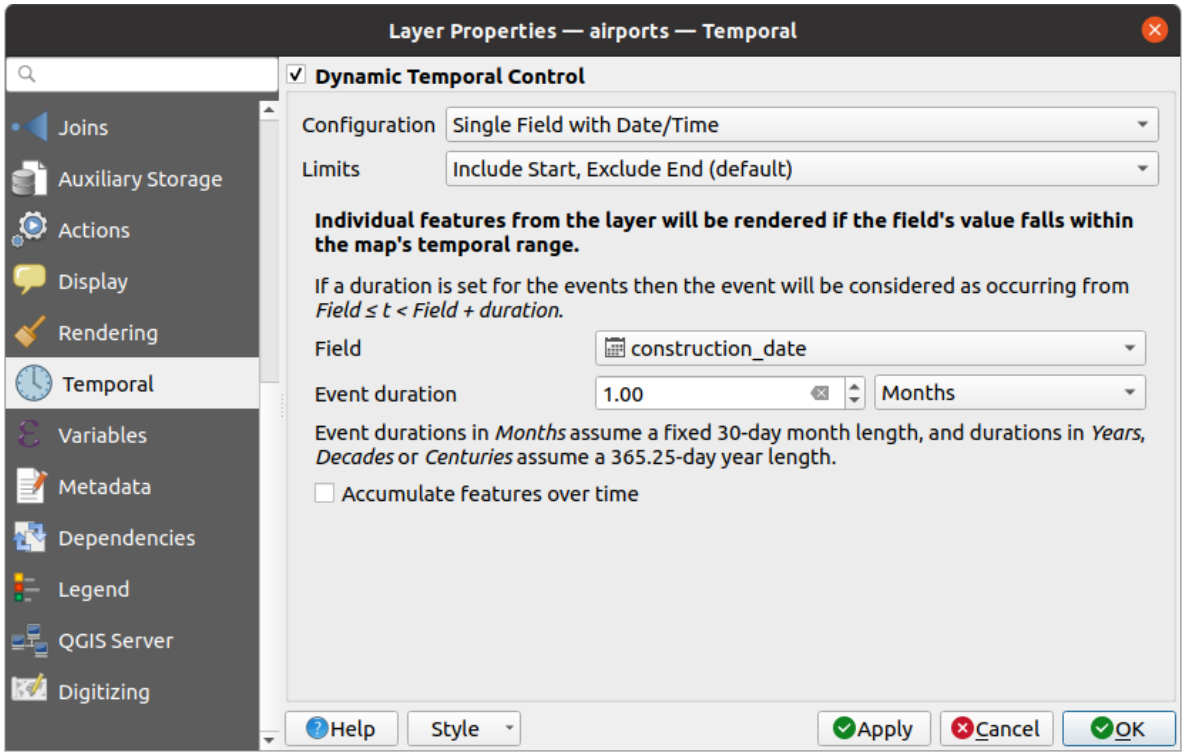


Fig. 12.63: Dialog för vektorlagrets temporala egenskaper

Markera alternativet *Dynamic Temporal Control* för att konfigurera vektorlagrets temporala rendering. Beroende på strukturen i din dataset kanske du vill använda något av de angivna *Konfiguration*-alternativen:

- *Fixed time range*: alla funktioner återges om kartans tidsram överlappar de angivna intervallen *Startdatum* och *Slutdatum*.
- *Single field with date/time*: objekt återges om deras *Field*-värde faller inom kartans tidsram. En *Event duration* kan ställas in. Om du markerar alternativet *Accumulate features over time*, kommer alla funktioner som inträffar före eller inom kartans tidsintervall att fortsätta att återges. Händelsens varaktighet ignoreras alltså.
- *Separata fält för start- och slutdatum/tid*: funktioner återges om det intervall som anges av deras värden för *Startfält* och *Slutfält* överlappar kartans temporala canvas.
- *Separata fält för start och händelsevaraktighet*: funktioner återges om det intervall som definieras av deras värden för *Startfält* och *Händelsevaraktighetsfält* överlappar kartans temporala canvas.
- *Start- och slutdatum/tid från uttryck*: funktioner återges om det tidsintervall som anges i fälten *Startuttryck* och *Slututtryck* överlappar kartans temporala canvas.
- *Redraw layer only*: lagret ritas om vid varje ny animationsram men ingen tidsbaserad filtrering tillämpas på funktionerna. Det är användbart när lagret använder tidsbaserade uttrycksvärden för renderarinställningar (t.ex. datadefinierad symbologi).

Det är också möjligt att ställa in *Limits* för funktionernas tidsintervall som:

- *Inkludera start, exkludera slut*
- *Inkludera början, inkludera slutet*

12.1.16 Variabler Egenskaper

Fliken *Variables* listar alla variabler som är tillgängliga på lagrets nivå (vilket inkluderar alla globala variabler och projektvariabler).

Det gör det också möjligt för användaren att hantera variabler på lagernivå. Klicka på knappen för att lägga till en ny anpassad variabel på lagernivå. På samma sätt kan du välja en anpassad variabel på lagernivå i listan och klicka på knappen för att ta bort den.

Mer information om användning av variabler finns i avsnittet Allmänna verktyg *Lagring av värden i variabler*.

12.1.17 Elevation Egenskaper

Fliken *Elevation* innehåller alternativ för att styra lagrets höjdegenskaper i en *3D-kartvy* och dess utseende i *profilverktygets diagram*. Specifikt kan du ställa in:

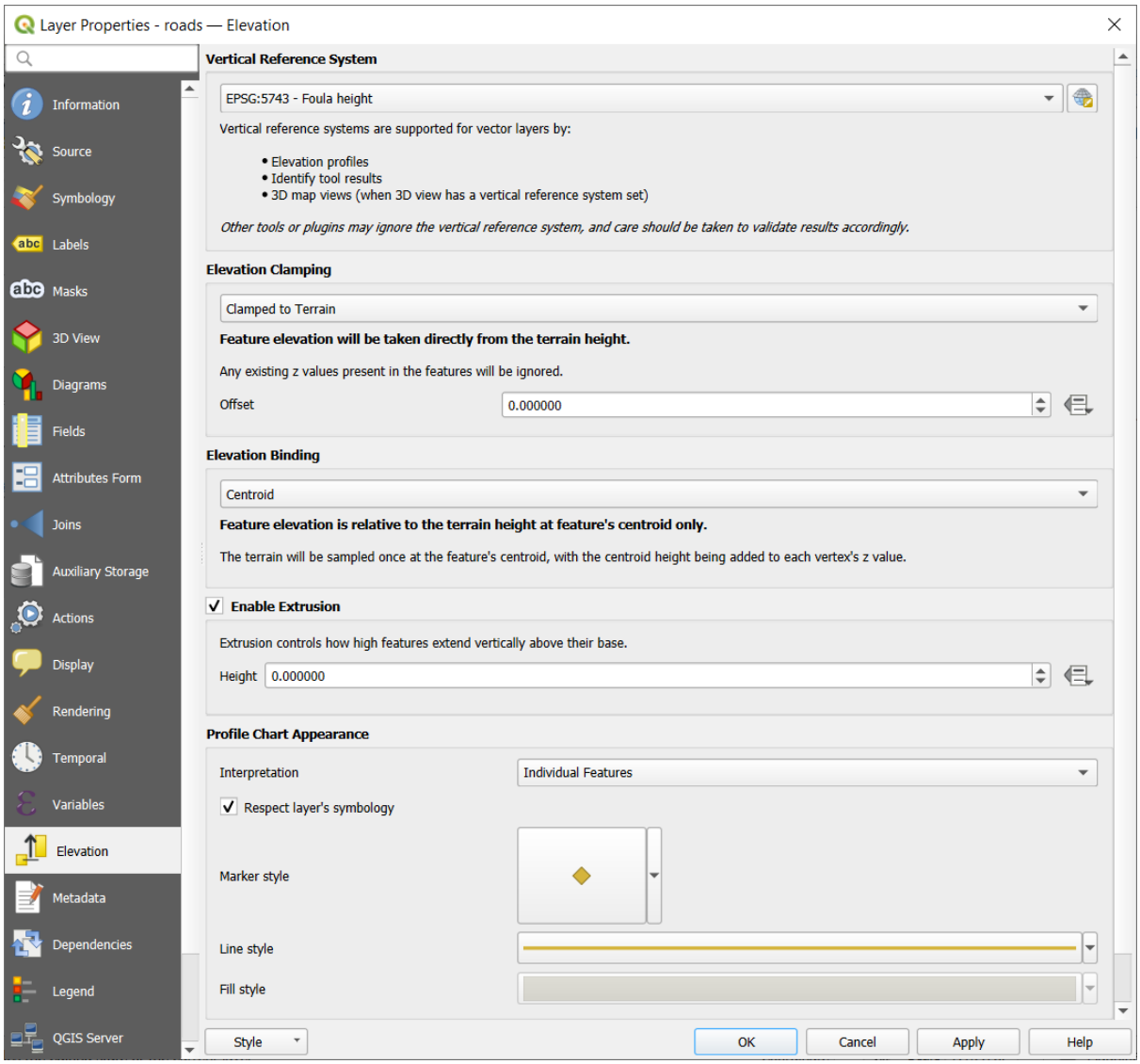


Fig. 12.64: Dialog för egenskaper för vektorlagrets höjd

- *Vertikalt referenssystem*: Om vektorlagrets CRS är ett sammansatt CRS (med en Z-dimension), kommer det vertikala CRS som används för lagret att vara den vertikala komponenten i lagrets CRS. I detta fall kan du

inte manuellt ställa in ett annat vertikalt CRS. Om lagrets CRS är horisontellt (2D) kan du välja ett specifikt vertikalt CRS genom att klicka på Select CRS. Vertikala referenssystem stöds för vektorlager av:

  - *Höjningsprofiler*
  - *Identifiera verktygsresultat*
  - *3D-kartvyer*

- *Elevation Clamping*: definierar hur och om objektets höjd ska vara:
  - *Clamped to terrain*: tar höjd direkt från terrängens höjd och ignorerar eventuella befintliga Z-värden i funktionerna. Ett datadefinierat *Offset*-värde från terrängen kan också fyllas i.
  - *Relativ till terräng*: alla befintliga Z-värden i objekten läggs till terränghöjden. En *Scale*-faktor följt av en datadefinierad *Offset* kan användas för att justera höjden. Det här alternativet är inte tillgängligt för 2D-geometrilager.
  - *Absolute*: ignorerar terränghöjden och tar direkt Z-värden från objekten för höjden. En faktor *Scale* följd av en datadefinierad *Offset* kan användas för att justera höjden. För 2D-geometrilager (utan Z-värden) kan istället en datadefinierad *Base height* anges.
- *Aktivera extrusion*: du kan ställa in en *Höjd* för att kontrollera hur högt objekten sträcker sig vertikalt över sin bas. Detta är praktiskt för att ange att ett 2D-geometrilager, t.ex. ett polygonlager med fotavtryck från en byggnad, faktiskt representerar 3D-objekt.
- *Elevation Binding*: Detta alternativ är endast relevant när man kombinerar en *Elevation clamping* som bygger på terrängen med ett linje- eller polygonlager och styr hur objektets höjd ställs in i förhållande till terrängens höjd. Terrängen kan samplas:
  - vid objektets *Centroid*, varvid centroidhöjden läggs till varje toppunkts z-värde
  - vid varje enskilt *Vertex* innan det läggs till vertexens z-värde
- *Profile Chart Appearance*: styr hur funktioner återges när ett profildiagram ritas. Två huvudsakliga *Tolkningslägen* är tillgängliga:
  - as *Individual features*: samplar diskreta positioner där tvärsnittets profillinje skär vektorelementen. Denna skärningspunkt kan representeras som en punkt, linje eller yta beroende på lagertyp och om en extrudering tillämpas.

    Med markeringen *Respect layer symbology*, kommer funktioner att återges på profildiagrammet med motsvarande *layer styling* (vilket gör att t.ex. kategoriserade klasser kan synas på profildiagrammet). Om profilens symboltyp inte matchar lagrets symboltyper i renderingsprogrammet används endast symbolfärgen från renderingsprogrammet på profilsymbolen.

    Beroende på lagerinställningarna kan profilsymboler representeras med en anpassad stil med hjälp av:

    * *Markeringsstil*: för icke-extruderade punkt- och linjeobjekt, och för icke-extruderade polygonobjekt som berörs av profillinjen
    * *Line style*: för extruderade punkt- och linjeelement, och för icke-extruderade polygonelement som skärs av profillinjen
    * *Fyllstil*: för extruderade polygonfunktioner
  - as *Kontinuerlig yta (t.ex. konturer)*: höjddiagrammet återges som en yta i stället för som separata funktioner genom att de samplade höjdresultaten sammanfogas till en kontinuerlig linje. Detta kan förbättra visualiseringen och är avsett för vektorlager som representerar en kontinuerlig höjdyta, t.ex. konturlinjer eller inmätta höjdpunkter. Profilen *Style* kan ställas in som:
    * en *Linje* med en specifik *Linjestil*
    * en höjdyta som återges med en fyllnadssymbol antingen ovanför (*Fill above*) eller nedanför (*Fill below*) höjdkurvans linje. Ytsymboliken representeras med hjälp av:
      · a *Fyllstil*

- och en *Limit*: den maximala (respektive minimala) höjden som bestämmer hur hög fyllnadsytan kommer att vara

Du kan också markera *Show markers at sampled points*` för att göra dem synliga över tolkningslinjen och tilldela dem en *Marker style*`.

12.1.18 Egenskaper för metadata

Fliken *Metadata* ger dig alternativ för att skapa och redigera en metadatarapport om ditt lager. Se *Metadata* för mer information.

12.1.19 Egenskaper för beroenden

Fliken *Beroenden* gör det möjligt att deklarera databeroenden mellan lager. Ett databeroende uppstår när en datamodifiering i ett lager, som inte sker genom direkt användarmanipulation, kan modifiera data i andra lager. Detta är t.ex. fallet när geometrin i ett lager uppdateras av en databastrigger eller ett anpassat PyQGIS-skript efter modifiering av ett annat lagers geometri.

På fliken *Dependencies* kan du välja alla lager som externt kan ändra data i det aktuella lagret. Om du anger beroende lager på rätt sätt kan QGIS ogiltigförklara cacheminnet för detta lager när de beroende lagren ändras.

12.1.20 Förklaringsegenskaper

Egenskapsfliken *Förklaring* ger dig avancerade inställningar för *Lagerpanelen* och/eller *skriv ut layoutförklaring*. Dessa alternativ inkluderar:

- Beroende på vilken symbologi som används i lagret kan du få flera poster i förklaringen, som inte nödvändigtvis är läsbara/användbara att visa. Med *Legend placeholder image* kan du *välja en bild* för ersättning, som visas både i panelen *Layers* och i utskriftslayoutens förklaring.
- *Visa etikettlegenden*: Visar översikter över de olika etikettinställningarna som poster i legenderna. *etikettstil* förhandsvisas tillsammans med beskrivningen.
- *Texter på symboler*: I vissa fall kan det vara bra att lägga till extra information till symbolerna i teckenförklaringen. Med den här ramen kan du tilldela alla symboler som används i lagersymbologin en text som visas över symbolen, både i *Layers*-panelen och i teckenförklaringen för utskriftslayouten. Denna mappning görs genom att skriva in varje text bredvid symbolen i tabellwidgeten eller fylla i tabellen med hjälp av knappen *Set Labels from Expression*. Textens utseende hanteras med hjälp av widgetarna för val av teckensnitt och färg på knappen *Text Format*.

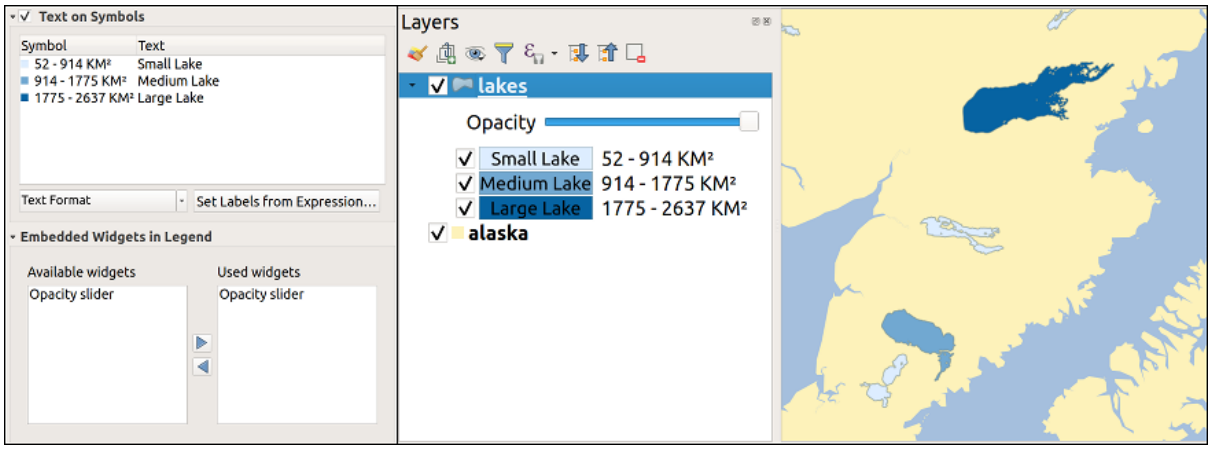


Fig. 12.65: Inställning av text på symboler (vänster) och dess återgivning i *Lager*-panelen (höger)

- en lista med widgetar som du kan bädda in i lagerträdet i panelen Lager. Tanken är att det ska finnas ett sätt att snabbt komma åt vissa åtgärder som ofta används med lagret (inställning av transparens, filtrering, urval, stil eller andra saker …).

  Som standard tillhandahåller QGIS en widget för transparens, men detta kan utökas genom att tillägg registrerar sina egna widgets och tilldelar anpassade åtgärder till lager som de hanterar.

12.1.21 Egenskaper för QGIS-server

Fliken *QGIS Server* består av avsnitten *Description*, *Attribution*, *Metadata URL* och *Legend URL*.

I avsnittet *Beskrivning* kan du ändra *Kortnamn* som används för att referera till lagret i förfrågningar (om du vill veta mer om kortnamn kan du läsa services_basics_short_name). Du kan också lägga till eller redigera en *Title*, en alternativ *WFS-titel* och *Abstract* för lagret, eller definiera en *Nyckelordslista* här. Dessa nyckelordslistor kan användas i en metadatakatalog. Om du vill använda en titel från en XML-metadatafil måste du fylla i en länk i fältet *Data URL*.

Använd *Attribution* för att hämta attributdata från en XML-metadatakatalog.

I *Metadata URL* kan du lägga till de allmänna sökvägarna till XML-metadatakatalogen. Denna information kommer att sparas i QGIS-projektfilen för efterföljande sessioner och kommer att användas för QGIS Server.

I avsnittet *Legend URL* kan du ange webbadressen till en legendbild i fältet url. Du kan använda rullgardinsalternativet Format för att tillämpa lämpligt format för bilden. För närvarande stöds bildformaten png, jpg och jpeg.

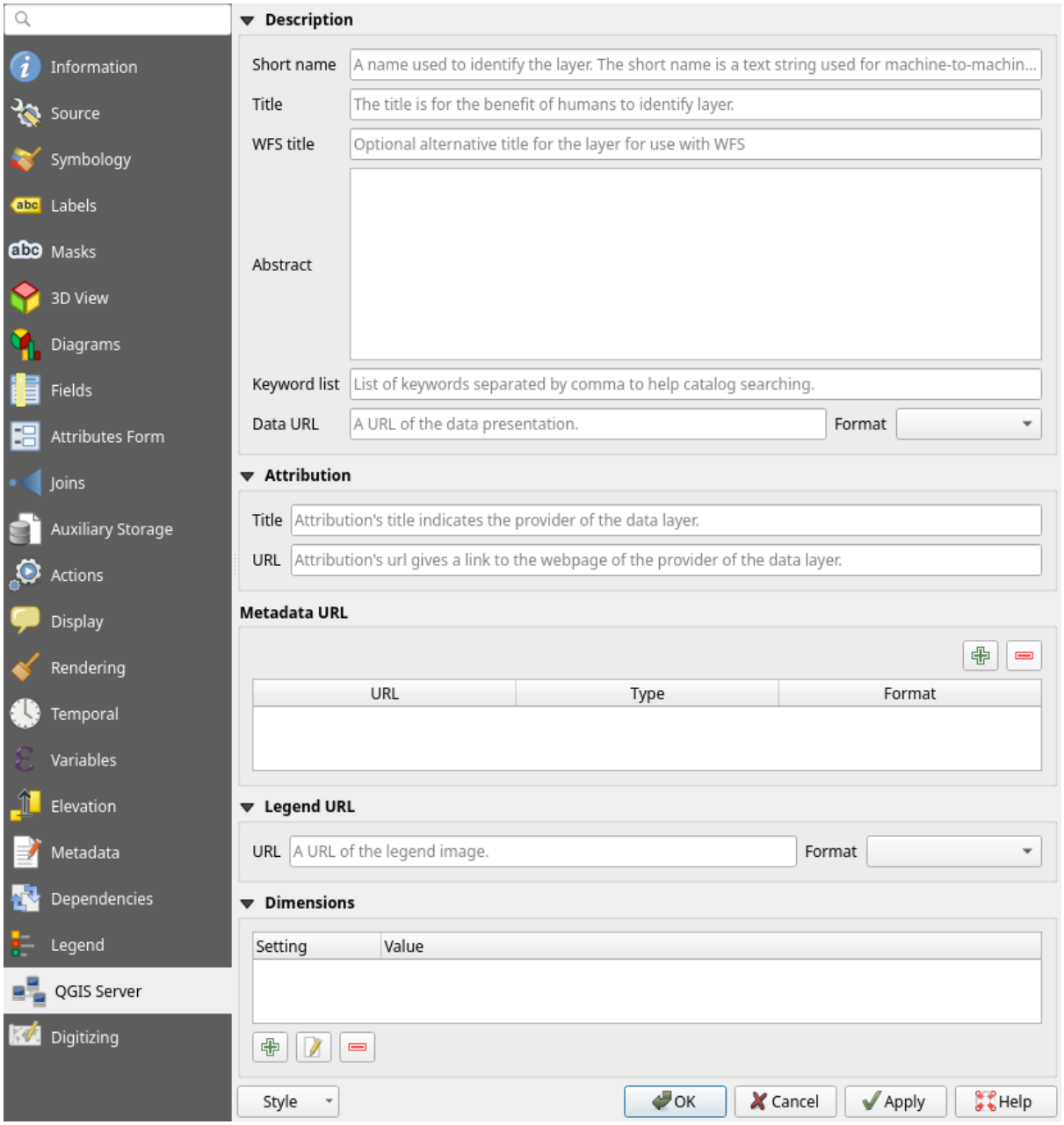


Fig. 12.66: Fliken QGIS Server i dialogen Egenskaper för vektorlager

Om du vill veta mer om QGIS Server kan du läsa QGIS-Server-manual.

12.1.22 Egenskaper för digitalisering

Fliken *Digitizing* ger tillgång till alternativ som hjälper till att säkerställa kvaliteten på digitaliserade geometrier.

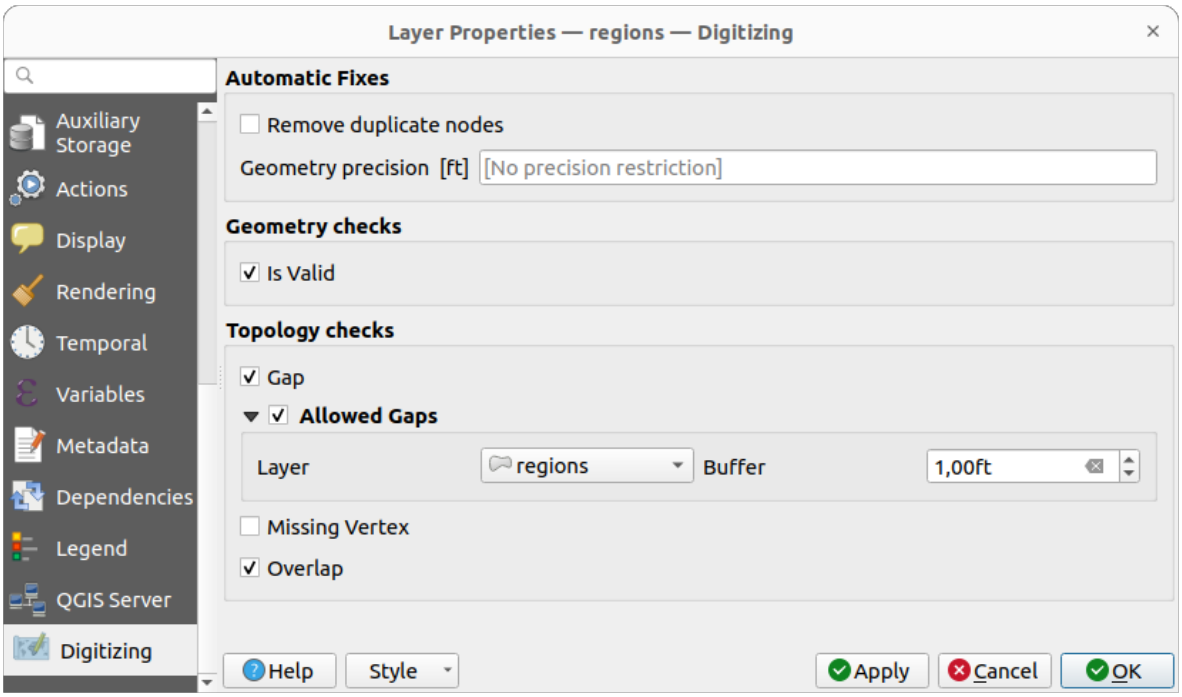


Fig. 12.67: Fliken QGIS-digitalisering i dialogrutan Egenskaper för vektorlager

Automatiska korrigeringar

Alternativen i avsnittet *Automatic Fixes* kommer direkt att påverka vertikalerna i alla geometrier som läggs till eller ändras. Om alternativet *Remove duplicate nodes* är markerat kommer alla två efterföljande hörn med exakt samma koordinater att tas bort. Om *Geometry precision* är inställt kommer alla hörn att avrundas till närmaste multipel av den konfigurerade geometriska precisionen. Avrundningen sker i referenssystemet för lagrets koordinater. Z- och M-värden avrundas inte. Med många kartverktyg visas ett rutnät på duken under digitaliseringen.

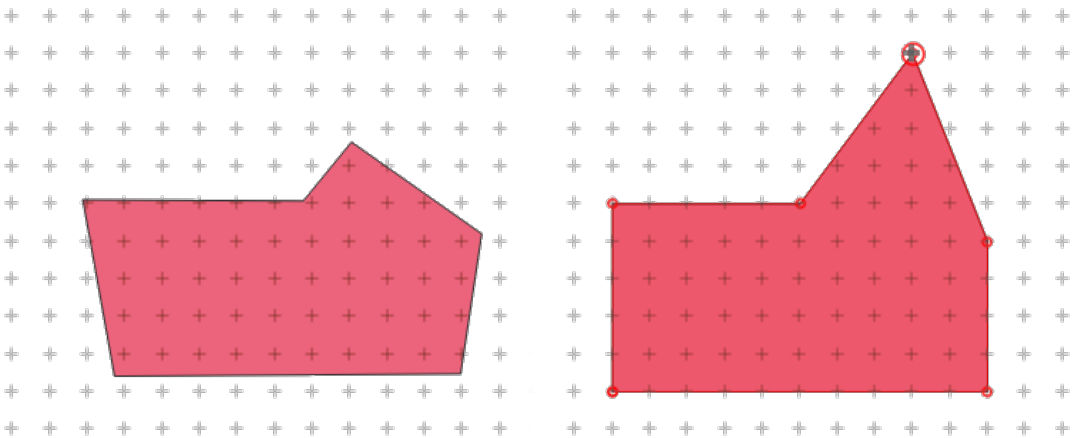


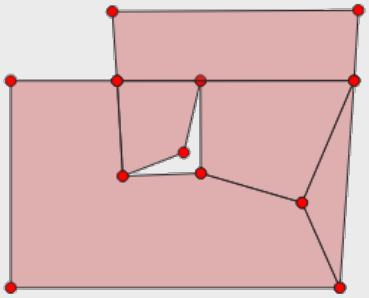
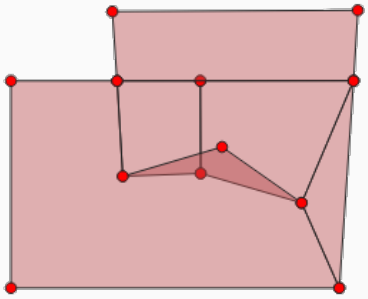
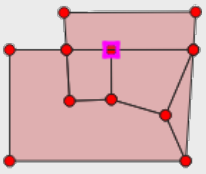
Fig. 12.68: När du flyttar den översta toppunkten fästs alla toppunkterna i rutnätet

Geometri-kontroller

I avsnittet *Geometry checks* kan ytterligare valideringar per geometri aktiveras. Omedelbart efter varje geometriändring rapporteras fel i dessa kontroller till användaren i panelen *Geometry validation*. Så länge en kontroll misslyckas är det inte möjligt att spara lagret. Kontrollen *Is valid* kör grundläggande giltighetskontroller som självskärning på geometrier.

Topologikontroller

I avsnittet *Topology checks* kan ytterligare valideringskontroller av topologin aktiveras. Topologikontrollerna utförs när användaren sparar lagret. Kontrollfel kommer att rapporteras i panelen *Geometry validation*. Så länge det finns valideringsfel kan lagret inte sparas. Topologikontroller utförs inom området för de modifierade objektens avgränsningsbox. Eftersom det kan finnas andra funktioner i samma område rapporteras topologiska fel som rör dessa funktioner, liksom fel som introducerats under den aktuella redigeringssessionen.

Alternativ för topologikontroll	Illustration
Kontrollen *Gap* kontrollerar om det finns mellanrum mellan närliggande polygoner.	
Kontrollen *Overlap* söker efter överlappningar mellan närliggande polygoner.	
Kontrollen *Missing vertex* kontrollerar om det finns delade gränser mellan angränsande polygoner där den ena gränsen saknar en vertex som finns på den andra.	

Undantag för glappkontroll

Ibland är det önskvärt att behålla luckor inom ett område i ett polygonlager som annars helt täcks av polygoner. Ett markanvändningsskikt kan t.ex. ha acceptabla hål för sjöar. Det är möjligt att definiera områden som ignoreras i gap-kontrollen. Eftersom luckor inom dessa områden är tillåtna kommer vi att kalla dem för *Allowed Gaps*-områden.

I alternativen för gap-kontrollerna under *Allowed Gaps* kan ett lager med *Allowed Gaps* konfigureras.

När gap-kontrollen utförs rapporteras inte gap som täcks av en eller flera polygoner i *Allowed Gaps Layer* som topologifel.

Det är också möjligt att konfigurera ytterligare en *Buffert*. Denna buffert tillämpas på varje polygon på *Allowed Gaps Layer*. Detta gör det möjligt att göra testerna mindre känsliga för små förändringar i konturerna vid gränserna för luckor.

När *Allowed Gaps* är aktiverat finns en extra knapp (*Add Allowed Gap*) för upptäckta gapfel i dockningsstationen för geometrivalidering, där gap rapporteras under digitaliseringen. Om du trycker på knappen *Add Allowed Gap* infogas en ny polygon med geometrin för det upptäckta gapet i *Allowed Gaps Layer*. Detta gör det möjligt att snabbt flagga luckor som tillåtna.

Panel för validering av geometri

Panelen *Geometry Validation* aktiveras när någon av de ovan nämnda digitaliseringskontrollerna hittar ett fel. I dialogrutan visas en lista över fel och deras beskrivning, och du kan bläddra i listan med hjälp av tangentbordets pilar eller dedikerade pilar.

Du måste ta itu med alla problem innan du kan spara ändringar i lagret. För att göra det:

1. Välj ett fel, och det är möjligt att göra det:
   - *Zooma till valda funktion(er)*
   - *Zoom till problem*
2. Välj de vanliga *digitaliseringsverktygen* för att lösa problemet.

12.2 Arbeta med attributtabellen

Attributtabellen visar information om funktioner i ett valt lager. Varje rad i tabellen representerar en funktion (med eller utan geometri) och varje kolumn innehåller en viss information om funktionen. Funktionerna i tabellen kan sökas, markeras, flyttas eller till och med redigeras.

12.2.1 Förord: Spatiala och icke-spatiala tabeller

Med QGIS kan du ladda spatiala och icke-spatiala lager. Detta inkluderar för närvarande tabeller som stöds av GDAL och avgränsad text, samt leverantörerna PostgreSQL, MS SQL Server, SpatiaLite och Oracle. Alla laddade lager listas i panelen *Layers*. Huruvida ett lager är spatialt aktiverat eller inte avgör om du kan interagera med det på kartan.

Icke-spatiala tabeller kan bläddras igenom och redigeras med hjälp av attributtabellvyn. Dessutom kan de användas för fältuppslagningar. Du kan t.ex. använda kolumner i en icke spatial tabell för att definiera attributvärden, eller ett intervall av tillåtna värden, som ska läggas till i ett specifikt vektorlager under digitaliseringen. Ta en närmare titt på redigeringswidgeten i avsnitt *Attribut Formulär Egenskaper* för att ta reda på mer.

12.2.2 Introduktion till gränssnittet för attributtabeller

För att öppna attributtabellen för ett vektorlager, aktivera lagret genom att klicka på det i *Panel för lager*. Från huvudmenyn *Layer* väljer du sedan *Open Attribute Table*. Det går också att högerklicka på lagret och välja *Open Attribute Table* från rullgardinsmenyn, eller att klicka på knappen *Open Attribute Table* i verktygsfältet Attributes. Om du föredrar genvägar kan du öppna attributtabellen med `F6`, attributtabellen filtrerad till valda funktioner med `Shift+F6` och attributtabellen filtrerad till synliga funktioner med `Ctrl+F6`.

Detta öppnar ett nytt fönster som visar funktionens attribut för lagret (*figure_attributes_table*). Enligt inställningen i menyn :menuselection:` *Inställningar –> Alternativ –> Datakällor* öppnas attributtabellen i ett dockat fönster eller i ett vanligt fönster. Det totala antalet objekt i lagret och antalet för närvarande valda/filtrerade objekt visas i attributtabellens titel, liksom om lagret är spatialt begränsat.

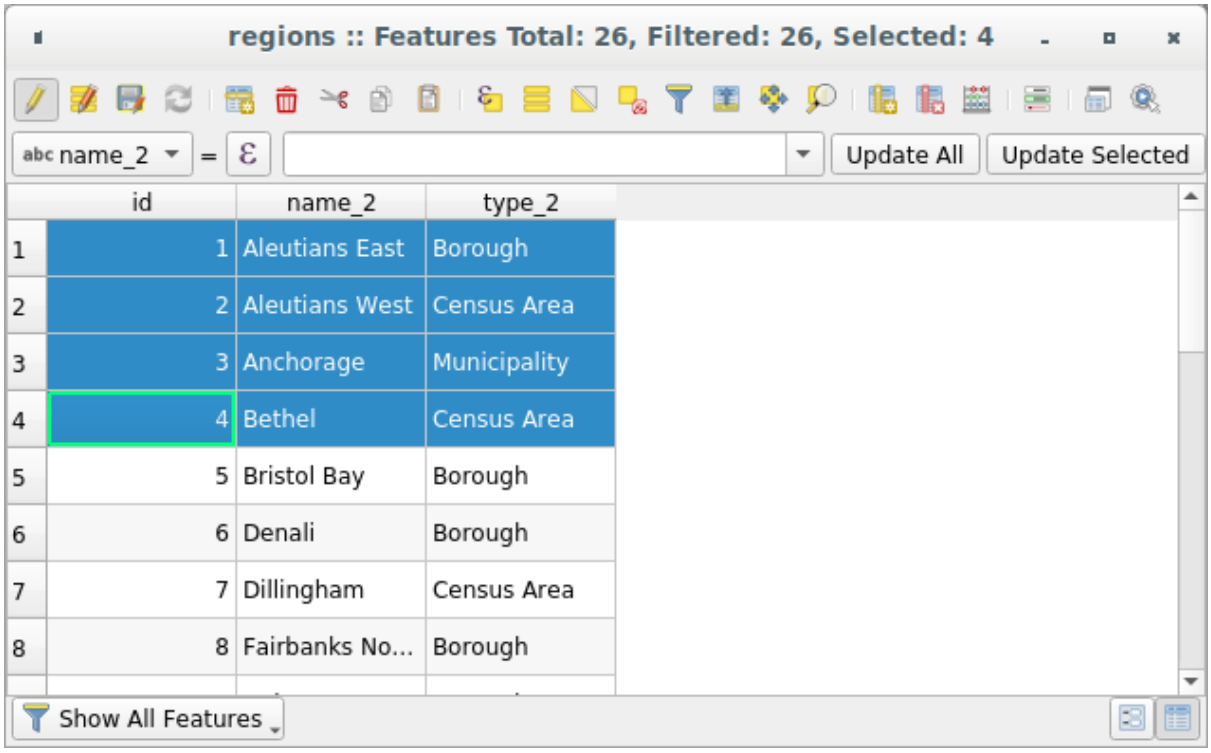


Fig. 12.69: Attributtabell för regionlager

Knapparna längst upp i fönstret för attributtabellen har följande funktioner:

Table 12.1: Tillgängliga verktyg

Ikon	Etikett	Syfte	Standardgenväg
	Växla redigeringsläge	Aktivera redigeringsfunktioner	`Ctrl+E`
	Växla till multiredigeringsläge	Uppdatera flera fält med många funktioner	
	Spara redigeringar	Spara aktuella ändringar	
	Ladda om tabellen		
	Lägg till funktion	Lägg till ny geometrilös funktion	
	Ta bort valda funktioner	Ta bort valda funktioner från lagret	
	Klipp ut valda funktioner till klippbordet		`Ctrl+X`

continues on next page

Table 12.1 – fortsättning från föregående sida

Ikon	Etikett	Syfte	Standardgenväg
	Kopiera valda funktioner till urklipp		`Ctrl+C`
	Klistra in funktioner från urklipp	Infoga nya funktioner från kopierade funktioner	`Ctrl+V`
	Välj funktioner med hjälp av en Expression		
	Välj alla	Välj alla funktioner i lagret	`Ctrl+A`
	Avmarkera alla	Invertera det aktuella valet i lagret	`Ctrl+R`
	Avmarkera alla	Avmarkera alla funktioner i det aktuella lagret	`Ctrl+Shift+A`
	Filtrera/välja funktioner med hjälp av formulär		`Ctrl+F`
	Flytta markerat till toppen	Flytta markerade rader till toppen av tabellen	
	Panoramakarta till de valda raderna		`Ctrl+P`
	Zooma kartan till de markerade raderna		`Ctrl+J`
	Nytt fält	Lägg till ett nytt fält i datakällan	`Ctrl+W`
	Ta bort fält	Ta bort ett fält från datakällan	
	Organisera kolumner	Visa/dölj fält från attributtabellen	
	Kalkylator för öppet fält	Uppdatera fält för många funktioner i rad	`Ctrl+I`
	Villkorsstyrd formatering	Aktivera tabellformatering	
	Tabell över dockningsattribut	Gör det möjligt att docka/avdocka attributtabellen	
	Actions	Visar de åtgärder som är relaterade till lagret	

Observera

Beroende på dataformat och GDAL-biblioteket som byggts med din QGIS-version kan det hända att vissa verktyg inte är tillgängliga.

Under dessa knappar finns fältet Quick Field Calculation (endast aktiverat i *edit mode*), som gör det möjligt att snabbt tillämpa beräkningar på alla eller delar av objekten i lagret. Det här fältet använder samma *uttryck* som Field Calculator (se *Redigera attributvärden*).

Tabellvy vs formulärvy

QGIS tillhandahåller två visningslägen för att enkelt manipulera data i attributtabellen:

- Table view, visar värden för flera funktioner i tabellform, där varje rad representerar en funktion och varje kolumn ett fält. Genom att högerklicka på kolumnrubriken kan du *konfigurera tabellvisningen* medan du genom att högerklicka på en cell kan *interagera med funktionen*.

  Attributtabellen stöder `Shift+Mouse Wheel`-skrollning i tabellvisningsläget för att växla mellan vertikala och horisontella rullningsrörelser. Detta kan också uppnås genom att ersätta musen med styrplattan på macOS.

- Form view visar *feature identifiers* i en första panel och visar bara attributen för den klickade identifieraren i den andra. Det finns en rullgardinsmeny längst upp i den första panelen där ”identifieraren” kan anges med hjälp av ett attribut (*Column preview*) eller ett *Expression*. Rullgardinsmenyn innehåller också de 10 senaste uttrycken för återanvändning. Formulärvyn använder konfigurationen för lagerfält (se *Attribut Formulär Egenskaper*).

Du kan bläddra bland funktionsidentifierarna med pilarna längst ned i den första panelen. Objektens attribut uppdateras i den andra panelen allteftersom du går igenom dem. Det är också möjligt att identifiera eller flytta till den aktiva funktionen i kartbilden genom att trycka på någon av knapparna längst ned:

- Highlight current feature om den är synlig i kartbilden
- Automatisk panorering till aktuell funktion
- Zoom till aktuell funktion

Du kan växla från ett läge till ett annat genom att klicka på motsvarande ikon längst ned till höger i dialogrutan.

Du kan också ange läget *Standardvy* när attributtabellen öppnas i :menuselection:` *Inställningar* –> *Alternativ* –> *Datakällor* menyn. Det kan vara ”Kom ihåg senaste vyn”, ”Tabellvy” eller ”Formulärvy”.

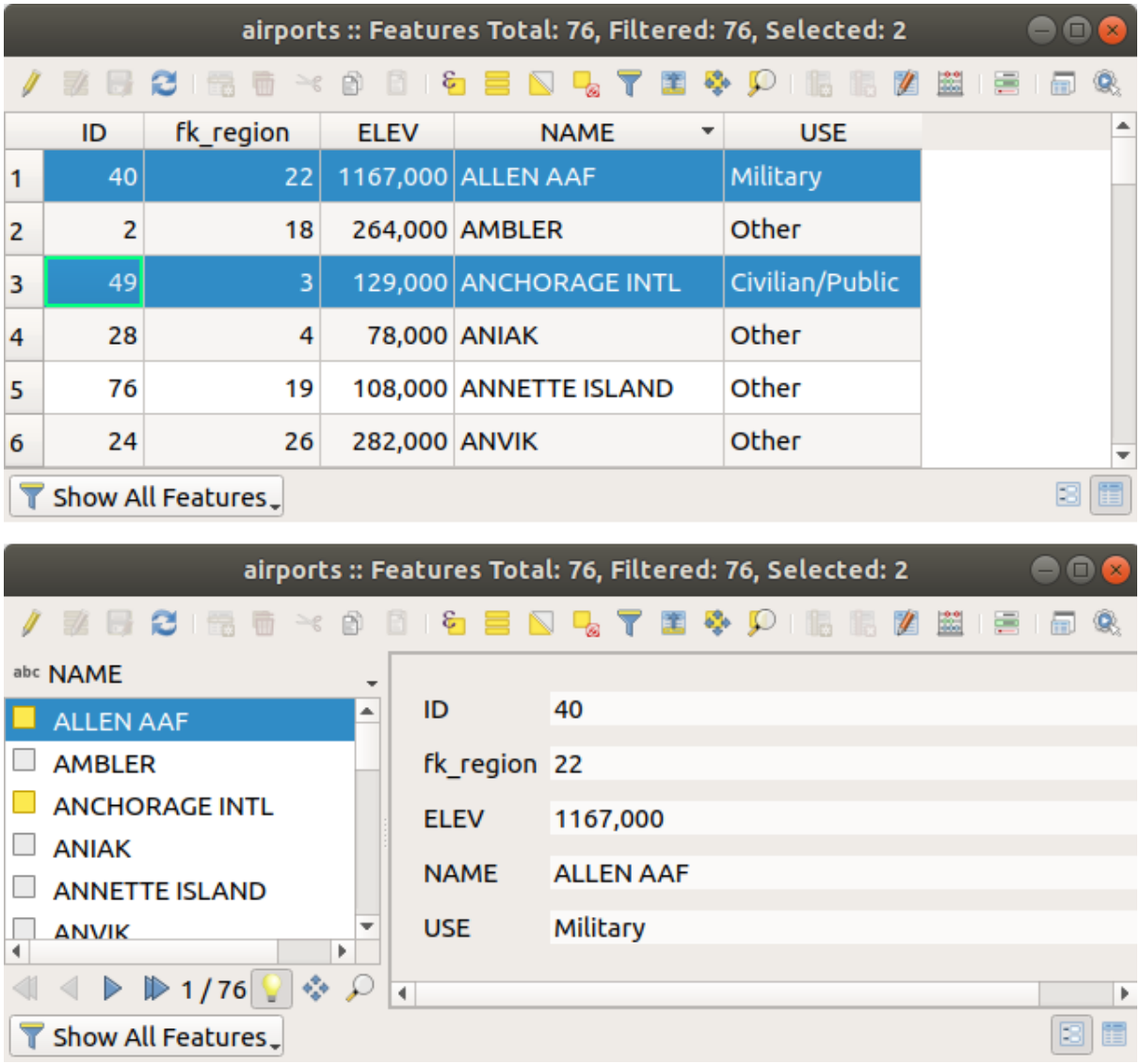


Fig. 12.70: Attributtabell i tabellvy (överst) jämfört med formulärvy (nederst)

Konfigurera kolumnerna

Högerklicka i en kolumnrubrik i tabellvyn för att få tillgång till verktyg som hjälper dig att styra:

- *kolumnstorlek*
- *kolumners synlighet och ordning*
- *Sorteringsordning för data*

Ändra storlek på kolumnernas bredd

Kolumnbredden kan ställas in genom att högerklicka på kolumnrubriken och välja antingen:

- *Set width…* för att ange önskat värde. Som standard visas det aktuella värdet i widgeten
- *Ställ in alla kolumnbredder…* till samma värde
- *Autosize* för att ändra storlek på kolumnen så att den passar bäst.
- *Automatisera alla kolumner*

Du kan också ändra storleken på en kolumn genom att dra i gränsen till höger om rubriken. Den nya storleken på kolumnen bibehålls för lagret och återställs vid nästa öppning av attributtabellen.

I *Data Source Settings* kan du välja att *Autosize all columns by default when opening attribute table*, vilket gör att ”Autosize All Columns” blir standardvy varje gång attributtabeller öppnas i QGIS.

Dölja och organisera kolumner och aktivera åtgärder

Genom att högerklicka i en kolumnrubrik kan du välja att *Dölja kolumn* från attributtabellen (i läget ”tabellvy”). För mer avancerade kontroller, tryck på knappen Organisera kolumner… i dialogverktygsfältet eller välj *Organisera kolumner…* i en kontextuell meny för kolumnrubriken.

I den nya dialogrutan kan du:

- markera/avmarkera kolumner som du vill visa eller dölja: en dold kolumn kommer att försvinna från varje instans av attributtabelldialogen tills den aktivt återställs. Det är också möjligt att:
  - välj *Show All* för att visa alla fält (kolumner) och åtgärder i tabellen
  - välj *Hide All* för att dölja alla fält (kolumner) och åtgärder i tabellen
  - använd *Toggle selection* för att invertera synligheten för det aktuella urvalet av kolumner. Du kan använda *tangentbordskombination* för att välja flera kolumner.
- dra-och-släpp-objekt för att ändra ordningen på kolumnerna i attributtabellen. Observera att denna ändring gäller tabellrenderingen och inte ändrar fältordningen i skiktdatakällan
- lägg till en ny virtuell *Actions*-kolumn som i varje rad visar en rullgardinsmeny eller en knapplista med aktiverade åtgärder. Se *Åtgärder Egenskaper* för mer information om åtgärder.

Sortering av rader

Raderna kan sorteras efter valfri kolumn genom att klicka på kolumnrubriken. En liten pil anger sorteringsordningen (nedåtpekande betyder fallande värden från den översta raden och nedåt, uppåtpekande betyder stigande värden från den översta raden och nedåt). Du kan också välja att sortera raderna med alternativet *Sort...* i kolumnhuvudets snabbmeny och skriva ett uttryck. Om du t.ex. vill sortera raderna med hjälp av flera kolumner kan du skriva `concat(col0, col1)`.

I formulärvyn kan identifierare av funktioner sorteras med hjälp av alternativet *Sort by preview expression*.

Observera att sortering av raderna endast påverkar tabellrenderingen och inte ändrar funktionernas ordning i skiktdatakällan.

Tips

Sortering baserad på kolumner av olika typer

Om du försöker sortera en attributtabell baserad på kolumner av sträng- och numeriska typer kan det leda till oväntade resultat eftersom uttrycket `concat("USE", "ID")` returnerar strängvärden (t.ex. `'Borough105' < 'Borough6'`). Du kan kringgå detta genom att t.ex. använda `concat("USE", lpad("ID", 3, 0))` som returnerar `'Borough105' > 'Borough006'`.

Formatering av tabellceller med hjälp av villkor

Inställningar för villkorlig formatering kan användas för att i attributtabellen markera funktioner som du kanske vill lägga särskilt fokus på, med hjälp av anpassade villkor för funktioner:

- geometri (t.ex. identifiering av objekt som består av flera delar, små områden eller i en definierad kartutsträckning…);
- eller fältvärde (t.ex. jämföra värden med ett tröskelvärde, identifiera tomma celler, dubbletter, …).

Du kan aktivera panelen för villkorlig formatering genom att klicka på Conditional formatting-knappen längst upp till höger i attributfönstret i tabellvyn (utlöses inte i formulärvyn).

Den nya panelen gör det möjligt för användaren att lägga till nya regler för formatåtergivning av *Field* eller *Full row*. Om du lägger till en ny regel öppnas ett formulär för att definiera:

- namnet på regeln;
- ett villkor med hjälp av någon av funktionerna *uttrycksbyggare*;
- formateringen: den kan väljas från en lista med fördefinierade format eller skapas baserat på egenskaper som t.ex:
  - bakgrunds- och textfärger;
  - användning av ikon;
  - fetstil, kursiv stil, understrykning eller genomstrykning;
  - Misslyckades med att läsa in teckensnitt: teckensnittstyp okänd

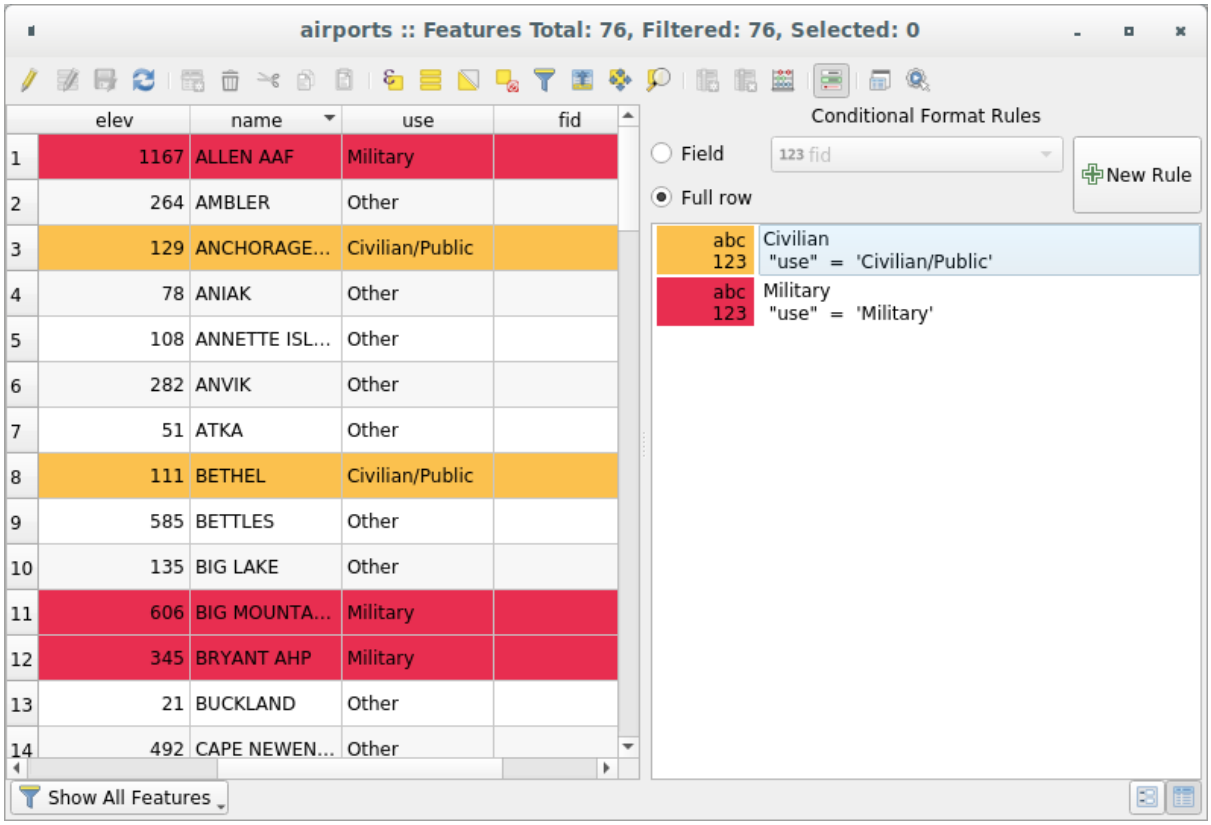


Fig. 12.71: Villkorlig formatering av en attributtabell

12.2.3 Interagera med funktioner i en attributtabell

Välja funktioner

I tabellvyn visar varje rad i attributtabellen attributen för en unik funktion i lagret. Om du väljer en rad väljs funktionen och på samma sätt väljs raden i attributtabellen om du väljer en funktion i kartbilden (om lagret är aktiverat för geometri). Om den uppsättning funktioner som valts i kartbilden (eller attributtabellen) ändras, uppdateras också valet i attributtabellen (eller kartbilden) i enlighet med detta.

Rader kan väljas genom att klicka på radnumret till vänster om raden. **Flera rader** kan markeras genom att hålla ned `Ctrl`-tangenten. Ett **kontinuerligt urval** kan göras genom att hålla ned `Shift`-tangenten och klicka på flera radhuvuden till vänster om raderna. Alla rader mellan den aktuella markörpositionen och den klickade raden väljs. Om markörens position flyttas i attributtabellen genom att klicka på en cell i tabellen ändras inte radvalet. Om du ändrar urvalet i huvudfönstret flyttas inte markörens position i attributtabellen.

I formulärvyn för attributtabellen identifieras objekten som standard i den vänstra panelen med värdet för det fält som visas (se *Egenskaper för display*). Denna identifierare kan ersättas med hjälp av rullgardinsmenyn längst upp i panelen, antingen genom att välja ett befintligt fält eller genom att använda ett eget uttryck. Du kan också välja att sortera listan med funktioner från rullgardinsmenyn.

Klicka på ett värde i den vänstra panelen för att visa funktionens attribut i den högra. För att välja en funktion måste du klicka inuti den fyrkantiga symbolen till vänster om identifieraren. Som standard blir symbolen gul. Precis som i tabellvyn kan du välja flera funktioner med hjälp av de tangentbordskombinationer som beskrivits tidigare.

Förutom att välja objekt med musen kan du göra automatiska urval baserat på objektets attribut med hjälp av verktyg som finns i attributtabellens verktygsfält, t.ex. (se avsnitt *Automatisk markering* och följande för mer information och användningsfall):

- *Välj efter uttryck…*

- *Välj funktioner efter värde…*
- *Avmarkera alla funktioner från lagret*
- *Välj alla funktioner*
- *Invertera funktionsval.*

Det är också möjligt att *välja funktioner med hjälp av formulär*.

Funktioner för filtrering

När du har valt ut funktioner i attributtabellen kanske du vill visa endast dessa poster i tabellen. Detta kan enkelt göras med hjälp av *Show Selected Features* i rullgardinsmenyn längst ned till vänster i dialogrutan för attributtabellen. Denna lista innehåller följande filter:

- *Visa alla funktioner*
- *Show Selected Features* - samma som att använda *Open Attribute Table (Selected Features)* från *Layer*-menyn eller *Attributes Toolbar* eller trycka på `Shift+F6`
- *Visa funktioner som är synliga på kartan* - samma som att använda *Öppna attributtabell (synliga funktioner)* från menyn *Layer* eller verktygsfältet *Attributes* eller trycka på `Ctrl+F6`
- *Visa funktioner med misslyckade begränsningar* - funktioner filtreras för att endast visa de som har misslyckade *begränsningar*. Beroende på om den ouppfyllda begränsningen är hård eller mjuk visas fältvärden som inte uppfylls i mörka respektive ljusorange celler.
- *Visa redigerade och nya funktioner* - samma som att använda *Öppna attributtabell (redigerade och nya funktioner)* från menyn *Layer* eller verktygsfältet *Attributes*
- *Field Filter* - låter användaren filtrera baserat på värdet i ett fält: välj en kolumn från en lista, skriv eller välj ett värde och tryck på `Enter` för att filtrera. Därefter visas endast de funktioner som matchar uttrycket `num_field = value` eller `string_field ilike '%value%'` i attributtabellen. Du kan markera *Case sensitive* för att vara mindre tillåtande med strängar.
- *Avancerat filter (Uttryck)* - Öppnar dialogrutan för uttrycksbyggare. I den kan du skapa *komplexa uttryck* för att matcha rader i tabellen. Du kan t.ex. filtrera tabellen med hjälp av mer än ett fält. När filteruttrycket tillämpas visas det längst ned i formuläret.
- *Lagrade filteruttryck* ►: en genväg till *Lagrade uttryck* som ofta används för att filtrera din attributtabell.

Det är också möjligt att *filtrera funktioner med hjälp av formulär*.

Observera

Att filtrera bort poster från attributtabellen innebär inte att funktioner filtreras bort från lagret; de är helt enkelt tillfälligt dolda från tabellen och kan nås från kartbilden eller genom att ta bort filtret. För filter som döljer funktioner från lagret ska du använda *Query Builder*.

Tips

Uppdatera datakällans filtrering med `Visa funktioner som är synliga på kartan`

När funktioner som visas i attributtabellen av prestandaskäl är spatialt begränsade till kanvasens utsträckning när den öppnas (se *Data Source Options* för en beskrivning), uppdateras den spatiala begränsningen om du väljer *Show Features Visible on Map* på en ny kanvasutsträckning.

Lagring av filteruttryck

Uttryck som du använder för filtrering av attributtabeller kan sparas för senare användning. När du använder posterna *Field Filter* eller *Advanced Filter (expression)* visas det använda uttrycket i en textwidget längst ned i attributtabelldialogen. Tryck på ☆ Spara uttryck med text som namn bredvid rutan för att spara uttrycket i projektet. Om du trycker på rullgardinsmenyn bredvid knappen kan du spara uttrycket med ett eget namn (*Save expression as…*).

När ett sparat uttryck visas aktiveras knappen ★ och i rullgardinsmenyn kan du välja *Redigera uttrycket* och namnet om det finns något, eller *Radera sparat uttryck*.

Sparade filteruttryck sparas i projektet och är tillgängliga via menyn *Stored filter expressions* i attributtabellen. De skiljer sig från *användaruttryck*, som delas av alla projekt i den aktiva användarprofilen.

Filtrering och urval av funktioner med hjälp av formulär

Om du klickar på Filter/Select features using form eller trycker på `Ctrl+F` kommer attributtabelldialogen att växla till formulärvy och ersätta varje widget med dess sökvariant.

Från och med nu liknar verktygets funktionalitet den som beskrivs i *Välj funktioner efter värde*, där du kan hitta beskrivningar av alla operatörer och urvalslägen.

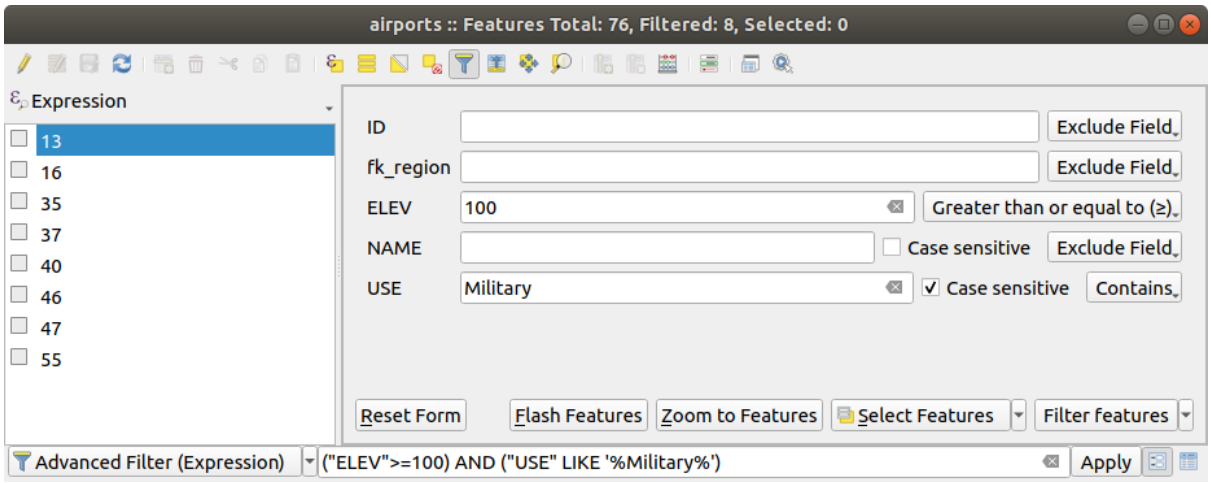


Fig. 12.72: Attributtabell filtrerad med filterformuläret

När du väljer/filtrerar egenskaper från attributtabellen finns det en knapp *Filter features* som gör det möjligt att definiera och förfina filter. Om du använder den aktiveras alternativet *Avancerat filter (uttryck)* och motsvarande filteruttryck visas i en redigerbar textwidget längst ned i formuläret.

Om det redan finns filtrerade funktioner kan du förfina filtret med hjälp av listrutan bredvid knappen *Filter features*. Alternativen är:

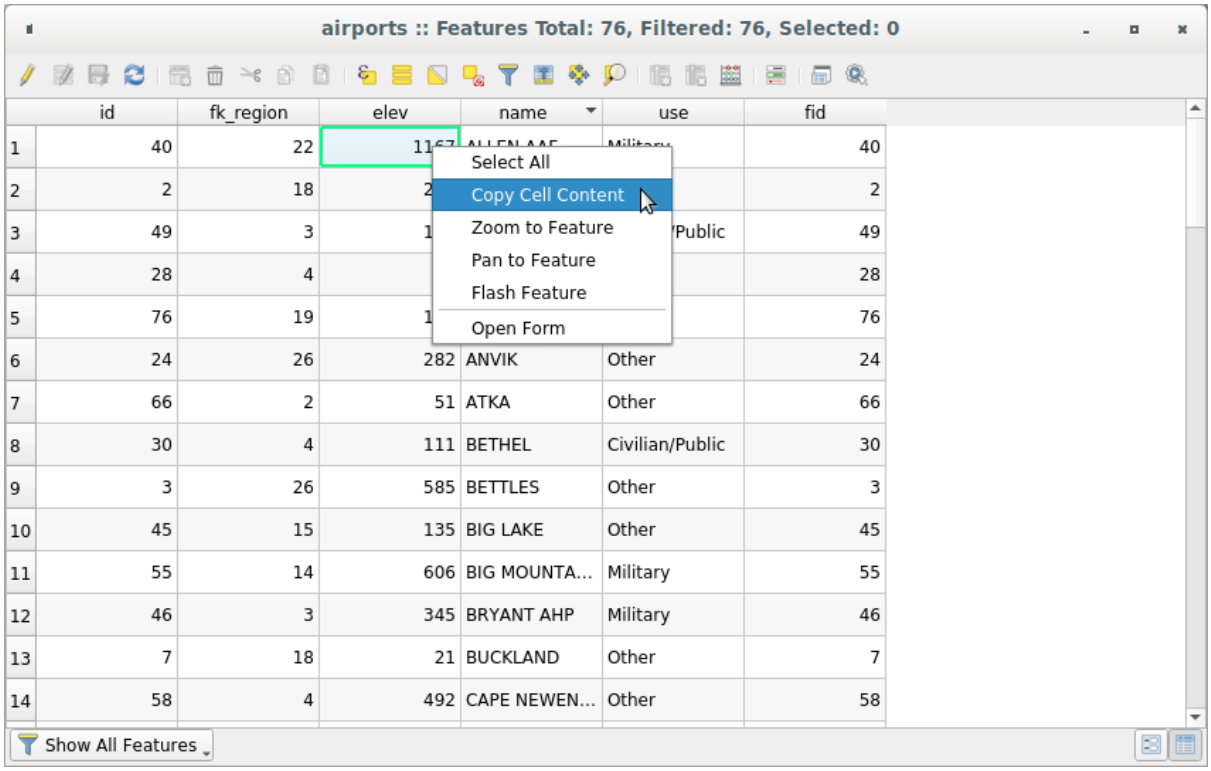
- *Filter inom ("AND")*
- *Förläng filter ("OR")*

För att rensa filtret väljer du antingen alternativet *Visa alla funktioner* i rullgardinsmenyn längst ned till vänster, eller rensar uttrycket och klickar på *Apply* eller trycker på `Enter`.

Fler åtgärder på funktioner

Användare har flera möjligheter att manipulera funktioner i en attributtabell. Högerklicka i en cell och du kan:

- *Välj alla* (`Ctrl+A`) funktioner;
- Kopiera innehållet i en cell till urklipp med *Copy cell content*;
- *Zoom to feature* utan att behöva välja det i förväg;
- *Pan to feature* utan att behöva välja den i förväg;
- *Flash feature*, för att markera den i kartbilden;
- *Open form*: den växlar attributtabell till formulärvy med fokus på den klickade funktionen.
- Visa en *lista över åtgärder*, tidigare aktiverad i *Lageregenskaper* ► *Actions*-fliken.



	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

Fig. 12.73: Knappen Kopiera cellinnehåll

Om du vill använda attributdata i externa program (t.ex. Excel, LibreOffice eller en egen webbapplikation) markerar du en eller flera rader och använder knappen Kopiera markerade rader till urklipp eller trycker på `Ctrl+C`. Dessutom kan du i menyn *Inställningar* ► *Alternativ* ► *Datakällor* definiera formatet att klistra in till med alternativet *Kopiera funktioner som*. Mer information finns på *Inställningar för datakällor*.

12.2.4 Redigera attributvärden

För att kunna ändra data i en attributtabell måste du först växla lagret till redigering. Tryck på knappen Toggle Editing. Beroende på lagrets geometrityp och klippbordets status aktiveras några fler verktyg i attributtabellens övre verktygsfält.

Redigering av attributvärden kan sedan göras genom att:

- skriver in det nya värdet direkt i cellen, oavsett om attributtabellen är i tabell- eller formulärvy. Ändringar görs alltså cell för cell, funktion för funktion;
- med hjälp av *field calculator*: uppdatera i en rad ett fält som kanske redan finns eller ska skapas men för flera funktioner. Det kan användas för att skapa virtuella fält;
- med hjälp av snabbfältet *calculation bar*: samma som ovan men för endast befintligt fält;
- eller med läget *multi edit*: uppdatera flera fält för flera funktioner på en rad.

Om du sätter lagret i redigeringsläge kan du också göra följande: Limma in funktioner från urklipp (`Ctrl+V`) Klipp ut valda rader till urklipp (`Ctrl+X`) eller Radera valda funktioner. Mer information finns på *Ändrar*.

Använda fältkalkylatorn

Med knappen Field Calculator i attributtabellen kan du utföra beräkningar på basis av befintliga attributvärden eller definierade funktioner, t.ex. för att beräkna längd eller area för geometriska objekt. Resultaten kan användas för att uppdatera ett befintligt fält eller skrivas till ett nytt fält (som kan vara ett *virtuellt* ett).

Fältkalkylatorn är tillgänglig på alla lager som stöder redigering. När du klickar på ikonen för fältkalkylatorn öppnas dialogrutan (se Fig. 12.74). Om lagret inte är i redigeringsläge visas en varning och om du använder fältkalkylatorn kommer lagret att sättas i redigeringsläge innan beräkningen görs.

Dialogrutan Field Calculator bygger på dialogrutan *Expression Builder* och erbjuder ett komplett gränssnitt för att definiera ett uttryck och tillämpa det på ett befintligt eller nyskapat fält. För att använda dialogrutan för fältkalkylator måste du välja om du vill:

1. tillämpa beräkning på hela lagret eller endast på utvalda funktioner
2. skapa ett nytt fält för beräkningen eller uppdatera ett befintligt.

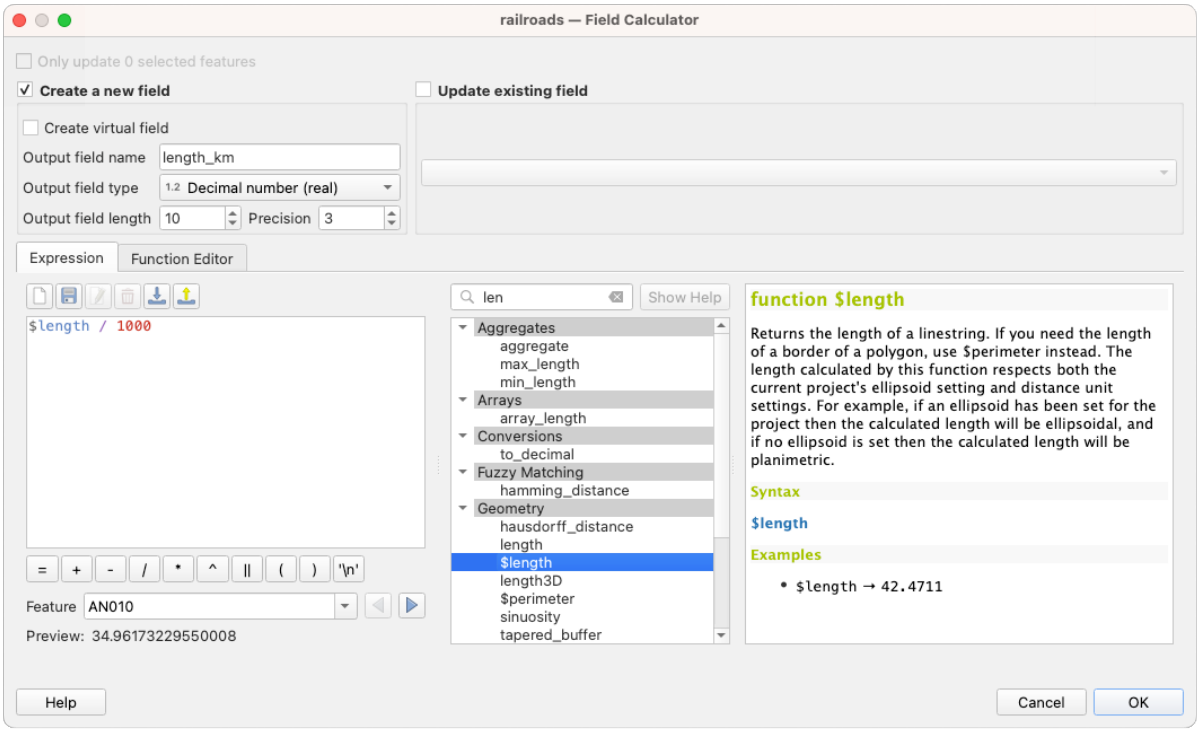


Fig. 12.74: Fältkalkylator

Om du väljer att lägga till ett nytt fält måste du ange ett fältnamn, en fälttyp (heltal, real, datum eller sträng) och vid behov den totala fältlängden och fältprecisionen. Om du t.ex. väljer en fältlängd på 10 och en fältprecision på 3 innebär det att du har 7 siffror före punkten och 3 siffror för decimaldelen.

Ett kort exempel illustrerar hur fältkalkylatorn fungerar när du använder fliken *Expression*. Vi vill beräkna längden i km för lagret `railroads` från QGIS exempeldataset:

1. Ladda shapefilen `railroads.shp` i QGIS och tryck på Öppna attributtabell.
2. Klicka på Toggle editing mode och öppna dialogrutan Field Calculator.
3. Markera kryssrutan *Create a new field* för att spara beräkningarna i ett nytt fält.
4. Ställ in *Output field name* till `length_km`
5. Välj `Decimaltal (verkligt)` som *Fälttyp för utdata*
6. Ställ in *Output field length* till `10` och *Precision* till `3`
7. Dubbelklicka på `$length` i gruppen *Geometry*` för att lägga till geometrins längd i Field calculators uttrycksruta (du kommer att se en förhandsgranskning av resultatet, upp till 60 tecken, under uttrycksrutan som uppdateras i realtid när uttrycket sammanställs).
8. Fyll i uttrycket genom att skriva `/ 1000` i rutan Field calculator expression och klicka på *OK*.
9. Du kan nu hitta ett nytt fält *length_km* i attributtabellen.

Skapa ett virtuellt fält

Ett virtuellt fält är ett fält som baseras på ett uttryck som beräknas i farten, vilket innebär att dess värde automatiskt uppdateras så snart en underliggande parameter ändras. Uttrycket gäller för alla funktioner i lagret och ställs in en gång; du behöver inte längre räkna om fältet varje gång de underliggande värdena ändras. Du kan t.ex. använda ett virtuellt fält om du vill att området ska utvärderas när du digitaliserar funktioner eller för att automatiskt beräkna en varaktighet mellan datum som kan ändras (t.ex. med funktionen `now()`).

Att skapa ett virtuellt fält görs via dialogen *Field calculator* och följer *samma procedur* som för vanliga fält. Kom bara ihåg att markera alternativet *Create virtual field* och använd en fälttyp som är kompatibel med de data som ditt uttryck skulle generera.

Redigering av ett virtuellt fält görs via fliken *Fält* i dialogen för lagrets egenskaper (se *Fältegenskaper*). Uttrycket som definierar fältet visas i kolumnen *Kommentar*, och om du trycker på knappen bredvid öppnas ett fönster med en uttrycksredigerare för uppdatering.

Observera

Användning av virtuella fält

- Ett fält kan bara ställas in virtuellt när det skapas.
- Virtuella fält är inte permanenta i lagerattributen, vilket innebär att de endast sparas och är tillgängliga i den projektfil där de har skapats.

Använda fältet för snabbfältsberäkningar

Field calculator är alltid tillgänglig, men fältberäkningsfältet för snabb fältberäkning ovanpå attributtabellen är endast synligt om lagret är i redigeringsläge. Tack vare uttrycksmotorn går det snabbare att redigera ett redan befintligt fält:

1. Välj det fält som ska uppdateras i rullgardinsmenyn.
2. Fyll textrutan med ett värde, ett uttryck som du skriver direkt eller skapar med hjälp av uttrycksknappen .
3. Klicka på knappen *Update All*, *Update Selected* eller *Update Filtered* beroende på dina behov.

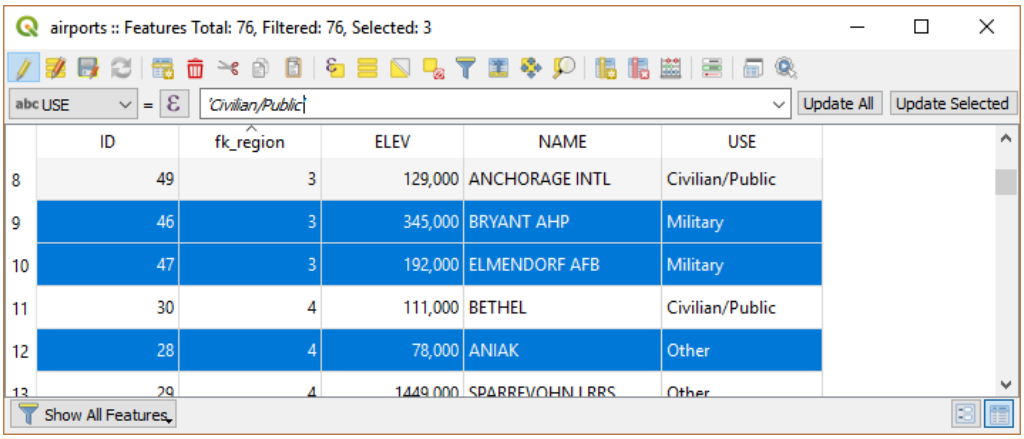


Fig. 12.75: Snabb fältberäkning Bar

Redigera flera fält

Till skillnad från de tidigare verktygen kan du i multiredigeringsläget redigera flera attribut för olika funktioner samtidigt. När lagret växlas till redigering är multiredigeringsfunktionerna tillgängliga:

- med hjälp av knappen Toggle multi edit mode i verktygsfältet i dialogrutan för attributtabellen;
- eller genom att välja *Redigera* ► *Modify attributes of selected features* menyn.

Observera

Till skillnad från verktyget från attributtabellen får du genom att trycka på alternativet *Edit* ► *Modify Attributes of Selected Features* en modal dialogruta för att fylla i attributändringar. Därför krävs val av funktioner före körning.

För att kunna redigera flera fält i en rad:

1. Välj de funktioner som du vill redigera.
2. Klicka på knappen i verktygsfältet för attributtabeller. Detta växlar dialogrutan till formulärvyn. I det här steget kan du också göra ett funktionsval.
3. På höger sida av attributtabellen visas fält (och värden) för valda funktioner. Nya widgetar visas bredvid varje fält så att det aktuella multiredigeringsläget kan visas:
   - Fältet innehåller olika värden för valda funktioner. Det visas tomt och varje funktion behåller sitt ursprungliga värde. Du kan återställa fältets värde från widgetens rullgardinslista.
   - Alla valda funktioner har samma värde för det här fältet och det värde som visas i formuläret kommer att behållas.
   - Fältet har redigerats och det inmatade värdet kommer att tillämpas på alla valda funktioner. Ett meddelande visas längst upp i dialogrutan, där du uppmanas att antingen tillämpa eller återställa din ändring.

   Om du klickar på någon av dessa widgetar kan du antingen ange det aktuella värdet för fältet eller återställa det till det ursprungliga värdet, vilket innebär att du kan återställa ändringar för varje enskilt fält.

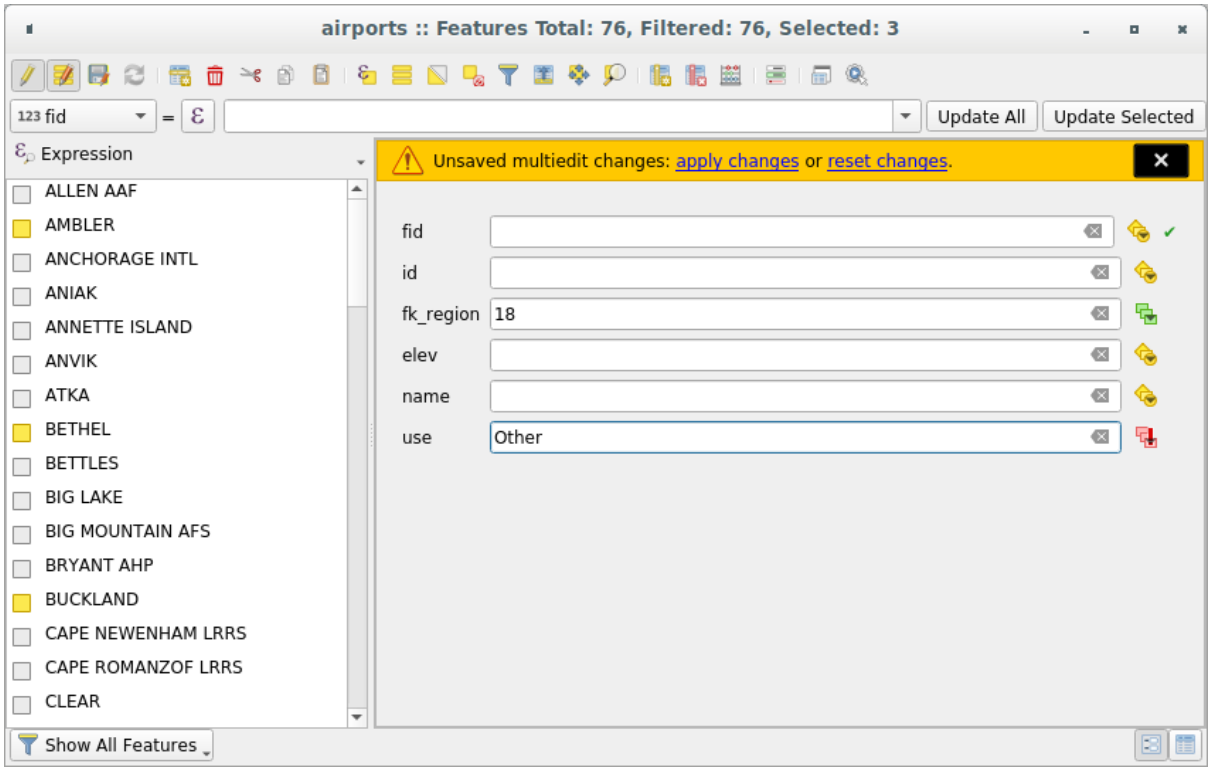


Fig. 12.76: Redigering av fält för flera funktioner

4. Gör ändringarna i de fält du vill ha.
5. Klicka på **Apply changes** i den övre meddelandetexten eller i någon annan funktion i den vänstra panelen.

Ändringarna gäller **alla valda funktioner**. Om ingen funktion är vald uppdateras hela tabellen med dina ändringar. Ändringar görs som ett enda redigeringskommando. Om du trycker på Undo återställs alltså attributändringarna för alla markerade funktioner på en gång.

Observera

Multiredigeringsläget är endast tillgängligt för autogenererade formulär och dra-och-släpp-formulär (se *Anpassa ett formulär för dina data*); det stöds inte av anpassade ui-formulär.

12.2.5 Utforska funktionsattribut med hjälp av Identify Tool

Verktyget *Identify features* kan användas för att visa alla attribut för en funktion i kartbilden. Det är ett snabbt sätt att visa och verifiera alla data utan att behöva söka efter dem i attributtabellen.

Följ dessa steg för att använda verktyget *Identify features* för vektorlager:

1. Markera vektorlagret i lagerpanelen.
2. Klicka på verktyget *Identify features* i verktygsfältet eller tryck på `Ctrl+Shift+I`.
3. Klicka på en funktion i kartvyn.

Panelen *Identify results* visar information om olika egenskaper beroende på lagertyp. Det finns två kolumner i panelen, på vänster sida kan du se *Feature* och på höger sida *Value*. Under kolumnen *Feature* kommer panelen att visa följande information:

- **Derived** section - detta är den information som beräknas eller härleds från annan information i lagret. Till exempel arean på en polygon eller längden på en linje. Allmän information som kan hittas i detta avsnitt:
  - Beroende på geometrityp, kartesiska mått på längd, omkrets eller area i lagrets CRS-enheter. För 3D-linjevektorer är den cartesiska linjelängden tillgänglig.
  - Beroende på geometrityp och om en ellipsoid är inställd i dialogen *Projektegenskaper* (*Allmänt* ► *Mätningar*), ellipsoidvärden för längd, omkrets eller area med de angivna enheterna.
  - Antalet geometriska delar i funktionen och numret på den del som klickades på.
  - Antal toppar i objektet.

  Samordna information som finns i detta avsnitt:
  - X- och Y-koordinatvärden för den klickade punkten.
  - Numret på den vertex som ligger närmast den klickade punkten.
  - X- och Y-koordinatvärden för det närmaste toppunktet.
  - Om du klickar på ett krökt segment visas även radien för den delen.
- **Dataattribut**: Detta är en lista över attributfält och värden för den funktion som har klickats på.
- information om den relaterade underordnade funktionen om du har definierat en *relation*:
  - namnet på relationen
  - posten i referensfältet, t.ex. namnet på den relaterade underordnade funktionen
  - **Actions**: listar åtgärder som definierats i dialogen för lagrets egenskaper (se *Åtgärder Egenskaper*) och standardåtgärden är `View feature form`.
  - **Dataattribut**: Detta är en lista över attributfält och värden för den relaterade underordnade funktionen.

12.2.6 Lagring och hämtning av en extern resurs

Ett fält kan vara inriktat på en resurs som lagras i ett externt lagringssystem. Attributformulär kan konfigureras så att de fungerar som en klient till ett externt lagringssystem för att lagra och hämta dessa resurser, på användarnas begäran, direkt från formulären.

Konfigurera en extern lagringsenhet

För att konfigurera en extern lagring måste du först konfigurera den från vektorn *attribute form properties* och välja widgeten *Attachment*.

Fig. 12.77: Redigera en extern WebDAV-lagring för ett visst fält

I widgeten *Attachment* måste du först välja *Storage type*:

- *Välj befintlig fil*: Mål-URL:en finns redan. När du väljer en resurs sker ingen lagring, utan attributet uppdateras helt enkelt med URL-adressen.
- *Enkel kopia*: Lagrar en kopia av resursen på en fildiskdestination (som kan vara ett lokalt eller delat filsystem i nätverket) och attributet uppdateras med sökvägen till kopian.
- *WebDAV-lagring*: Resursen flyttas till en HTTP-server som stöder protokollet WebDAV och attributet uppdateras med dess URL. Nextcloud, Pydio eller annan programvara för filhosting stöder detta protokoll.
- *AWS S3*: Resursen flyttas till en server som stöder protokollet AWS Simple Storage Service och attributet uppdateras med dess URL. Amazon Web Service och MinIO värdprogramvara stöder detta protokoll.

Sedan måste du ställa in parametern *Store URL*, som anger den URL som ska användas när en ny resurs behöver lagras. Det är möjligt att ställa in ett uttryck med *data defined override widget* för att få specifika värden enligt funktionsattribut.

Variabeln **@selected_file_path** kan användas i det uttrycket och representerar den absoluta filsökvägen till den fil som användaren har valt (med hjälp av filväljaren eller drag'n drop).

Observera

Om du använder den externa lagringen **WebDAV** eller **AWS S3** och URL:en slutar med "/" betraktas den som en mapp och det valda filnamnet läggs till för att få den slutliga URL:en.

Om det externa lagringssystemet behöver det, är det möjligt att konfigurera en *authentication*.

Observera

Om du vill använda extern lagring **AWS S3** måste du använda en autentiseringstyp **AWS S3**.

Använda en extern lagringsenhet

När konfigurationen är klar kan du välja en lokal fil med knappen … när du redigerar en funktions attribut. Beroende på den konfigurerade *lagringstypen* lagras filen i det externa lagringssystemet (utom om *Välj befintlig fil* har valts) och fältet uppdateras med den nya resurs-URL:en.

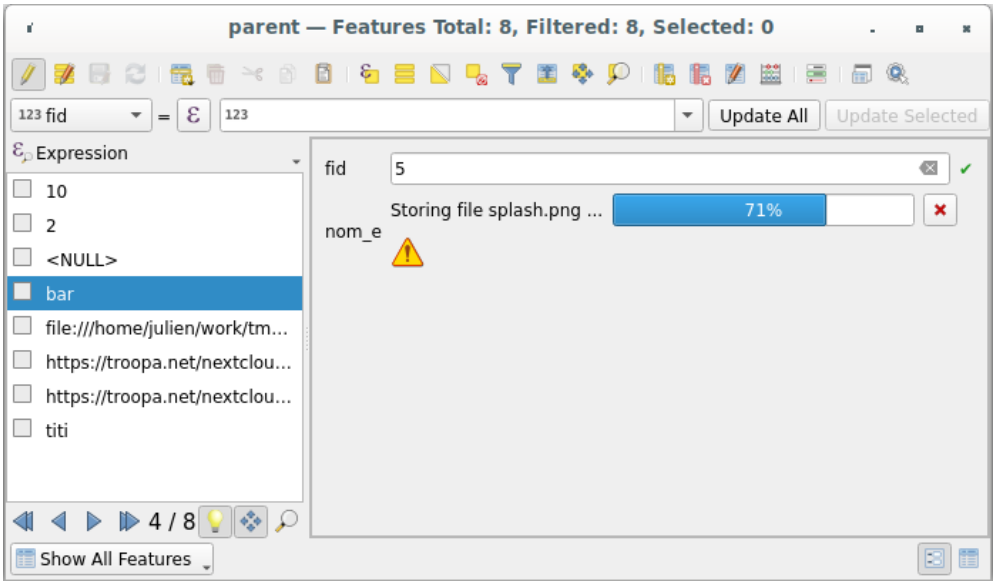


Fig. 12.78: Lagra en fil i en extern WebDAV-lagringsplats

Observera

Användaren kan också uppnå samma resultat om han drar och släpper en fil på hela widgeten för bilagor.

Använd knappen Cancel för att avbryta lagringsprocessen. Det är möjligt att konfigurera en visare med hjälp av *Integrated document viewer* så att resursen automatiskt hämtas från det externa lagringssystemet och visas direkt under URL:en. Varningsikonen ovan indikerar att resursen inte kan hämtas från det externa lagringssystemet. I så fall kan mer information visas i *Panel för loggmeddelanden*.

12.3 Ändrar

QGIS har olika funktioner för redigering av OGR, SpatiaLite, PostGIS, MS SQL Server och Oracle Spatial vektorlager och tabeller. De kan vara av typen 2D- eller 3D-geometri.

Observera

Proceduren för att redigera GRASS-lager är annorlunda - se avsnitt *Digitalisering och redigering av ett GRASS-vektorlager* för detaljer.

Uppmärksamma

Samtidiga redigeringar

QGIS spårar inte om någon annan redigerar samma funktion samtidigt som du gör det. Den sista personen som sparar redigeringarna vinner.

Tips

Validerande redigeringar

Kontinuerlig validering kan aktiveras för varje lager i fliken *Lageregenskaper* ► *Digitizing*. Mer information finns på *Egenskaper för digitalisering*.

12.3.1 Inställning av snäpptolerans och sökradie

Under menyn *Inställningar* ► *Alternativ…* ► *Digitizing* tillhandahåller QGIS ett antal parametrar för att konfigurera standardbeteendet för redigeringsverktyg. Mer information finns på *Inställningar för digitalisering*.

För optimal och exakt redigering av geometrier i vektorlager måste vi ställa in ett lämpligt värde för snäpptolerans och sökradie för funktionernas hörn. I gruppen *Snapping* finns relaterade alternativ, nämligen hantering av snäpptoleransen och sökradien.

- Tolerans för snäppning: När du lägger till en ny vertex eller flyttar en befintlig vertex är snäppningstoleransen det avstånd som QGIS använder för att söka efter den närmaste vertex eller det närmaste segment som du försöker ansluta till. Om du inte befinner dig inom snäpptoleransen kommer QGIS att lämna vertexen där du släppte musknappen, istället för att snäppa den till en befintlig vertex eller ett befintligt segment.

  Toleransinställningen påverkar alla verktyg som arbetar med snapping och gäller som standard för nya lager och projekt. Den kan dock åsidosättas på lagernivå (se *Alternativ för snäppning och digitalisering*).

- Sökradie: *Sökradie för vertexredigeringar* är det avstånd som QGIS använder för att `söka` efter det vertex som ska väljas när du klickar på kartan. Om du inte befinner dig inom sökradien kommer QGIS inte att hitta och välja något vertex för redigering.

Snap-tolerans och sökradie ställs in i ”mapp-enheter” eller ”pixlar”. Du kan behöva experimentera för att få dem rätt. Om du anger en för stor tolerans kan QGIS snäppa till fel toppunkt, särskilt om du har att göra med ett stort antal toppunkter i närheten av varandra. Ju mindre sökradie, desto svårare blir det att träffa det du vill flytta.

12.3.2 Alternativ för snäppning och digitalisering

Globala *inställningar för snäppning och digitalisering* (snäppningsläge, toleransvärde och enheter…) kan åsidosättas i projektet från menyn *Project* ► *Snapping Options…*. I *Snapping and Digitizing Options* kan du också konfigurera några andra egenskaper (snapping-lager, skalgräns, topologi…) *Snapping Toolbar* ger tillgång till de flesta av dessa funktioner.

Som standard är snapping inaktiverat i ett projekt tills du trycker på knappen Enable snapping eller trycker på `S`. Snapping-läget, toleransvärdet och enheterna kan också konfigureras i det här verktygsfältet.

Egenskaper för snäppning

Det finns tre alternativ för att välja det eller de lager som ska fästas på:

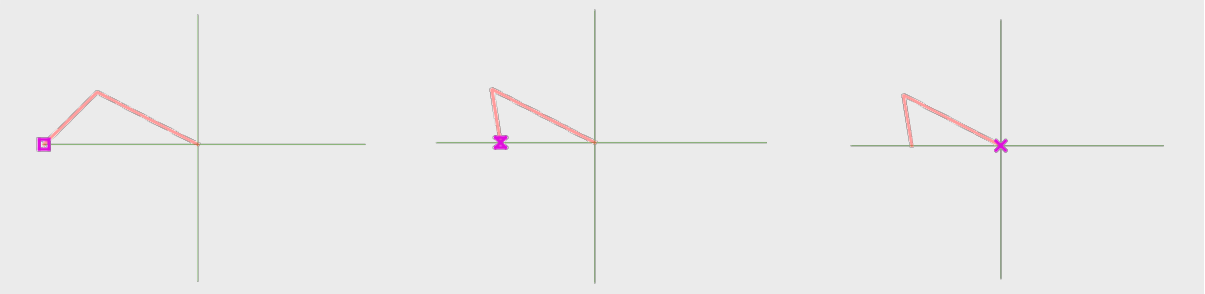
- *Alla lager*: snabbinställning för alla synliga lager i projektet så att pekaren snäpper till alla toppar och/eller segment. I de flesta fall är det tillräckligt att använda detta snapping-läge, men var försiktig när du använder det för projekt med många vektorlager, eftersom det kan påverka prestandan.
- *Aktuellt lager*: endast det aktiva lagret används, ett bekvämt sätt att säkerställa topologisk konsistens inom det lager som redigeras.
- *Advanced Configuration*: gör att du kan aktivera och justera snapping-läge, tolerans och enheter, överlappningar och skalor för snapping på lagerbasis (se Fig. 12.79). Om du behöver redigera ett lager och snäppa dess hörn till ett annat, se till att mållagret är markerat och öka snäppningstoleransen till ett högre värde. Snapping kommer inte att ske till ett lager som inte är markerat i dialogen för snapping-alternativ.

När du flyttar eller skapar en vertex kan du välja mellan följande snapping-lägen:

- *Vertex*
- *Segment*: snäpper längs en linje eller en polygons omkrets. Om topologisk redigering är aktiverad läggs en ny vertex till vid snäppstället.
- *Area*: garanterar att snap-punkten ligger var som helst på en polygons yta, inte nödvändigtvis på dess gräns
- *Centroid*: snäpper till centroiden i geometrin för en funktion. Om geometrin består av flera delar kan målpunkten skilja sig från de befintliga delarna.
- *Mitten av segmenten* på linje eller polygon
- *Line Endpoints*: snäpper till det första eller sista toppunktet i varje del av en linje eller polygon.

QGIS kommer att visa olika *snap*-ikoner beroende på vilken typ av *snap* det är:

Table 12.2: Snappa ikoner



Snappa till en vertex: box-ikon	Snappa till ett segment: timglasikon	Snappa till en korsning: kors-ikon

Observera att det är möjligt att ändra färgen på dessa ikoner i *Digitizing*-delen av de globala inställningarna.

Toleransvärdena kan ställas in antingen i projektets `kartenheter` eller i `pixlar`. Fördelen med att välja ”pixlar” är att snappningen hålls konstant i olika kartskalor. 10 till 12 pixlar är normalt ett bra värde, men det beror på DPI på din skärm. Genom att använda kartans enheter kan toleransen relateras till verkliga markavstånd. Om du t.ex. har ett minsta avstånd mellan element kan det här alternativet vara användbart för att säkerställa att du inte lägger till hörnpunkter för nära varandra.

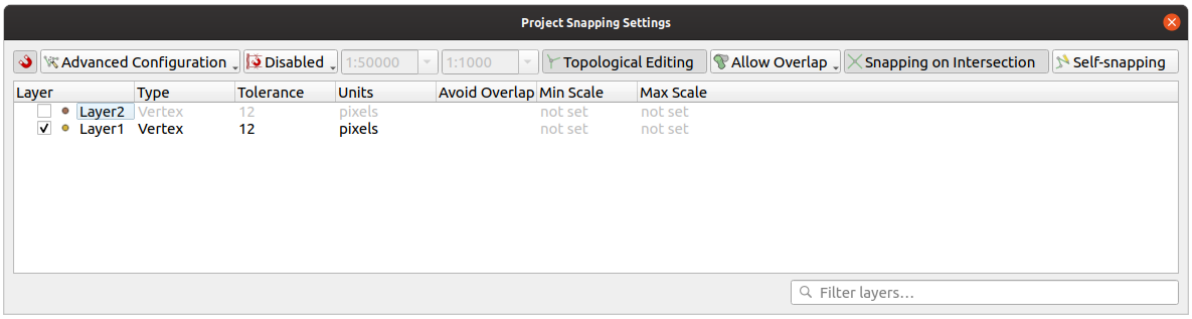


Fig. 12.79: Snapping-alternativ (Avancerat konfigurationsläge)

Observera

Som standard kan endast synliga objekt (de objekt vars stil visas, utom för lager där symbologin är ”Inga symboler”) snappas. Du kan aktivera snäppning av osynliga funktioner genom att markera *Aktivera snäppning av osynliga funktioner* under fliken *Inställningar* ► *Alternativ* ► *Digitalisering*.

Tips

Aktivera snapping som standard

Du kan ställa in att snäppning ska vara aktiverat som standard i alla nya projekt på fliken *Inställningar* ► *Alternativ* ► *Digitalisering*. Du kan också ställa in standardläget för snäppning, toleransvärde och enheter, som kommer att fylla i dialogrutan *Snapping Options*.

Aktivera snapping på korsningar

Ett annat alternativ är att använda *snapping on intersection*, vilket gör att du kan snäppa till geometriska skärningspunkter mellan snäppaktiverade lager, även om det inte finns några toppar i skärningspunkterna.

Begränsa snäppningen till ett skalområde

I vissa fall kan snapping bli mycket långsamt. Detta orsakas ofta av mängden funktioner i vissa lager som kräver ett tungt index för att beräknas och underhållas. Det finns vissa parametrar som gör att snapping endast aktiveras när kartvyn befinner sig inom ett relevant skalintervall. Detta gör det möjligt att endast göra den kostsamma indexberäkningen i samband med snäppning i en skala där ritningen är relevant.

Skalbegränsning för snäppning konfigureras i *Project* ► *Snapping Options*.... Begränsning av snäppning till skala är endast tillgänglig i läget *Avancerad konfiguration*.

För att begränsa snappningen till ett skalområde finns tre lägen tillgängliga:

- *Avaktiverad*: Snapping är aktiverat oavsett den aktuella kartskalan. Detta är standardläget.
- *Global*: Snapping är begränsad och aktiveras endast när kartans aktuella skala ligger mellan ett globalt minimi- och ett globalt maximivärde. När du väljer det här läget blir två widgetar tillgängliga för att konfigurera det skalintervall där snapping är aktiverat.
- *Per lager*: Gränsen för snappingskalan definieras för varje lager. När detta läge väljs blir två kolumner tillgängliga för att konfigurera minsta och största skalor för varje lager.

Observera att minimi- och maximiskalorna följer QGIS-konventionen: minimiskalan är den mest ”utzoomade” skalan medan maximiskalan är den mest ”inzoomade”. En minimi- eller maximiskala som är inställd på ”0” eller ”ej inställd” anses inte vara begränsande.

Självhäftande

Alternativet Self-snapping gör att du kan snäppa till den geometri som redigeras. Kombinerat med *advanced digitizing panel* är detta ett praktiskt sätt att digitalisera nya kanter i förhållande till tidigare kanter eller hörn. Självgnagning kan orsaka ogiltiga geometrier, använd med försiktighet.

Snapping på anpassat rutnät

Ett snäppavstånd kan också anpassas för ett enskilt lager på fliken *Digitizing* i dialogrutan för lageregenskaper. Genom att ställa in avståndet *Geometry precision* aktiverar du ett prickat rutnät som är synligt när kartbilden har en sammanhängande skala för visning. Snapping kan sedan utföras på punkterna i rutnätet: en geometri som läggs till eller ändras kommer att få alla sina hörn automatiskt snäppta till den närmaste noden i rutnätet. Mer information finns på *Egenskaper för digitalisering*.

12.3.3 Topologisk redigering

Förutom dessa snapping-alternativ kan du i dialogrutan *Snapping options…* (*Project* ► *Snapping options*) och verktygsfältet *Snapping* aktivera/inaktivera vissa andra topologiska funktioner.

Aktivera topologisk redigering

Knappen Topologisk redigering hjälper till när du redigerar och underhåller funktioner med gemensamma gränser. När detta alternativ är aktiverat ”upptäcker” QGIS delade gränser. När du flyttar gemensamma hörn/segment kommer QGIS även att flytta dem i geometrierna för de angränsande objekten.

Topologisk redigering fungerar med funktioner från olika lager, så länge lagren är synliga och i redigeringsläge.

I lager med Z- eller M-värden kommer topologisk redigering att interpolera vertexens Z- eller M-värde baserat på värdet för den kant som används för anslutningen.

Överlappande kontroll

Överlappning hindrar dig från att rita nya objekt som överlappar befintliga objekt i det markerade lagret, vilket påskyndar digitaliseringen av intilliggande polygoner. Det kan styras med överlappningsverktyget. Tre lägen finns tillgängliga:

1. *Allow Overlap* (standard)
2. *Avoid Overlap on Active Layer*: förhindrar överlappning med andra objekt från det lager som redigeras. Digitalisera de nya geometrierna så att de överlappar sina grannar och QGIS kommer att klippa de överlappande delarna av de nya geometrierna och fästa dem vid gränsen för de befintliga objekten. Fördelen är att du inte behöver digitalisera de gemensamma hörnen på gränsen.
3. *Follow Advanced Configuration*: gör att överlappningsinställningen kan ställas in för varje lager i visningsläget *Advanced configuration*.

Observera

Om den nya geometrin helt täcks av befintliga geometrier rensas den bort och QGIS visar ett felmeddelande.

Varning

Använd försiktigt *Avvik överlappning* **alternativ**

Eftersom det här alternativet skär nya överlappande geometrier i alla polygonlager kan du få oväntade geometrier om du glömmer att avmarkera det när det inte längre behövs.

Automatisk spårning

När du använder kartverktyg (lägg till funktion, lägg till del, lägg till ring, omforma och dela) måste du vanligtvis klicka på varje vertex i funktionen. Med det automatiska spårningsläget kan du påskynda digitaliseringsprocessen eftersom du inte längre behöver placera alla hörn manuellt under digitaliseringen:

1. Aktivera verktyget Tracing (i verktygsfältet *Snapping*) genom att trycka på ikonen eller trycka på `T`.
2. *Snap to* en topp eller ett segment av en funktion som du vill spåra längs.
3. För musen över ett annat toppunkt eller segment som du vill snäppa och istället för den vanliga raka linjen representerar det digitaliserande gummibandet en väg från den senaste punkten du snäppte till den aktuella positionen. Verktyget fungerar också med krökta geometrier.

   QGIS använder faktiskt de underliggande funktionernas topologi för att bygga den kortaste vägen mellan de två punkterna. Spårning kräver att snäppning aktiveras i spårbara lager för att bygga vägen. Du bör också snäppa till en befintlig vertex eller ett befintligt segment när du digitaliserar och se till att de två noderna är topologiskt anslutningsbara genom befintliga funktionskanter, annars kan QGIS inte ansluta dem och spårar därför en enda rak linje.
4. Klicka och QGIS placerar de mellanliggande topparna efter den visade banan.

Fäll ut ikonen Enable Tracing och ställ in alternativet *Offset* för att digitalisera en bana parallellt med objekten i stället för att spåra längs dem. Ett positivt värde flyttar den nya ritningen till vänster om spårningsriktningen och ett negativt värde gör det motsatta.

Observera

Justera kartskalan eller snäppinställningarna för en optimal avbildning

Om det finns för många funktioner i kartvisningen inaktiveras spårningen för att undvika potentiellt lång förberedelse av spårningsstrukturen och stora minneskostnader. Efter inzoomning eller inaktivering av vissa lager aktiveras spårningen igen.

Observera

Lägger inte till topologiska punkter

Det här verktyget lägger inte till punkter i befintliga polygongeometrier även om *Topologisk redigering* är aktiverat. Om geometriprecision är aktiverat på det redigerade lagret kanske den resulterande geometrin inte exakt följer en befintlig geometri.

Tips

Aktivera eller inaktivera automatisk spårning snabbt genom att trycka på `T` tangenten

Genom att trycka på `T` kan spårning aktiveras/avaktiveras när som helst (även när du digitaliserar ett objekt), så det är möjligt att digitalisera delar av objektet med spårning aktiverad och andra delar med spårning avaktiverad. Verktygen beter sig som vanligt när spårningen är inaktiverad.

Tips

Konvertera spårning till krökta geometrier

Genom att använda *Inställningar* ► *Alternativ* ► *Digitalisering* ► *Spårning* kan du skapa böjda geometrier när du digitaliserar. Se *Digitiseringsalternativ*.

12.3.4 Digitalisering av ett befintligt lager

Som standard laddar QGIS lager skrivskyddat. Detta är en säkerhetsåtgärd för att undvika att ett lager redigeras av misstag om musen skulle slinta. Du kan dock välja att redigera vilket lager som helst så länge dataleverantören stöder det (se *Utforska dataformat och fält*) och den underliggande datakällan är skrivbar (dvs. dess filer är inte skrivskyddade).

Tips

Begränsa redigeringsbehörigheten för lager inom ett projekt

Från *Projekt* ► *Egenskaper…* ► *Data Sources* ► *Layers Capabilities* kan du välja att göra alla lager skrivskyddade oavsett leverantörens behörighet. Detta kan vara ett praktiskt sätt att i en miljö med flera användare undvika att obehöriga användare av misstag redigerar lager (t.ex. Shapefile) och därmed potentiellt korrumperar data. Observera att den här inställningen endast gäller i det aktuella projektet.

I allmänhet är verktygen för redigering av vektorlager indelade i ett digitaliseringsverktygsfält och ett avancerat digitaliseringsverktygsfält, som beskrivs i avsnitt *Avancerad digitalisering*. Du kan välja och avmarkera båda under *View* ► *Toolbars* ►.

Med hjälp av de grundläggande digitaliseringsverktygen kan du utföra följande funktioner:

Table 12.3: Verktygsfält för grundläggande redigering av vektorlager

Verktyg	Syfte	Verkt	Syfte
	Möjlighet att spara, rulla tillbaka eller avbryta ändringar i alla eller valda lager samtidigt		Aktivera eller inaktivera redigeringsstatus för valda lager baserat på statusen för aktivt lager
	Spara ändringar i det aktiva lagret		
	Digitalisera med hjälp av raka segment		Digitalisera med hjälp av kurvlinjer
	Aktivera frihandsdigitalisering		Digitalisera polygon av regelbunden form
	Lägg till ny post		Lägg till funktion: Fångstpunkt
	Lägg till funktion: Fånga upp linje		Lägg till funktion: Fånga polygon
	Vertex-verktyg (alla lager)		Vertex-verktyg (aktuellt lager)
Visa vertexredigera	Ställ in om panelen för vertexredigering ska öppnas automatiskt		Ändra attributen för alla valda funktioner samtidigt
	Ta bort valda funktioner från det aktiva lagret		Klipp bort funktioner från det aktiva lagret
	Kopiera valda funktioner från det aktiva lagret		Klistra in funktioner i det aktiva lagret
	Ångra ändringar i det aktiva lagret		Gör om ändringar i det aktiva lagret

Observera att när du använder något av digitaliseringsverktygen kan du fortfarande *zooma eller panorera* i kartbilden utan att tappa fokus på verktyget.

Alla redigeringssessioner börjar med att du väljer alternativet Växla redigering som finns i kontextmenyn för ett visst lager, i dialogen för attributtabeller, i digitaliseringsverktygsfältet eller i menyn *Lager*.

När lagret är i redigeringsläge blir ytterligare verktygsknappar i redigeringsverktygsfältet tillgängliga och markörer visas vid alla funktioners hörn om inte alternativet *Visa markörer endast för valda funktioner* under *Inställningar* ► *Alternativ…* ► *Digitalisering* är markerat.

Tips

Spara regelbundet

Kom ihåg att Save Layer Edits regelbundet. Detta kommer också att kontrollera att din datakälla kan acceptera alla ändringar.

Tekniker för redigering av geometri

När ett geometriskt ritverktyg (främst de som lägger till, delar upp och omformar funktioner) är aktiverat för ett linje- eller polygonbaserat lager kan du välja teknik för att lägga till nya hörn:

- Digitize with Segment: ritar ett rakt segment vars start- och slutpunkter definieras genom vänsterklick.
- Digitize with Curve: ritar en kurvlinje baserad på tre på varandra följande noder som definieras med vänsterklick (start, punkt längs bågen, slut). Om geometritypen inte stöder kurvor används på varandra följande mindre segment för att approximera krökningen.
- Stream Digitizing: ritar linjer i frihandsläge, dvs. noder läggs till efter markörens rörelse i kartbilden och en *Streaming Tolerance*. Streamingtoleransen definierar avståndet mellan på varandra följande hörn. För närvarande är den enda enhet som stöds pixlar (`px`). Endast vänsterklicket i början och högerklicket i slutet är nödvändiga i det här läget.
- Digitize Shape: aktiverar verktyg i *Shape Digitizing Toolbar* för att rita en polygon av en regelbunden form.

Den valda tekniken behålls när du växlar mellan digitaliseringsverktygen. Du kan kombinera vilken som helst av de tre första metoderna när du ritar samma geometri.

Lägga till funktioner

Beroende på lagertyp kan du använda ikonerna Lägg till post, Lägg till punktfunktion, Lägg till linjefunktion eller Lägg till polygonfunktion i verktygsfältet för att lägga till nya funktioner i det aktuella lagret.

För att lägga till en geometrilös funktion klickar du på knappen Lägg till post och du kan ange attribut i funktionens formulär som öppnas.

För att skapa funktioner med de spatialt aktiverade verktygen digitaliserar du först geometrin och anger sedan dess attribut. Så här digitaliserar du geometrin:

1. (Valfritt eftersom det är standard) Välj geometriritningsmetoden Digitize With Segment (Digitalisera med segment)
2. Vänsterklicka på kartområdet för att skapa den första punkten i din nya funktion. För punktobjekt bör detta vara tillräckligt och utlösa, om så krävs, objektformuläret för att fylla i deras attribut.

3. För linje- eller polygongeometrier fortsätter du att vänsterklicka för varje ytterligare punkt som du vill fånga. Du kan förlita dig på alternativen *snapping to features*, *snap-to-grid* eller *advanced digitizing* för att placera varje vertex korrekt.

   Förutom att rita raka segment mellan noder som du klickar på en efter en, kan linjer och polygoner ritas:

   - *traced automatically*, vilket påskyndar digitaliseringen. Detta kommer att skapa raka linjer mellan de hörnpunkter du placerar, efter befintliga funktioner.
   - frihandsdigitaliseras, tryck på `R` eller aktivera Stream Digitizing.
   - ritas som en kurva, tryck på `Ctrl+Shift+G` eller aktivera Digitize with Curve.

   Observera

   När du digitaliserar linje- eller polygongeometrier kan du växla fram och tillbaka mellan geometriritningsmetoderna, vilket gör att du kan skapa funktioner som blandar raka segment, frihandssegment och krökta delar.

4. Tryck på `Delete` eller `Backspace` för att återställa den eller de sista noderna som du felaktigt har lagt till.
5. När du är klar med att lägga till punkter högerklickar du var som helst på kartområdet för att bekräfta att du är klar med att ange geometrin för den funktionen.

   Tips

   Anpassa det digitaliserande gummibandet

   När du fångar en polygon kan det röda gummibandet som standard dölja underliggande funktioner eller platser där du vill fånga en punkt. Detta kan åtgärdas genom att ställa in en lägre opacitet (eller alfakanal) för gummibandets *Fill Color* i :menuselection:` *Inställningar –> Alternativ –> Digitizing*-menyn. Du kan också undvika att använda gummibandet genom att markera *Don't update rubber band during node editing*.

6. För linjefunktioner stängs linjen automatiskt om du trycker på `Shift` + högerklick.
7. Attributfönstret visas, så att du kan ange informationen för den nya funktionen. Fig. 12.80 visar inställning av attribut för en fiktiv ny flod. Men i menyn *Digitizing* under menyn *Inställningar* ► *Alternativ* kan du också göra det:

   - *Undertrycka attribut popup-fönster efter varje skapad funktion* för att undvika att formuläret öppnas;
   - eller *Återanvänd senast angivna attributvärden* för att ha fält som fylls i automatiskt när formuläret öppnas och bara behöva skriva in ändringsvärden.

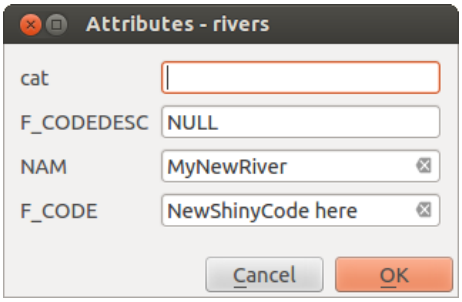


Fig. 12.80: Dialogrutan Enter Attribute Values efter digitalisering av en ny vektorfunktion

Vertex-verktyg

QGIS tillhandahåller två verktyg för att interagera med vektorobjektens vertikaler:

- Vertex Tool (Current Layer): endast överlagrade funktioner i det aktiva lagret (i panelen *Layers*) påverkas
- Vertex Tool (All Layers): alla överlagrade funktioner i alla redigerbara lager påverkas. Detta gör att du kan redigera funktioner utan att byta det aktiva lagret eller redigera flera lager samtidigt (t.ex. land och deras regiongränser)

För alla redigerbara vektorlager ger vertexverktygen manipuleringsmöjligheter för objektets hörnpunkter på liknande sätt som i CAD-program. Det går att välja flera vertex samtidigt och flytta, lägga till eller ta bort dem helt och hållet. Vertexverktygen stöder också den topologiska redigeringsfunktionen. De är valbeständiga, så när någon åtgärd utförs förblir valet aktivt för den här funktionen och det här verktyget.

Tips

Rita en serie nya hörnpunkter

Vertexverktyget stöder inte *automatisk spårning* och är optimerat för redigering av enskilda vertex och för att flytta eller ta bort ett godtyckligt urval av flera vertex. Försök använda verktyget Reshape Features när du behöver ersätta eller infoga en serie nya vertex, särskilt om du vill spåra andra funktioner.

Det är viktigt att ställa in egenskapen *Settings* ► *Options* ► *Digitizing* ► *Search Radius:* till ett tal större än noll. Annars kommer QGIS inte att kunna avgöra vilket vertex som redigeras och kommer att visa en varning.

Tips

Vertexmarkörer

QGIS stöder olika typer av vertexmarkörer: 'Semi-transparent cirkel', 'Kors' och 'Ingen'. För att ändra markörstil, välj *Alternativ* från menyn *Inställningar*, klicka på fliken *Digitizing* och välj lämplig post.

Grundläggande funktioner

Om du har ett lager i redigeringsläge börjar du med att aktivera vertexverktyget. Röda cirklar kommer att visas när du håller muspekaren över vertexen.

- **Välja hörnpunkter**: Du kan välja toppar genom att:
  - Klicka på dem en i taget med `Shift`-tangenten nedtryckt
  - Klicka och dra i en rektangel som omger målets hörnpunkter
  - Rita en polygon som omger målvertikalerna: Håll in `Alt` och klicka med vertexverktyget för att börja digitalisera en polygon. Varje efterföljande klick lägger till en ny vertex till gummibandspolygonen. `Backspace` eller `Delete` tar bort den senast tillagda gummibandsvertexen. `Esc` avbryter polygonmarkeringsläget, liksom backspacing/radering av gummibandets alla vertexer. Högerklick slutför polygondigitaliseringen och väljer alla toppar inom gummibandspolygonen.

  När en vertex är markerad ändras dess färg till blått. Om du vill lägga till fler toppar i det aktuella urvalet håller du ned `Shift` samtidigt som du fortsätter enligt ovan. Om du vill ta bort toppar från markeringen håller du ned `Ctrl`.

  Tips

  Vertexverktyget begränsar urvalet av funktioner

Hörnpunkter kan väljas över olika funktioner (eller lager). Om du letar efter hörnpunkter för en viss funktion på en plats där det är mycket folk, markerar du först den funktionen. Dra sedan rektangel- eller polygonväljaren med vertexverktyget runt topparna: endast den valda funktionens toppar väljs.

Detta gäller även om du visar funktionen i panelen *vertex editor*.

- **Batch-läge för val av vertex**: Batchval-läget kan aktiveras genom att trycka på `Shift+R`. Välj en första nod med ett enda klick och håll sedan muspekaren över en annan vertex **utan att klicka**. Detta kommer dynamiskt att välja alla noder däremellan med hjälp av den kortaste vägen (för polygoner).

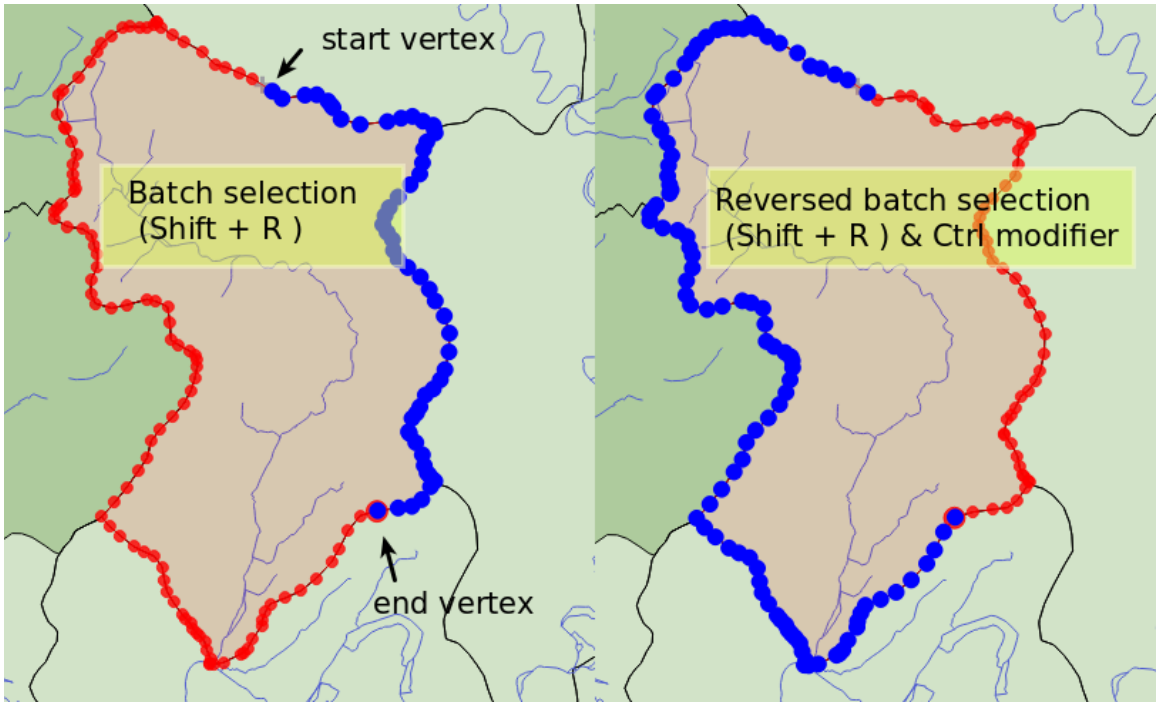


Fig. 12.81: Batchval av vertex med hjälp av `Shift+R`

Tryck på `Ctrl` för att invertera urvalet och välja den längsta vägen längs funktionsgränsen. Avsluta nodvalet med ett andra klick eller tryck på `Esc` för att lämna batch-läget.

- **Lägga till hörnpunkter**: För att lägga till en toppunkt till en linje eller polygongeometri, håll in `Shift` och dubbelklicka på platsen på segmentet.

  När du håller muspekaren över ett segment visas en virtuell ny nod i mitten. Klicka på den, flytta markören till en ny plats och klicka igen för att lägga till ett nytt toppunkt. För linjer föreslås också en virtuell nod i båda ändarna: klicka på den, gör efterföljande klick och avsluta med ett högerklick; detta gör det enkelt att förlänga en befintlig linje.

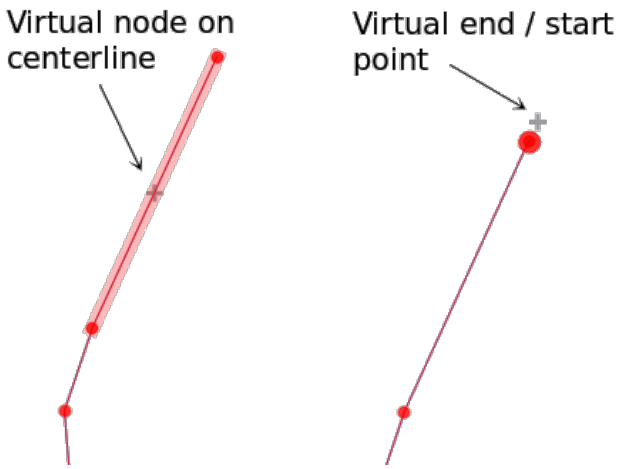


Fig. 12.82: Virtuella noder för att lägga till toppar

- **Radera toppar**: Markera vertikalerna och klicka på `Delete`-tangenten. Om du raderar alla hörn i en funktion genereras en geometrilös funktion, om den är kompatibel med datakällan. Observera att detta inte raderar hela objektet, utan bara geometridelen. Om du vill radera en hel funktion använder du verktyget Delete Selected.
- **Flytta toppar**: Markera alla toppar som du vill flytta, klicka på en markerad topp eller kant och klicka på den önskade nya platsen. Du kan använda *snapping to feature capabilities* och *Advanced Digitizing Panel* för avstånd, vinklar, exakt X- och Y-placering före det andra klicket. Alla valda hörnpunkter kommer att översättas.

  Men om alternativet *snap-to-grid* är aktiverat, snapsas valda vertikaler till den närmaste rutnätskorsningen i förhållande till sin översatta position. Ovalda hörn flyttas också till närmaste rutnätsintersektion. Det finns ingen enkel översättning.

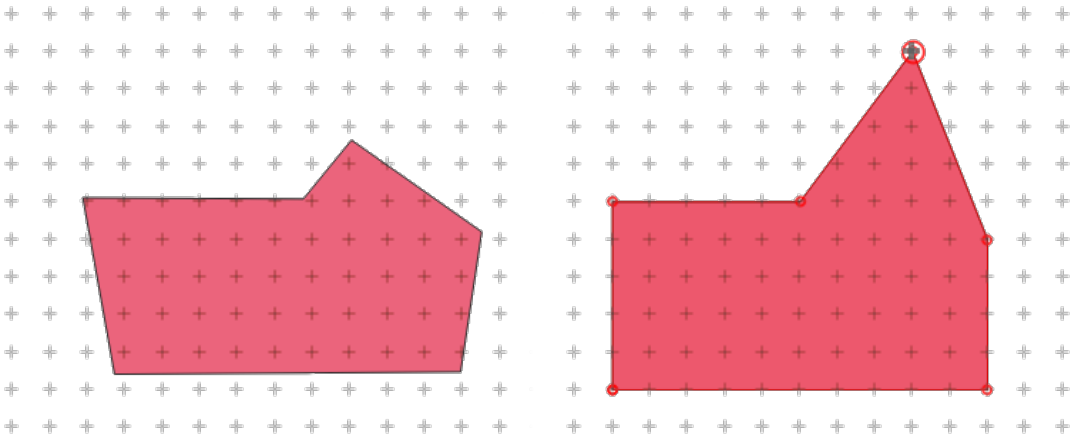


Fig. 12.83: När du flyttar den översta toppunkten fästs alla toppunkterna i rutnätet

- **Konvertera intilliggande segment till/från kurva**: Välj mittpunkten i det segment som du vill konvertera och tryck på bokstavstangenten `O`. Om toppunktet låg i en kurva omvandlas kurvan till raka linjer. Om toppunkten låg mellan två raka linjer omvandlas de till en kurva. Ett första eller sista vertex på en linje kan inte konverteras till en kurva med mittvertex. Lagret måste vara kompatibelt med geometritypen kurva.

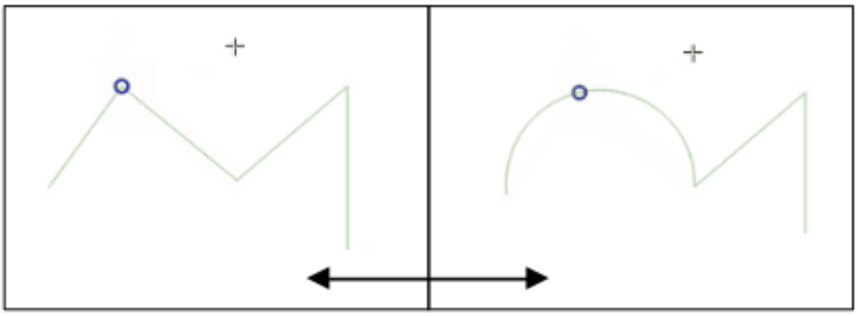


Fig. 12.84: Växla från kurva till raka linjer med ○ brev

Varje ändring som görs med vertexverktyget sparas som en separat post i dialogrutan *Undo*. Kom ihåg att alla operationer stöder topologisk redigering när detta är aktiverat. Projektion i farten stöds också.

Vertexredigerarens panel

När du aktiverar ett vertexverktyg öppnar du också panelen *Vertex Editor*. Om du högerklickar på en funktion fylls panelen med en lista över funktionens alla hörnpunkter med deras *x*, *y* (*z*, *m* om tillämpligt) koordinater och *r* (för radien, vid cirkulär geometri). Funktionen görs också exklusiv för redigering, vilket innebär att redigering av alla andra funktioner inaktiveras:

- Om du markerar en rad i tabellen markeras motsvarande vertex i kartbilden, och vice versa.
- Om du klickar eller drar över kartbilden kommer du bara att välja eller flytta hörnpunkter och segment för den funktionen
- Ändra en koordinat i tabellen så uppdateras vertexpositionen. Detta är ett bekvämt sätt att redigera Z-koordinat eller M-värde på vertexar.
- Du kan också markera flera rader och ta bort dem helt och hållet.
- Nya hörn kan bara läggas till i den avgränsade funktionen

Om du inte vill att panelen *Vertex Editor* ska visas omedelbart varje gång du använder vertexverktyg (och eventuellt dölja andra paneler eller störa placeringen av paneler), avmarkerar du *Auto-open table* i menyn ☰ Options högst upp i panelen. Du kan då också stänga panelen. För att öppna panelen igen måste du högerklicka på en panel eller ett verktygsfält och välja det i listan eller markera *Visa vertexredigerare* i *Digitaliseringsverktygsfält*.

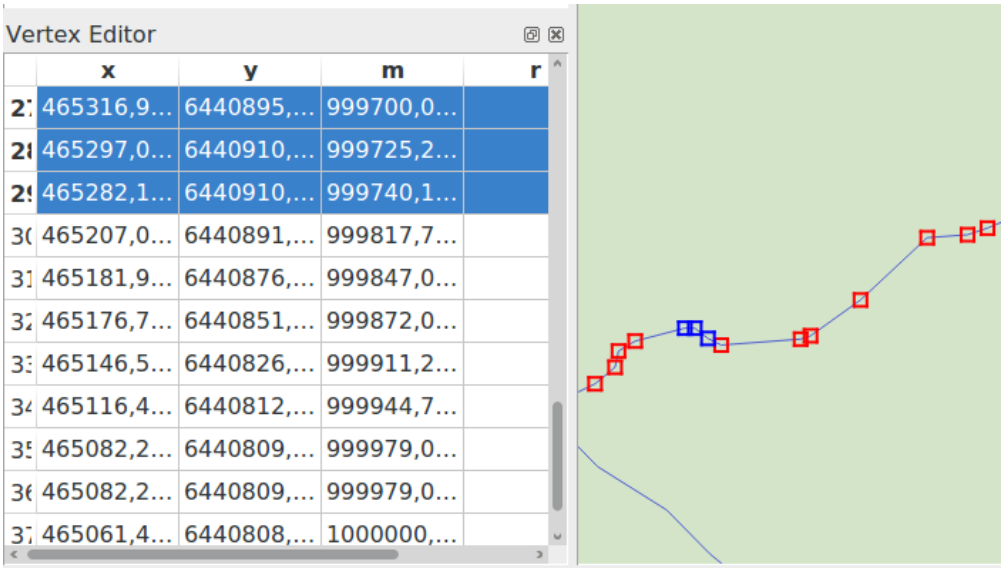


Fig. 12.85: Vertexredigeringspanelen visar valda noder

Regler för tilldelning av Z-koordinat eller M-värde

Att digitalisera 3D-vektorfunktioner eller funktioner med M-värde skiljer sig inte så mycket från (X,Y) 2D-lager. De verktyg och alternativ som beskrivs i det här kapitlet är fortfarande tillgängliga och hjälper dig att placera vertexet eller punkten i en plan miljö. Sedan kan du behöva hantera tilldelningen av Z-koordinaten (eller M-värdet):

- Som standard tilldelar QGIS nya hörn *Default Z value* (respektive *Default M value*) som anges i fliken :menuselection:` *Inställningar –> Alternativ –> Digitizing*. Om *Advanced Digitizing Panel* används, hämtas värdet från dess *z* (respektive *m*) widget.
- När du snäpper till en vertex får den nya eller flyttade vertexen den snäppta vertexens Z- eller M-värde.
- När du snäpper till ett segment medan den topologiska redigeringen är aktiverad interpoleras det nya Z- eller M-värdet för vertexen längs segmentet.
- Om widgeten *z* (respektive *m*) i *Advanced Digitizing Panel* är låst, tillämpas dess värde på vertexen och har företräde framför Z- eller M-värden för vertexen eller segmentet.

För att redigera Z- eller M-värden för en befintlig funktion kan du använda *Vertex editor panel*. Om du vill skapa objekt med anpassade Z- eller M-värden kan du använda *Advanced Digitizing Panel*.

Klipp ut, kopiera och klistra in funktioner

Markerade objekt kan klippas ut, kopieras och klistras in mellan lager i samma QGIS-projekt, förutsatt att destinationslagren i förväg är inställda på Toggle editing.

Tips

Transformera polygon till linje och vice versa med hjälp av kopiera/klistra in

Kopiera en linjefunktion och klistra in den i ett polygonlager: QGIS klistrar in en polygon i mållagret, vars gräns motsvarar den slutna geometrin för linjefunktionen. Detta är ett snabbt sätt att generera olika geometrier av samma data.

Funktioner kan också klistras in i externa applikationer som text. Det innebär att objekten representeras i CSV-format och att geometridata visas i OGC Well-Known Text (WKT)-format. WKT- och GeoJSON-objekt utanför QGIS kan också klistras in i ett lager i QGIS.

När skulle kopiera och klistra in-funktionen komma till nytta? Jo, det visar sig att du kan redigera mer än ett lager åt gången och kopiera/klistra in funktioner mellan lagren. Varför skulle vi vilja göra det här? Säg att vi behöver arbeta med ett nytt lager men bara behöver en eller två sjöar, inte de 5 000 som finns i vårt lager `big_lakes`. Vi kan skapa ett nytt lager och använda kopiera/klistra in för att lägga in de sjöar som behövs i det.

Som ett exempel ska vi kopiera några sjöar till ett nytt lager:

1. Ladda det lager som du vill kopiera från (källager)
2. Ladda eller skapa det lager som du vill kopiera till (mållager)
3. Börja redigera för mållagret
4. Gör källagret aktivt genom att klicka på det i teckenförklaringen
5. Använd verktyget Select Features by area or single click för att välja funktion(er) på källagret
6. Klicka på verktyget Kopiera funktioner
7. Gör destinationslagret aktivt genom att klicka på det i förklaringen
8. Klicka på verktyget Klistra in funktioner
9. Avbryt redigeringen och spara ändringarna

Vad händer om käll- och mållagren har olika scheman (fältnamn och typer är inte desamma)? QGIS fyller i det som matchar och ignorerar resten. Om du inte bryr dig om att attributen kopieras till mållagret spelar det ingen roll hur du utformar fälten och datatyperna. Om du vill vara säker på att allt - objektet och dess attribut - kopieras måste du se till att schemana matchar.

Observera

Kongruens av tidigare funktioner

Om käll- och destinationslagren använder samma projektion kommer de inklistrade objekten att ha en geometri som är identisk med källagrets. Men om destinationslagret har en annan projektion kan QGIS inte garantera att geometrin är identisk. Detta beror helt enkelt på att det förekommer små avrundningsfel vid konvertering mellan projektioner.

Tips

Kopiera strängattribut till ett annat

Om du har skapat en ny kolumn i din attributtabell med typen ”string” och vill klistra in värden från en annan attributkolumn som har en större längd kommer kolumnstorleken att förlängas med samma mängd. Detta beror på att GDAL Shapefile-drivrutinen vet att automatiskt förlänga sträng- och heltalsfält för att dynamiskt anpassa längden på de data som ska infogas.

Radera valda funktioner

Om vi vill ta bort en hel funktion (attribut och geometri) kan vi göra det genom att först välja geometrin med hjälp av det vanliga verktyget Select Features by area or single click. Urvalet kan också göras från attributtabellen. När du har gjort urvalet trycker du på `Delete` eller `Backspace` eller använder verktyget Delete Selected för att radera objekten. Flera valda funktioner kan tas bort samtidigt.

Verktyget Cut Features i digitaliseringsverktygsfältet kan också användas för att radera funktioner. Detta raderar effektivt funktionen men placerar den också på ett ”spatialt klippbord”. Vi klipper alltså ut den funktion som ska raderas. Vi kan sedan använda verktyget Paste Features för att sätta tillbaka den, vilket ger oss en ångerfunktion på en nivå. Klipp ut, kopiera och klistra in fungerar på de aktuella valda funktionerna, vilket innebär att vi kan arbeta med mer än en åt gången.

Ångra och gör om

Verktygen Undo och Redo gör att du kan ångra eller göra om vektorredigeringsoperationer. Det finns också en dockningsbar widget som visar alla operationer i ångra/gör om-historiken (se Fig. 12.86). Den här widgeten visas inte som standard, utan kan visas genom att högerklicka på verktygsfältet och aktivera kryssrutan *Undo/Redo Panel*. Funktionen Undo/Redo är dock aktiv även om widgeten inte visas.

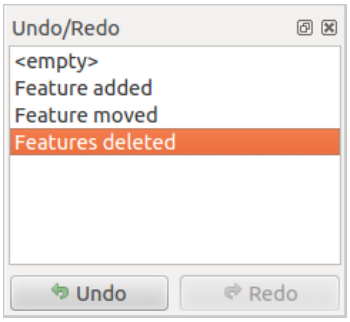


Fig. 12.86: Gör om och ångra digitaliseringssteg

När du trycker på Undo eller `Ctrl+Z` (eller `Cmd+Z`) återställs alla funktioner och attribut till det tillstånd som rådde innan den åtgärd som återställdes utfördes. Andra ändringar än normala vektorredigeringsoperationer (t.ex. ändringar som gjorts av ett tillägg) kan återställas eller inte, beroende på hur ändringarna utfördes.

Om du vill använda widgeten för ångra/återställ historik klickar du bara för att välja en åtgärd i historiklistan. Alla funktioner återställs till det tillstånd de befann sig i efter den valda åtgärden.

Spara redigerade lager

När ett lager är i redigeringsläge ligger alla ändringar kvar i QGIS minne. De överförs/sparas därför inte omedelbart till datakällan eller disken. Om du vill spara ändringar i det aktuella lagret men vill fortsätta redigera utan att lämna redigeringsläget kan du klicka på knappen Save Layer Edits. När du stänger av redigeringsläget med Toggle editing (eller avslutar QGIS för den delen), får du också frågan om du vill spara eller förkasta dina ändringar.

Om ändringarna inte kan sparas (t.ex. om disken är full eller om attributen har värden som ligger utanför intervallet), bevaras QGIS minnesstatus. Detta gör att du kan justera dina ändringar och försöka igen.

Tips

Dataintegritet

Det är alltid en god idé att säkerhetskopiera din datakälla innan du börjar redigera. Även om författarna till QGIS har gjort allt för att bevara integriteten hos dina data, erbjuder vi ingen garanti i detta avseende.

Spara flera lager på en gång

Den här funktionen gör det möjligt att digitalisera flera lager. Välj *Save for Selected Layers* för att spara alla ändringar du gjort i flera lager. Du har också möjlighet att *Rollback for Selected Layers*, så att digitaliseringen kan dras tillbaka för alla valda lager. Om du vill sluta redigera de valda lagren är *Cancel for Selected Layer(s)* ett enkelt sätt.

Samma funktioner finns tillgängliga för redigering av alla lager i projektet.

Tips

Använd transaktionsgrupp för att redigera, spara eller återställa flera lagerändringar samtidigt

När du arbetar med lager från samma PostGreSQL-databas, aktivera alternativet *Automatiskt skapa transaktionsgrupper där det är möjligt* i *Projekt* ► *Egenskaper…* ► *Datakällor* för att synkronisera deras beteende (gå in eller ut ur redigeringsläget, spara eller rulla tillbaka ändringar samtidigt).

12.3.5 Avancerad digitalisering

Table 12.4: Verktygsfält för avancerad redigering av vektorlager

Ikon	Syfte	Ikon	Syfte
	Aktivera avancerade digitaliseringsverktyg		
	Flytta funktion(er)		Kopiera och flytta funktion(er)
	Rotera funktion(er)		Förenkla funktionen
	Skalfunktion		
	Lägg till ring		Lägg till del
	Fyllningsring		Byt riktning
	Radera ring		Ta bort del
	Offset-kurva		Omforma funktioner
	Delade delar		Delade funktioner
	Slå samman attribut för valda funktioner		Slå samman valda funktioner
	Rotera punktsymboler		Symboler för offsetpunkter
	Trimma eller utöka funktionen		

Flytta funktion(er)

Med verktyget Move Feature(s) kan du flytta befintliga funktioner:

1. Välj den eller de funktioner som ska flyttas.
2. Klicka på kartbilden för att ange förskjutningens ursprungspunkt; du kan förlita dig på snäppfunktionerna för att välja en exakt punkt.

   Du kan också använda *advanced digitizing constraints* för att exakt ställa in koordinaterna för ursprungspunkten. I det fallet:

   1. Klicka först på knappen för att aktivera panelen.
   2. Skriv `x` och ange motsvarande värde för den ursprungspunkt du vill använda. Tryck sedan på knappen bredvid alternativet för att låsa värdet.
   3. Gör samma sak för koordinaten `y`.
   4. Klicka på kartbilden och din startpunkt placeras vid de angivna koordinaterna.
3. Flytta över kartbilden för att ange destinationspunkten för förskjutningen, fortfarande med hjälp av snäppläge eller, som ovan, använd den avancerade digitaliseringspanelen som skulle ge kompletterande `avstånd` och `vinkel` placeringsbegränsningar för att placera slutpunkten för översättningen.
4. Klicka på kartbilden: alla funktioner flyttas till den nya platsen.

På samma sätt kan du skapa en översatt kopia av funktionen/funktionerna med hjälp av verktyget Copy and Move Feature(s).

Observera

Om ingen funktion är markerad när du först klickar på kartbilden med något av verktygen *Move Feature(s)* eller *Copy and Move Feature(s)* påverkas endast den funktion som är under musen av åtgärden. Om du vill flytta flera objekt måste du alltså välja dem först.

Rotera funktion(er)

Använd verktyget Rotate Feature(s) för att rotera en eller flera funktioner i kartbilden:

1. Tryck på ikonen Rotera funktion(er)
2. Klicka sedan på den funktion som ska roteras. Funktionens centroid refereras till som rotationscentrum, en förhandsgranskning av den roterade funktionen visas och en widget öppnas som visar den aktuella *Rotation*-vinkeln.
3. Klicka på kartbilden när du är nöjd med den nya placeringen eller ange rotationsvinkeln manuellt i textrutan. Du kan också använda rutan *Snap to* ° för att begränsa rotationsvärdena.
4. Om du vill rotera flera objekt samtidigt ska de väljas först, och rotationen sker som standard runt centroiden för deras kombinerade geometrier.

Du kan också använda en annan ankarpunkt än standardfunktionens centroid: tryck på knappen `Ctrl`, klicka på kartbilden och den punkten kommer att användas som nytt rotationscentrum.

Om du håller in `Shift` innan du klickar på kartan kommer rotationen att göras i 45 graders steg, som kan ändras i efterhand i widgeten för användarinmatning.

Om du vill avbryta rotationen av funktioner trycker du på knappen `ESC` eller klickar på ikonen Rotera funktion(er).

Skalfunktion

Verktyget Scale Feature liknar verktyget Rotate feature. Men istället för att utföra en rotation av valda funktioner skalar det om deras geometri. Ändringen utförs i förhållande till ankarpunkten och skalförhållandet kan anges manuellt i widgeten som visas i det övre hörnet av arbetsytan.

Förenkla funktionen

Med verktyget Simplify Feature kan du interaktivt omforma en linje- eller polygongeometri genom att minska eller förtäta antalet hörn, så länge geometrin är giltig:

1. Välj verktyget Förenkla funktion.
2. Klicka på objektet eller dra en rektangel över objektet.
3. En dialogruta dyker upp där du kan definiera vilken *Method* som ska tillämpas, dvs. om du vill göra det:
   - *Förenkla geometrin*, vilket innebär färre hörn än originalet. Tillgängliga metoder är `Förenkla efter avstånd`, `Simplify by snapping to grid` eller `simplify by area (Visvalingam)`. Du måste sedan ange värdet på *Tolerans* i `Layer units`, `Pixels` eller `map units` som ska användas för förenkling. Ju högre toleransen är desto fler toppar kan tas bort.
   - eller *förtäta geometrierna* med nya toppar tack vare alternativet `Smooth`: för varje befintlig topp placeras två toppar på vart och ett av de segment som härrör från den, på ett *Offset*-avstånd som representerar procentandelen av segmentlängden. Du kan också ställa in antalet *Iterations* som placeringen ska bearbetas: ju fler iterationer, desto fler vertex och jämnare blir funktionen.

   Inställningar som du har använt sparas när du lämnar ett projekt eller en redigeringssession. På så sätt kan du gå tillbaka till samma parametrar nästa gång du förenklar en funktion.
4. En sammanfattning av de ändringar som skulle gälla visas längst ned i dialogrutan, med en lista över antal features och antal vertices (före och efter operationen och det förhållande som ändringen motsvarar). I kartbilden visas också den förväntade geometrin över den befintliga med hjälp av gummibandsfärgen.
5. När den förväntade geometrin passar dina behov klickar du på *OK* för att tillämpa modifieringen. Om du vill avbryta åtgärden kan du antingen trycka på *Cancel* eller högerklicka i kartbilden.

Observera

Till skillnad från alternativet för förenkling av objekt i menyn *Inställningar* ► *Alternativ* ► *Rendering* som förenklar geometrin bara för rendering, ändrar verktyget Simplify Feature objektets geometri permanent i datakällan.

Lägg till del

Du kan Add Part till en vald funktion som genererar en multipunkt-, multilinje- eller multipolygonfunktion. Den nya delen måste digitaliseras utanför den befintliga delen som bör väljas i förväg.

Verktyget Add Part kan också användas för att lägga till en geometri till en geometrilös funktion. Välj först objektet i attributtabellen och digitalisera den nya geometrin med verktyget Add Part.

Observera

Ordningsföljd för hörn i polygondelar

Till skillnad från OGC-standarderna kräver inte QGIS att hörnpunkterna i den yttre gränsen för en polygonfunktion ska ordnas moturs. Du kan alltså hitta båda riktningarna i ett lager. Alla delar av samma multipolygonfunktion kommer dock att ha sina yttre hörn ordnade efter samma riktning.

Du kan dock använda *Kraft högerhands-regel*-algoritmen för att begränsa funktioner i ett lager så att topparna i deras yttre gränser ordnas i medurs riktning.

Ta bort del

Med verktyget Delete Part kan du ta bort delar från flerdelade objekt (t.ex. ta bort polygoner från ett objekt med flera polygoner). Verktyget fungerar med alla flerdelade geometrier: punkt, linje och polygon. Dessutom kan det användas för att helt ta bort den geometriska komponenten i en funktion. För att ta bort en del klickar du helt enkelt inom måldelen.

Lägg till ring

Du kan skapa ringpolygoner med hjälp av ikonen Add Ring i verktygsfältet. Detta innebär att det inom ett befintligt område är möjligt att digitalisera ytterligare polygoner som kommer att uppstå som ett ”hål”, så att endast området mellan gränserna för de yttre och inre polygonerna kvarstår som en ringpolygon.

För att lägga till en ring:

1. Välj den eller de funktioner som ska modifieras
2. Aktivera verktyget Add Ring
3. Rita en polygon inom de valda geometrierna med hjälp av ovannämnda *tekniker*. Ett hål visas i de valda geometrierna.
4. Om ingen geometri är vald när ringen ritas, läggs ett hål till i var och en av de polygoner som ringen ritas över.

Observera

Ordningsföljd för hörn i polygonringar

Till skillnad från OGC-standarderna kräver inte QGIS att hörnpunkterna i den yttre gränsen för en polygonfunktion ska ordnas moturs. Du kan alltså hitta båda riktningarna i ett lager. Alla ringar i samma (multi)polygonfunktion kommer dock att ha sina toppar ordnade i motsatt riktning mot den yttre gränsen.

Du kan dock använda algoritmen *Kraft högerhands-regel* för att begränsa funktioner i ett lager så att hörnpunkterna i deras yttre gränser ordnas medurs och hörnpunkterna i deras inre ringar ordnas moturs.

Fyllningsring

Verktyget Fill Ring hjälper dig att skapa polygonelement som helt faller inom ett annat utan överlappande område; det vill säga det nya elementet täcker ett hål i det befintliga. För att skapa en sådan funktion:

1. Välj verktyget Fill Ring.
2. Rita en ny polygon över den befintliga funktionen: QGIS lägger till en ring till dess geometri (som om du använde verktyget Add Ring) och skapar en ny funktion vars geometri matchar ringen (som om du *traced* över de inre gränserna med verktyget Add polygon feature).

3. Eller alternativt, om ringen redan finns på objektet, placera musen över ringen och vänsterklicka medan du trycker på `Shift`: ett nytt objekt som fyller hålet ritas på den platsen.

   Formuläret *Feature Attributes* för den nya funktionen öppnas, förifyllt med värden för den ”överordnade” funktionen och/eller *fields constraints*.

Radera ring

Med verktyget Delete Ring kan du ta bort ringar inom en befintlig polygon genom att klicka inuti hålet. Detta verktyg fungerar endast med polygon- och multipolygonfunktioner. Det ändrar ingenting när det används på polygonens yttre ring.

Omforma funktioner

Du kan omforma linje- och polygonfunktioner med hjälp av verktyget Reshape Features i verktygsfältet. För linjer ersätter verktyget linjedelen från den första till den sista skärningspunkten med den ursprungliga linjen.

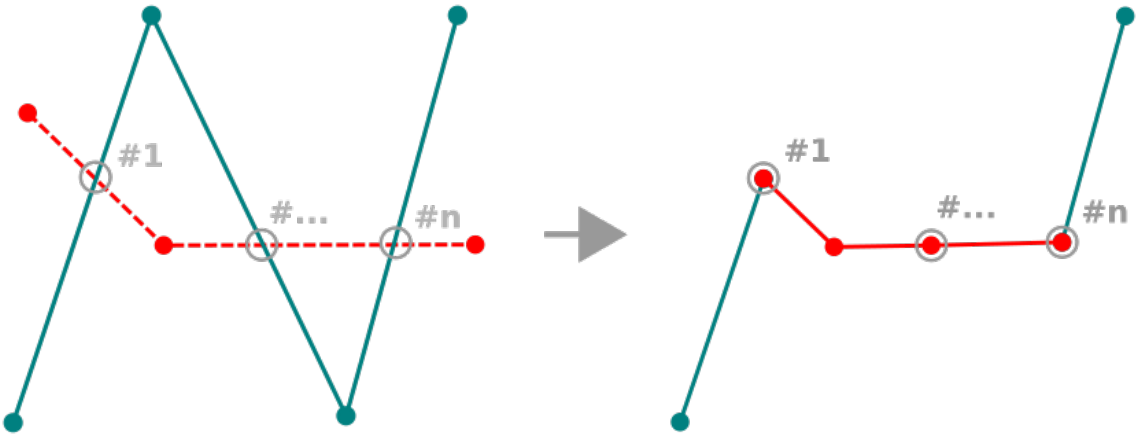


Fig. 12.87: Omforma linjen

Tips

Utöka geometrier med linjestring med verktyget reshape

Använd verktyget Reshape Features för att utöka befintliga linjegeometrier: snäpp till linjens första eller sista toppunkt och rita en ny. Validera och objektets geometri blir en kombination av de två linjerna.

För polygoner kommer det att omforma polygonens gräns. För att det ska fungera måste omformningsverktygets linje korsa polygonens gräns minst två gånger. För att rita linjen klickar du på kartans canvas för att lägga till vertexer. För att avsluta det är det bara att högerklicka. Precis som med linjerna är det bara segmentet mellan den första och den sista korsningen som beaktas. De segment av omformningslinjen som ligger innanför polygonen kommer att resultera i att den beskärs, medan de som ligger utanför polygonen kommer att förlänga den.

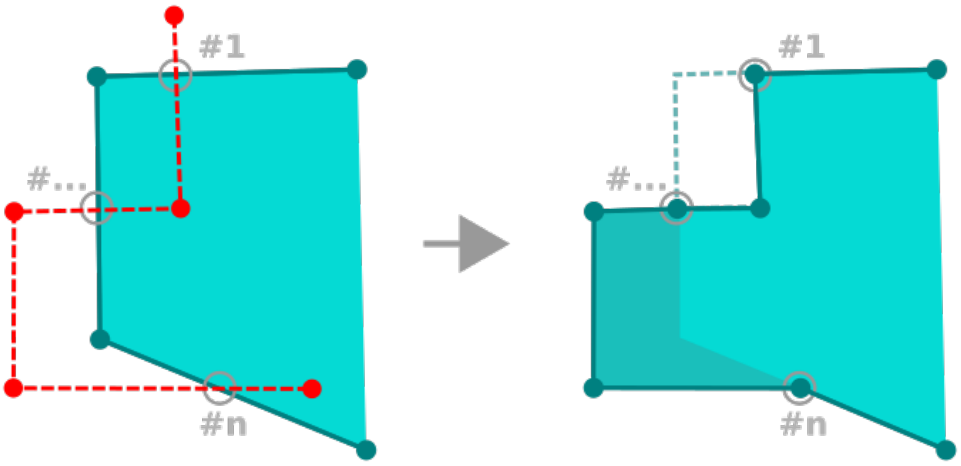


Fig. 12.88: Omforma polygon

Med polygoner kan omformning ibland leda till oavsiktliga resultat. Det är främst användbart för att ersätta mindre delar av en polygon, inte för större översyner, och omformningslinjen får inte korsa flera polygonringar, eftersom detta skulle generera en ogiltig polygon.

Observera

Verktyget reshape kan ändra startpositionen för en polygonring eller en sluten linje. Den punkt som representeras ”två gånger” kommer alltså inte längre att vara densamma. Detta är kanske inte ett problem för de flesta applikationer, men det är något att tänka på.

Offset-kurvor

Verktyget Offset Curve skapar parallella förskjutningar av linjelager. Verktyget kan användas på det redigerade lagret (geometrierna ändras) eller på bakgrundslager (i så fall skapas kopior av linjerna/ringarna och läggs till i det redigerade lagret). Den är därför idealisk för att skapa lager med avståndslinjer. Dialogrutan *User Input* dyker upp och visar förskjutningsavståndet och andra inställningar.

För att skapa en förskjutning av ett linje- eller polygonlager måste du först gå till redigeringsläget och aktivera verktyget Offset Curve. Klicka sedan på en funktion för att skifta den. Flytta musen och klicka där du vill eller ange önskat avstånd i widgeten för användarinmatning. Håll in `Ctrl` under det andra klicket för att göra en offsetkopia. Dina ändringar kan sedan sparas med verktyget Save Layer Edits.

För geometrier på bakgrundslager, se till att snapping är aktiverat och håll in `Ctrl` för att välja geometrin från bakgrunden. Håll även inne `Ctrl` när du gör det andra klicket. Geometrierna kommer att konverteras till mållagrets geometrityp.

I QGIS alternativdialog (fliken Digitalisering och avsnittet **Värktyg för kurvoffset**) eller ikonen i dialogrutan *User Input* kan du konfigurera *vissa parametrar* som **Join style**, **Quadrant segments**, **Miter limit** och **End cap style**.

Omvänd linje

Att ändra riktning på en linjegeometri kan vara användbart för kartografiska ändamål eller vid förberedelser för nätverksanalys.

Ändra riktning på en linje:

1. Aktivera verktyget för omvänd linje genom att klicka på Omvänd linje.
2. Klicka på linjen. Linjens riktning vänds.

Delade funktioner

Använd verktyget Split Features för att dela upp en funktion i två eller flera nya och oberoende funktioner, dvs. varje geometri motsvarar en ny rad i attributtabellen.

För att dela linje- eller polygonfunktioner:

1. Välj verktyget Split Features.
2. Dra en linje över den eller de funktioner som du vill dela. Om en markering är aktiv delas endast markerade features. Den ursprungliga funktionen tilldelas sedan den största geometri som blir resultatet av delningen, och nya funktioner skapas för de återstående delarna. Fälten i funktionerna fylls i/uppdateras enligt reglerna för datakällans leverantör eller deras *splitting policy*.
3. Du kan sedan som vanligt modifiera alla attribut för den resulterande funktionen.

Tips

Dela upp en polylinje i nya funktioner med ett klick

Använd verktyget Split Features och klicka på ett befintligt toppunkt i en polylinjefunktion för att dela upp funktionen i två nya funktioner.

Delade delar

I QGIS är det möjligt att dela upp delarna i en flerdelad funktion så att antalet delar ökar. Dra bara en linje över den del som du vill dela med hjälp av ikonen Split Parts.

Tips

Dela upp en polylinje i nya delar med ett klick

Använd verktyget Split Parts och klicka på ett befintligt toppunkt i en polylinjefunktion för att dela upp funktionen i två nya polylinjer som tillhör samma funktion.

Slå samman utvalda funktioner

Med verktyget Merge Selected Features kan du skapa en ny funktion genom att slå samman befintliga funktioner: deras geometrier slås samman för att generera en ny. Om funktionerna inte har gemensamma gränser skapas en multipolygon/multipolylinje/multipunktsfunktion.

1. Välj först de funktioner som du vill kombinera.
2. Tryck sedan på knappen Merge Selected Features.
3. I den nya dialogrutan visar *Merge*-raden längst ned i tabellen attributen för den resulterande funktionen. Du kan ändra något av dessa värden antingen genom att:
   - ersätter manuellt värdet i motsvarande cell;
   - välja en rad i tabellen och trycka på *Take attributes from selected feature* för att använda värdena för denna första funktion;
   - tryck på *Take attributes from the largest geometry* för att använda attributen från den längsta linjefunktionen, den största polygonen eller multipunkterna med flest delar;
   - tryck på *Skippa alla fält* för att använda tomma attribut;
   - expandera rullgardinsmenyn längst upp i tabellen och välj något av ovanstående alternativ som endast ska gälla för motsvarande fält. Där kan du också välja att aggregera de ursprungliga funktionernas attribut (Minimum, Maximum, Median, Sum, Count, Concatenation … beroende på fältets typ. se *Panel för statistisk sammanfattning* för en fullständig lista över funktioner).

   Observera

   Om lagret har standardvärden eller klausuler för fält används dessa som det ursprungliga värdet för den sammanslagna funktionen.
4. Tryck på *OK* för att tillämpa ändringarna. En enskild (multi)feature skapas i lagret och ersätter de tidigare valda.

Slå samman attribut för utvalda funktioner

Med verktyget Merge Attributes of Selected Features kan du använda samma attribut på objekt utan att sammanfoga deras gränser. Dialogrutan är densamma som för verktyget `Merge Selected Features`, men till skillnad från det verktyget behåller valda objekt sin geometri medan vissa av deras attribut görs identiska.

Rotera punktsymboler

Med funktionen Rotate Point Symbols kan du individuellt ändra rotationen för punktsymboler i kartbilden.

1. Först måste du ange det fält som rotationsvärdet ska lagras i. Detta görs genom att tilldela ett fält till symbolen *data-defined* rotationsegenskap:
   1. I dialogrutan *Lageregenskaper* ► *Symbology* bläddrar du till dialogrutan för symbolredigering.
   2. Klicka på widgeten *Data-defined override* nära alternativet *Rotation* på den översta *Marker*-nivån (helst) i symbollagren.
   3. Välj ett fält i kombinationsrutan *Field Type*. Värdena i detta fält används för att rotera varje objekts symbol i enlighet med detta.

      Du kan också kontrollera Lagra data i projekt för att generera ett *auxiliary data storage*-fält för att styra rotationsvärdet.

> **Observera**
>
> **Se till att samma fält tilldelas alla symbollager**
>
> Om det datadefinierade rotationsfältet ställs in på symbolträdets översta nivå sprids det automatiskt till alla symbollager, vilket är en förutsättning för att kunna utföra grafisk symbolrotation med verktyget *Rotate Point Symbols*. Om ett symbollager inte har samma fält kopplat till sin rotationsegenskap kommer verktyget inte att fungera.

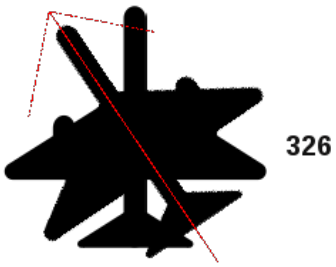


Fig. 12.89: Rotera en punktsymbol

2. Klicka sedan på en punktsymbol i kartbilden med verktyget Rotate Point Symbols (rotera punktsymboler)
3. Flytta musen runt. En röd pil med rotationsvärdet kommer att visualiseras (se Fig. 12.89). Om du håller ned `Ctrl`-tangenten medan du rör dig kommer rotationen att ske i steg om 15 grader.
4. När du får det förväntade vinkelvärdet klickar du igen. Symbolen renderas med den nya rotationen och det tillhörande fältet uppdateras i enlighet med detta.

   Du kan högerklicka för att avbryta symbolrotationen.

Symboler för offsetpunkter

Med Offset Point Symbols kan du interaktivt ändra den återgivna positionen för punktsymboler i kartbilden. Det här verktyget fungerar som verktyget Rotate Point Symbols förutom att du måste ansluta ett fält till den datadefinierade egenskapen *Offset (X,Y)* för varje lager i symbolen. Fältet kommer sedan att fyllas med offsetkoordinaterna för de objekt vars symbol flyttas i kartbilden.

1. Associera ett fält till den datadefinierade widgeten för egenskapen *Offset (X,Y)* i symbolen. Om symbolen består av många lager kanske du vill tilldela fältet till vart och ett av dem
2. Välj verktyget Offset Point Symbols
3. Klicka på en punktsymbol
4. Flytta till en ny plats
5. Klicka igen. Symbolen flyttas till den nya platsen. Offset-värden från den ursprungliga positionen lagras i det länkade fältet.

   Du kan högerklicka för att avbryta symboloffset.

> **Observera**

Verktyget Offset Point Symbols flyttar inte själva punktfunktionen; du bör använda verktyget Vertex Tool (Current Layer) eller Move Feature för detta ändamål.

Trim/Extend-funktion

Med verktyget Trim/Extend kan du förkorta eller förlänga segment av en (multi)linje- eller (multi)polygongeometri så att de konvergerar med ett valt segment (skärlinjen). Detta resulterar i en modifierad geometri med en vertex som är knäppt till målsegmentet eller i dess förlängning. Beroende på hur de valda geometrierna är placerade i förhållande till varandra kommer verktyget antingen:

- **Trim**: tar bort delar av linjesegmentet eller polygongränsen, bortom skärlinjen
- **Extend**: förlänger polygongränser eller linjesegment så att de kan snäppas till skärlinjen.

För att trimma eller utöka befintliga geometrier:

1. Aktivera lämpliga *snapping-inställningar* på segmentet för det/de berörda lagret/lagren
2. Välj verktyget Trim/Extend
3. Klicka på målbegränsningssegmentet, dvs. det segment i förhållande till vilket du vill förlänga eller trimma ett annat segment. Det visas markerat.
4. Flytta till det segment som du vill trimma eller förlänga. Det behöver inte vara det sista segmentet i geometrin, men det måste finnas på det aktiva lagret.
5. Håll muspekaren över segmentet så visar QGIS en förhandsgranskning av hur funktionens geometri skulle se ut. Om OK, klicka på segmentet. När det gäller en trimning måste du välja den del som ska förkortas.
6. När båda segmenten är i 3D utför verktyget en interpolering på gränsegmentet för att få fram Z-värdet.

Uppmärksamma

Var uppmärksam på den modifierade geometrin när du använder verktyget Trim/Extend. Beroende på indata kan det skapa ogiltiga geometrier, vilket kan leda till fel vid lagersparandet.

12.3.6 Digitalisering av form

Verktygsfältet *Shape Digitizing* erbjuder en uppsättning verktyg för att rita linjer eller polygoner med regelbunden form. Det är synkroniserat med Digitize Shape *geometri ritmetod* som du kan välja på *Digitizing Toolbar*. För att använda den:

1. Visa verktygsfältet: *Visa* ► *Verktygsfält* ► *Shape Digitizing*
2. Välj ett verktyg som skapar eller ändrar formen på en geometri, t.ex. Add line feature, Add polygon feature, Add part, Add ring, Reshape Features, …
3. Knappen Digitalisera med segment i verktygsfältet *Digitalisering* är aktiverad. Första gången kan du behöva ändra den till Digitize Shape för att aktivera verktygen i verktygsfältet *Shape Digitizing*.
4. Välj ett digitaliseringsverktyg för form och rita.

Cirkulär sträng efter radie

Knappen Cirkulär sträng med radie gör det möjligt att lägga till linje- eller polygonfunktioner med en cirkulär geometri, givet två noder på kurvan och en radie:

1. Vänsterklicka två gånger för att placera de två punkterna på geometrin.
2. En *Radius*-widget i det övre högra hörnet av kartbilden visar aktuell radie (motsvarar avståndet mellan punkterna). Redigera det fältet till det värde du vill ha.
3. En översikt över de bågar som matchar dessa begränsningar visas medan markören flyttas runt. Högerklicka för att validera när den förväntade bågen visas.
4. Lägg till en ny punkt för att börja forma en ny båge.

Observera

Krökta geometrier lagras som sådana endast i kompatibla dataleverantörer

Även om QGIS gör det möjligt att digitalisera krökta geometrier i alla redigerbara dataformat måste du använda en dataleverantör (t.ex. PostGIS, minneslager, GML eller WFS) som stöder kurvor för att få objekt lagrade som krökta, annars segmentiserar QGIS cirkelbågarna.

Rita cirklar

Det finns en uppsättning verktyg för att rita cirklar. Verktygen beskrivs nedan.

Cirklar konverteras till cirkulära strängar. Därför, som förklaras i *Cirkulär sträng efter radie*, om det tillåts av dataleverantören, kommer det att sparas som en krökt geometri, om inte, kommer QGIS att segmentera de cirkulära bågarna.

- Cirkel från 2 punkter: De två punkterna definierar cirkelns diameter och orientering. (Vänsterklicka, högerklicka)
- Cirkel från 3 punkter: Ritar en cirkel från tre kända punkter på cirkeln. (Vänsterklicka, vänsterklicka, högerklicka)
- Cirkel med en mittpunkt och en annan punkt: Ritar en cirkel med en given mittpunkt och en punkt på cirkeln (vänsterklicka, högerklicka). När det används med *Panelen för avancerad digitalisering* kan detta verktyg bli ett ”Lägg till cirkel från centrum och radie”-verktyg genom att ställa in och låsa avståndsvärdet efter första klicket.
- Cirkel från 3 tangenter: Ritar en cirkel som är tangentiell mot tre segment. **Observera att du måste aktivera snäppning till segment** (Se *Inställning av snäpptolerans och sökradie*). Klicka på ett segment för att lägga till en tangent. Om två tangenter är parallella, används koordinaterna för klicket på den första parallella tangenten för att bestämma cirkelns positionering. Om tre tangenter är parallella, visas ett felmeddelande och inmatningen rensas. (Vänsterklick, vänsterklick, högerklick)
- Cirkel från 2 tangenter och en punkt: Liknar cirkel från 3 tangenter, förutom att du måste välja två tangenter, ange en radie och välja önskat centrum.

Rita ellipser

Det finns en uppsättning verktyg för att rita ellipser. Verktygen beskrivs nedan.

Ellipser kan inte omvandlas till cirkulära strängar, så de kommer alltid att vara segmenterade.

- Ellips från centrum och två punkter: Ritar en ellips med ett givet centrum, huvudaxel och sidoaxel. (Vänsterklicka, vänsterklicka, högerklicka)
- Ellips från centrum och en punkt: Rita en ellips i en avgränsande box med centrum och ett hörn. (Vänsterklicka, högerklicka)
- Ellips från utsträckning: Ritar en ellips i en begränsningsbox med två motsatta hörn. (Vänsterklicka, högerklicka)
- Ellips från foci: Ritar en ellips med 2 punkter för foci och en punkt på ellipsen. (Vänsterklicka, vänsterklicka, högerklicka)

Rita rektanglar

Det finns en uppsättning verktyg för att rita rektanglar. Verktygen beskrivs nedan.

- Rektangel från centrum och en punkt: Ritar en rektangel från mitten och ett hörn. (Vänsterklicka, högerklicka)
- Rektangel från utsträckning: Ritade en rektangel från två motsatta hörn. (Vänsterklicka, högerklicka)
- Rektangel från 3 punkter (avstånd): Ritade en orienterad rektangel från tre punkter. Den första och andra punkten bestämmer längden och vinkeln på den första kanten. Den tredje punkten bestämmer längden på den andra kanten. Man kan använda *Panelen för avancerad digitalisering* för att ställa in längden på kanterna. (Vänsterklicka, vänsterklicka, högerklicka)
- Rektangel från 3 punkter (projicerad): Samma som föregående verktyg, men längden på den andra kanten beräknas utifrån projektionen av den tredje punkten på den första kanten. (Vänsterklicka, vänsterklicka, högerklicka)

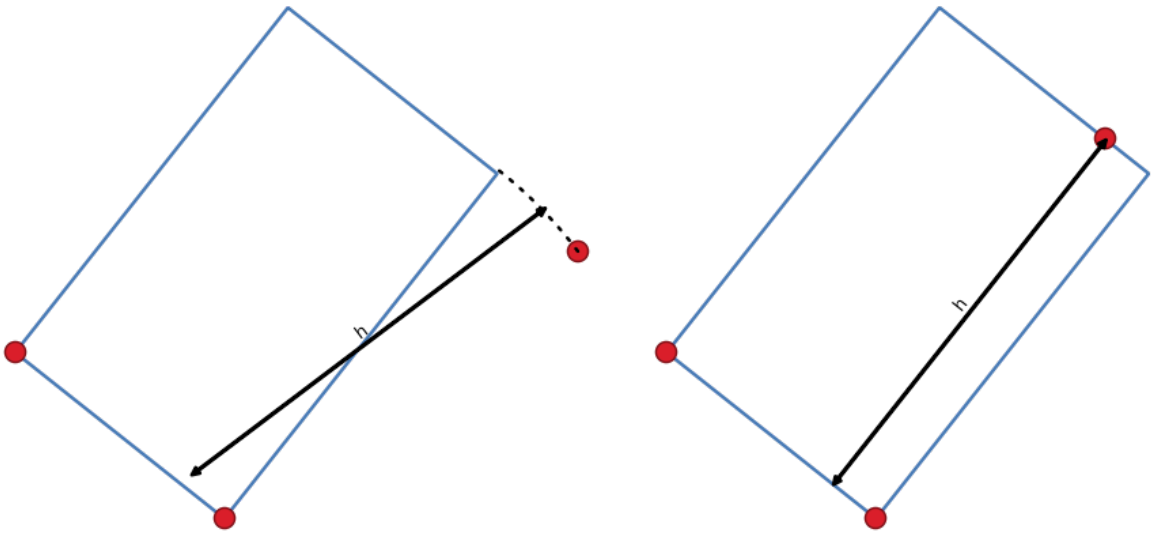


Fig. 12.90: Rita en rektangel från 3 punkter med hjälp av avstånd (höger) och projektion (vänster)

Rita regelbundna polygoner

Det finns en uppsättning verktyg för att rita regelbundna polygoner. Verktygen beskrivs nedan. Vänsterklicka för att placera den första punkten. En dialogruta visas där du kan ställa in antalet polygonkanter. Högerklicka för att avsluta den regelbundna polygonen.

- Regelbunden polygon från två punkter: Tecknar en regelbunden polygon där de två punkterna bestämmer längden och vinkeln på den första kanten.
- Regelbunden polygon från centrum och en punkt: Ritar en regelbunden polygon från den angivna mittpunkten. Den andra punkten bestämmer vinkeln och avståndet till mitten av en kant.
- Regelbunden polygon från centrum och ett hörn: Samma som föregående verktyg, men den andra punkten bestämmer vinkeln och avståndet till en vertex.

12.3.7 Panelen för avancerad digitalisering

När du registrerar, omformar eller delar upp nya eller befintliga geometrier har du också möjlighet att använda panelen Advanced Digitizing. Du kan digitalisera linjer som är exakt parallella eller vinkelräta mot en viss vinkel eller låsa linjer till specifika vinklar. Dessutom kan du göra en exakt definition av din nya geometri genom att ange X- och Y-koordinater samt Z för 3D-funktioner eller M-värden.

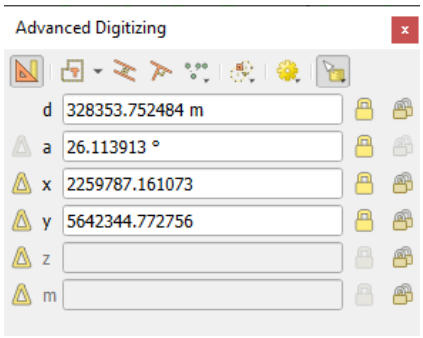


Fig. 12.91: Panelen för avancerad digitalisering

Panelen *Advanced Digitizing* kan öppnas antingen genom att högerklicka på verktygsfältet, från menyn *View* ► *Panels* ► eller genom att trycka på `Ctrl+4`. När panelen är synlig klickar du på knappen Enable advanced digitizing tools för att aktivera verktygsuppsättningen.

Observera

Verktygen är inte aktiverade om kartvyn är i geografiska koordinater.

Syftet med verktyget Advanced Digitizing är att låsa koordinater, längder och vinklar när du flyttar musen under digitaliseringen i kartbilden.

Du kan också skapa constraints med relativ eller absolut referens. Relativ referens innebär att värdena för nästa vertexbegränsning är relativa i förhållande till föregående vertex eller segment.

Verktygsrad

Längst upp på *Digitizing panel* finns följande knappar:

- Aktivera avancerade digitaliseringsverktyg
- Construction mode: gör det möjligt att fånga klickens positioner och återanvända dem som referenspunkter för att låsa relativa värden för avstånd, vinkel, X, Y, Z eller M. Mer information finns på *Konstruktionsläge*.
- Parallel för att dra en linje parallellt med en befintlig (mer på *Parallella och vinkelräta linjer*)
- Perpendicular för att rita en linje vinkelrätt mot en befintlig (mer på *Parallella och vinkelräta linjer*)
- Construction Tools innehåller ett par alternativ som begränsar topparnas placering baserat på extrapolerade koordinater för befintliga element:
  - *Line Extension*: håll muspekaren över ett segment så får du en lila prickad linje som förlänger segmentet över kartbilden. Du kan fästa vertexen var som helst på denna virtuella linje.
  - *X/Y Point*: håll muspekaren över en vertex så får du en lila prickad linje längs dess X- eller Y-koordinat, över kartbilden. Du kan fästa vertexen var som helst på denna virtuella linje. Det är till och med möjligt att hålla muspekaren över två olika toppar, vilket genererar virtuella koordinatlinjer för båda, och snäppa till deras skärningspunkt.
- 2-circle Point Intersection: låter dig digitalisera en punkt eller ett toppunkt i skärningspunkten mellan två cirklar. (mer på *2-cirkel punkt skärningspunkt*).
- Snap to common angles: när du flyttar markören visas en virtuell linje som du kan snappa till för att lägga till nästa vertex. Snäpplinjen definieras av det senast tillagda toppunktet och en vinkel (absolut eller relativ till föregående segment) från en förinställd lista (i steg om 0,1°, 0,5°, 1°, 5°, 10°, 15°, 18°, 22,5°, 30°, 45° eller 90°). Välj *Do not snap to common angles* för att inaktivera den här funktionen.

  Snapping to features kan användas tillsammans med snapping till vanliga vinklar för noggrann digitalisering. För att få en finkornig kontroll över hur målelementet som ska snäppas till behålls kan du ange om snäppning till funktioner ska prioriteras framför vanliga vinklar och vice versa under *Snäppningsprioritet*. Du kan växla från den ena metoden till den andra under digitaliseringsoperationen, och på så sätt undviker du att inaktivera något av snäppalternativen under tiden. Tryck på `N` (eller `Shift+N`) under en digitaliseringsoperation för att bläddra igenom vinkellistan.
- Floater settings: om alternativet *Show floater* är markerat följer en kontextuell meny med digitaliseringsinformation markören under digitaliseringen. Värdena kan nås med hjälp av *panelens genvägar*, redigeras och Låst efter validering (tryck på `Enter`). Vilken typ av information som ska visas kan väljas i den nedre delen av menyn:
  - *Visa avstånd*
  - *Visa vinkel*
  - *Visa XY-koordinater*
  - *Visa Z-värde*
  - *Visa M-värde*
  - *Visa bäring/azimut*
  - *Visa vanlig snäppvinkel*

Under verktygsfältet finns ett antal textrutor vars värde som standard återspeglar markörens position eller rörelse i kartbilden. Genom att redigera dessa värden kan du begränsa positionen för de objekt du redigerar:

- *d* för avståndet från en referensposition, vanligtvis det senast redigerade toppunktet

- *a* för vinkeln (absolut eller relativ) från en referensposition, vanligtvis det senast redigerade segmentet
- *x* för X-koordinaten för pekaren
- *y* för pekarens Y-koordinat
- *z* för standardvärdet för Z eller Z-koordinaten för toppunkten eller segmentet under pekaren
- *m* för standard-M-värdet eller M-värdet för toppunkten eller segmentet under pekaren

Tangentbordsgenvägar

För att påskynda användningen av Advanced Digitizing Panel finns det ett par kortkommandon tillgängliga:

Table 12.5: Kortkommandon för verktygen i den avancerade digitaliseringspanelen

Nyckel	Enkel	`Ctrl+` eller `Alt+`	`Shift+`
`D`	Ställ in avstånd	Avstånd till lås	
`A`	Ställ in vinkel	Låsvinkel	Växla relativ vinkel till sista segmentet
`X`	Ställ in X-koordinat	Lås X koordinat	Växla relativ X till sista toppunkten
`Y`	Ställ in Y-koordinat	Lås Y-koordinat	Växla relativ Y till sista toppunkten
`Z`	Ställ in Z-koordinaten	Lås Z-koordinat	Växla relativ Z till sista vertexen
`M`	Ställ in M-värde	Lås M-värde	Växla relativ M till sista toppunkten
`C`	Växla byggläge		
`P`	Växla mellan vinkelrätt och parallellt läge		

Observera

Alternativen Z-koordinat och M-värde är endast tillgängliga om de är kompatibla med lagrets geometridimension.

Digitalisering av absolut referens

När du ritar en ny geometri från grunden är det mycket användbart att ha möjlighet att börja digitalisera vertexer vid givna koordinater.

Om du t.ex. vill lägga till en ny funktion i ett polygonalt lager klickar du på knappen . Du kan ange de exakta koordinaterna där du vill börja redigera funktionen, t.ex:

1. Klicka på textrutan *x* (eller använd kortkommandot `X`).
2. Skriv in det X-koordinatvärde du vill ha och tryck på `Enter` eller klicka på knappen till höger för att låsa musen till X-axeln på kartbilden.
3. Klicka på textrutan *y* (eller använd kortkommandot `Y`).
4. Skriv in det Y-koordinatvärde du vill ha och tryck på `Enter` eller klicka på knappen till höger för att låsa musen till Y-axeln på kartbilden.
5. Om lagret har Z-koordinater eller M-värden aktiveras motsvarande widget *z* eller *m* och visar dess standardvärde, enligt inställningen i fliken *Inställningar* ► *Alternativ* ► *Digitalisering*.
    1. Klicka på textrutan *z* eller *m* (eller använd kortkommandot `Z` eller `M`).
    2. Skriv in det koordinatvärde du vill ha och tryck på `Enter` eller klicka på knappen till höger för att låsa värdet i widgeten.

> **Observera**
>
> Läs *Regler för tilldelning av Z-koordinat eller M-värde* för detaljer om hur Z-koordinater och M-värden automatiskt bestäms från befintliga funktioner.

6. Två blå streckade linjer och ett grönt kors identifierar de exakta koordinater som du angav. Klicka på kartbilden för att lägga till en toppunkt vid det gröna korsets position.

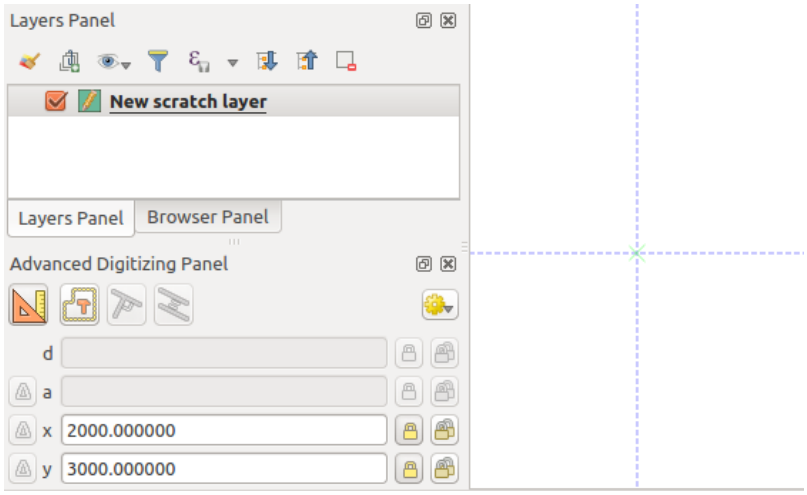


Fig. 12.92: Börja rita vid angivna koordinater

7. Du kan fortsätta som ovan och lägga till en ny uppsättning koordinater för nästa toppunkt, eller byta till en annan *digitaliseringsmetod* (t.ex. segment, kurva eller ström).
8. Om du vill rita ett segment med en viss längd:
    1. Klicka på textrutan *d (avstånd)* (kortkommando `D`)
    2. Skriv in avståndsvärdet (i kartenheter)
    3. Tryck på `Enter` eller klicka på knappen till höger för att låsa musen i kartbilden till segmentets längd. I kartbilden omges det senaste toppunktet av en cirkel vars radie är det värde som angetts i textrutan för avstånd. Ett kors i cirkeln visar positionen för nästa toppunkt om du klickar.

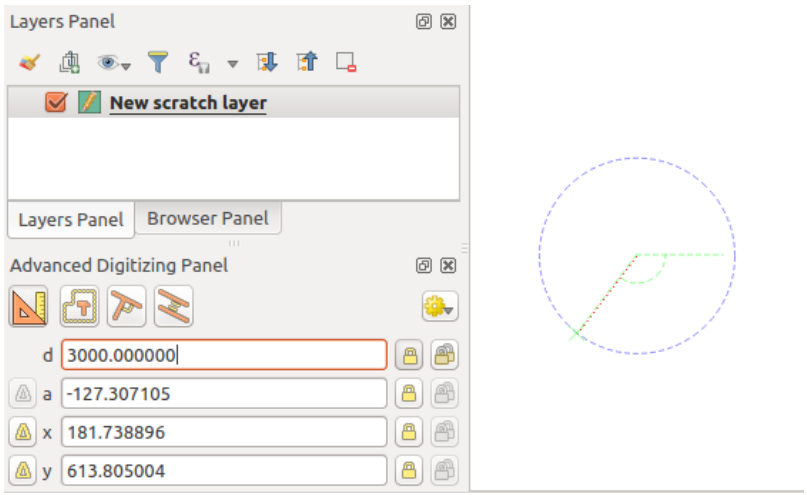


Fig. 12.93: Segment med fast längd

9. Du kan också begränsa vertexpositionen genom att ställa in segmentets vinkel. Som beskrivits tidigare:
    1. Klicka på textrutan *a (vinkel)* (kortkommando `A`)
    2. Skriv in vinkelvärdet (i grader)
    3. Tryck på `Enter` eller klicka på knappen till höger för att låsa den. En linje som går genom det senaste toppunktet och roteras enligt den inställda vinkeln visas i kartbilden och ett kors på den visar nästa toppunktsposition om du klickar.

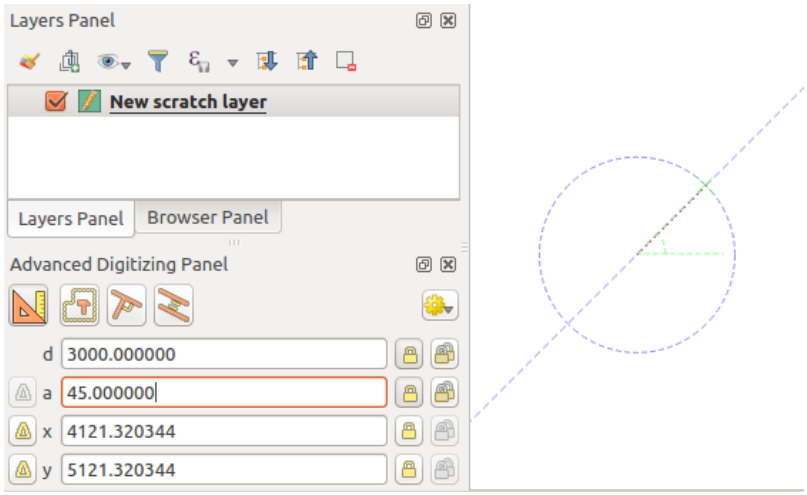


Fig. 12.94: Segment med fast vinkel

Råd

Genom att trycka på `Ctrl+<key>` eller `Alt+<key>` låses automatiskt målegenskapen och dess värde läggs in i edit. Ändra, tryck på `Enter` och du är klar. Kombinerat med Toggle floater, kan detta vara en riktig tidsbesparare, med tangentbordsdigitalisering.

Digitalisering av relativa referenser

I stället för att använda absoluta värden för vinklar eller koordinater kan du också använda värden som är relativa till den senast digitaliserade toppunkten eller segmentet.

För vinklar kan du klicka på -knappen till vänster om textrutan *a* (eller trycka på `Shift+A`) för att växla relativa vinklar till föregående segment. När detta alternativ är aktiverat mäts vinklarna mellan det sista segmentet och muspekaren.

För koordinater, klicka på -knapparna till vänster om textrutorna *x*, *y*, *z* eller *m* (eller tryck på `Shift+<key>`) för att växla relativa koordinater till föregående toppunkt. När dessa alternativ är aktiverade kommer koordinatmätningen att betrakta det sista toppunktet som ursprunget för de inställda koordinaterna.

Kontinuerligt lås

Både vid digitalisering med absolut eller relativ referens kan begränsningar för vinkel, avstånd, X, Y, Z och M låsas kontinuerligt genom att klicka på knapparna *Continuous lock*. Genom att använda kontinuerligt lås kan du digitalisera flera punkter eller vertexer med samma begränsningar.

Konstruktionsläge

Du kan aktivera och inaktivera *konstruktionsläget* genom att klicka på ikonen Construction mode eller med kortkommandot `C`. När du är i konstruktionsläge lägger du inte till nya vertexer genom att klicka på kartbilden, men klickens positioner registreras så att du kan använda dem som referenspunkter för att låsa avstånd, vinkel eller relativa värden för X, Y, Z, M.

I Construction mode hittar du en rullgardinsmeny där du kan välja att:

- *Record Construction Guides*: Alla byggsteg återges som streckade linjer. Varje steg som du utför representeras visuellt, så att du kan följa byggprocessen. Guiderna visas så länge som *Avancerad digitalisering* är aktiv.
- *Visa byggguider*: Gör det möjligt att synliggöra eller dölja konstruktionsguider på duken. När detta är aktiverat visas alla aktiva guider, vilket ger bättre spatial orientering och precision under digitaliseringsprocessen.
- *Snap to Visible Construction Guides*: Guiderna är snäppbara, vilket gör att du kan starta nya byggsteg från vilken punkt som helst längs de befintliga guiderna.
- Välj *Clear Construction Guides* för att ta bort alla guider från duken.

Konstruktionsläget kan t.ex. användas för att rita en punkt på exakt avstånd från en befintlig punkt.

Med en befintlig punkt i kartbilden och korrekt aktiverat snapping-läge kan du enkelt rita andra punkter på givna avstånd och vinklar från den. Förutom knappen måste du också aktivera *konstruktionsläget* genom att klicka på ikonen Konstruktionsläge eller med kortkommandot `C`.

Klicka bredvid den punkt från vilken du vill beräkna avståndet och klicka på rutan *d* (`D` genväg) skriv in önskat avstånd och tryck på `Enter` för att låsa musens position i kartbilden:

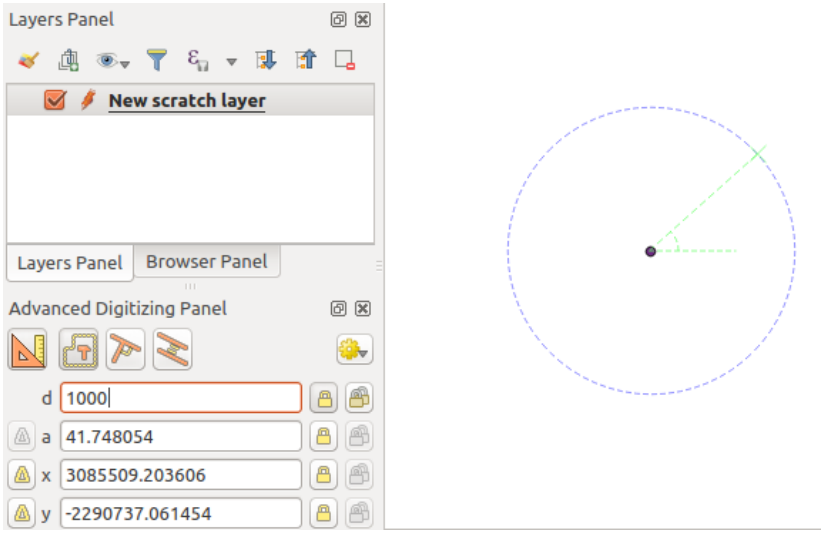


Fig. 12.95: Avstånd från punkt

Innan du lägger till den nya punkten trycker du på `C` för att avsluta konstruktionsläget. Nu kan du klicka på kartbilden och punkten kommer att placeras på det avstånd som angetts.

Du kan också använda vinkelbegränsningen för att t.ex. skapa en annan punkt på samma avstånd som den ursprungliga, men i en viss vinkel från den nytillkomna punkten. Klicka på ikonen Construction mode eller med kortkommandot `C` för att gå till konstruktionsläget. Klicka på den nyligen tillagda punkten och sedan på den andra för att ange ett riktningssegment. Klicka sedan på textrutan *d* (`D` genväg), skriv in önskat avstånd och tryck på `Enter`. Klicka på textrutan *a* (`A` genväg) skriv in den vinkel du vill ha och tryck på `Enter`. Musens position kommer att låsas både i avstånd och vinkel.

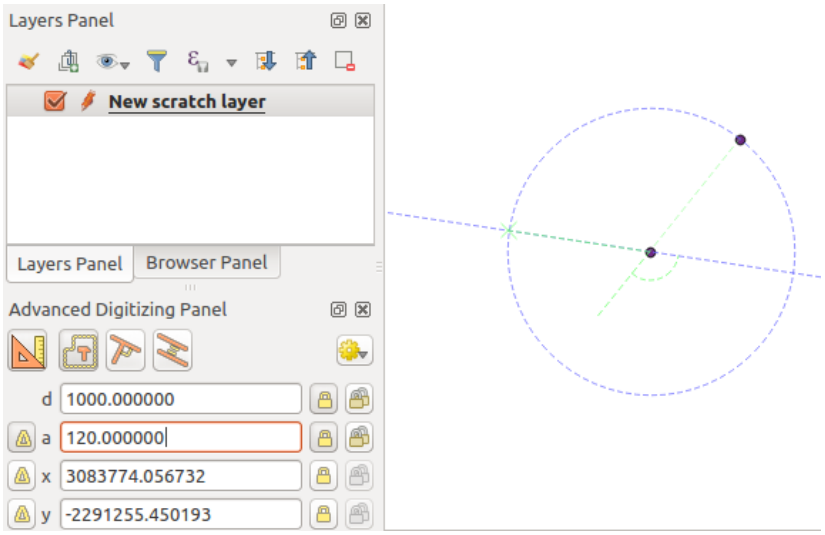


Fig. 12.96: Avstånd och vinkel från punkter

Innan du lägger till den nya punkten trycker du på `C` för att avsluta konstruktionsläget. Nu kan du klicka på kartbilden och punkten kommer att placeras på det avstånd och i den vinkel som angetts. Genom att upprepa processen kan flera punkter läggas till.

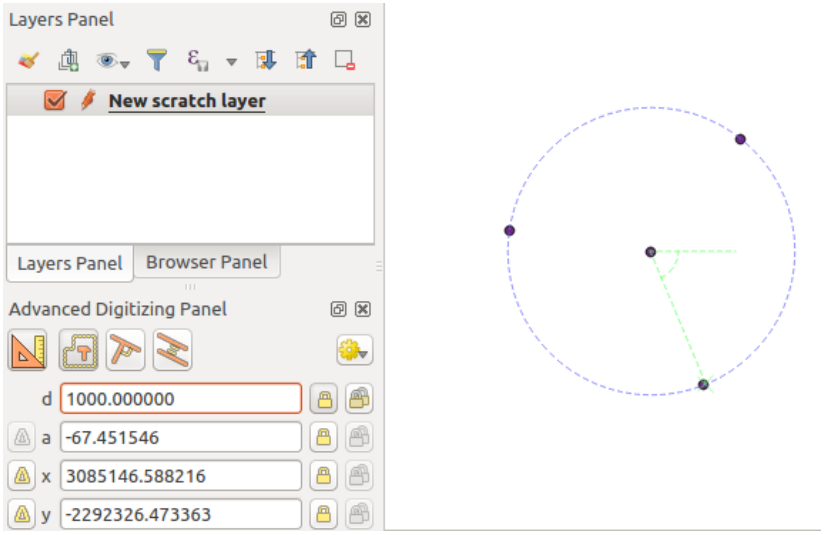


Fig. 12.97: Punkter på givet avstånd och vinkel

Parallella och vinkelräta linjer

Alla verktyg som beskrivs ovan kan kombineras med verktygen Perpendicular och Parallel. Dessa två verktyg gör det möjligt att rita segment som är helt vinkelräta eller parallella med ett annat segment. Målsegmentet kan vara i ett annat lager, en annan funktion i lagret eller den funktion som digitaliseras (kräver *self-snapping option*).

För att rita ett *vinkelrätt* segment:

1. Lägg först till en av segmentets hörn.
2. Klicka på ikonen Perpendicular (kortkommando `P`) för att aktivera den.
3. Klicka på det segment som du vill vara vinkelrät mot.
4. En virtuell streckad linje vinkelrät mot segmentet genom föregående vertex visas. Vinkelegenskapen är låst, vilket begränsar nästa vertex på den linjen och ett kryss anger markörens projicerade position på linjen. Klicka för att placera det nya toppunktet.

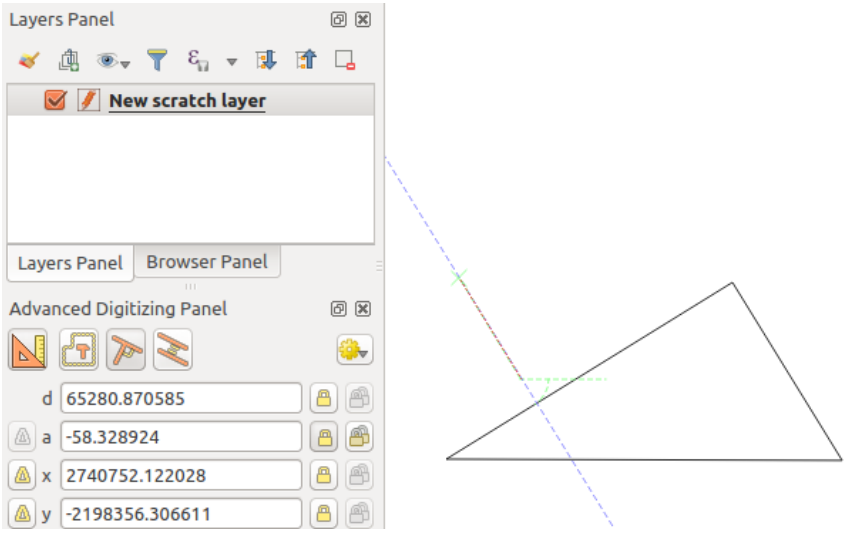


Fig. 12.98: Vinkelrät digitalisering

För att rita ett *parallellt* segment är stegen desamma förutom att du måste klicka på ikonen Parallel (kortkommando `P` två gånger).

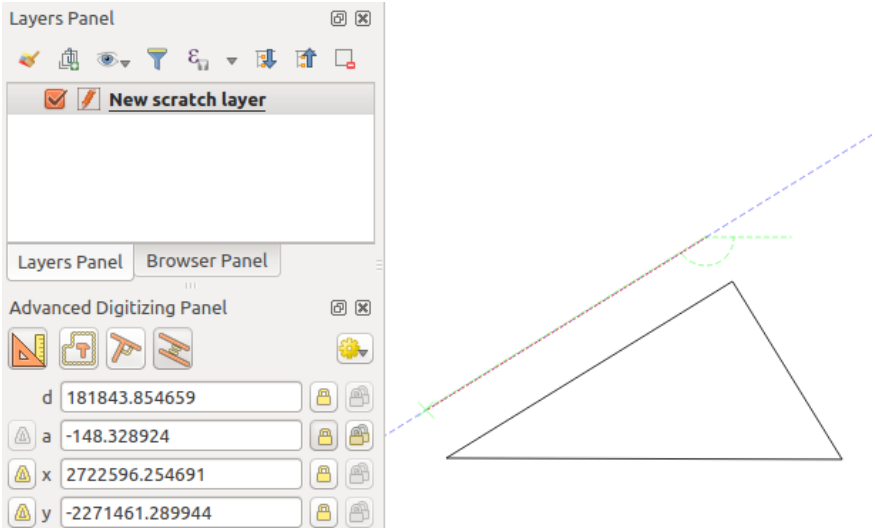


Fig. 12.99: Parallelldigitalisering

Dessa två verktyg hittar bara den rätta vinkeln för den vinkelräta och parallella vinkeln och låser denna parameter under redigeringen. Lås upp vinkelparametern för att avbryta användningen av dem mitt i processen.

2-cirkel punkt skärningspunkt

Följ dessa steg för att lägga till en toppunkt i skärningspunkten mellan två cirklar:

1. Klicka på ikonen 2-cirklars punktkorsning.
2. En dialogruta öppnas där du kan definiera parametrarna för *Circle #1* och *Circle #2*.
3. Klicka på kartbilden så beräknar verktyget automatiskt *X*- och *Y*-koordinaterna för de båda cirklarnas centrum.
4. Ange avståndet *d* för varje cirkel.
5. Verktyget beräknar och visar de två skärningspunkterna mellan cirklarna.
6. Klicka på en av skärningspunkterna för att lägga till den nya vertexen.

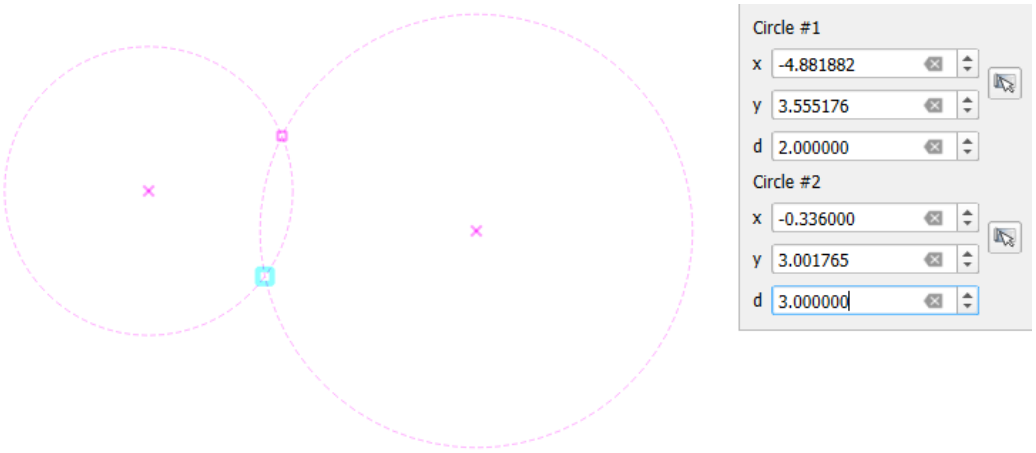


Fig. 12.100: 2-cirkel punkt korsning

12.3.8 Modifieraren för bearbetning av lager på plats

Menyn *Processing* ger tillgång till en stor uppsättning verktyg för att analysera och skapa nya funktioner baserat på egenskaperna hos de ingående funktionerna eller deras relationer med andra funktioner (inom samma lager eller inte). Det vanliga beteendet är att skapa nya lager som utdata, men vissa algoritmer tillåter också ändringar av inmatningslagret. Detta är ett praktiskt sätt att automatisera modifiering av flera funktioner med hjälp av avancerade och komplexa operationer.

För att redigera funktioner på plats:

1. Välj det lager som ska redigeras i panelen *Layers*.
2. Välj de berörda funktionerna. Du kan hoppa över det här steget, i så fall kommer ändringen att gälla hela lagret.
3. Tryck på knappen Edit Features In-Place högst upp i *Processing toolbox*. Listan över algoritmer filtreras och visar endast de som är kompatibla med modifieringar på plats, dvs:
   - De arbetar vid funktionskällan och inte på lagernivå.
   - De ändrar inte lagerstrukturen, t.ex. genom att lägga till eller ta bort fält.
   - De ändrar inte geometritypen, t.ex. från linje- till punktlager.

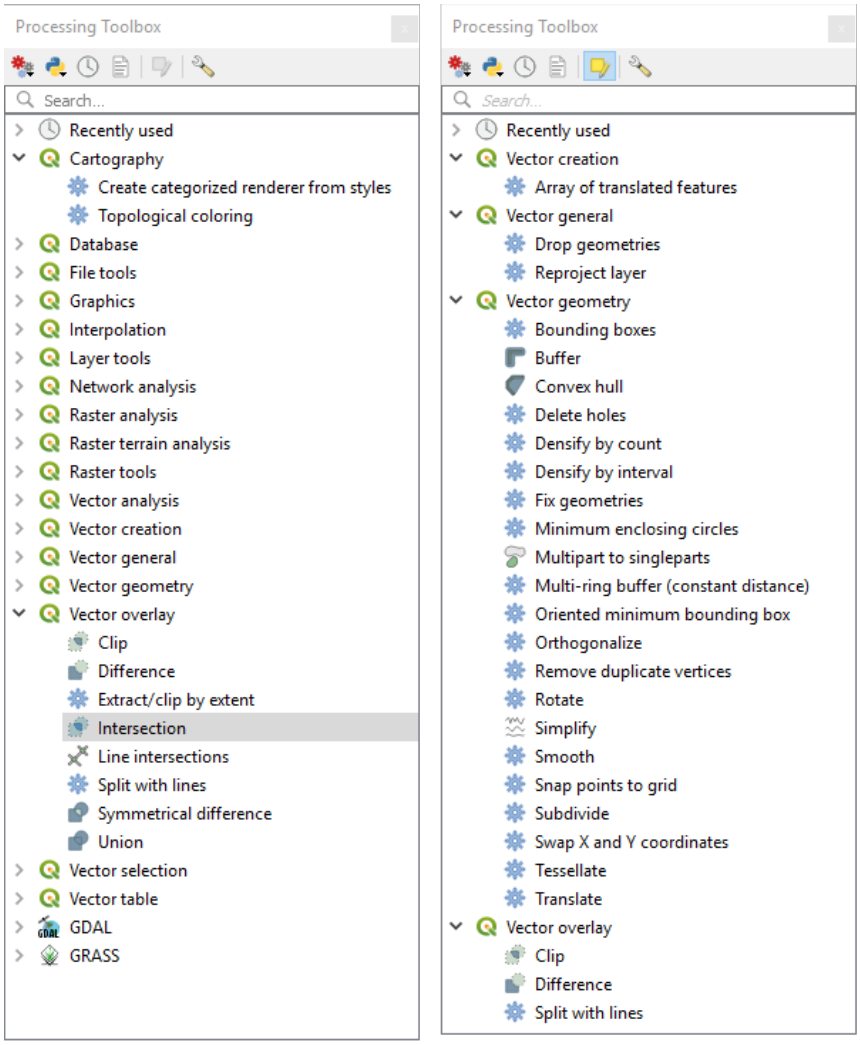


Fig. 12.101: Bearbetningsalgoritmer: alla (vänster) vs polygon in-place-editorer (höger)

4. Leta reda på den algoritm du vill köra och dubbelklicka på den.

> **Observera**
>
> Om algoritmen inte behöver några ytterligare parametrar som användaren ställer in (förutom de vanliga parametrarna för in- och utlagret) körs algoritmen omedelbart utan att någon dialogruta visas.

   1. Om det behövs andra parametrar än de vanliga inmatnings- eller utmatningslagren visas algoritmdialogen. Fyll i den information som krävs.
   2. Klicka på *Modify Selected Features* eller *Modify All Features* beroende på om det finns ett aktivt val.

   Ändringar tillämpas på lagret och placeras i redigeringsbufferten: lagret växlas till redigeringsläge med osparade ändringar, vilket indikeras av ikonen bredvid lagrets namn.

5. Som vanligt trycker du på Save layer edits för att göra ändringarna i lagret. Du kan också trycka på Undo för att återställa hela ändringen.

12.4 Koppla samman och redigera data i olika lager

En av uppgifterna i ett GIS-program är att kunna koppla samman data från olika lager. En sådan koppling kan baseras på det spatiala förhållandet mellan objekten eller på deras gemensamma attribut. QGIS tillhandahåller verktyg för att hantera alla dessa associationer, t.ex:

- Bearbetningsalgoritmer som kan skapa ett nytt lager som ett resultat av anslutningen, nämligen *Anslut attribut efter plats*, *Sammanfoga attribut efter närmaste*, *Sammanfoga attribut efter fältvärde*, …
- SQL-frågor för att skapa ett nytt lager från *DB Manager* eller som ett *virtuellt lager*
- *Joins-egenskaper* eller *relationsinställningar* som tillfälligt utökar attributen för objekt i ett visst lager med attributen för objekt i ett annat lager baserat på något eller några matchande attribut.

  Joins och relationer är tekniska begrepp som lånats från databaser för att få ut mesta möjliga av data som lagras i tabeller genom att kombinera deras innehåll. Tanken är att funktioner (rader) i olika lager (tabeller) kan associeras med varandra. Antalet rader som matchar varandra kan ha vilket värde som helst (noll, en, många).

12.4.1 Sammanfogning av funktioner mellan två lager

*med *Joins** i QGIS kan du associera funktioner i det aktuella lagret med funktioner från ett annat laddat vektorlager. Det spelar ingen roll om de är spatialt aktiverade eller vilken typ av geometri det rör sig om. Sammankopplingen baseras på ett attribut som delas av lagren, i ett ett-till-ett-förhållande.

För att skapa en join på ett lager (identifieras nedan som `mållager`):

1. Gå till lagret *Egenskaper* ► *Joins* fliken
2. Klicka på knappen Lägg till ny länk. Dialogrutan *Add vector join* visas.
3. Välj det *Join-lager* som du vill ansluta till målvektorlagret
4. Ange *Join field* (från `join layer`) och *Target field* (från `target layer`). Det är dessa fält som används för att hitta matchande objekt i båda lagren och de bör därför ha gemensamma värden.
5. Tryck på *OK* och en sammanfattning av de valda parametrarna läggs till i *Join*-panelen.

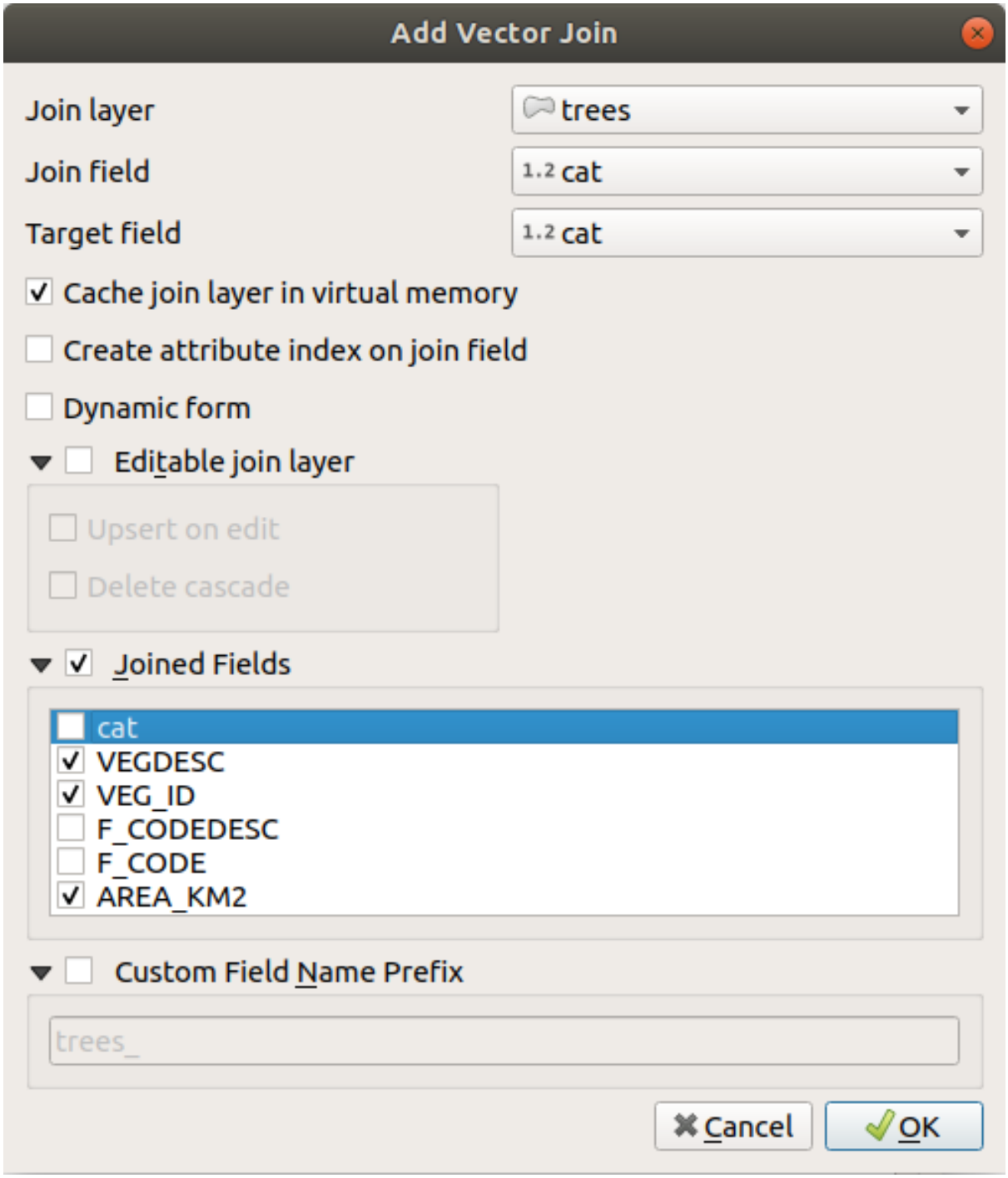


Fig. 12.102: Anslut en attributtabell till ett befintligt vektorlager

Stegen ovan skapar en join, där **ALLA** attributen för den **första matchande funktionen** i join-lagret läggs till i mållagrets funktion. Följande logik används för att para ihop funktioner under en join-process:

- Alla funktioner i mållagret returneras, oavsett om de har en matchning
- Om målfältet innehåller duplicerade värden tilldelas dessa funktioner samma funktion från det gemensamma lagret.
- Om join-fältet innehåller dubbla matchande värden väljs endast den först hämtade funktionen.

> **Observera**
>
> Sammanfogningar i QGIS baseras på matchning av ett enda fält, så för det mesta vill du se till att värdena i de matchande fälten är unika.

QGIS ger några fler alternativ för att justera sammanfogningen:

- *Cache join layer in virtual memory*: gör att du kan cacha värden i minnet (utan geometrier) från det sammanfogade lagret för att snabba upp sökningar.
- *Skapa attributindex på join-fältet* för att snabba upp sökningar
- *Dynamic form*: hjälper till att synkronisera join-fält i farten, enligt *Target field*. På så sätt uppdateras även begränsningarna för join-fält korrekt. Observera att den är avaktiverad som standard eftersom den kan vara mycket tidskrävande om du har många funktioner eller en mängd olika kopplingar.
- Om mållagret är redigerbart kommer vissa ikoner att visas i attributtabellen bredvid fälten för att informera om deras status:
  - : länklagret är inte konfigurerat för att kunna redigeras. Om du vill kunna redigera join-funktioner från målattributtabellen måste du markera alternativet *Editable join layer*.
  - : det gemensamma lagret är väl konfigurerat för att vara redigerbart, men dess aktuella status är skrivskyddad.
  - : det sammanfogade lagret är redigerbart, men synkroniseringsmekanismerna är inte aktiverade. Om du automatiskt vill lägga till en funktion i det anslutande lagret när en funktion skapas i mållagret måste du markera alternativet *Upsert on edit*. Symmetriskt kan alternativet *Delete cascade* aktiveras om du automatiskt vill ta bort join-funktioner.
- *Joined fields*: istället för att lägga till alla fält från det sammanfogade lagret kan du ange en delmängd.
- *Prefix för anpassat fältnamn* för sammanfogade fält, för att undvika namnkollision

12.4.2 Inställning av relationer mellan flera lager

Till skillnad från joins som definierar en en-till-en-länk mellan funktioner i två lager, hjälper relationer dig att skapa sammankopplingar mellan flera funktioner i två eller flera lager. Relationer är alltså inställningar på projektnivå och anges i fliken *Projekt* ► *Egenskaper* ► *Relationer*. Därifrån kan du:

- *Lägg till relation* vars typ kan vara:
  - *en till många-relation*
  - *många till många-relation*
  - *polymorfisk relation* som du kan lägga till eller redigera med de dedikerade verktygen i rullgardinsmenyn för åtgärder.

> **Observera**
>
> Det finns ännu inget enkelt sätt att redigera en icke-polymorf relation när den väl har skapats. Endast namnet kan redigeras med ett dubbelklick. För alla andra parametrar i en sådan relation måste du ta bort den och skapa den på nytt.

- *Upptäck relationer*: QGIS kan upptäcka befintliga relationer från databasformat som stöds (PostgreSQL, GeoPackage, ESRI File Geodatabase, …). Detta kan vara ett bra sätt att underlätta relationsdefinitionen.

- *Ta bort relation*

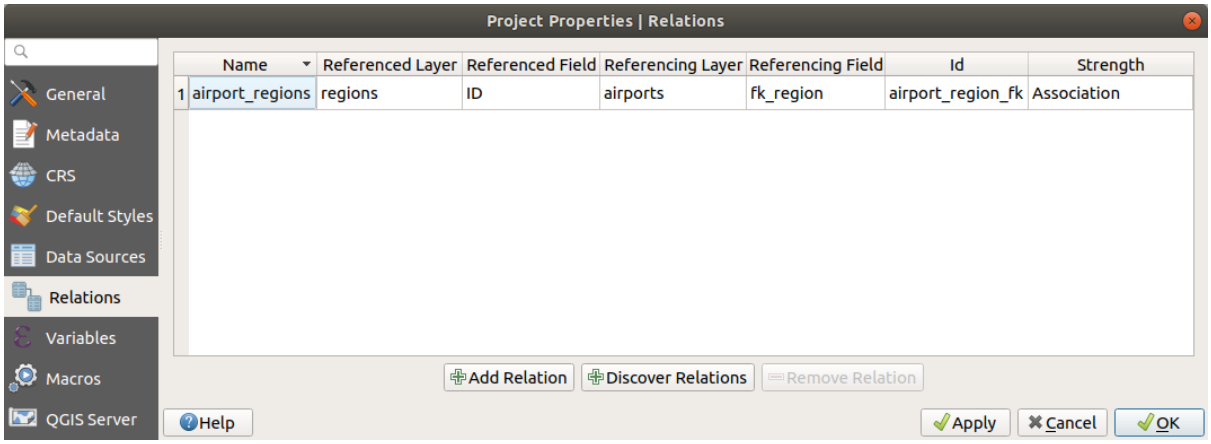


Fig. 12.103: Fliken Relationer

En till många (1-N) relationer

Som ett exempel har du ett lager med alla regioner i Alaska (polygon) som ger några attribut om dess namn och regiontyp och ett unikt id (som fungerar som primärnyckel).

Sedan får du ett annat punktlager eller en tabell med information om flygplatser som ligger i regionerna och du vill också hålla reda på dessa. Om du vill lägga till dem i regionlagret måste du skapa en en till många-relation med hjälp av främmande nycklar, eftersom det finns flera flygplatser i de flesta regioner.

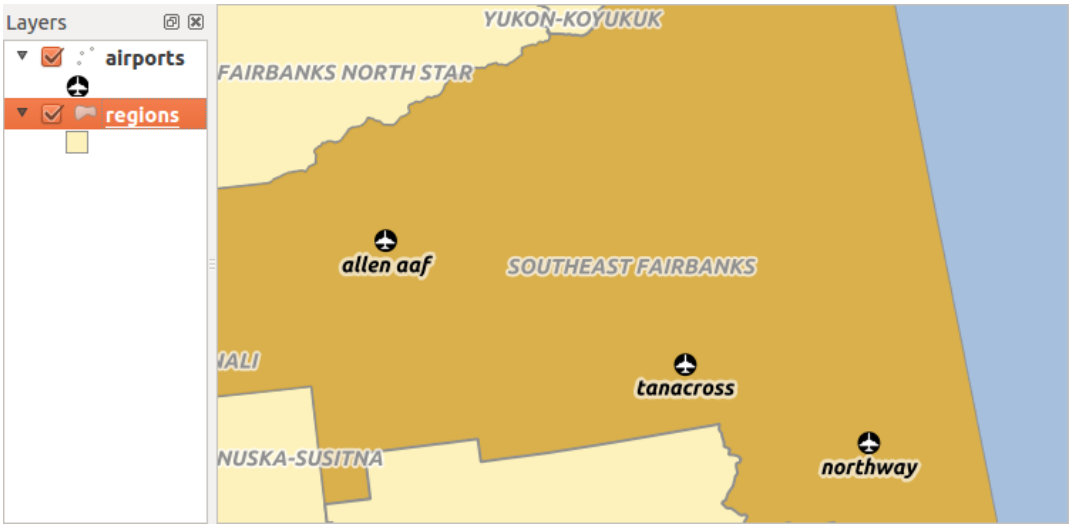


Fig. 12.104: Alaska regionen med flygplatser

Lager och nycklar

QGIS gör ingen skillnad mellan en tabell och ett vektorlager. I grund och botten är ett vektorlager en tabell med en geometri. Så du kan lägga till din tabell som ett vektorlager. För att demonstrera 1-n-relationen kan du ladda lagren `regions` och `airports` i exempeldatasetet. I praktiken hör varje flygplats till exakt en region medan varje region kan ha ett valfritt antal flygplatser (en typisk en till många-relation).

som har ett främmande nyckelfält (`fk_region`) till lagerregionerna.

Förutom de attribut som beskriver flygplatserna har aiports-lagret ytterligare ett fält, `fk_region`, som fungerar som en främmande nyckel (om du har en databas vill du förmodligen definiera en begränsning för det). Fältet `fk_region` kommer alltid att innehålla ett id för en region. Det kan ses som en pekare till den region som det tillhör.

Allt du behöver göra är att tala om för QGIS relationen mellan lagren så att du kan utforma ett anpassat redigeringsformulär för redigering och QGIS tar hand om installationen. Det fungerar med olika leverantörer (så du kan också använda det med shape- och csv-filer).

Definiera 1-N-relationer

Det första vi ska göra är att berätta för QGIS om relationerna mellan lagren. Detta görs i *Projekt* ► *Egenskaper…*.

Öppna fliken *Relationer* och klicka på *Lägg till relation*.

- **Namn** kommer att användas som en titel. Det ska vara en mänskligt läsbar sträng som beskriver vad relationen används till. I det här fallet säger vi bara **airport_relation**.
- **Referenced Layer (Parent)** betraktas också som överordnat lager, är det med primärnyckeln, pekade på, så här är det `regioner`-lagret. Du måste definiera den primära nyckeln för det refererade lagret, så det är `ID`.
- **Referencing Layer (Child)** betraktas också som barnlager, det är det som har det främmande nyckelfältet på sig. I vårt fall är detta `flygplatser`-lagret. För detta lager måste du lägga till ett referensfält som pekar på det andra lagret, så det här är `fk_region`.

> **Observera**
>
> Ibland behöver du mer än ett enda fält för att unikt identifiera funktioner i ett lager. För att skapa en relation med ett sådant lager krävs en **kompositnyckel**, dvs. mer än ett enda par matchande fält. Använd knappen Lägg till nytt fältpar som en del av en sammansatt främmande nyckel för att lägga till så många par som behövs.

- **Id** kommer att användas för interna ändamål och måste vara unikt. Du kan behöva det för att bygga *custom forms*. Om du lämnar det tomt kommer ett att genereras åt dig, men du kan tilldela ett själv för att få ett som är lättare att hantera
- **Relationship strength** anger styrkan i relationen mellan det överordnade och det underordnade lagret. Standardtypen *Association* innebär att det överordnade lagret är *enkelt* länkat till det underordnade medan typen *Composition* gör att du kan duplicera även de underordnade funktionerna när du duplicerar de överordnade och när du tar bort en funktion tas även de underordnade bort, vilket resulterar i en kaskad över alla nivåer (vilket innebär att barn till barn till… också tas bort).

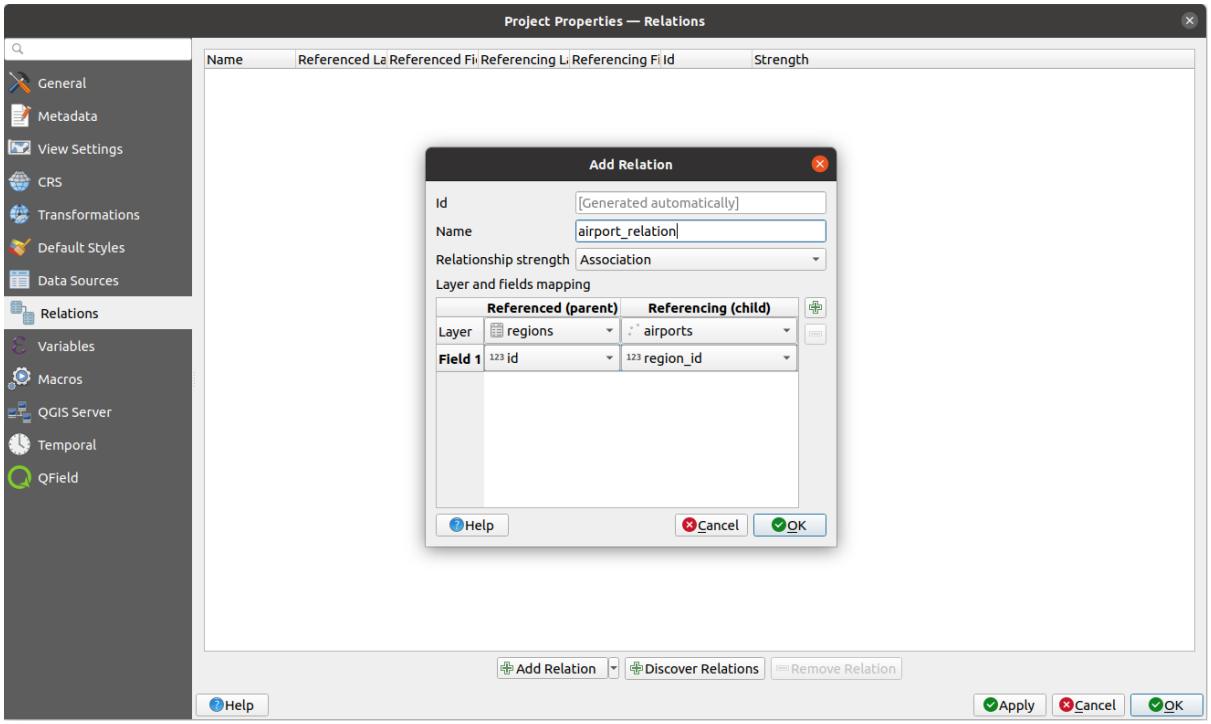


Fig. 12.105: Lägga till en relation mellan region- och flygplatslager

Från fliken *Relations* kan du också trycka på knappen *Discover Relation* för att hämta de relationer som är tillgängliga från leverantörerna av de laddade lagren. Detta är möjligt för lager som lagras i dataleverantörer som PostgreSQL eller SpatiaLite.

Formulär för 1-N-relationer

Nu när QGIS känner till relationen kommer den att användas för att förbättra de formulär som genereras. Eftersom vi inte har ändrat standardmetoden för formuläret (autogenererat) kommer det bara att lägga till en ny widget i vårt formulär. Så låt oss välja lagerregionen i förklaringen och använda identifieringsverktyget. Beroende på dina inställningar kan formuläret öppnas direkt eller så måste du välja att öppna det i identifieringsdialogen under åtgärder.

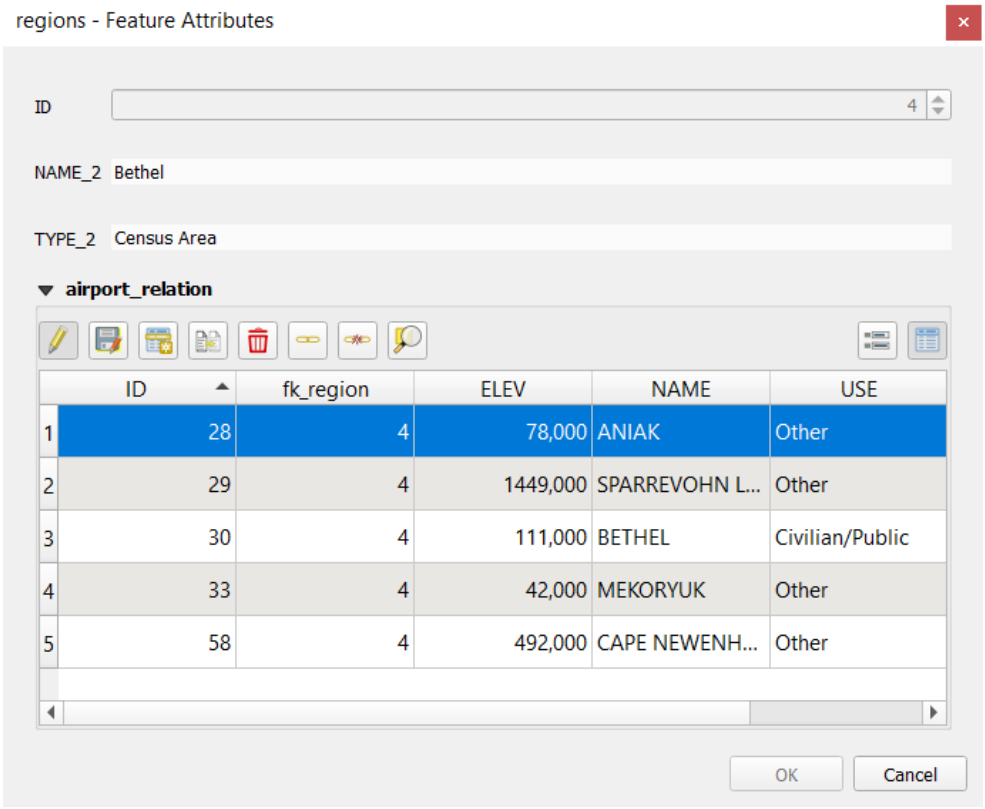


Fig. 12.106: Identifiering av dialogregioner i förhållande till flygplatser

Som du kan se visas alla flygplatser som tilldelats den här regionen i en tabell. Och det finns också några knappar tillgängliga. Låt oss granska dem kort:

- Knappen är till för att växla redigeringsläge. Tänk på att den växlar redigeringsläget för flygplatslagret, även om vi befinner oss i funktionsformatet för en funktion från regionlagret. Men tabellen representerar funktioner i flygplatslagret.
- Knappen används för att spara alla redigeringar i underordnat lager (flygplats).
- Med knappen kan du digitalisera flygplatsgeometrin i kartbilden och tilldela den nya funktionen till den aktuella regionen som standard. Observera att ikonen kommer att ändras beroende på geometritypen.
- Knappen lägger till en ny post i attributtabellen för flygplatslagret och tilldelar den nya funktionen till den aktuella regionen som standard. Geometrin kan ritas senare med digitaliseringsverktyget *Add part*.
- Med knappen kan du kopiera och klistra in en eller flera underordnade funktioner i det underordnade lagret. De kan senare tilldelas en annan överordnad funktion eller få sina attribut modifierade.
- Knappen raderar den eller de valda flygplatserna permanent.
- Symbolen öppnar en ny dialogruta där du kan välja en befintlig flygplats som sedan kommer att tilldelas den aktuella regionen. Detta kan vara praktiskt om du råkade skapa flygplatsen i fel region.
- Symbolen kopplar bort den eller de valda flygplatserna från den aktuella regionen och lämnar dem utan tilldelning (främmande nyckel sätts till NULL) i praktiken.
- Med knappen kan du zooma kartan till de valda barnfunktionerna.

- De två knapparna och till höger växlar mellan *tabellvy och formulärvy* för de relaterade underordnade funktionerna.

Om du använder *Drag and Drop Designer* för regionfunktionen kan du välja vilka verktyg som ska vara tillgängliga. Du kan även bestämma om ett nytt formulär ska öppnas när en ny funktion läggs till med hjälp av alternativet *Force hide form on add feature*. Tänk på att detta alternativ innebär att not null-attribut måste ha ett giltigt standardvärde för att fungera korrekt.

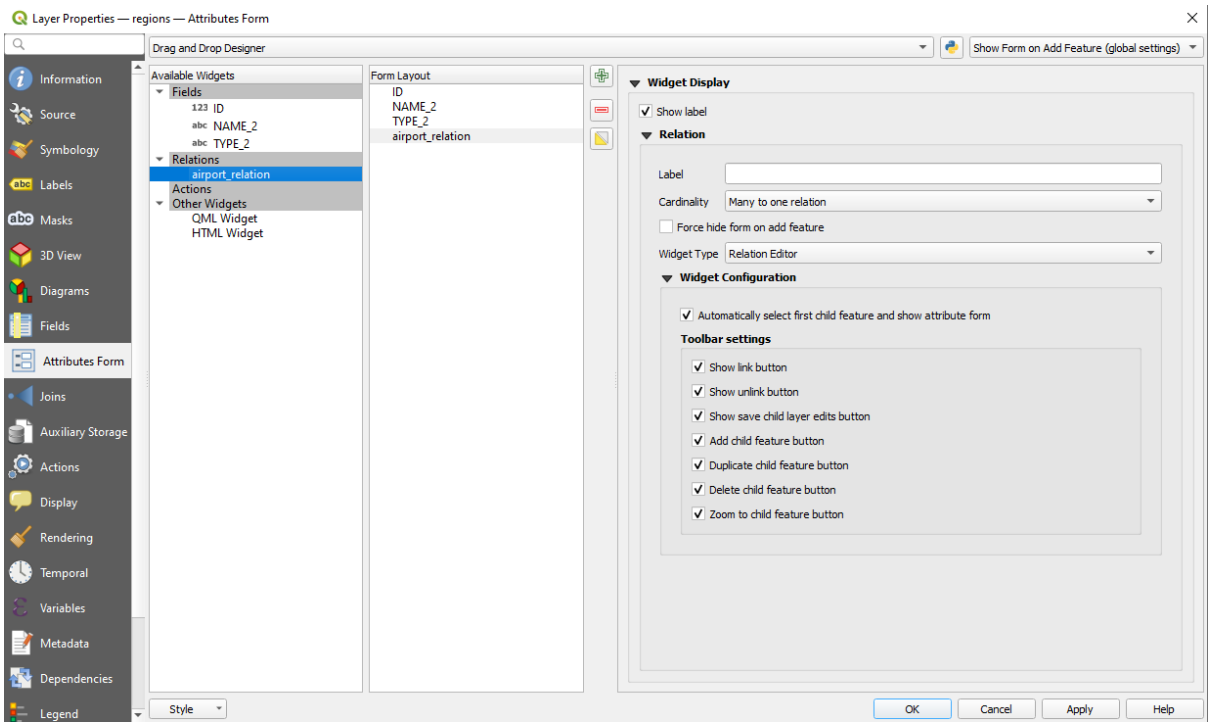


Fig. 12.107: Drag and Drop Designer för att konfigurera verktyg för relationer mellan regioner och flygplatser

I exemplet ovan har referenslagret geometrier (så det är inte bara en alfanumerisk tabell) så stegen ovan kommer att skapa en post i lagerattributtabellen som inte har någon motsvarande geometrisk funktion. För att lägga till geometrin:

1. Välj *Open Attribute Table* för referenslagret.
2. Välj den post som tidigare har lagts till i det refererade lagrets funktionsform.
3. Använd digitaliseringsverktyget Add Part för att koppla en geometri till den valda attributtabellposten.

Om du arbetar med flygplatstabellen konfigureras widgeten Relation Reference automatiskt för fältet `fk_region` (det som används för att skapa relationen), se *Relation Reference widget*.

I flygplatsformuläret ser du knappen till höger om fältet `fk_region`: om du klickar på knappen öppnas formuläret för regionlagret. Med den här widgeten kan du enkelt och snabbt öppna formulären för de länkade överordnade funktionerna.

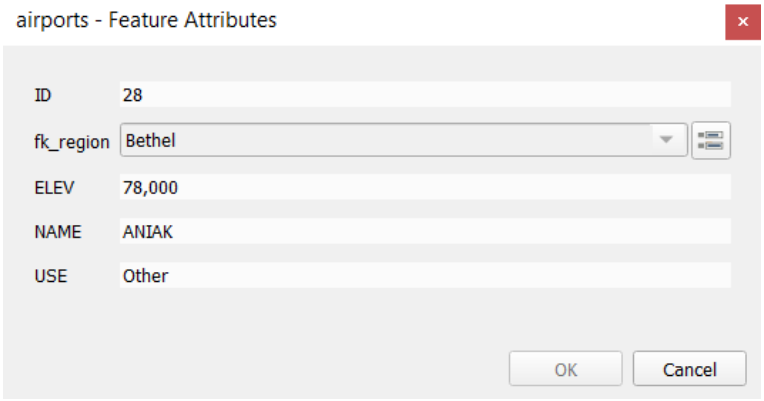
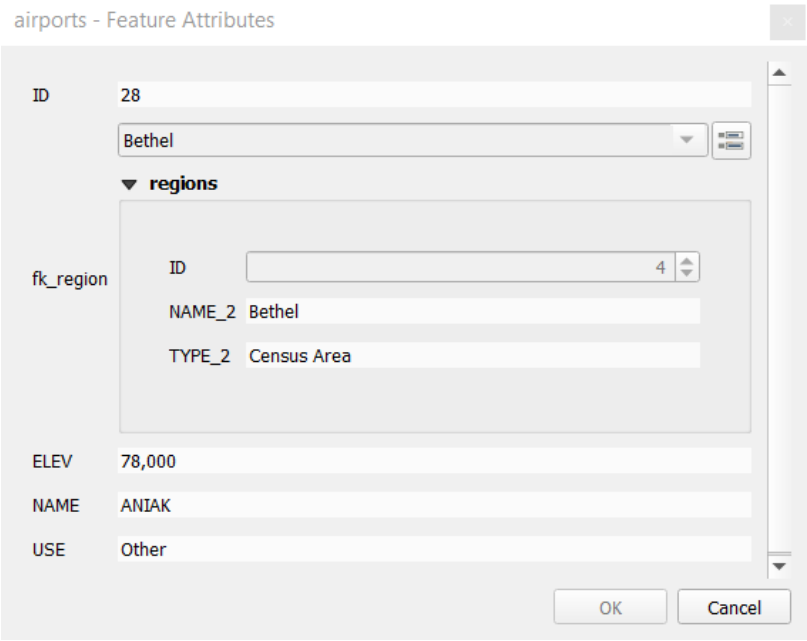


Fig. 12.108: Identifieringsdialog flygplats med relation till regioner

Widgeten Relationsreferens har också ett alternativ för att bädda in formuläret för det överordnade lagret i det underordnade. Det finns tillgängligt i menyn *Egenskaper* ► *Attribut Form* i flygplatslagret: välj fältet `fk_region` och markera alternativet `Visa inbäddat formulär`.

Om du tittar på funktionsdialogen nu kommer du att se att regionformuläret är inbäddat i flygplatsformuläret och till och med har en kombinationsruta som gör att du kan tilldela den aktuella flygplatsen till en annan region.



Om du växlar redigeringsläget för flygplatslagret har fältet `fk_region` dessutom en autokompletteringsfunktion: när du skriver kommer du att se alla värden i fältet `id` för regionlagret. Här är det möjligt att digitalisera en polygon för regionlagret med hjälp av knappen om du har valt alternativet `Allow adding new features` i menyn *Egenskaper* ► *Attributformulär* i flygplatslagret.

Barnskiktet kan också användas i verktyget *Välj funktioner efter värde* för att välja funktioner i det överordnade lagret baserat på attribut för deras barn.

I Fig. 12.109 väljs alla regioner där flygplatsernas medelhöjd är större än 500 meter över havet.

Du kommer att upptäcka att många olika aggregeringsfunktioner finns tillgängliga i formuläret.

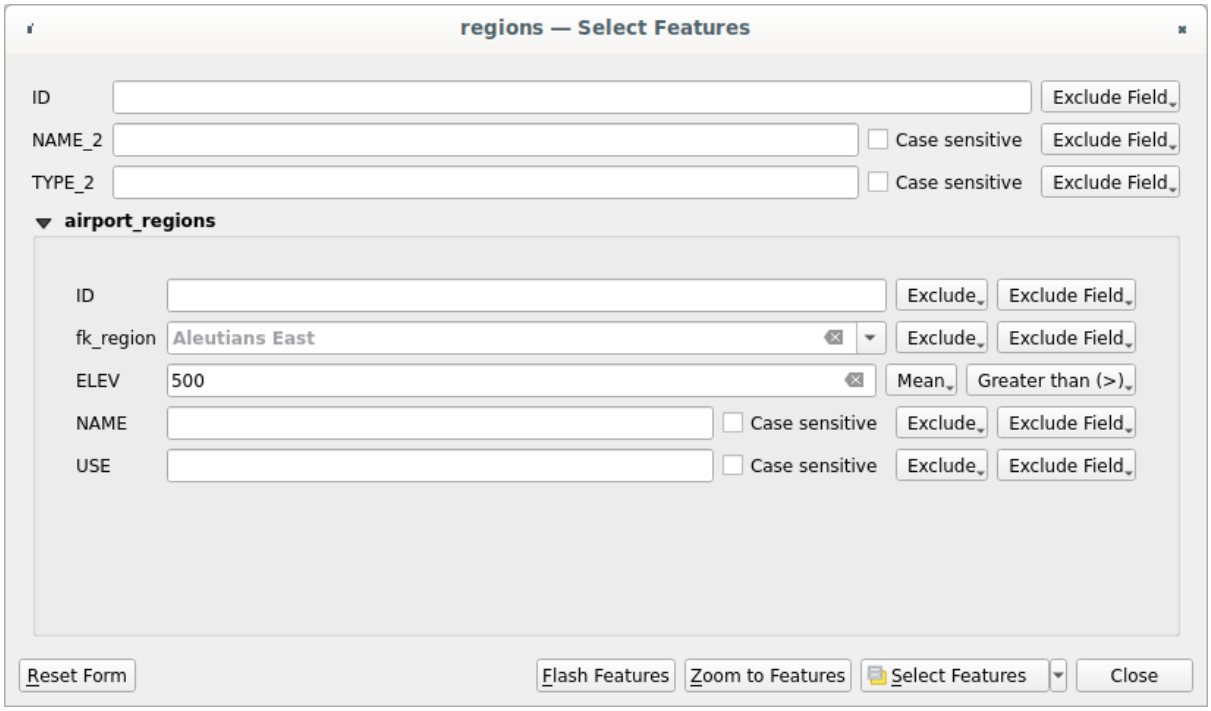


Fig. 12.109: Välj föräldrafunktioner med barnvärden

Många-till-många-relationer (N-M)

N-M-relationer är många-till-många-relationer mellan två tabeller. Till exempel lagren `flygplatser` och `flygbolag`: en flygplats tar emot flera flygbolag och ett flygbolag flyger till flera flygplatser.

Denna SQL-kod skapar de tre tabellerna vi behöver för en N-M-relation i ett PostgreSQL / PostGIS-schema med namnet *locations *. Du kan köra koden med hjälp av :menuselection:`Database –> DB Manager…` för PostGIS eller externa verktyg som `pgAdmin <https://www.pgadmin.org>`_. Flygplatstabellen lagrar ``airports``-lagret och flygbolagstabellen lagrar ``airlines``-lagret. I båda tabellerna används få fält för tydlighetens skull. Den * knepiga * delen är tabellen ``airports_airlines``. Vi behöver den för att lista alla flygbolag för alla flygplatser (eller vice versa). Den här typen av tabell kallas för en *pivottabell*.* Begränsningarna i den här tabellen gör att en flygplats kan associeras med ett flygbolag endast om båda redan finns i sina lager.

```
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)
);
```

(continues on next page)

(fortsättning från föregående sida)

```
CREATE INDEX airlines_geom_idx ON locations.airlines USING gist (geom);

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
 );
```

Istället för PostgreSQL kan du också använda GeoPackage. I det här fallet kan de tre tabellerna skapas manuellt med hjälp av *Databas* ► *DB Manager*…. I GeoPackage finns det inga scheman så prefixet *locations* behövs inte.

Foreign key constraints i tabellen `airports_airlines` kan inte skapas med hjälp av *Table* ► *Create Table*… eller *Table* ► *Edit Table*… så de bör skapas med hjälp av *Database* ► *SQL Window*…. GeoPackage stöder inte *ADD CONSTRAINT*-satser så tabellen `airports_airlines` bör skapas i två steg:

1. Ställ in tabellen endast med fältet `id` med hjälp av *Tabell* ► *Skapa tabell*…
2. Använd *Database* ► *SQL Window*…, skriv och kör denna SQL-kod:

```
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;
```

I QGIS ska du sedan ställa in två *one-to-many-relationer* enligt förklaringen ovan:

- en relation mellan `airlines` tabellen och pivottabellen;
- och en andra mellan `airports` tabellen och pivottabellen.

Ett enklare sätt att göra det (endast för PostgreSQL) är att använda *Discover Relations* i *Projekt* ► *Egenskaper* ► *Relations*. QGIS kommer automatiskt att läsa alla relationer i din databas och du behöver bara välja de två du behöver. Kom ihåg att först ladda in de tre tabellerna i QGIS-projektet.

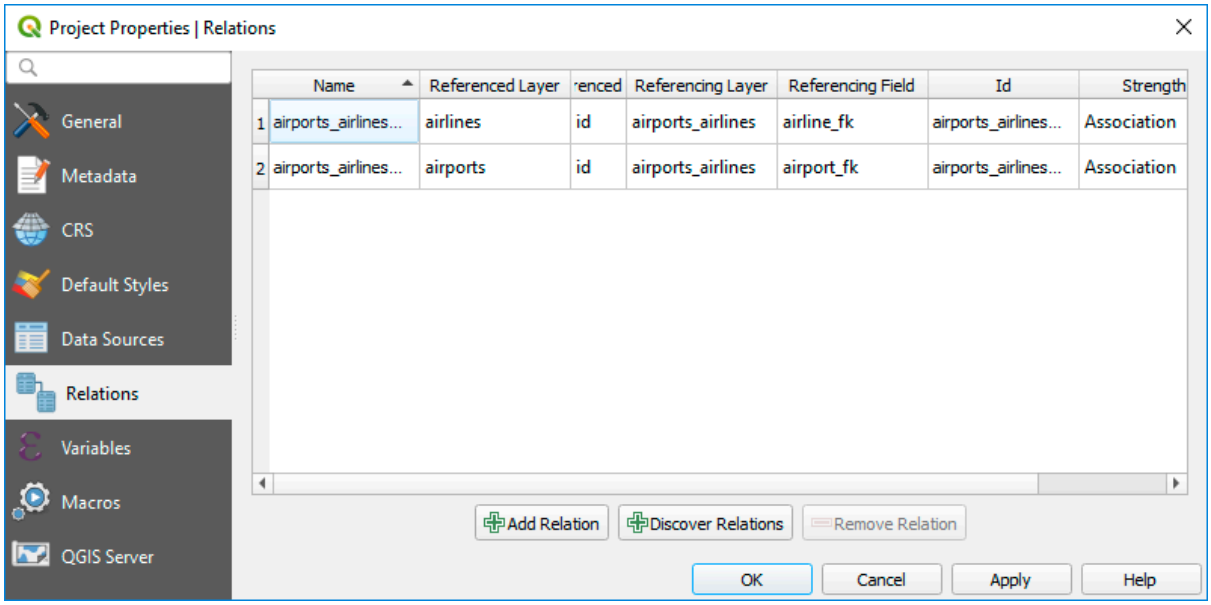


Fig. 12.110: Relationer och autodiscover

Om du vill ta bort en ”flygplats” eller ett ”flygbolag” kommer QGIS inte att ta bort de tillhörande posterna i tabellen ”airports_airlines”. Denna uppgift kommer att utföras av databasen om vi anger rätt *begränsningar* i skapandet av pivottabellen som i det aktuella exemplet.

Observera

Kombinera N-M-relation med automatisk transaktionsgrupp

Du bör aktivera transaktionsläget i *Projektegenskaper* ► *Datakällor* ► när du arbetar med en sådan kontext. QGIS bör kunna lägga till eller uppdatera rad(ar) i alla tabeller (flygbolag, flygplatser och pivottabeller).

Slutligen måste vi välja rätt kardinalitet i *Lageregenskaper* ► *Attributes Form* för lagren `airports` och `airlines`. För det första bör vi välja alternativet **airlines (id)** och för det andra alternativet **airports (id)**.

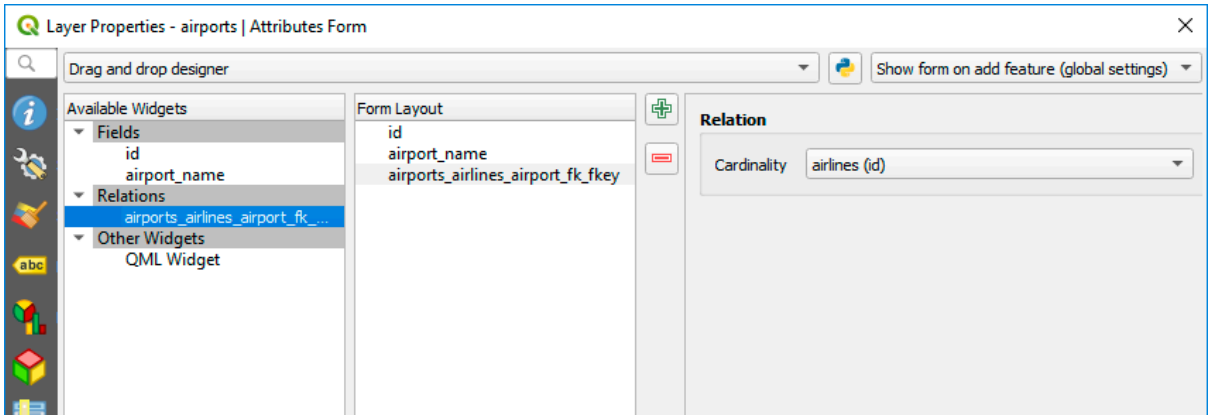


Fig. 12.111: Ange kardinalitet för relation

Nu kan du koppla ihop en flygplats med ett flygbolag (eller ett flygbolag med en flygplats) med hjälp av *Add child feature* eller *Link existing child feature* i underformulären. En post kommer automatiskt att infogas i tabellen `airports_airlines`.

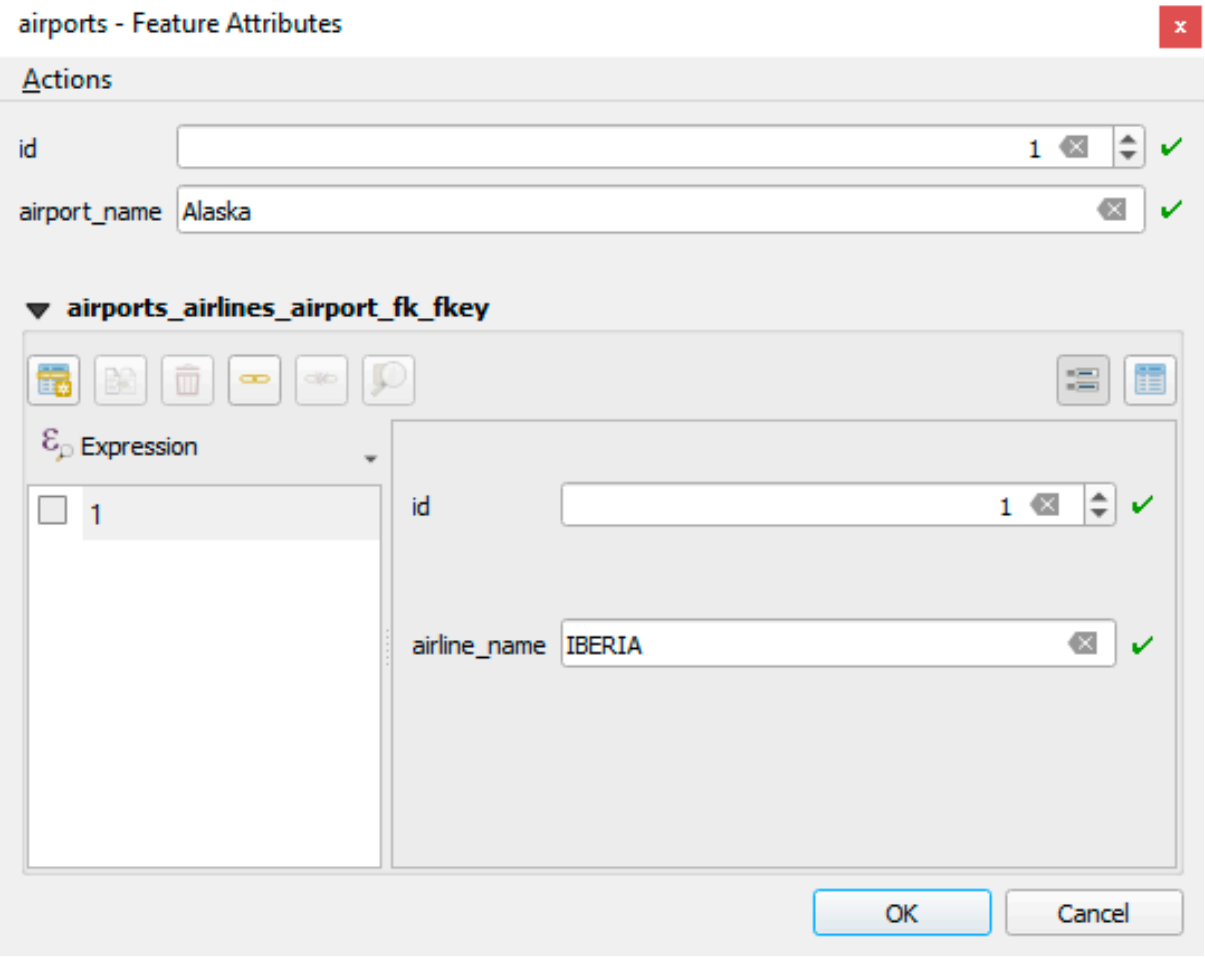


Fig. 12.112: N-M-förhållande mellan flygplatser och flygbolag

Observera

Använda **Många till en relation** kardinalitet

Ibland är det inte önskvärt att dölja pivottabellen i en N-M-relation. Främst för att det finns attribut i relationen som bara kan ha värden när en relation är etablerad. Om dina tabeller har ett geometrifält kan det vara intressant att aktivera alternativet *On map identification* (*Lageregenskaper* ► *Attributes Form* ► *Available widgets* ► *Fields*) för fälten med främmande nycklar i pivottabellen.

Observera

Pivottabellens primärnyckel

Undvik att använda flera fält i den primära nyckeln i en pivottabell. QGIS förutsätter en enda primärnyckel, så en begränsning som `constraint airports_airlines_pkey primary key (airport_fk, airline_fk)` fungerar inte.

Polymorfa relationer

Syftet med

Polymorfa relationer är ett specialfall av 1-N-relationer, där ett enda refererande (dokument)lager innehåller egenskaper för flera refererade lager. Detta skiljer sig från normala relationer som kräver olika referenslager för varje refererat lager. Ett enda referenslager (dokumentlager) uppnås genom att lägga till en extra kolumn ”layer_field” i referenslagret (dokumentlagret) som lagrar information för att identifiera det refererade lagret. I sin enklaste form kommer det refererande lagret (dokumentlagret) bara att infoga lagernamnet för det refererade lagret i detta fält.

För att vara mer exakt är en polymorf relation en uppsättning normala relationer som har samma referenslager men där det refererade lagret definieras dynamiskt. Den polymorfa inställningen av lagret löses genom att använda ett uttryck som måste matcha vissa egenskaper hos det refererade lagret, t.ex. tabellnamn, lager-ID, lagernamn.

Tänk dig att vi går till parken och vill ta bilder av olika arter av ”växter” och ”djur” som vi ser där. Varje växt eller djur har flera bilder associerade med sig, så om vi använder de normala 1:N-relationerna för att lagra bilder skulle vi behöva två separata tabeller, `animal_images` och `plant_images`. Detta kanske inte är ett problem för 2 tabeller, men tänk om vi vill ta separata bilder för svampar, fåglar etc.

Polymorfiska relationer löser detta problem eftersom alla refererande funktioner lagras i samma tabell `documents`. För varje funktion lagras det refererade lagret i fältet `referenced_layer` och den refererade funktionens id i fältet `referenced_id`.

Definiera polymorfa relationer

Låt först QGIS veta om de polymorfa relationerna mellan lagren. Detta görs i *Projekt* ► *Egenskaper*…. Öppna fliken *Relationer* och klicka på den lilla nedåtpilen bredvid knappen *Lägg till relation*, så att du kan välja alternativet *Lägg till polymorfisk relation* i den nya rullgardinsmenyn.

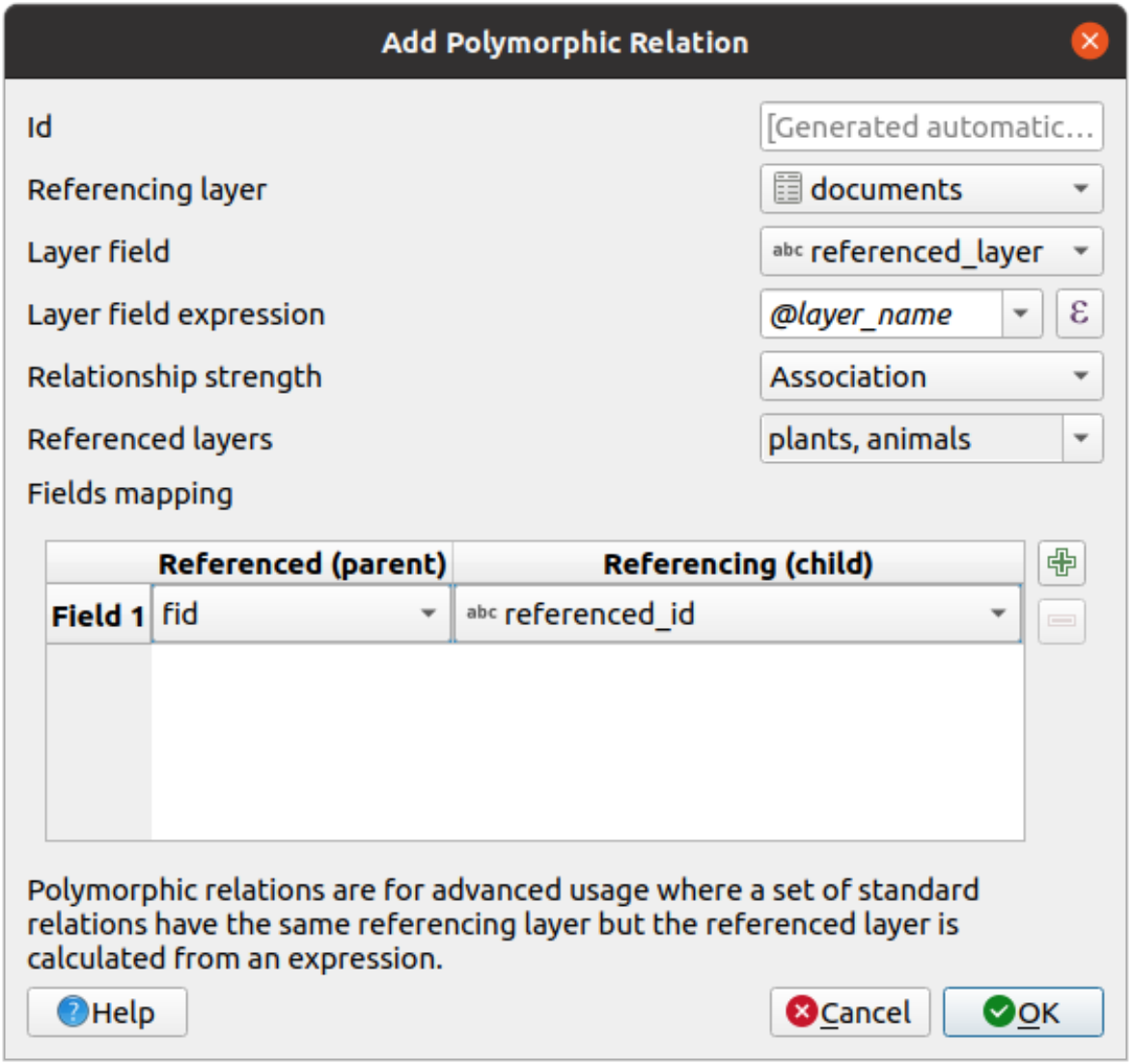


Fig. 12.113: Lägga till en polymorf relation med lagret `dokument` som refererande och lagren `djur` och `växter` som refererande.

- **Id** kommer att användas för interna ändamål och måste vara unikt. Du kan behöva det för att bygga *custom forms*. Om du lämnar det tomt kommer ett att genereras åt dig, men du kan tilldela ett själv för att få ett som är lättare att hantera
- **Referencing Layer (Child)** betraktas också som barnlager och är det lager som innehåller fältet med främmande nyckel. I vårt fall är detta `documents`-lagret. För detta lager måste du lägga till ett referensfält som pekar på det andra lagret, så detta är `referenced_id`.

  > **Observera**
  >
  > Ibland behöver du mer än ett enda fält för att unikt identifiera funktioner i ett lager. För att skapa en relation med ett sådant lager krävs en **kompositnyckel**, dvs. mer än ett enda par matchande fält. Använd knappen Lägg till nytt fältpar som en del av en sammansatt främmande nyckel för att lägga till så många par som behövs.

- **Lagerfält** är det fält i referenstabellen som lagrar resultatet av det utvärderade lageruttrycket, vilket är den referenstabell som den här funktionen tillhör. I vårt exempel skulle detta vara fältet `referenced_layer`.
- **Layer expression** utvärderas till en unik identifierare för lagret. Detta kan vara lagernamnet `@layer_name`,

lagrets id `@layer_id`, lagrets tabellnamn `decode_uri(@layer, 'table')` eller något annat som kan identifiera ett lager på ett unikt sätt.

- **Relationship strength** anger styrkan i de genererade relationerna mellan det överordnade och det underordnade lagret. Standardtypen *Association* innebär att det överordnade lagret är *enkelt* länkat till det underordnade medan typen *Composition* gör att du kan duplicera även de underordnade funktionerna när du duplicerar de överordnade och när du tar bort en funktion tas även de underordnade bort, vilket resulterar i en kaskad över alla nivåer (vilket innebär att barn till barn till… också tas bort).
- ** Refererade lager ** betraktas också som föräldralager, är de med den primära nyckeln, pekade på, så här skulle de vara `` växter `` och `` djur `` lager. Du måste definiera den primära nyckeln för de refererade lagren från rullgardinsmenyn, så det är `fid`. Observera att definitionen av en giltig primärnyckel kräver att alla refererade lager har ett fält med det namnet. Om det inte finns något sådant fält kan du inte spara en polymorf relation.

När den polymorfa relationen har lagts till kan den redigeras via menyalternativet *Edit Polymorphic Relation*.

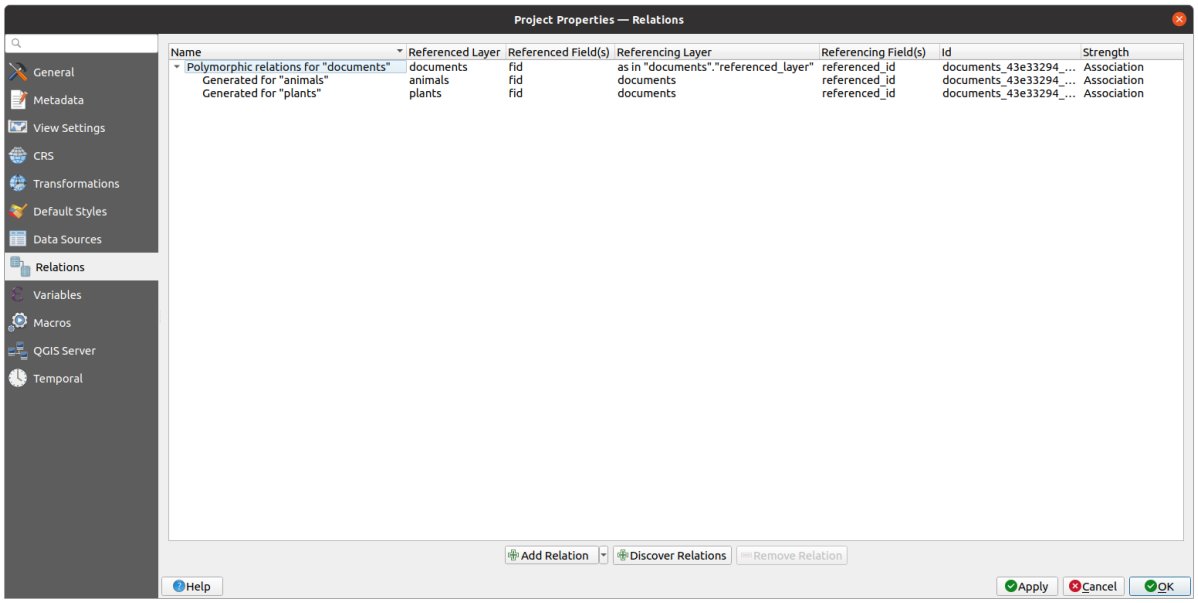


Fig. 12.114: Förhandsgranskning av den nyskapade polymorfa relationen och dess underordnade relationer för djur och växter.

I exemplet ovan används följande databasschema:

```
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)

(fortsättning från föregående sida)

```
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)
);
```

KAPITEL 13

Arbeta med rasterdata

13.1 Dialog för egenskaper för raster

Rasterdata består av pixlar (eller celler) och varje pixel har ett värde. Det används ofta för att lagra olika typer av data, t.ex:

- Bildmaterial, t.ex. satellitbilder, digitala flygfoton, skannade kartor
- Höjddata, t.ex. digitala höjdmodeller (DEM), digitala terrängmodeller (DTM)
- Andra typer av data, t.ex. marktäcke, jordtyper, nederbörd och mycket annat.

Rasterdata kan lagras i flera format som stöds, inklusive GeoTIFF, ERDAS Imagine, ArcInfo ASCII GRID, PostGIS Raster och andra. Se mer på *Öppningsdata*.

För att visa och ställa in egenskaperna för ett rasterlager, dubbelklicka på lagrets namn i kartans teckenförklaring eller högerklicka på lagrets namn och välj *Egenskaper* från snabbmenyn. Då öppnas dialogrutan *Rasterlageregenskaper*.

Det finns flera flikar i dialogrutan:

Information	*Källa*	*Symbologi*[1]
Transparency[1]	*Histogram*[1]	*Rendering*
Temporal	*Pyramider*	*Elevation*
Metadata	*Förklaring*	*Display*
QGIS Server	*Externa tillägg*[2] flikar	

[1] Finns även i *Layer stylingpanel*

[2] *Externa tillägg* som du installerar kan lägga till flikar i den här dialogen. Dessa presenteras inte i detta dokument. Hänvisa till deras dokumentation.

Tips

Live uppdatering rendering

Panel för lagerstyling ger dig några av de vanliga funktionerna i dialogen Lageregenskaper och är en bra modellfri widget som du kan använda för att påskynda konfigurationen av lagerstilarna och visa dina ändringar på kartbilden.

Observera

Eftersom egenskaper (symbologi, etikett, åtgärder, standardvärden, formulär …) för inbäddade lager (se *Inbäddning av lager från externa projekt*) hämtas från den ursprungliga projektfilen, och för att undvika ändringar som kan bryta detta beteende, görs dialogen för lageregenskaper otillgänglig för dessa lager.

13.1.1 Informationsegenskaper

Fliken *Information* är skrivskyddad och utgör en intressant plats för att snabbt hämta sammanfattad information och metadata för det aktuella lagret. Tillhandahållen information är:

- allmänna uppgifter som namn i projektet, källsökväg, lista över hjälpfiler, senaste spartid och storlek, den använda leverantören
- anpassade egenskaper, som används för att lagra ytterligare information om lagret i det aktiva projektet. Standardinställda egenskaper inkluderar *Identify/format*, som påverkar hur resultaten från att använda verktyget *Identify features* över ett rasterlager formateras. Fler egenskaper kan skapas och hanteras med hjälp av PyQGIS, särskilt genom metoden setCustomProperty().
- baserat på lagrets leverantör: omfattning, bredd och höjd, datatyp, GDAL-drivrutin, bandstatistik
- det koordinatsatta referenssystemet: namn, enheter, metod, noggrannhet, referens (dvs. om det är statiskt eller dynamiskt)
- läsa från lagrets egenskaper: datatyp, omfattning, bredd/höjd, komprimering, pixelstorlek, statistik över band, antal kolumner, rader och värden utan data för rastret…
- hämtad från *fyllda metadata*: tillgång, utbredning, länkar, kontakter, historik…

13.1.2 Källegenskaper

Fliken *Källa* visar grundläggande information om det valda rastret, inklusive:

- det *Layernamn* som ska visas i *Layers Panel*;
- *Koordinatreferenssystem*: Visar lagrets *Koordinatreferenssystem (CRS)*. Du kan ändra lagrets CRS genom att välja ett nyligen använt i listrutan eller klicka på knappen Välj CRS (se *Väljare för referenssystem för koordinater*). Använd denna process endast om lagrets CRS är fel eller inte har angetts. Om du vill återprojicera dina data använder du en återprojekteringsalgoritm från Processing eller *Spara det som ny dataset*.

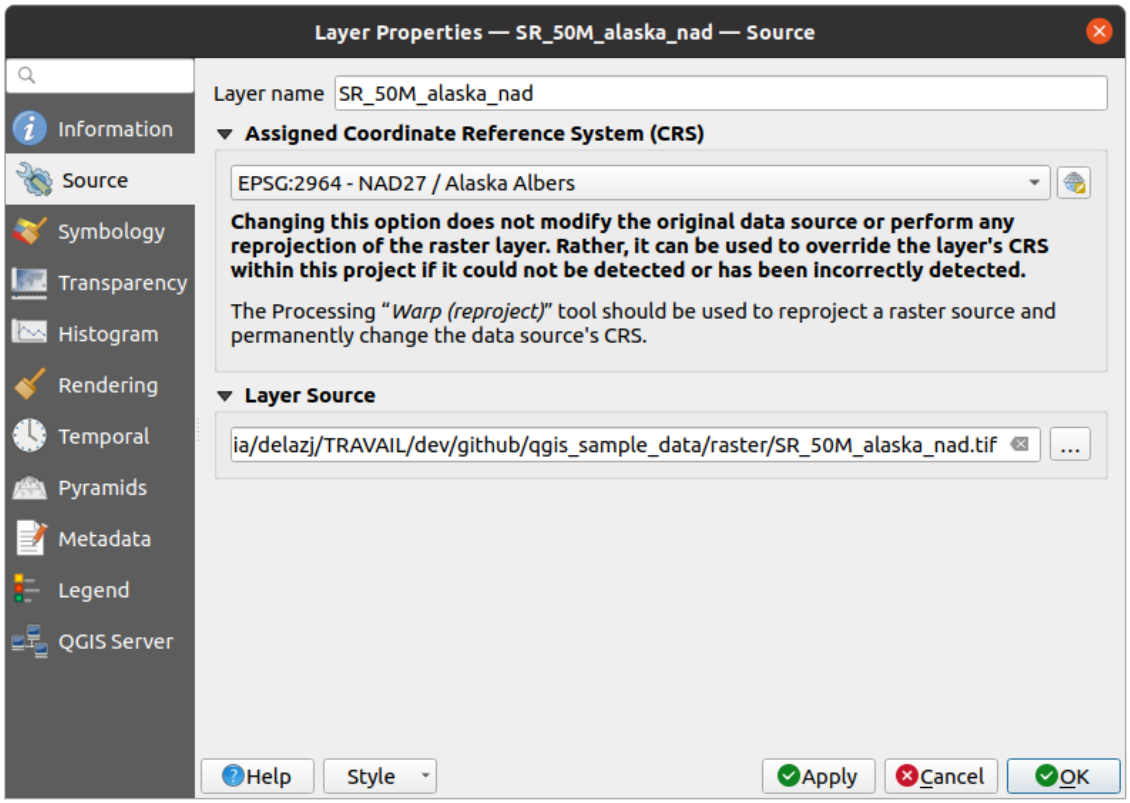


Fig. 13.1: Egenskaper för rasterlager - dialogrutan Källa

13.1.3 Egenskaper för symbologi

Fliken för rasterlagersymbologi består av tre olika delar:

- *Band rendering* där du kan styra vilken typ av rendering som ska användas
- *Layer rendering* för att tillämpa effekter på renderade data
- Metoderna *Resampling* för att optimera rendering på karta

Rendering av band

QGIS erbjuder många olika *Renderertyper*. Valet av renderingstyp beror på datatypen och den information som du vill lyfta fram.

1. *Multiband color* - om filen levereras med flera band (t.ex. en satellitbild med flera band).
2. *Paletted/Unique values* - för single band-filer som levereras med en indexerad palett (t.ex. en digital topografisk karta) eller för allmän användning av paletter för rendering av rasterlager.
3. *Singleband gray* - (ett band av) bilden kommer att återges som grå. QGIS väljer denna renderingsmetod om filen varken är multiband eller paletterad (t.ex. en skuggad reliefkarta).
4. *Singleband pseudocolor* - den här renderingen kan användas för filer med en kontinuerlig palett eller färgkarta (t.ex. en höjdkarta).
5. *Enkel färg* - rasterlagret kommer att återges med en enda färg.
6. *Terrängskuggning* - Skapar terrängskuggning från ett band.
7. *Contours* - Genererar konturer i farten för ett källrasterband.

Multiband färg

Med multiband-färgåtergivningen kommer tre valda band från bilden att användas som den röda, gröna eller blå komponenten i färgbilden. QGIS hämtar automatiskt värdena *Min* och *Max* för varje band i rastret och skalar färgläggningen i enlighet med detta. Du kan styra värdeintervallen i avsnittet *Min/Max Value Settings*.

En *Kontrastförbättringsmetod* kan tillämpas på värdena: ”Ingen förstärkning”, ”Sträck till MinMax”, ”Sträck och klipp till MinMax” och ”Klipp till min max”.

Observera

Kontrastförstärkning

När du lägger till GRASS-raster kommer alternativet *Kontrastförstärkning* alltid att automatiskt ställas in på *stretch to min max*, även om detta är inställt på ett annat värde i QGIS allmänna alternativ.

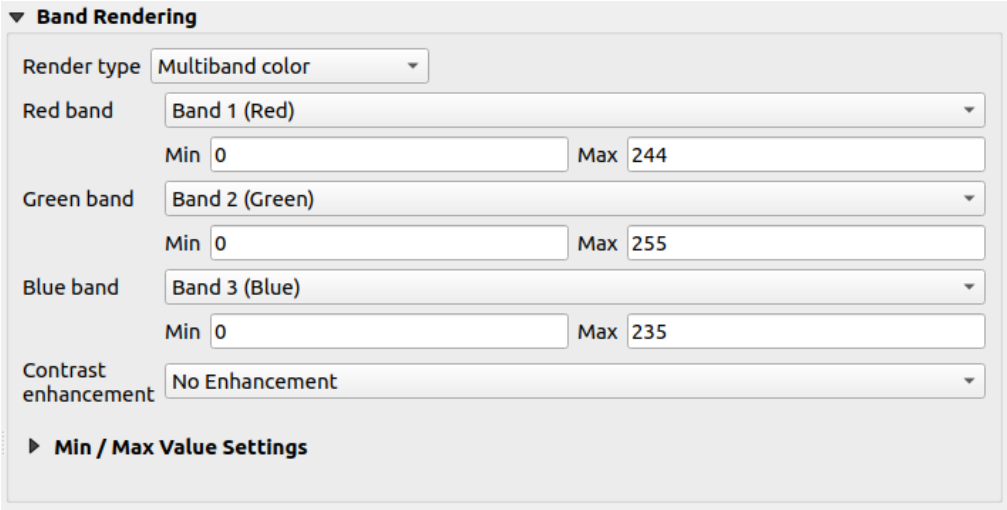


Fig. 13.2: Rastersymbologi - Multiband färgåtergivning

Tips

Visning av ett enstaka band i en multibandsraster

Om du vill visa ett enda band i en flerbandsbild (t.ex. rött) kan du tro att du ska ställa in de gröna och blå banden på *Not Set*. Men det bästa sättet att göra detta är att ställa in bildtypen till *Singleband gray* och sedan välja Red som det *Gray band* som ska användas.

Paletterade/unika värden

Detta är standardrenderingsalternativet för singleband-filer som innehåller en färgtabell, där en viss färg tilldelas varje pixelvärde. I så fall renderas paletten automatiskt.

Den kan användas för alla typer av rasterband och tilldelar en färg till varje unikt rastervärde.

Om du vill ändra en färg dubbelklickar du bara på färgen och dialogrutan *Välj färg* visas.

Det är också möjligt att tilldela färgerna etiketter. Etiketten visas då i rasterlagrets teckenförklaring.

Om du högerklickar på markerade rader i färgkartan visas en kontextuell meny för att:

- *Ändra färg…* för urvalet

- *Ändra opacitet…* för urvalet
- *Ändra etikett…* för urvalet

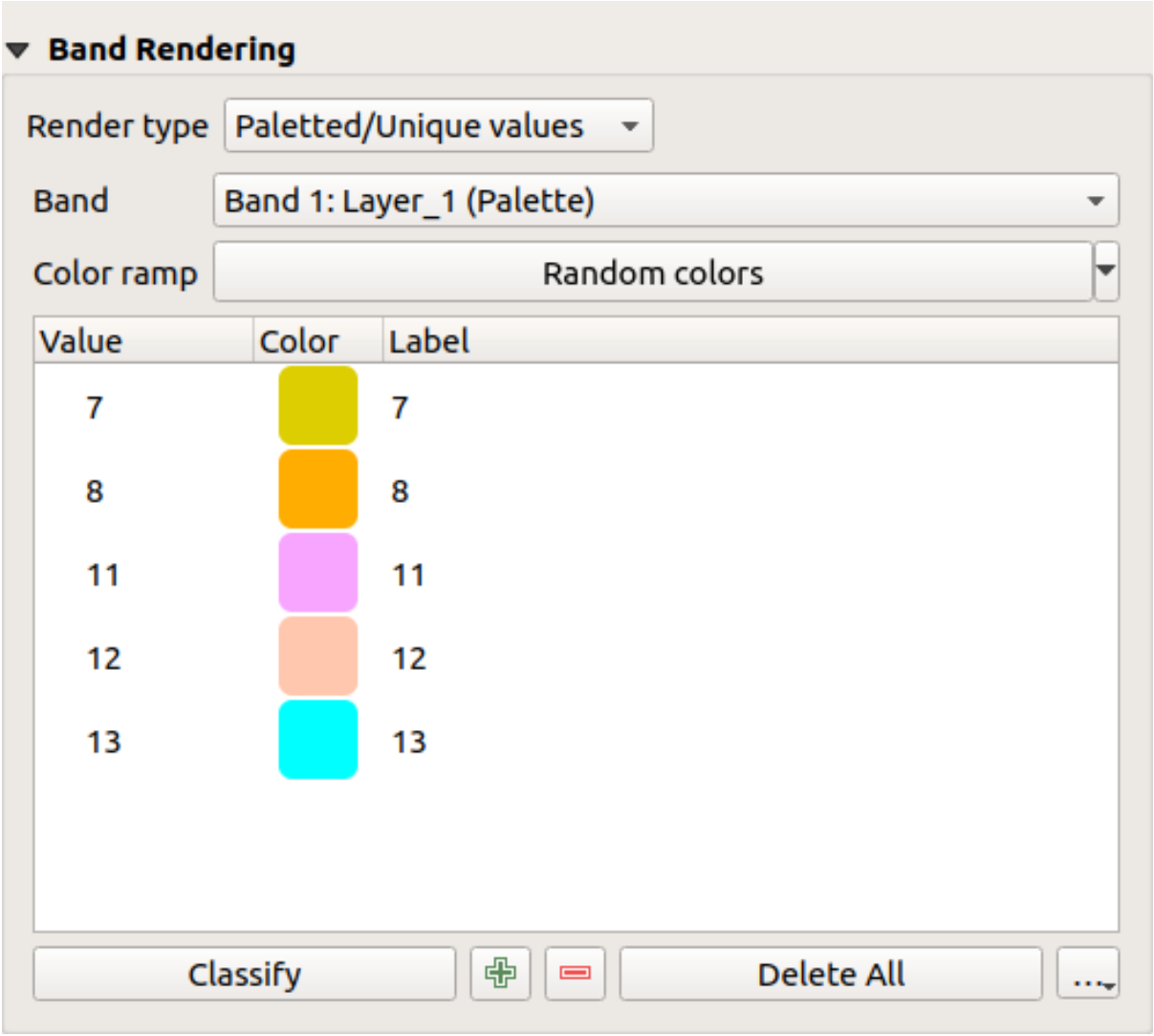


Fig. 13.3: Rastersymbologi - Paletterad unik värdeåtergivning

Rullgardinsmenyn, som öppnas när du klickar på knappen … (Advanced options) under färgkartan till höger, erbjuder laddning av färgkartor (*Load Color Map from File…*) och export (*Export Color Map to File…*), samt laddning av klasser (*Load Classes from Layer*).

Enbands grå

Med denna renderare kan du rendera ett lager med endast ett band med en *Color gradient*: ’Svart till vitt’ eller ’Vitt till svart’. Du kan ändra intervallet för färgvärden (*Min* och *Max*) i *Min/Max Value Settings*.

En *Kontrastförbättringsmetod* kan tillämpas på värdena: ”Ingen förstärkning”, ”Sträck till MinMax”, ”Sträck och klipp till MinMax” och ”Klipp till min max”.

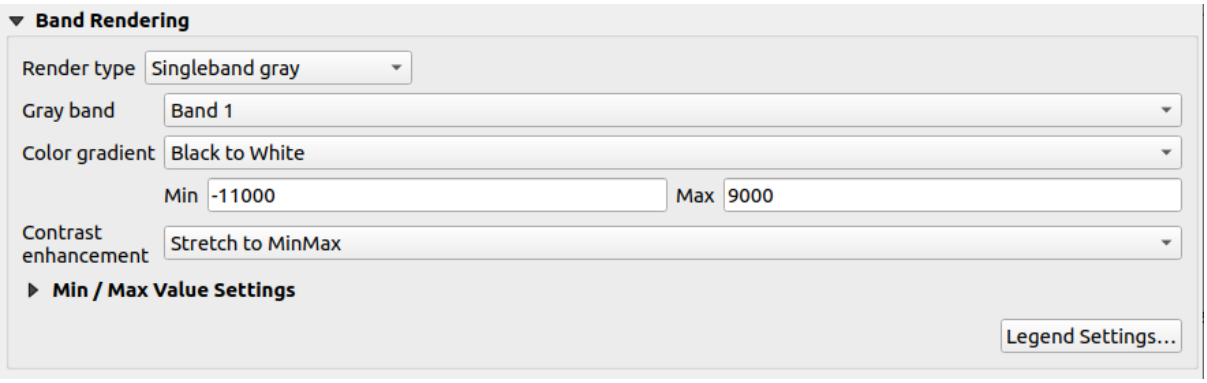


Fig. 13.4: Rastersymbologi - Grå återgivning med ett band

Pixlarna tilldelas en färg som baseras på den valda färggradienten och lagrets förklaring (i panelen *Layers* och layouten *förklaring item*) visas med en kontinuerlig färgramp. Tryck på *Legend settings...* om du vill justera inställningarna. Mer information finns på *Anpassa rasterförklaring*.

Enbands pseudofärg

Detta är ett renderingsalternativ för filer med enstaka band som innehåller en kontinuerlig palett. Du kan också skapa färgkartor för ett band i ett flerbandsraster.

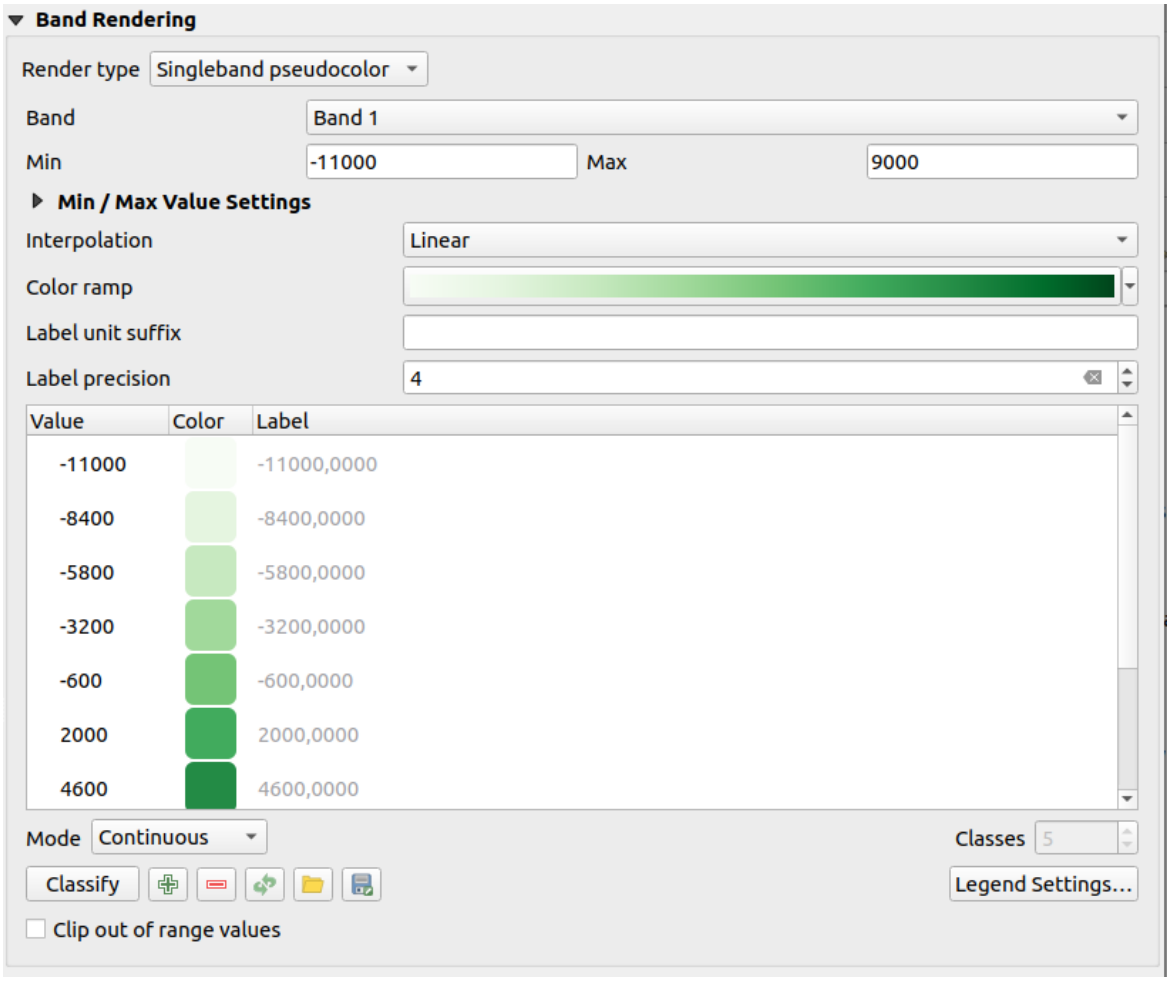


Fig. 13.5: Rastersymbologi - Rendering av pseudofärg med ett band

Med hjälp av en *Band* i lagret och en *values range* kan du nu interpolera och tilldela representationsfärg till pixlar inom klasser. Mer på *Klassificering av färgramp shader*.

Pixlarna tilldelas en färg som baseras på den valda färgrampen och lagrets teckenförklaring (i panelen *Layers* och layouten *förklaring item*) visas med en kontinuerlig färgramp. Tryck på *Legend settings...* om du vill justera inställningarna eller istället använda en förklaring med separerade klasser (och färger). Mer information finns på *Anpassa rasterförklaring*.

En färg

Med den här renderingen kan du rendera ett rasterlager med *Single color*. Den här typen av rendering är användbar när du vill visa ett rasterlager på ett enhetligt sätt, utan någon variation i färg baserat på pixelvärden.

Renderingen med en enda färg kan användas med både rasterlager med enstaka band och flerbandsraster. När den används med multibandraster kan du välja vilket band som ska få en enda färg, vilket gör att det specifika bandet visas enhetligt över hela lagret.

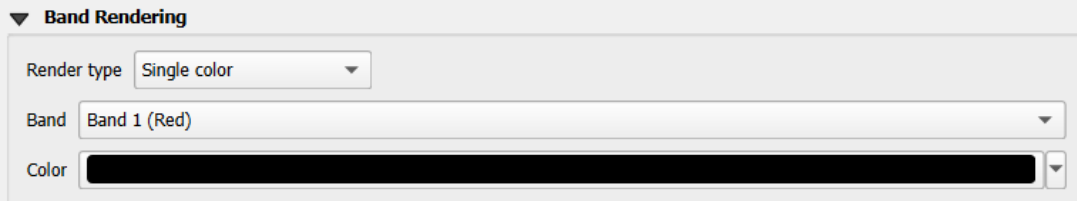


Fig. 13.6: Rastersymbologi - Rendering med en enda färg

Terrängskuggning

Rendera ett band av rasterlagret med hjälp av hillshading.

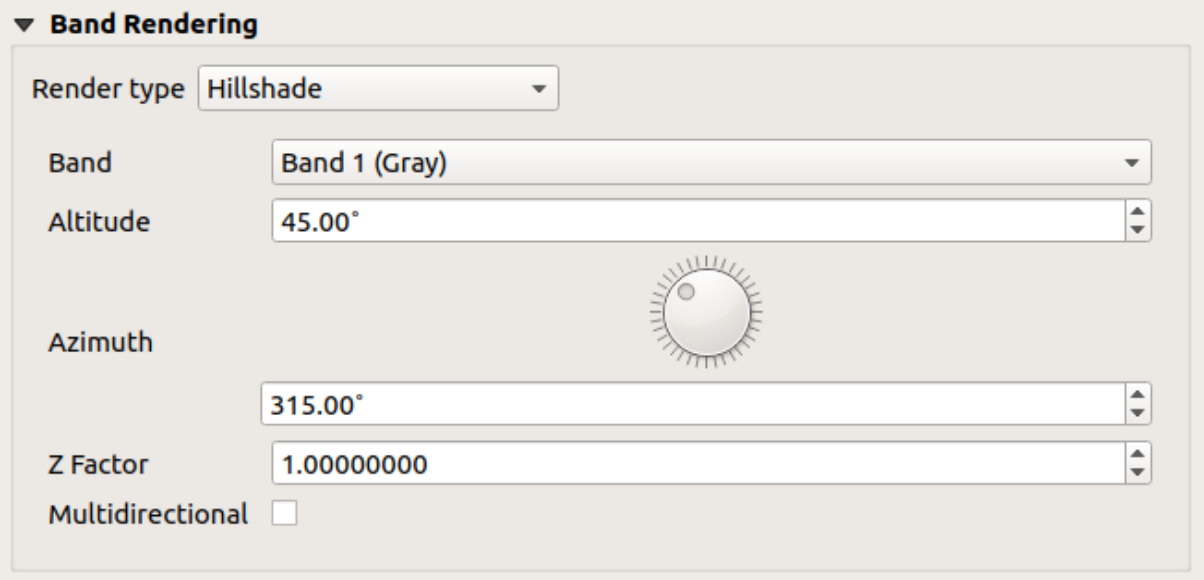


Fig. 13.7: Rastersymbologi - Rendering av terrängskuggning

Alternativ:

- *Band*: Det rasterband som ska användas.
- *Altitude*: Ljuskällans höjdvinkel (standard är `45°`).

- *Azimuth*: Ljuskällans azimut (standard är `315°`).
- *Z Factor*: Skalningsfaktor för värdena i rasterbandet (standard är `1`).
- *Multidirectional*: Ange om multidirektionell hillshading ska användas (standard är `off`).

Konturer

Denna renderare ritar konturlinjer som beräknas i farten från källans rasterband.

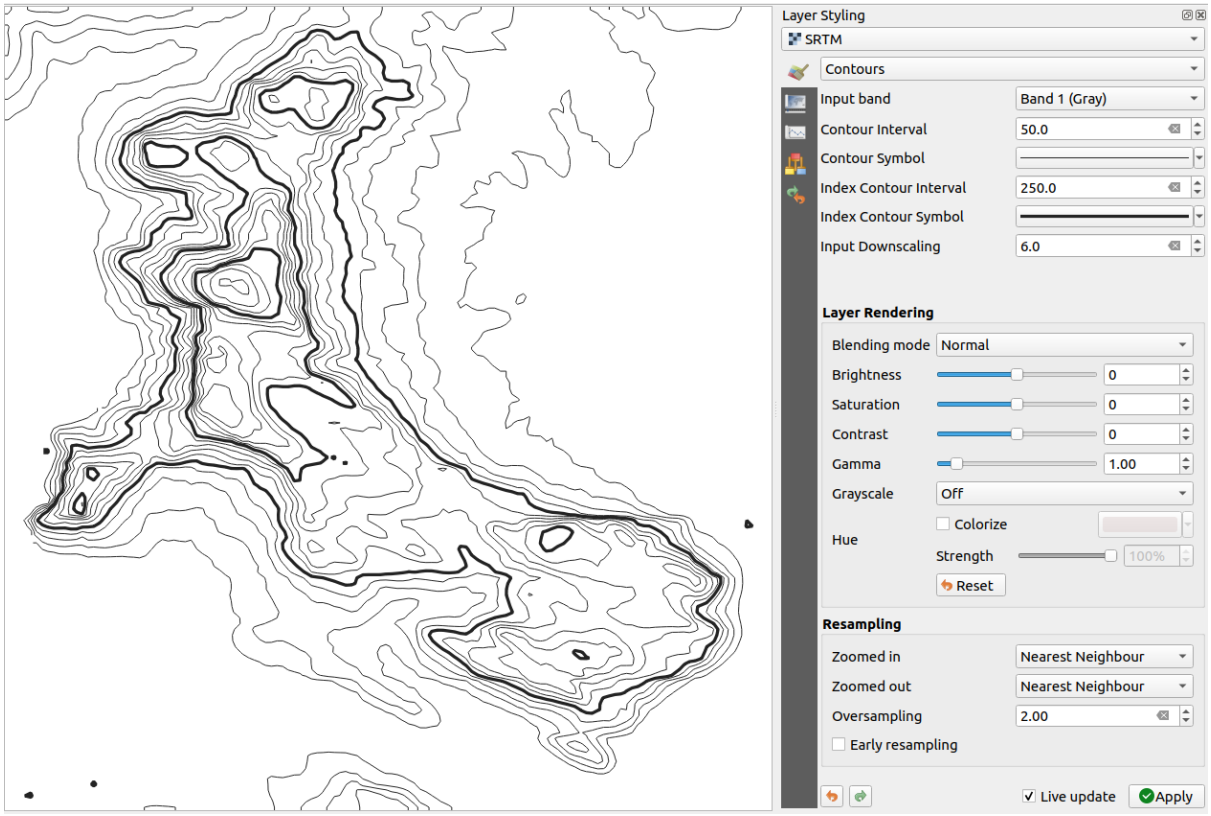


Fig. 13.8: Rastersymbologi - Rendering av konturer

Alternativ:

- *Input band*: det rasterband som ska användas.
- *Konturintervall*: avståndet mellan två på varandra följande konturlinjer
- *Contour symbol*: den *symbol* som ska tillämpas på de gemensamma konturlinjerna.
- *Indexkonturintervall*: avståndet mellan två på varandra följande **indexkonturer**, det vill säga linjer som visas på ett distinkt sätt för att underlätta identifiering, som vanligen trycks kraftigare än andra konturlinjer och vanligen märks med ett värde längs sin sträckning.
- *Indexkontursymbol*: den symbol som ska användas för indexkonturlinjerna
- *Nedskalning av indata*: Anger hur mycket renderingsprogrammet kommer att skala ner begäran till dataleverantören (standard är `4.0`).

  Om du t.ex. genererar konturlinjer på ett ingående rasterblock som har samma storlek som det utgående rasterblocket, kommer de genererade linjerna att innehålla för många detaljer. Denna detaljrikedom kan minskas med faktorn ”nedskalning”, vilket innebär att man begär en lägre upplösning på källrastret. För ett rasterblock 1000x500 med nedskalning 10 kommer renderingsverktyget att begära raster 100x50 från

leverantören. Högre nedskalning gör konturlinjerna mer förenklade (på bekostnad av att vissa detaljer går förlorade).

Ställa in min- och maxvärden

Som standard rapporterar QGIS *Min*- och *Max*-värdena för rastrets band. Ett fåtal mycket låga och/eller höga värden kan ha en negativ inverkan på renderingen av rastret. Ramen *Min/Max Value Settings* hjälper dig att kontrollera renderingen.

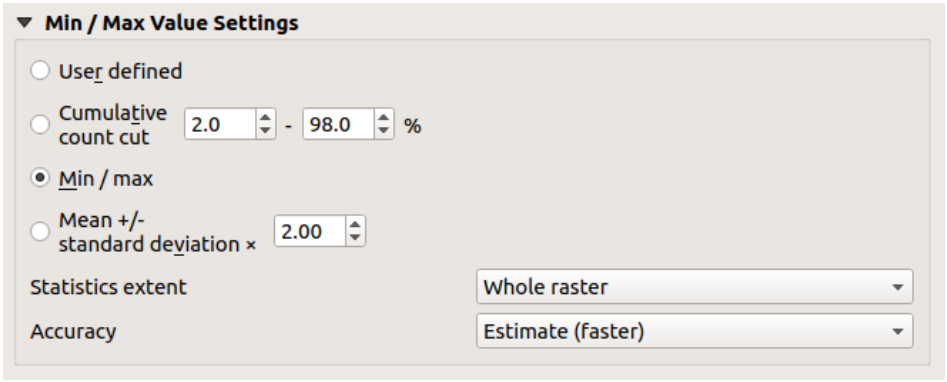


Fig. 13.9: Rastersymbologi - Inställningar för min- och maxvärden

Tillgängliga alternativ är:

- *Användardefinierad*: Standardvärdena *Min* och *Max* för bandet/banden kan åsidosättas
- *Kumulativ räkningsavskärning*: Tar bort outliers. Standardintervallet för värden är `2%` till `98%`, men det kan anpassas manuellt.
- *Min / max*: Använder hela intervallet av värden i bildbandet.
- *Medelvärde +/- standardavvikelse x*: Skapar en färgtabell som endast tar hänsyn till värden inom standardavvikelsen eller inom flera standardavvikelser. Detta är användbart när du har en eller två celler med onormalt höga värden i ett rasterlager som påverkar renderingen av rastret negativt.

Beräkningar av bandens min- och maxvärden görs baserat på:

- *Statistics extent*: det kan vara *Whole raster*, *Current canvas* eller *Updated canvas*. *Updated canvas* innebär att min/max-värden som används för rendering ändras med rasterutbredningen (dynamisk sträckning).
- *Accuracy*, som kan vara antingen *Estimate (snabbare)* eller *Actual (långsammare)*.

Observera

För vissa inställningar kan du behöva trycka på knappen *Apply* i dialogrutan för lageregenskaper för att visa de faktiska min- och maxvärdena i widgetarna.

Klassificering av färgramp shader

Denna metod kan användas för att klassificera och representera skalära dataset (raster eller mesh-kontur) baserat på deras värden. Givet en *color ramp* och ett antal klasser, genererar den mellanliggande färgkartposter för klassgränser. Varje färg mappas med ett värde som interpoleras från ett intervall av värden och i enlighet med ett klassificeringsläge. De skalära datasetelementen tilldelas sedan sin färg baserat på sin klass.

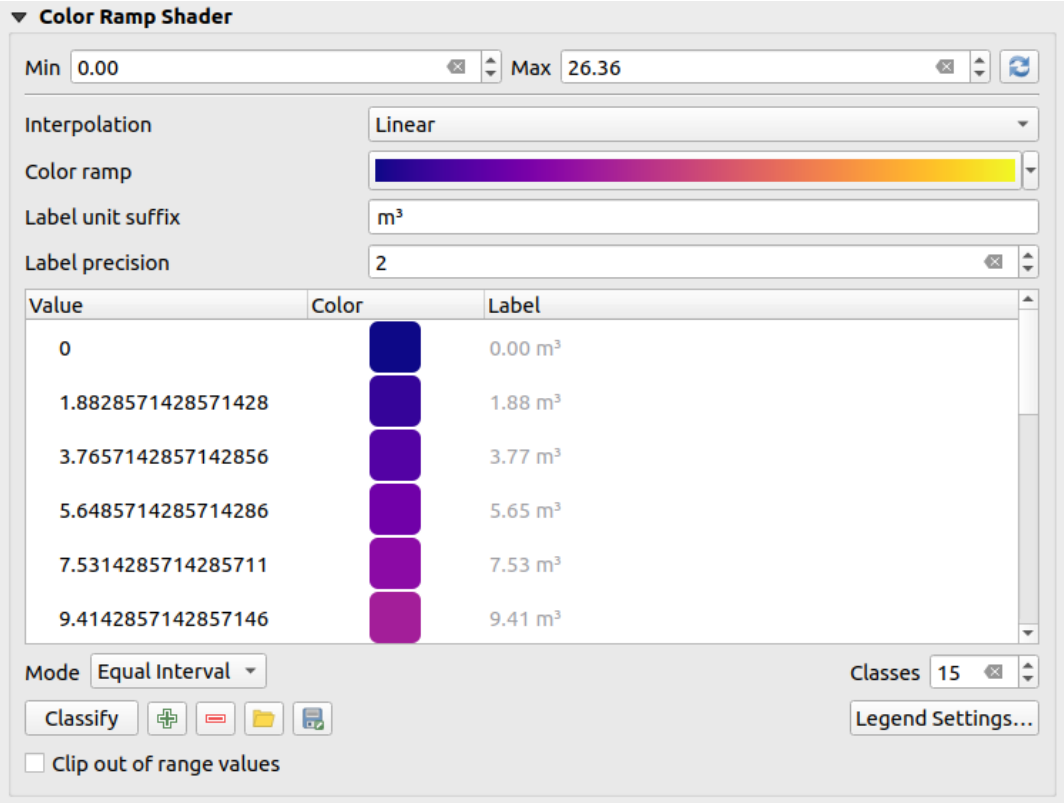


Fig. 13.10: Klassificering av ett dataset med en färgrampshader

1. Ett *Min*- och *Max*-värde måste definieras och användas för att interpolera klassgränser. Som standard detekterar QGIS dem från datasetet, men de kan modifieras.
2. I *Interpolation* anges hur skalära element tilldelas sin färg :
   - *Discrete* (en <= symbol visas i rubriken på kolumnen *Value*): Färgen tas från den närmaste färgkartsposten med lika eller högre värde
   - *Lineär*: Färgen interpoleras linjärt från färgkartans poster över och under pixelvärdet, vilket innebär att varje värde i datasetet motsvarar en unik färg
   - *Exact* (en = symbol visas i rubriken på kolumnen *Value*): Endast pixlar med ett värde som är lika med en post i färgkartan får en färg; andra pixlar återges inte.
3. Widgeten *Color ramp* hjälper dig att välja den färgramp som ska tilldelas datasetet. Som vanligt med *den här widgeten* kan du skapa en ny och redigera eller spara den som är vald för tillfället. Namnet på färgrampen kommer att sparas i konfigurationen.
4. Med *Label unit suffix* läggs en etikett till efter värdet i teckenförklaringen och med *Label precision* styrs antalet decimaler som ska visas.
5. Klassificeringen *Mode* hjälper dig att definiera hur värdena fördelas mellan klasserna:
   - *Ekvivalent intervall*: Förutsatt *Antal klasser*, definieras gränsvärden så att klasserna alla har samma storlek.
   - *Kontinuerlig*: Klasserna nummer och färg hämtas från färgrampens stopp; gränsvärden sätts efter stoppens fördelning i färgrampen.

- *Quantile*: Förutsatt *Antal klasser*, definieras gränsvärden så att klasserna har samma antal element. Ej tillgängligt med *mesh layers*.

6. Du kan sedan *Classify* eller justera klasserna:

   - Knappen Lägg till värden manuellt lägger till ett värde i tabellen.
   - Knappen Remove selected row tar bort markerade värden från tabellen.
   - Genom att dubbelklicka i kolumnen *Value* kan du ändra klassvärdet.
   - Genom att dubbelklicka i kolumnen *Color* öppnas dialogrutan *Change color*, där du kan välja en färg som ska gälla för det värdet.
   - Dubbelklicka i kolumnen *Label* för att ändra klassens etikett, men detta värde kommer inte att visas när du använder verktyget identify feature.
   - Om du högerklickar på markerade rader i färgtabellen visas en kontextuell meny med *Change Color…* och *Change Opacity…* för markeringen.

   Du kan använda knapparna Ladda färgkarta från fil eller Exportera färgkarta till fil för att ladda en befintlig färgtabell eller för att spara färgtabellen för senare användning.

7. Med linjär *Interpolation* kan du också konfigurera:

   - *Klipp bort värden utanför intervallet*: Som standard tilldelar den linjära metoden den första klassens (respektive den sista klassens) färg till värden i datasetet som är lägre än det inställda *Min*-värdet (respektive högre än det inställda *Max*-värdet). Markera denna inställning om du inte vill rendera dessa värden.
   - *Legend settings*, för visning i panelen *Layers* och layouten *förklaring item*. Mer information finns på *Anpassa rasterförklaring*.

Anpassa rasterförklaring

När du tillämpar en färgramp på ett raster- eller mesh-lager kanske du vill visa en förklaring som visar klassificeringen. Som standard visar QGIS en kontinuerlig färgramp med min- och maxvärden i panelen *Layers* och i layouten *förklaring item*. Detta kan anpassas med hjälp av knappen *Legend settings* i klassificeringswidgeten.

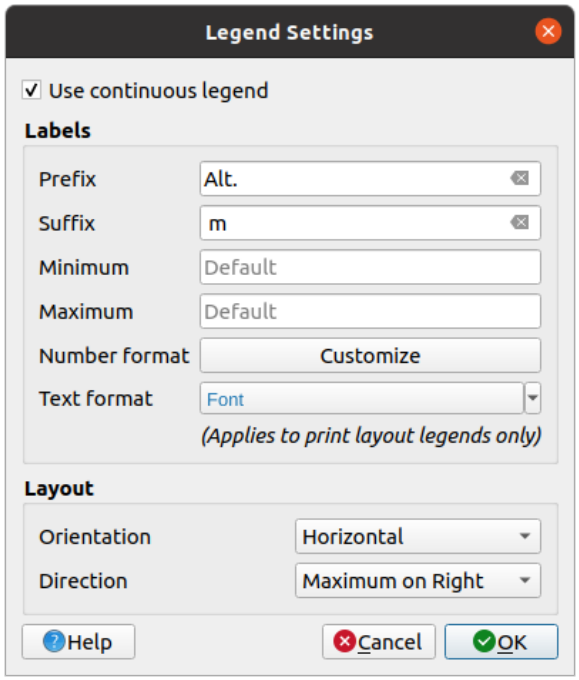


Fig. 13.11: Ändra en rasterförklaring

I den här dialogrutan kan du ange om du vill att *Use continuous förklaring*: om den inte är markerad visar teckenförklaringen separerade färger som motsvarar de olika klasser som tillämpas. Detta alternativ är inte tillgängligt för raster *singleband gray* symbologi.

Om du markerar *Use continuous förklaring* kan du konfigurera både etiketter och layoutegenskaper för teckenförklaringen.

Märkningar

- Lägg till en *Prefix* och en *Suffix* till etiketterna
- Ändra värdena för *Minimum* och *Maximum* så att de visas i teckenförklaringen
- *Anpassa Nummerformat*
- *Customize Textformat* som ska användas i teckenförklaringen för utskriftslayouten.

Layout

- Styr *Orientation* av färgrampen för förklaringen; den kan vara **Vertikal** eller **Horisontell**
- Styr *Direction* för värdena beroende på orientering:
  - Om den är vertikal kan du visa **Maximum överst** eller **Minimum överst**
  - Om det är horisontellt kan du visa **Maximum på höger sida** eller **Minimum på höger sida**

Rendering av lager

Med hjälp av den symbologityp som tillämpas på lagerbanden kan du uppnå speciella renderingseffekter för hela rasterfilen:

- Använd ett av blandningslägena (se *Blandningslägen*)
- Ställ in anpassade *Brightness*, *Saturation*, *Gamma* och *Contrast* till färger.
- Med *Invert colors* återges lagret med motsatta färger. Praktiskt, till exempel, för att växla OpenStreetMap-plattor till mörkt läge.
- Ställ in lagret på *Gråskala* antingen 'Efter ljushet', 'Efter ljusstyrka' eller 'Efter medelvärde'.

- *Colorize* och justera *Strength* av *Hue* i färgtabellen

Tryck på *Reset* för att ta bort alla egna ändringar i lagerrenderingen.

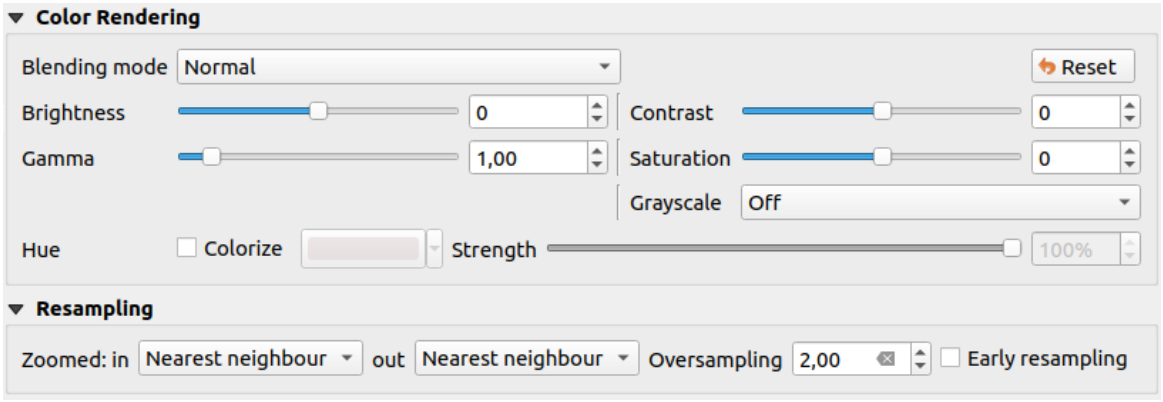


Fig. 13.12: Rastersymbologi - inställningar för lagerrendering och omsampling

Ny provtagning

Alternativet *Resampling* har effekt när du zoomar in och ut i en bild. Omsamplingslägen kan optimera kartans utseende. De beräknar en ny gråvärdesmatris genom en geometrisk transformation.

När man använder metoden ”Nearest neighbour” kan kartan få en pixlig struktur när man zoomar in. Detta utseende kan förbättras genom att använda metoderna ”Bilinear (2x2 kernel)” eller ”Cubic (4x4 kernel)”, som gör att skarpa kanter suddas ut. Effekten blir en mjukare bild. Den här metoden kan t.ex. tillämpas på digitala topografiska rasterkartor.

Early resampling: gör det möjligt att beräkna rasterrenderingen på leverantörsnivå där källans upplösning är känd, och säkerställer en bättre zoom i rendering med QGIS anpassad styling. Mycket praktiskt för kakelraster som laddats med en *tolkningsmetod*.

13.1.4 Egenskaper för genomskinlighet

QGIS tillhandahåller funktioner för att ställa in nivån *Transparency* för ett rasterlager.

Använd reglaget *Global opacity* för att ställa in i vilken utsträckning de underliggande lagren (om några) ska vara synliga genom det aktuella rasterlagret. Detta är mycket användbart om du överlagrar rasterlager (t.ex. en skuggad reliefkarta som överlagras av en klassificerad rasterkarta). På så sätt blir kartan mer tredimensionell. Rastrets opacitet kan vara datadefinierad och variera t.ex. beroende på hur synligt ett annat lager är, av tidsvariabler, på olika sidor i en atlas, …

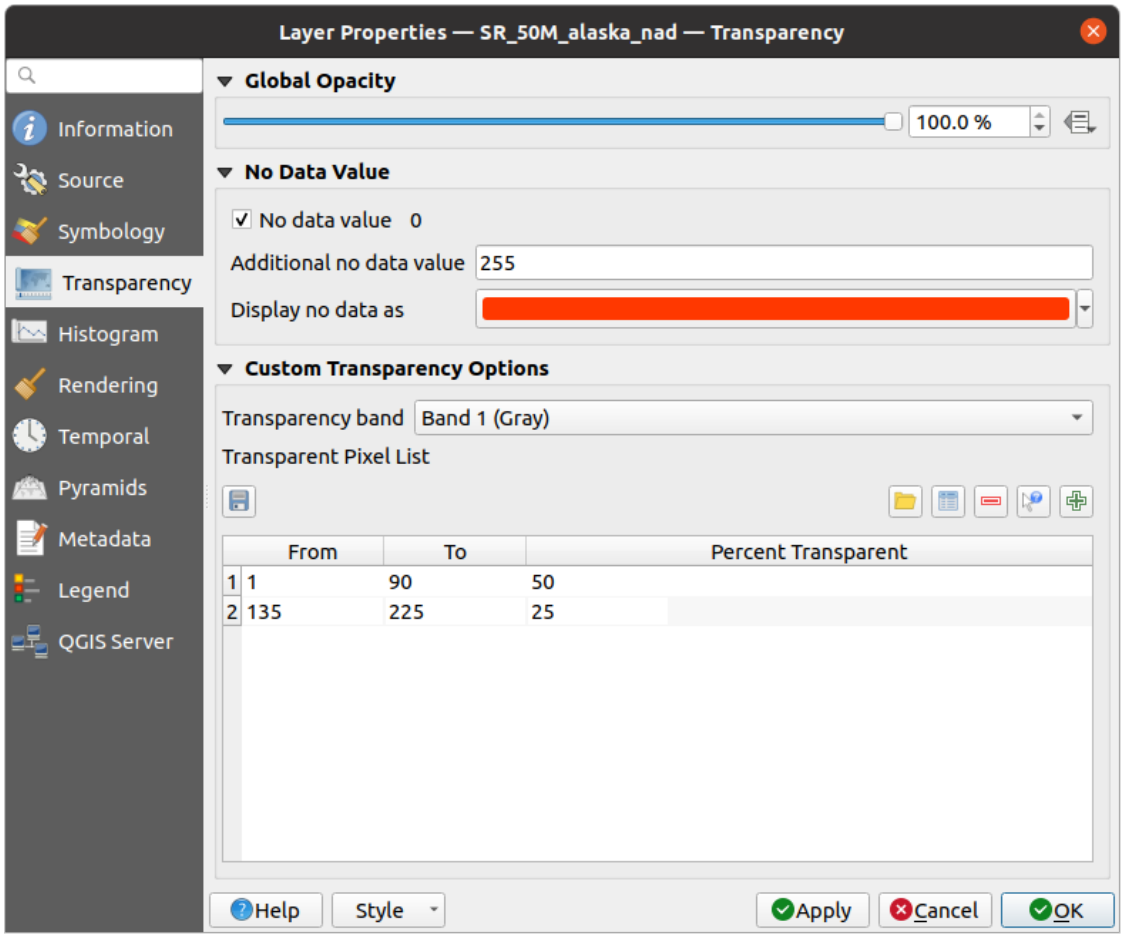


Fig. 13.13: Genomskinlighet för raster

Med *No data value* rapporterar QGIS originalkällans no data-värde (om definierat) som du kan betrakta som det är i renderingen. Dessutom kan du ange ett rastervärde som ska behandlas som ett *Addditional no data value*. Med färgväljaren *Display no data as* kan du använda en anpassad färg på pixlar utan data, i stället för den transparenta standardrenderingen.

Ett ännu mer flexibelt sätt att anpassa transparensen finns i avsnittet *Custom transparency options* (Anpassade transparensalternativ):

- Använd *Transparency band* för att tillämpa transparens för ett helt band.
- Ange en lista över pixlar som ska göras transparenta med motsvarande transparensnivåer:

  1. Klicka på knappen Lägg till värden manuellt. En ny rad kommer att visas i pixellistan.
  2. För enbandsbaserad symbologi (t.ex. DEM), ange värdena **From** och **To** och justera **Percent Transparent** för att tillämpa.
  3. För flerbandsbaserad symbologi (t.ex. RGB-bilder) anger du värdena **Röd**, **Grön** och **Blå** för pixeln och justerar **Percent Transparent** som ska tillämpas. QGIS stöder **Tolerans** för pixelvärden när man definierar transparens. Det innebär att pixlar med en färg som ligger nära de angivna RGB-värdena också kan göras transparenta. Observera att den här funktionen endast gäller för multibandraster.
  4. Alternativt kan du hämta pixelvärdena direkt från rastret med hjälp av knappen Lägg till värden från displayen. Ange sedan värdet för transparens.
  5. Upprepa stegen för att justera fler värden med anpassad transparens.
  6. Tryck på *Apply*-knappen och ta en titt på kartan.

Som du kan se är det ganska enkelt att ställa in anpassad transparens, men det kan vara ganska mycket arbete. Därför kan du använda knappen Exportera till fil för att spara din transparenslista i en fil. Knappen Importera från fil laddar dina transparensinställningar och tillämpar dem på det aktuella rasterlagret.

13.1.5 Egenskaper för histogram

På fliken *Histogram* kan du visa fördelningen av värdena i ditt raster. Histogrammet genereras när du trycker på knappen *Compute Histogram*. Alla befintliga band kommer att visas tillsammans. Du kan spara histogrammet som en bild med knappen .

Längst ned i histogrammet kan du välja ett rasterband i rullgardinsmenyn och *Set min/max style for* det. Rullgardinsmenyn *Prefs/Actions* ger dig avancerade alternativ för att anpassa histogrammet:

- Med alternativet *Visibility* kan du visa histogram för enskilda band. Du måste välja alternativet *Show selected band*.
- Med *Min/max options* kan du välja att ”Always show min/max markers”, att ”Zoom to min/max” och att ”Update style to min/max”.
- Med alternativet *Actions* kan du ”Reset” eller ”Recompute histogram” efter att du har ändrat min- eller maxvärdena för banden.

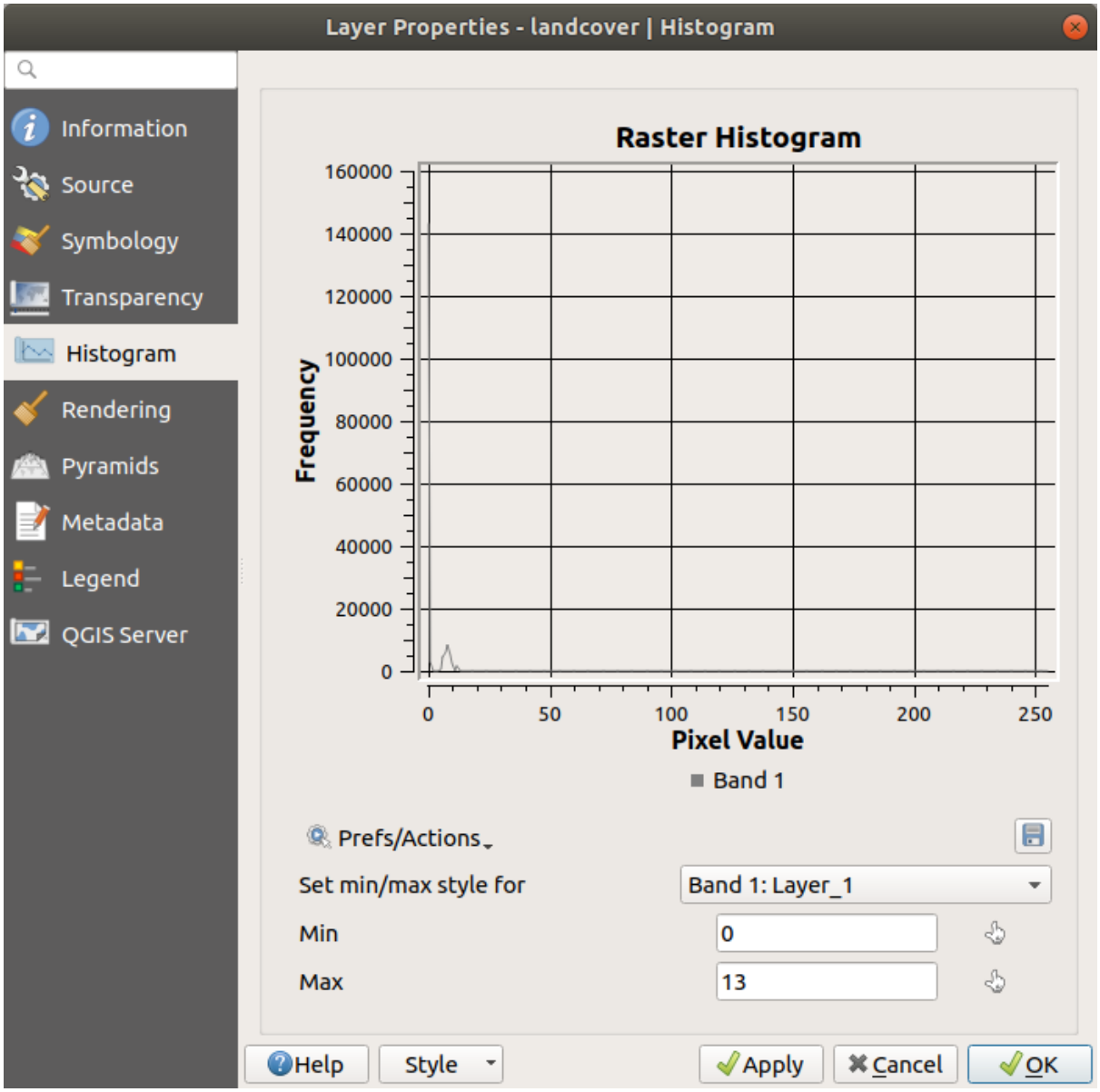


Fig. 13.14: Raster Histogram

13.1.6 Rendering av egenskaper

I fliken *Rendering* är det möjligt att:

- ställa in *Scale dependent visibility* för lagret: Du kan ställa in skalorna *Maximum (inklusive)* och *Minimum (exklusive)* och definiera ett skalområde där lagret är synligt. Utanför detta intervall är det dolt. Knappen Set to current canvas scale hjälper dig att använda den aktuella kartans canvasskala som en gräns. Se *Väljare för synlighetsskala* för mer information.

> **Observera**
>
> Du kan också aktivera skalberoende synlighet på ett lager från panelen *Layers*: högerklicka på lagret och välj *Set Layer Scale Visibility* i snabbmenyn.

- *Refresh layer at interval*: styr om och hur ofta ett lager kan uppdateras. Tillgängliga alternativ för *Konfiguration* är:
  - *Reload data*: lagret kommer att uppdateras helt och hållet. Eventuella cachelagrade data kasseras och hämtas på nytt från leverantören. Detta läge kan resultera i långsammare uppdateringar av kartan.
  - *Redraw layer only*: detta läge är användbart för animering eller när lagrets stil uppdateras med jämna mellanrum. Uppdateringar av Canvas skjuts upp för att undvika att uppdatera flera gånger om mer än ett lager har ett inställt intervall för automatisk uppdatering.

  Det är också möjligt att ställa in *Intervall (sekunder)* mellan på varandra följande uppfriskningar.

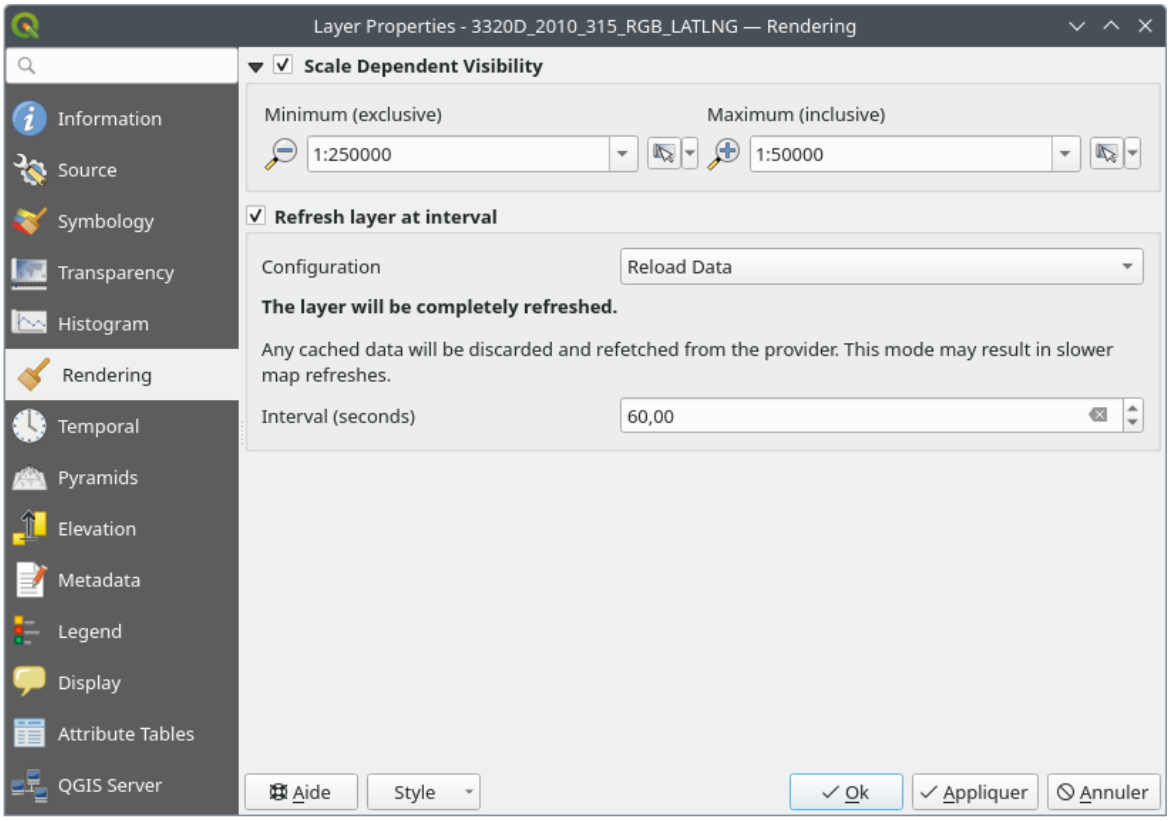


Fig. 13.15: Egenskaper för rasterrendering

13.1.7 Temporala egenskaper

Fliken *Temporal* innehåller alternativ för att styra återgivningen av lagret över tid. Sådan dynamisk rendering kräver att *temporal navigation* är aktiverad över kartans canvas.

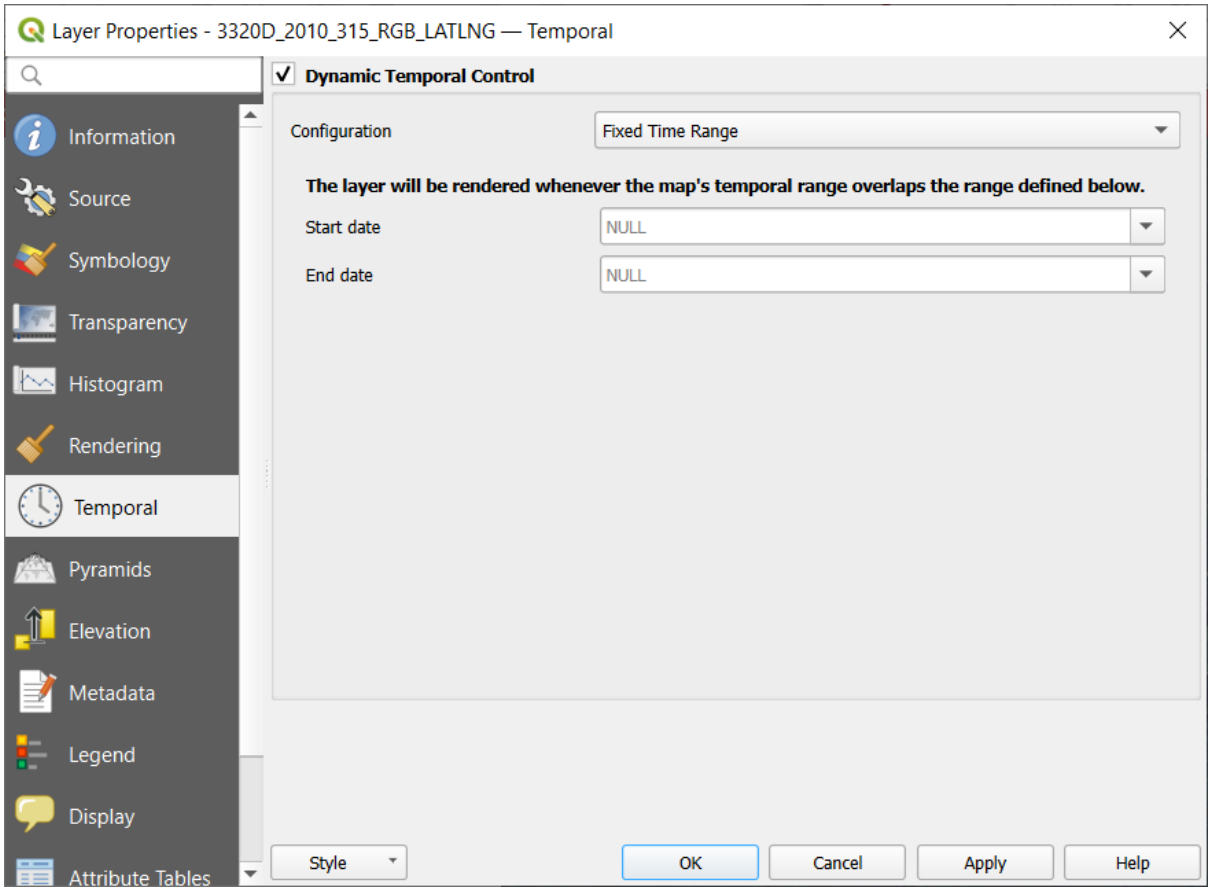


Fig. 13.16: Rasters temporala egenskaper

Markera alternativet *Dynamisk temporal kontroll* och ange om lagret ska ritas om:

- *Automatisk*: Renderingen styrs av den underliggande dataleverantören om den stöder temporal datahantering. Detta kan t.ex. användas med WMS-T-lager eller PostGIS-raster.
- *Fixed time range*: visar bara rasterlagret om animationstiden ligger inom intervallet *Start date* och *End date*
- *Fixed Time Range Per Band*: visar endast ett band när den aktuella animationstiden ligger mellan dess *Begin* och *End* datumintervall. Med det här alternativet kan du antingen manuellt ställa in dessa tidsintervall för varje band eller använda knappen för att automatiskt generera datatidsvärden, vilket möjliggör detaljerad temporal analys och visualisering. Det här läget är särskilt användbart när du arbetar med rasterlager där varje band motsvarar en viss tidsperiod, t.ex. NetCDF-filer.
- *Represents Temporal Values*: tolkar varje pixel i rasterlagret som ett datetime-värde. När detta temporala läge är aktivt döljs pixlar som inte faller inom det temporala intervall som anges i renderingskontexten, vilket säkerställer att endast temporalt relevanta data visas. Detta läge är effektivt för:
  - Analysera förändringar i markanvändning, t.ex. observera avskogningsmönster.
  - Studera översvämningar genom att jämföra vattentäckning under olika tider.
  - Utvärdering av rörelsekostnader i terränganalys, till exempel genom att använda GRASS GIS:s verktyg r.walk för att beräkna rörelsekostnader över ett landskap.
- *Redraw layer only*: lagret ritas om vid varje ny animationsram. Det är användbart när lagret använder tidsbaserade uttrycksvärden för renderarinställningar (t.ex. datadefinierad opacitet för renderaren, för att tona in/ut ett rasterlager).

13.1.8 Pyramidegenskaper

Rasterlager med hög upplösning kan göra navigeringen i QGIS långsammare. Genom att skapa kopior av data med lägre upplösning (pyramider) kan prestandan förbättras avsevärt, eftersom QGIS väljer den lämpligaste upplösningen att använda beroende på zoomnivån.

Du måste ha skrivrättigheter i den katalog där originaldata lagras för att kunna bygga pyramider.

I listan *Resolutions* väljer du de resolutioner där du vill skapa pyramidnivåer genom att klicka på dem.

Om du väljer **Internt (om möjligt)** i rullgardinsmenyn *Översiktsformat* försöker QGIS bygga pyramider internt.

> **Observera**
>
> Observera att byggandet av pyramider kan ändra den ursprungliga datafilen, och när de väl har skapats kan de inte tas bort. Om du vill bevara en ”icke-pyramidformad” version av ditt raster ska du göra en säkerhetskopia innan du bygger pyramiden.

Om du väljer **Externt** och **Externt (Erdas Imagine)** skapas pyramiderna i en fil bredvid originalrastret med samma namn och filtillägget `.ovr`.

Flera *Resampling methods* kan användas för pyramidberäkning:

- Närmaste granne
- Medel
- Gauss
- Kubisk
- Kubisk spline
- Laczos
- Läge
- Ingen

Slutligen klickar du på *Build Pyramids* för att starta processen.

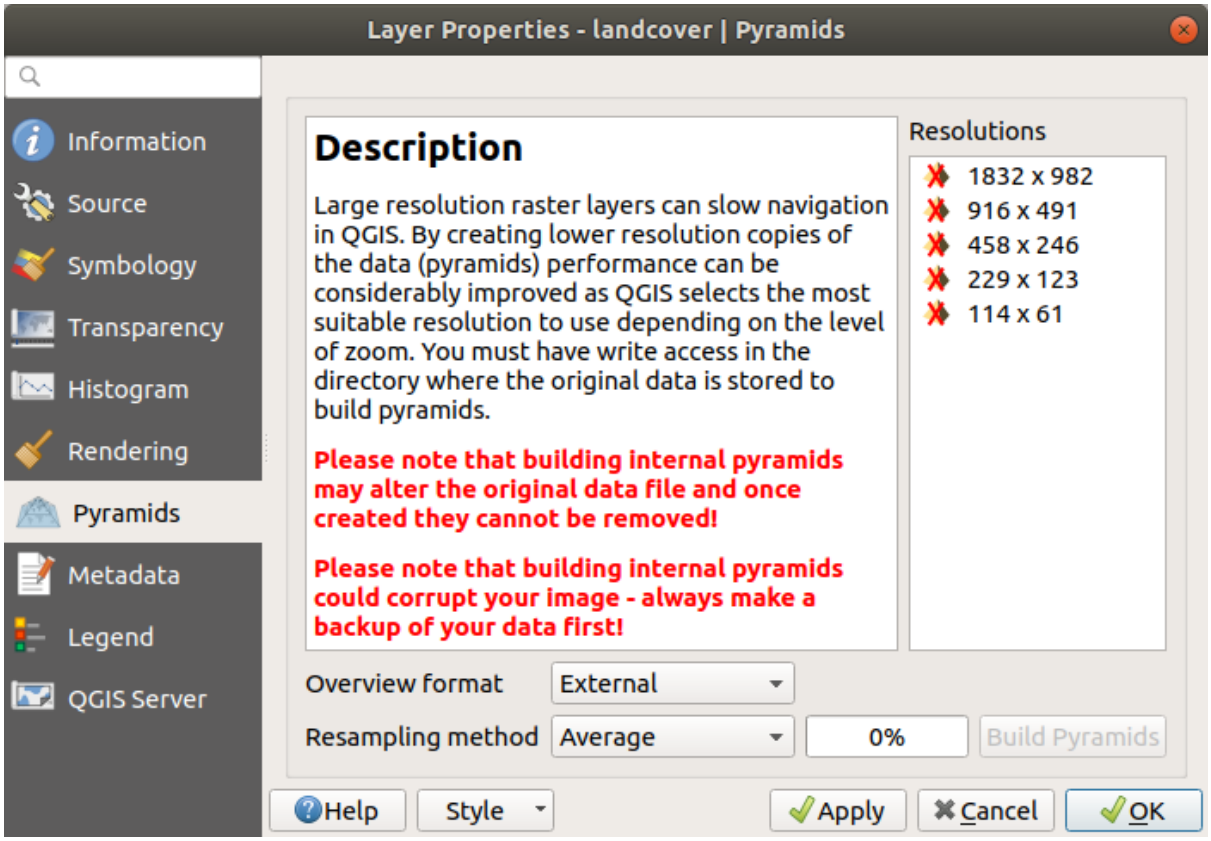


Fig. 13.17: Rasterpyramider

13.1.9 Elevation Egenskaper

Fliken *Elevation* innehåller alternativ för att styra lagrets höjdegenskaper i en *3D-kartvy* och dess utseende i *profilverktygets diagram*. Du kan välja att *Disable* den här konfigurationen om lagret inte innehåller höjddata eller så kan du ställa in:

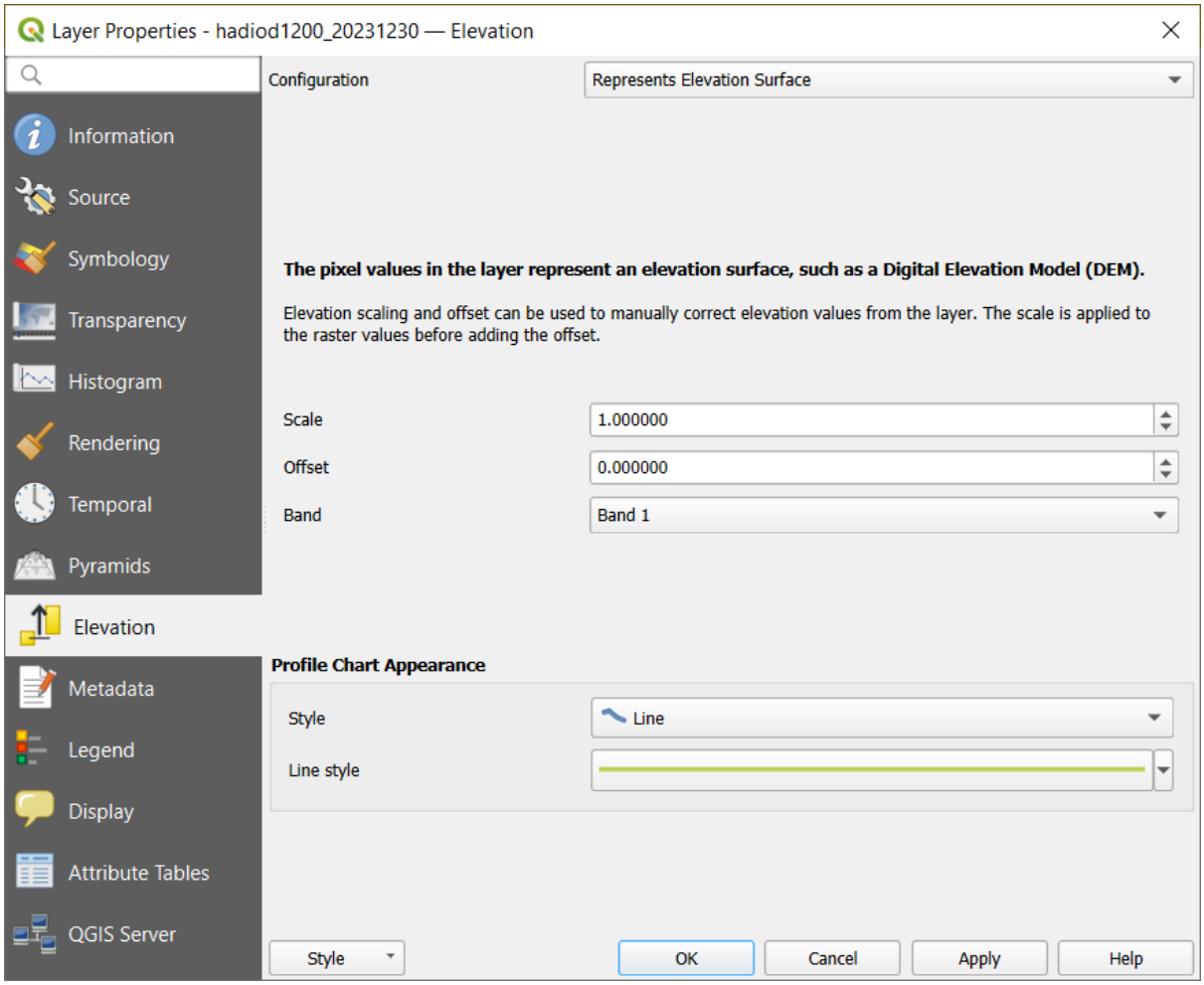


Fig. 13.18: Egenskaper för Rasterhöjd

- *Represents Elevation Surface*: om rasterlagret representerar en höjdyta (t.ex. DEM) och pixelvärdena ska tolkas som höjder. Välj det här alternativet om du vill visa ett raster i en *höjdprofilvy*. Du måste också fylla i *Band* för att välja värden från och kan tillämpa en *Scale*-faktor och en *Offset*.
- *Fixed Elevation Range*: Rasterlagret (eller det valda rasterbandet) är associerat med ett fast höjdintervall. Detta läge kan användas när ett lager har en enda fast höjd eller ett intervall (skiva) av höjdvärden. Om ett intervall anges kommer pixlar att extruderas över detta intervall. Du kan ställa in *Lower* och *Upper* höjdintervallvärden för lagret och ange om de nedre eller övre *Limits* är inkluderande eller exklusiva.
- *Fast höjdintervall per band*: Varje band i rastret kan ha ett fast höjdintervall associerat med sig. Detta är avsett för datakällor som exponerar höjdrelaterade data i band, t.ex. NetCDF-filer. Till exempel ett raster med temperaturdata på olika havsdjup. Vid rendering kommer det översta matchande bandet att väljas och användas för lagrets data. Den här funktionen exponeras som en användarredigerbar tabell för rasterband med lägre och högre värden. Användarna kan antingen fylla i de nedre och övre värdena manuellt eller använda ett ε *Expression* för att automatiskt fylla i alla bandvärden baserat på uttryck. Med den uttrycksbaserade ifyllningen kan du utforma uttryck som extraherar användbar information från bandnamn. Du kan t.ex. extrahera djupvärdet från ett bandnamn som ”Band 001: depth=-5500 (meters)”.
- *Profile Chart Appearance*: styr återgivningen av rasterhöjddata i profildiagrammet. Profilen *Style* kan ställas in som:
  - en *Linje* med en specifik *Linjestil*
  - en höjdyta som återges med en fyllnadssymbol antingen ovanför (*Fill above*) eller nedanför (*Fill below*) höjdkurvans linje. Ytsymboliken representeras med hjälp av:
    - en *Fyllningsstil*

* och en *Limit*: den maximala (respektive minimala) höjden som bestämmer hur hög fyllnadsytan kommer att vara

13.1.10 Egenskaper för metadata

Fliken *Metadata* ger dig alternativ för att skapa och redigera en metadatarapport om ditt lager. Se *Metadata* för mer information.

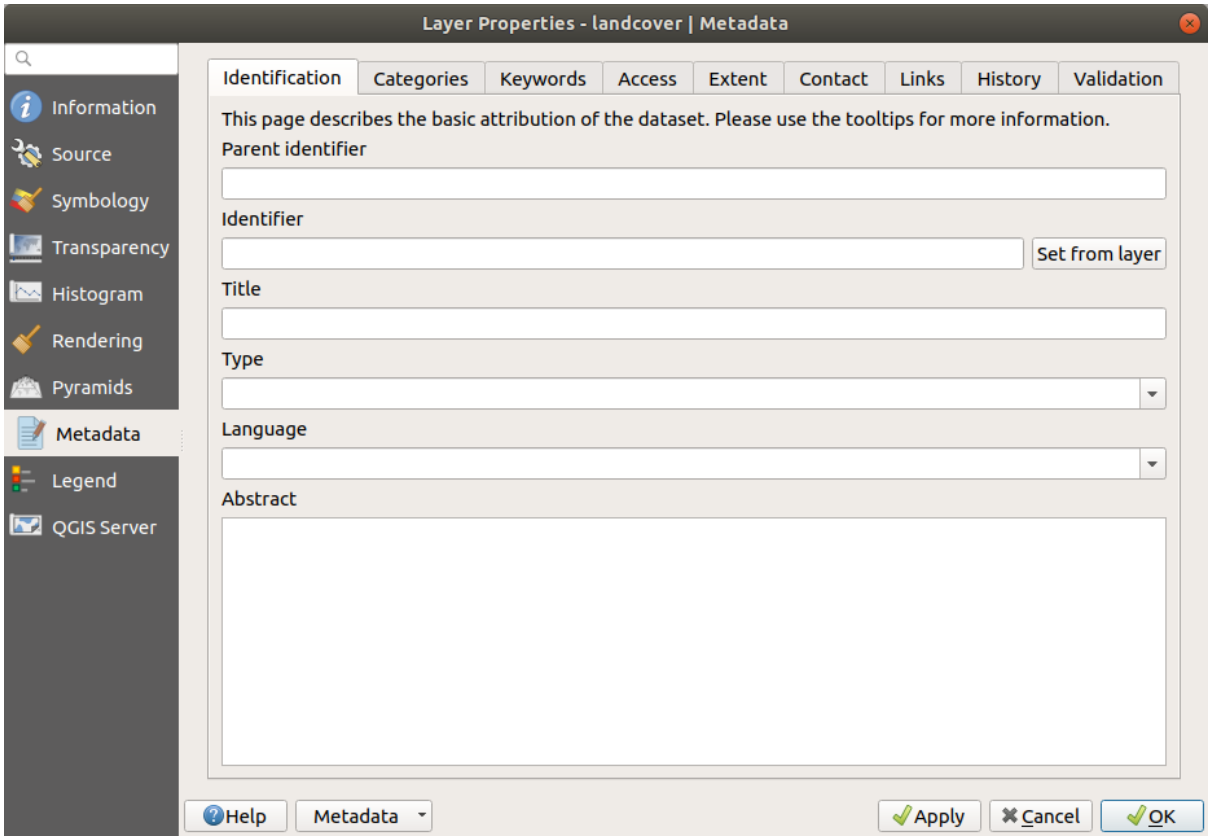


Fig. 13.19: Metadata för raster

13.1.11 Förklaringsegenskaper

Fliken *Förklaring* ger dig avancerade inställningar för *Lager-panelen* och/eller *print layout förklaring*. Dessa alternativ inkluderar:

- Beroende på vilken symbologi som används i lagret kan du få flera poster i förklaringen, som inte nödvändigtvis är läsbara/användbara att visa. Med *Legend placeholder image* kan du *välja en bild* för ersättning, som visas både i panelen *Layers* och i utskriftslayoutens förklaring.
- *Embedded widgets in Legend* ger dig en lista över widgets som du kan bädda in i lagerträdet i panelen Lager. Tanken är att det ska finnas ett sätt att snabbt komma åt vissa åtgärder som ofta används med lagret (inställning av transparens, filtrering, urval, stil eller andra saker …).

  Som standard tillhandahåller QGIS en widget för transparens, men denna kan utökas med tillägg som registrerar sina egna widgets och tilldelar anpassade åtgärder till lager som de hanterar.

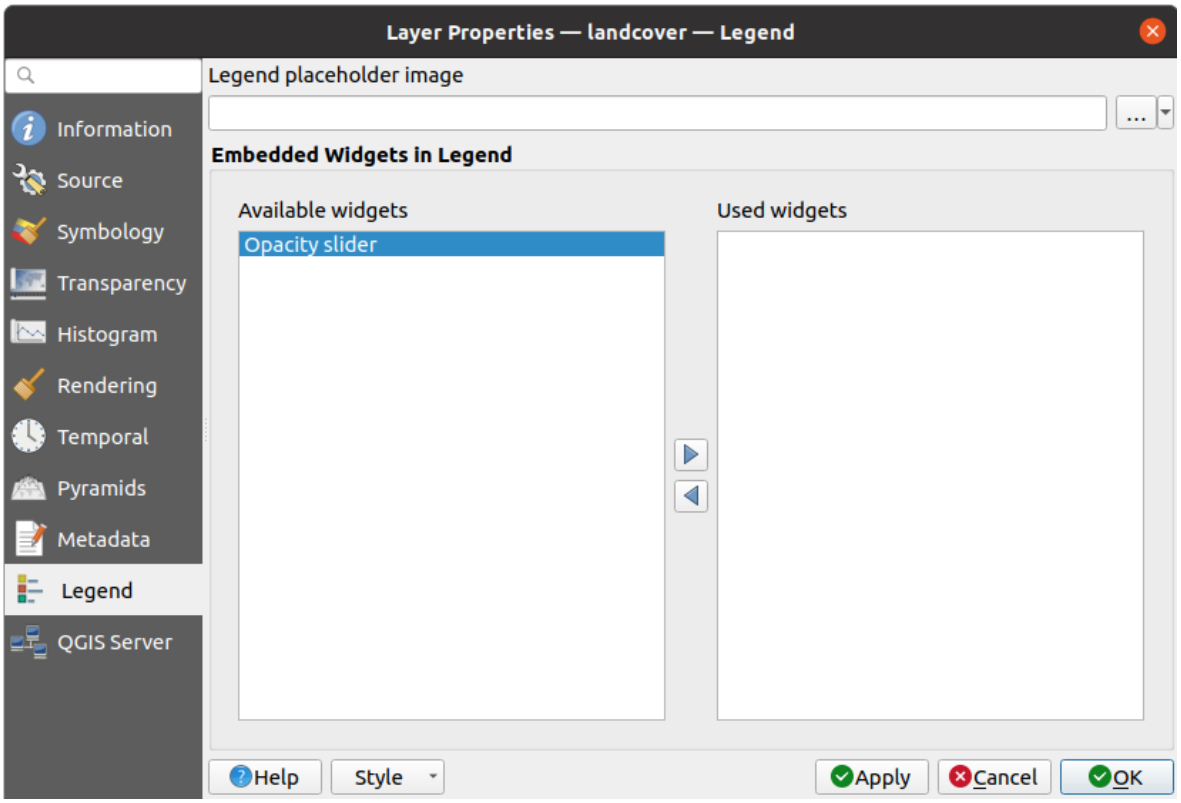


Fig. 13.20: Raster Legend

13.1.12 Egenskaper för display

Fliken *Display* hjälper dig att konfigurera HTML-karttips som ska användas för pixelidentifiering:

- *Enable Map Tips* styr om karttips ska visas för lagret
- *HTML Map Tip* ger en komplex och fullständig HTML-textredigerare för karttips, som blandar QGIS-uttryck och html-stilar och taggar (multilinjer, typsnitt, bilder, hyperlänkar, tabeller, …). Du kan kontrollera resultatet av ditt kodprov i ramen *Preview*. Du kan också välja och redigera befintliga uttryck med hjälp av knappen *Insert/Edit Expression*.

  Du kan leta efter uttryck i gruppen *Rasters* eller variabeln `@layer_cursor_point` i dialogrutan *Expressions*.

  Observera

  Förstå *Infoga/redigera uttryck* **knappens beteende**

  Om du markerar text inom ett uttryck (mellan ”[%” och ”%]”), eller om ingen text är markerad men markören befinner sig inom ett uttryck, markeras hela uttrycket automatiskt för redigering. Om markören eller en markerad text befinner sig utanför ett uttryck öppnas dialogrutan med markeringen.

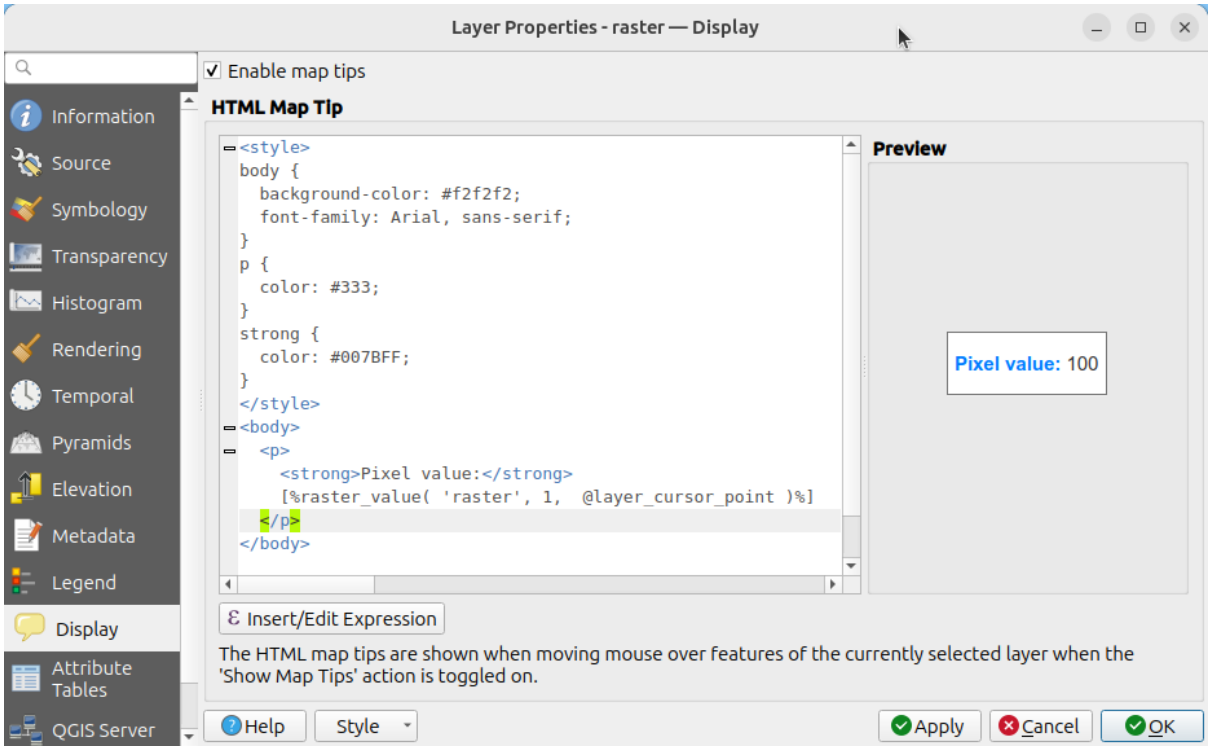


Fig. 13.21: Karttips med rasterlager

För att visa karttips:

1. Välj menyalternativet *View* ► *Show Map Tips* eller klicka på ikonen Show Map Tips i *Attributes Toolbar*.
2. Se till att det lager du vill använda är aktivt och att egenskapen *Aktivera karttips* är markerad.
3. Flytta över en pixel så visas motsvarande information över den.

Map tip är en funktion för flera lager, vilket innebär att när den har aktiverats förblir den aktiverad och gäller för alla lager i projektet som är aktiverade med Map tip tills den stängs av.

13.1.13 Egenskaper för QGIS-server

Fliken *QGIS Server* hjälper dig att konfigurera inställningar för data när de publiceras av QGIS Server. Konfigurationen gäller:

- *Description*: ger information för att beskriva data, såsom *Short name*, *Title*, *Summary*, en *List of Keywords* och en *Data URL* vars *Type* kan vara i `text/html`, `text/plain` eller `application/pdf`.
- *Attribution*: en *Title* och *URL* för att identifiera vem som tillhandahåller data
- *Metadata URL*: en lista med *URL* för metadata som kan vara av `FGDC` eller `TC211` *Type*, och i `text/plain` eller `text/xml` *Format*
- *Legend URL*: en *URL* för förklaringen, i antingen `image/png` eller `image/jpeg` *Format*

Observera

När det rasterlager som du vill publicera redan tillhandahålls av en webbtjänst finns ytterligare *egenskaper* tillgängliga för inställning.

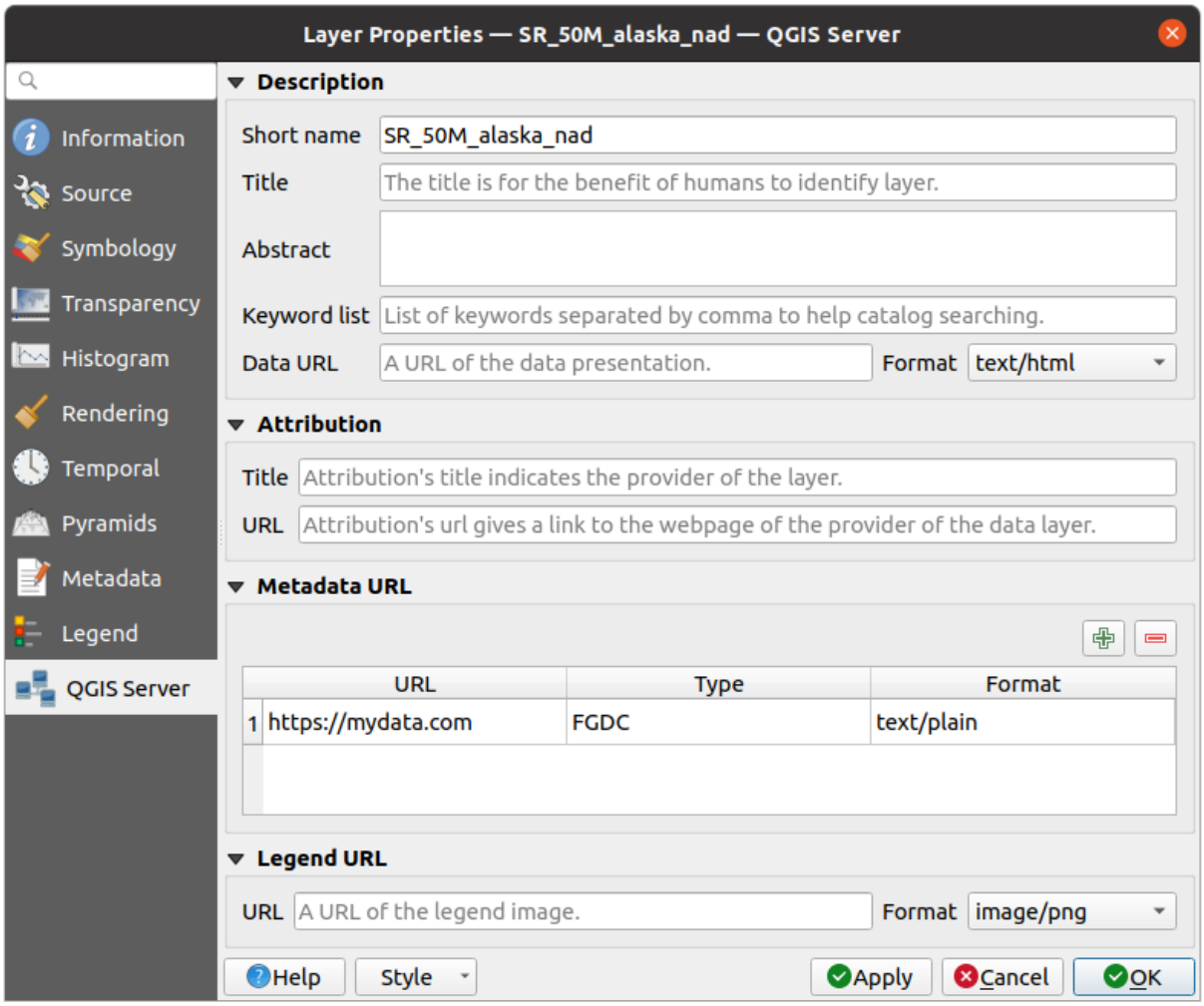


Fig. 13.22: QGIS Server i Egenskaper för Raster

13.1.14 Identifiera rasterceller

Med verktyget *identify features* kan du få information om specifika punkter i ett rasterlager.

För att använda verktyget *Identify features*:

1. Markera rasterlagret i panelen Lager.
2. Klicka på verktyget *Identify features* i verktygsfältet eller tryck på `Ctrl+Shift+I`.
3. Klicka på den punkt i rasterlagret som du vill identifiera.

Panelen Identify Results öppnas i standardvyn `Tree` och visar information om den klickade punkten. Formateringen av resultaten varierar beroende på vilken leverantör som tillhandahåller lagret. Till exempel

- För ett lokalt rasterlager: under lagrets namn har du till vänster banden för den klickade pixeln och till höger deras respektive värde.
- För ett fjärrlager som WMS kan du med hjälp av menyn *Format* välja om informationen ska visas som *HTML*, *Feature* eller *Text*.

Dessa värden kan också återges (från menyn *View* längst ner på panelen) i:

- en `Table`-vy - organiserar informationen om de identifierade funktionerna och deras värden i en tabell.
- en `Graph`-vy - organiserar informationen om de identifierade funktionerna och deras värden i en graf.

Under pixelattributen hittar du *Derived* information, t.ex:

- koordinatvärdena `X` och `Y` för den klickade punkten
- Kolumn och rad för den klickade punkten (pixel) när den är kompatibel

13.2 Rasteranalys

13.2.1 Rasterkalkylator

Med *Raster Calculator* i *Raster*-menyn kan du utföra beräkningar på grundval av befintliga rasterpixelvärden (se Fig. 13.23). Resultaten skrivs till ett nytt rasterlager i ett GDAL-stött format.

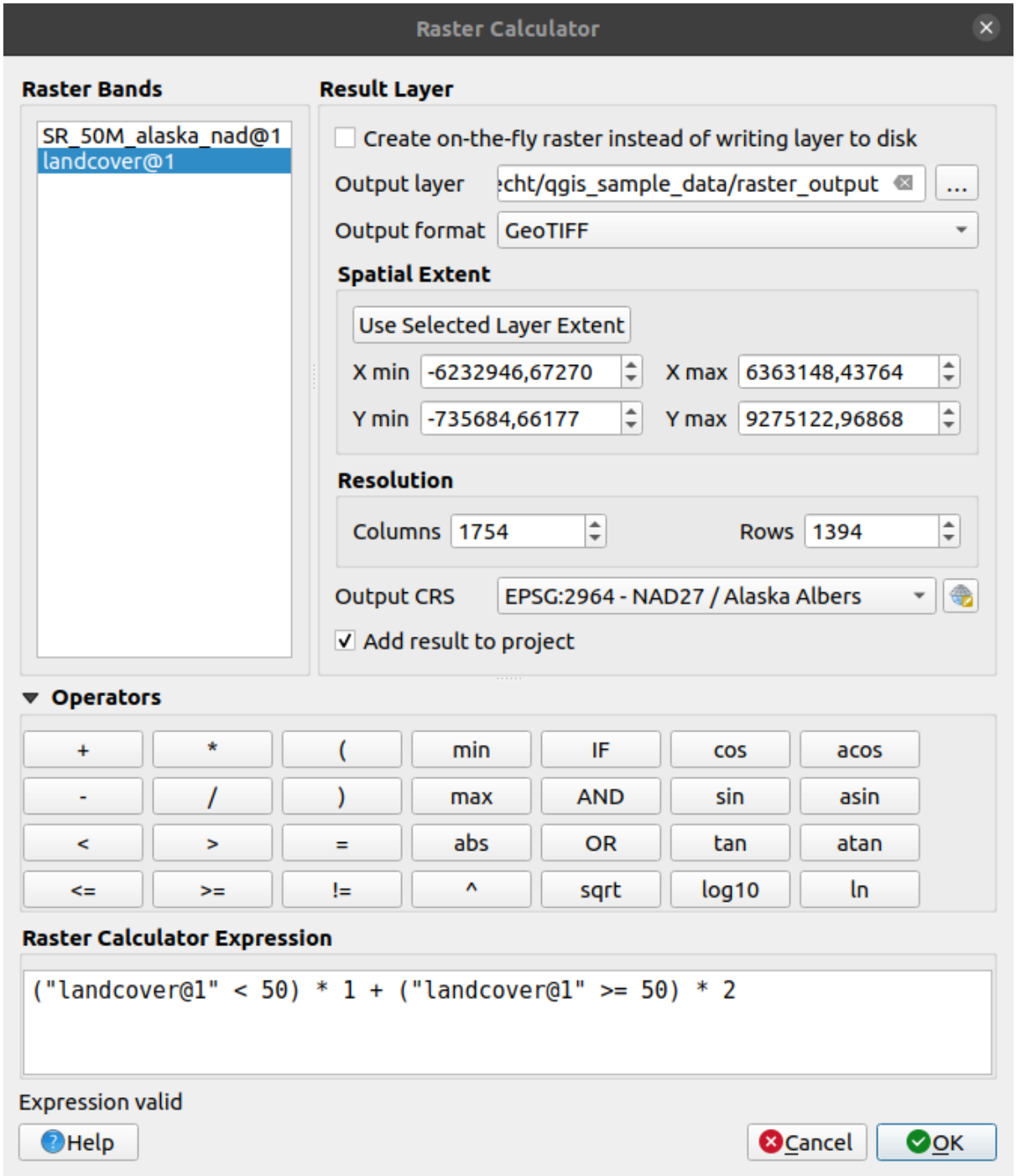


Fig. 13.23: Rasterkalkylator

Listan *Raster bands* innehåller alla inlästa rasterlager som kan användas. Om du vill lägga till ett raster i rasterkalkylatorns uttrycksfält dubbelklickar du på dess namn i listan Fields. Du kan sedan använda operatörerna för att konstruera beräkningsuttryck eller bara skriva in dem i rutan.

I avsnittet *Resultatlager* måste du definiera ett utmatningslager. Det kan du göra:

- *Skapa raster i farten i stället för att skriva lagret till disken*:
  - Om den inte är markerad lagras utdata på disken som en ny vanlig fil. En sökväg till *Output layer* och ett *Output format* krävs.
  - Om det är markerat skapas ett virtuellt rasterlager, dvs. ett rasterlager som definieras av sin URI och vars

pixlar beräknas i realtid. Det är inte en ny fil på disken; det virtuella lagret är fortfarande kopplat till de raster som används i beräkningen, vilket innebär att det skulle gå sönder om dessa raster raderades eller flyttades. En *Layer name* kan anges, annars används beräkningsuttrycket som sådant. Om det virtuella lagret tas bort från projektet raderas det, och det kan göras beständigt i filen med hjälp av lagrets *Export* ► *Save as…* kontextuella meny.

- Definiera *Spatial utsträckning* för beräkningen baserat på en utsträckning för ett rasterlager eller på anpassade X,Y-koordinater
- Ställ in *Resolution* för lagret med hjälp av kolumn- och radnummer. Om inmatningslagret har en annan upplösning kommer värdena att omsamplas med närmaste grannar-algoritmen.
- Med kryssrutan *Add result to project* läggs resultatlagret automatiskt till i legendområdet och kan visualiseras. Markerad som standard för virtuella raster.

Avsnittet *Operators* innehåller alla tillgängliga operatorer. Om du vill lägga till en operator i rasterkalkylatorns uttrycksruta klickar du på lämplig knapp. Matematiska beräkningar (+, −, *, …) och trigonometriska funktioner (`sin`, `cos`, `tan`, …) är tillgängliga. Villkorliga uttryck (=, !=, <, >=, …) returnerar antingen 0 för falskt eller 1 för sant, och kan därför användas med andra operatorer och funktioner.

Se även

algoritmerna *Rasterkalkylator* och *Rasterkalkylator (virtuell)*

Uttryck för rasterkalkylator

Dialogen

Dialogen *Raster calculator expression* ger möjlighet att skriva uttryck för pixelberäkningar mellan en uppsättning rasterlager.

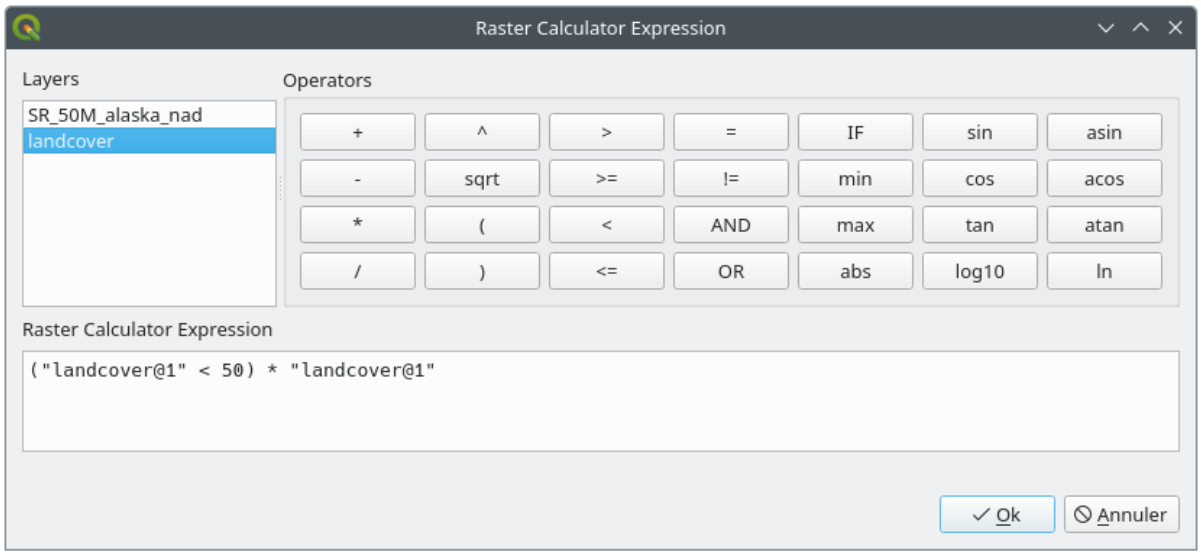


Fig. 13.24: Kalkylator för rasteruttryck

- *Lager*: Visar en lista över alla rasterlager som laddats i förklaringen. Dessa kan användas för att fylla uttrycksfältet (dubbelklicka för att lägga till). Rasterlager refereras till med sitt namn och bandets nummer: `layer_name@band_number`. Till exempel kommer det första bandet från ett lager med namnet `DEM` att refereras till som `DEM@1`.
- *Operators*: innehåller ett antal beräkningsoperatorer för pixelmanipulation:

  - Aritmetiska: +, –, *, `qrt`, `abs`, `ln`, ...
  - Trigonometriska: `sin`, `cos`, `tan`, ...
  - Jämförelse: =, `!=`, <, >=, ...
  - Logiska: `IF`, `AND`, `OR`, `(`, `)`
  - Statistik: `min`, `max`

  Om du vill lägga till en operator i uttrycksrutan för rasterkalkylatorn klickar du på lämplig knapp.

- *Rasterkalkylatoruttryck* är det område där uttrycket är sammansatt

Exempel

Konvertera höjdvärden från meter till fot

Om du skapar ett höjdraster i fot från ett raster i meter måste du använda omvandlingsfaktorn för meter till fot: 3.28. Uttrycket är:

```
"elevation@1" * 3.28
```

Använda en mask

Om du vill maskera bort delar av ett raster, t.ex. för att du bara är intresserad av höjder över 0 meter, kan du använda följande uttryck för att skapa en mask och tillämpa resultatet på ett raster i ett steg.

```
("elevation@1" >= 0) * "elevation@1"
```

Med andra ord, för varje cell som är större än eller lika med 0 utvärderas det villkorliga uttrycket till 1, vilket innebär att originalvärdet behålls genom att det multipliceras med 1. Annars utvärderas det villkorliga uttrycket till 0, vilket innebär att rastervärdet sätts till 0. Detta skapar masken i farten.

Klassificera ett raster

Om du vill klassificera ett raster, t.ex. i två höjdklasser, kan du använda följande uttryck för att skapa ett raster med de två värdena 1 och 2 i ett steg.

```
("elevation@1" < 50) * 1 + ("elevation@1" >= 50) * 2
```

Med andra ord, för varje cell som är mindre än 50 sätts värdet till 1. För varje cell som är större än eller lika med 50 sätts värdet till 2.

Eller så kan du använda operatorn `IF`.

```
if ( elevation@1 < 50 , 1 , 2 )
```

KAPITEL 14

Arbeta med Mesh-data

14.1 Vad är ett nät?

Ett mesh är ett ostrukturerat rutnät som vanligtvis har temporala och andra komponenter. Den spatiala komponenten innehåller en samling hörn, kanter och/eller ytor i 2D- eller 3D-rymd:

- **vertices** - XY(Z)-punkter (i lagrets referenssystem för koordinater)
- **kanter** - förbinder par av hörn
- **ytor** - en yta är en uppsättning kanter som bildar en sluten form - vanligtvis en triangel eller en fyrhörning (quad), sällan polygoner med fler hörn

På grundval av ovanstående kan nätskikten således ha olika typer av struktur:

- 1D Meshes: består av hörn och kanter. En kant förbinder två hörn och kan ha tilldelade data (skalärer eller vektorer) på sig. 1D mesh-nätverket kan t.ex. användas för modellering av ett urbant dräneringssystem.
- 2D-mesh: består av ytor med trianglar, regelbundna eller ostrukturerade kvadrater.
- 3D-skiktade nät: består av flera staplade ostrukturerade 2D-nät som var och en extruderas i vertikal riktning (nivåer) med hjälp av en vertikal koordinat. Topparna och ytorna har samma topologi i varje vertikal nivå. Nätdefinitionen (extrudering på vertikal nivå) kan i allmänhet ändras med tiden. Uppgifterna definieras vanligen i volymscentra eller genom någon parametrisk funktion.

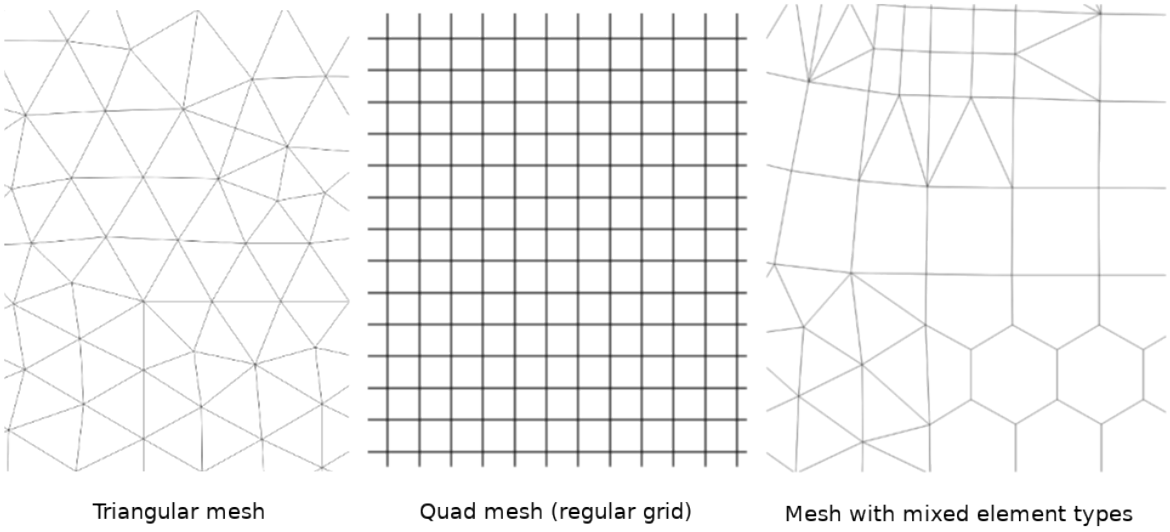


Fig. 14.1: Olika typer av nät

Mesh ger information om den spatiala strukturen. Dessutom kan mesh ha dataset (grupper) som tilldelar ett värde till varje vertex. Ett exempel är ett triangulärt nät med numrerade hörn enligt bilden nedan:

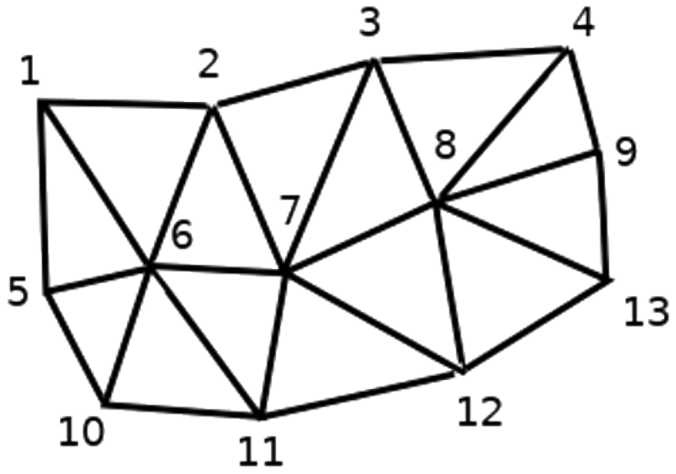


Fig. 14.2: Triangulärt rutnät med numrerade hörn

Varje vertex kan lagra olika dataset (vanligtvis flera kvantiteter), och dessa dataset kan också ha en tidsdimension. En enda fil kan alltså innehålla flera dataset.

Följande tabell ger en uppfattning om vilken information som kan lagras i mesh-dataset. Tabellens kolumner representerar index för nätets hörnpunkter, varje rad representerar en dataset. Datauppsättningar kan ha olika datatyper. I det här fallet lagras vindhastigheten på 10 m vid en viss tidpunkt (t1, t2, t3).

På liknande sätt kan mesh-datasetet också lagra vektorvärden för varje vertex. Till exempel vindriktningsvektorn vid de givna tidsstämplarna:

Table 14.1: Exempel på nätdataset

10 meter vind	1	2	3	...
10 meters hastighet vid tid=t1	17251	24918	32858	…
10 meters hastighet vid tiden=t2	19168	23001	36418	…
10 meters hastighet vid tiden=t3	21085	30668	17251	…
…	…	…	…	…
10m vindriktning tid=t1	[20,2]	[20,3]	[20,4.5]	…
10m vindriktning tid=t2	[21,3]	[21,4]	[21,5.5]	…
10m vindriktning tid=t3	[22,4]	[22,5]	[22,6.5]	…
…	…	…	…	…

Vi kan visualisera data genom att tilldela värdena färger (på samma sätt som görs med *Singleband pseudocolor* rasterrendering) och interpolera data mellan hörn enligt mesh-topologin. Det är vanligt att vissa storheter är 2D-vektorer i stället för enkla skalärvärden (t.ex. vindriktning). För sådana storheter är det önskvärt att visa pilar som anger riktningarna.

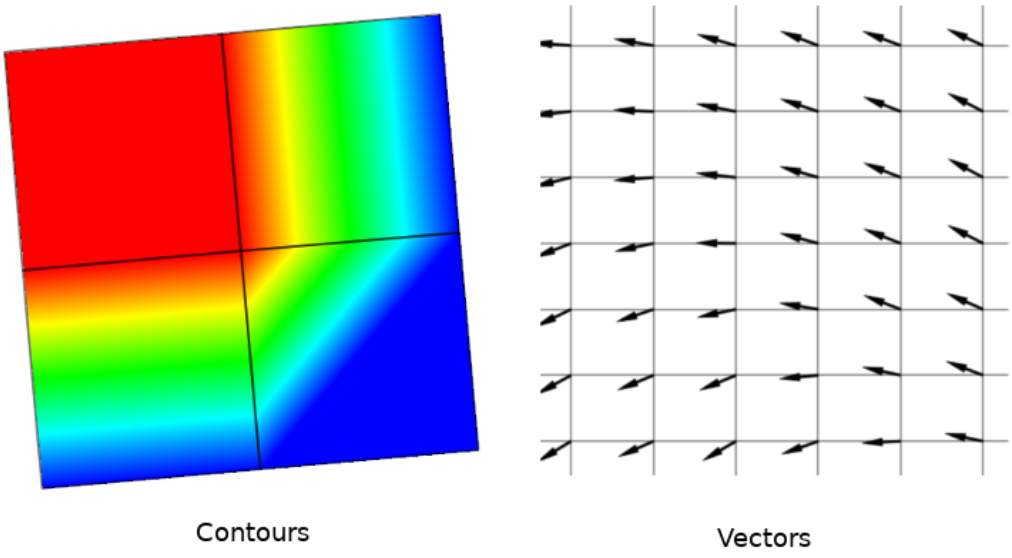


Fig. 14.3: Möjlig visualisering av maskdata

14.2 Format som stöds

QGIS har åtkomst till mesh-data med hjälp av MDAL-drivrutinerna och har inbyggt stöd för många olika format. Huruvida QGIS kan redigera ett mesh-lager beror på formatet och typen av mesh-struktur.

För att ladda en mesh-datauppsättning i QGIS använder du fliken *Mesh* i dialogrutan *Data Source Manager*. Läs *Ladda ett nätlager* för mer information.

14.3 Egenskaper för nätdataset

Dialogrutan *Lageregenskaper* för ett mesh-lager innehåller allmänna inställningar för att hantera lagrets datasetgrupper och deras rendering (aktiva datasetgrupper, symbologi, 2D- och 3D-rendering). Den innehåller också information om lagret.

För att komma till dialogrutan *Lageregenskaper*:

- Dubbelklicka på lagret i panelen *Layers* eller högerklicka och välj *Properties…* från popup-menyn;
- Gå till menyn *Layer* ► *Lageregenskaper…* när lagret är markerat.

Dialogrutan mesh *Lageregenskaper* innehåller följande avsnitt:

Table 14.2: Flikar för egenskaper för nätlager

Information	*Källa*	*Symbologi*[1]
3D-vy[1]	*Temporal*	*Labels*[1]
Elevation	*Rendering*	*Metadata*

[1] Finns även i *Layer stylingpanel*

Observera

De flesta av egenskaperna för ett mesh-lager kan sparas i eller laddas från en `.qml` med hjälp av *Style*-menyn längst ned i dialogrutan. Mer information finns på *Hantera anpassade stilar*.

14.3.1 Informationsegenskaper

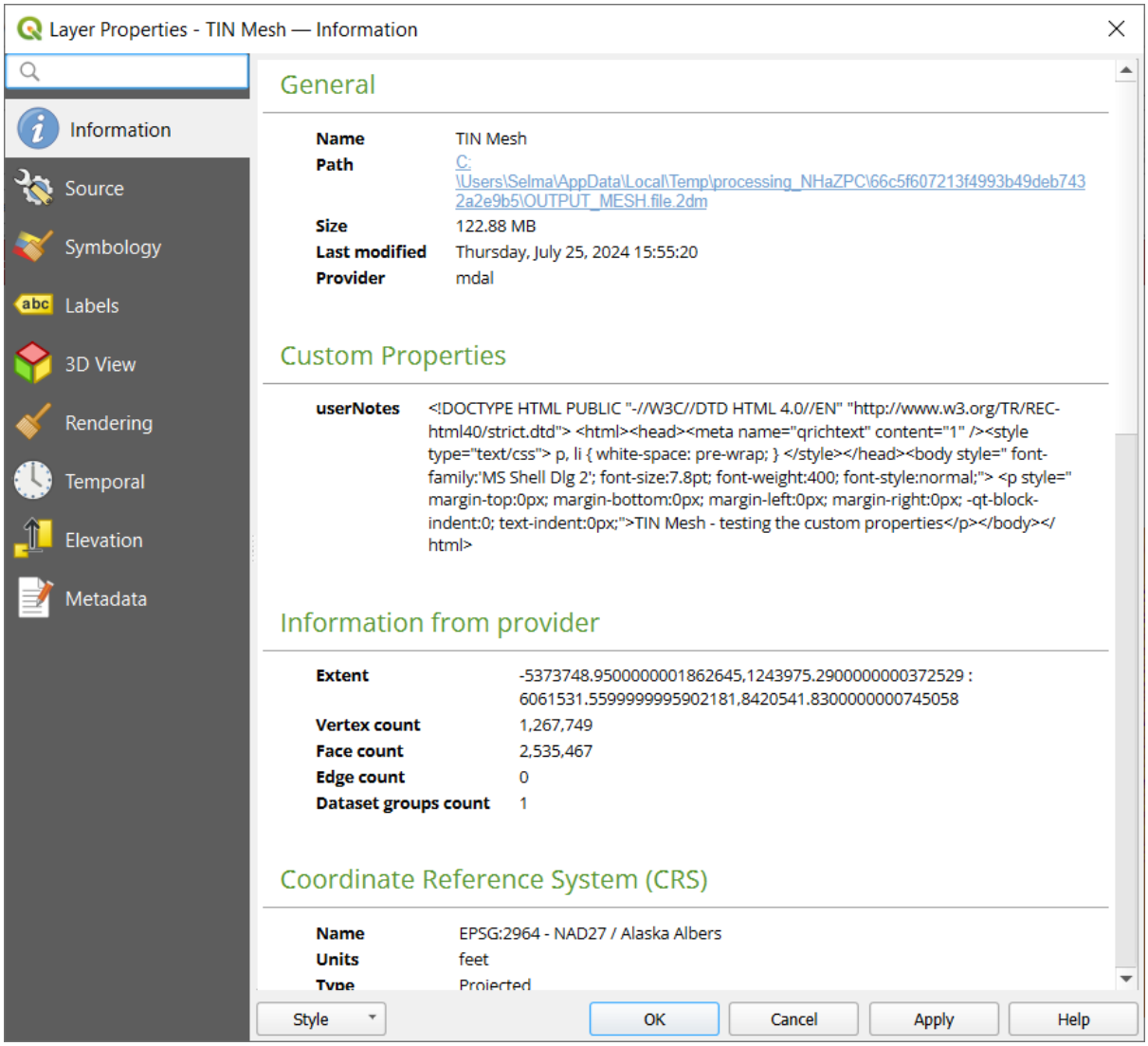


Fig. 14.4: Egenskaper för information om nätlager

Fliken *Information* är skrivskyddad och utgör en intressant plats för att snabbt hämta sammanfattad information och metadata om det aktuella lagret. Tillhandahållen information är:

- allmänna uppgifter som namn i projektet, källsökväg, lista över hjälpfiler, senaste spartid och storlek, den använda leverantören
- anpassade egenskaper som används för att lagra ytterligare information om lagret i det aktiva projektet. Standardanpassade egenskaper kan inkludera *layer notes*. Fler egenskaper kan skapas och hanteras med hjälp av PyQGIS, särskilt genom metoden setCustomProperty().
- baserat på lagrets leverantör: utsträckning, vertex, yta, kanter och/eller antal datasetgrupper
- det koordinatsatta referenssystemet: namn, enheter, metod, noggrannhet, referens (dvs. om det är statiskt eller dynamiskt)
- hämtat från fylld *metadata*: access, extents, länkar, kontakter, historik…

14.3.2 Källegenskaper

Fliken *Källa* visar grundläggande information om det valda nätet, inklusive:

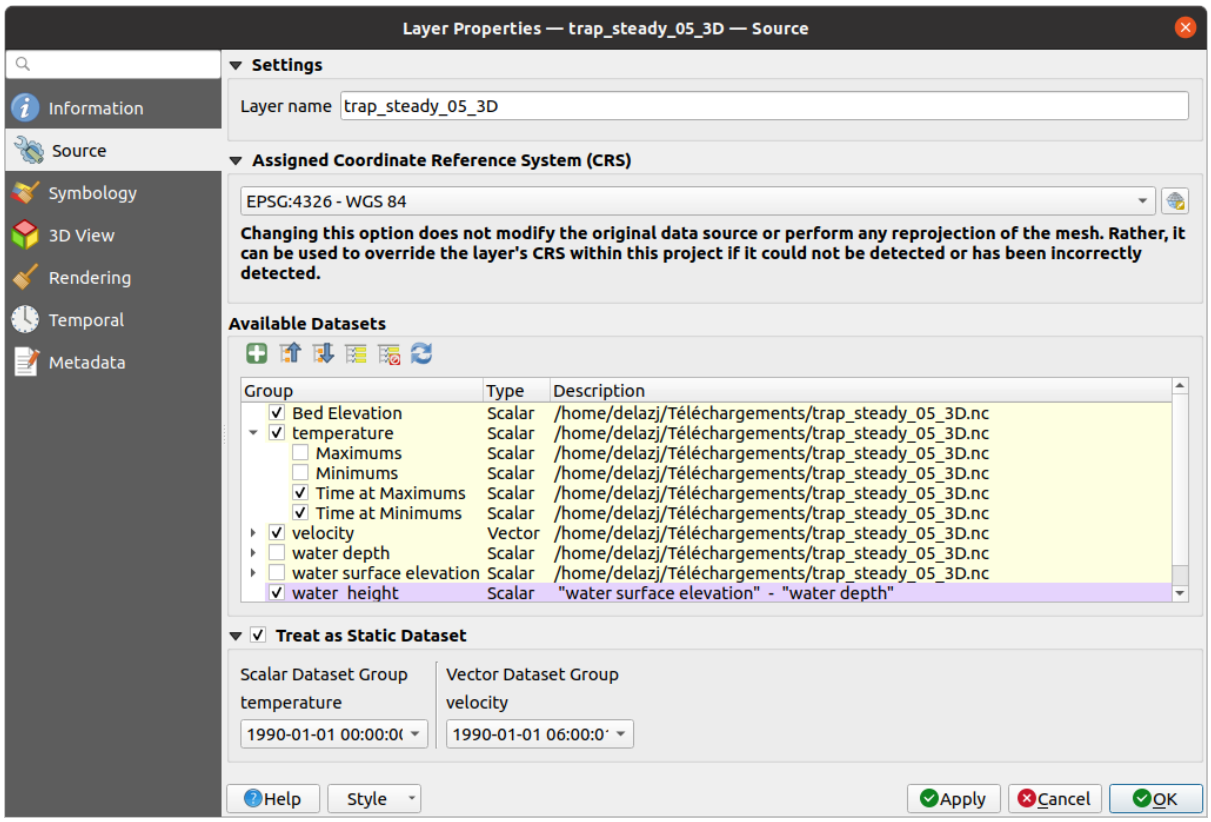


Fig. 14.5: Egenskaper för Mesh Layer Source

- det lagernamn som ska visas i *Layers*-panelen
- inställning av koordinatreferenssystem: Visar lagrets *Assigned Coordinate Reference System (CRS)*. Du kan ändra lagrets CRS genom att välja ett nyligen använt i listrutan eller klicka på *Välj CRS*-knappen (se *Väljare för referenssystem för koordinater*). Använd denna process endast om det CRS som används för lagret är fel eller om inget CRS har använts.
- I ramen *Available datasets* listas alla datasetgrupper (och undergrupper) i mesh-lagret, med typ och beskrivning i en trädvy. Både vanliga datamängder (dvs. deras data lagras i filen) och virtuella datamängder (som *beräknas i farten*) listas.
  - Använd knappen *Assign extra dataset to mesh* för att lägga till fler grupper i det aktuella mesh-lagret.
  - *Collapse all* och *Expand all* datasetets träd, i händelse av inbäddade grupper
  - Om du är intresserad av ett fåtal dataset kan du avmarkera de andra och göra dem otillgängliga i projektet
  - Dubbelklicka på ett namn så kan du byta namn på datasetet.
  - *Reset to defaults*: kontrollerar alla grupper och byter namn på dem tillbaka till deras ursprungliga namn i leverantören.
  - Högerklicka på en virtuell datasetgrupp så kan du:
    - *Remove dataset group* från projektet
    - *Spara datasetgruppen som…* en fil på disken, i valfritt format. Den nya filen hålls tilldelad det aktuella nätlagret i projektet.

- Om du markerar gruppen *Treat as static dataset* kan du ignorera egenskaperna *map temporal navigation* när du renderar nätlagret. För varje aktiv datasetgrupp (som valts i fliken *Symbologi* ► *Datasets*) kan du:
  - inställd på *None*: datasetgruppen visas inte alls
  - *Display dataset*: t.ex. för datasetet ”bed elevation” som inte är tidsmedvetet
  - extrahera ett visst datum och en viss tid: den datauppsättning som matchar den angivna tiden återges och förblir fixerad under kartnavigering.

14.3.3 Egenskaper för symbologi

Klicka på knappen *Symbology* för att aktivera dialogrutan. Symbologiegenskaperna är indelade i flera flikar:

- *Datauppsättningar*
- *Contours*
- *Vektorer*
- *Rendering*
- *Metod för medelvärdesberäkning av staplade nät*

Datamängder

Fliken Datasets är den viktigaste platsen för att kontrollera och ställa in vilka dataset som ska användas för lagret. Den presenterar följande objekt:

- *Grupper* tillgängliga i mesh-datasetet, med angivande av om de tillhandahåller:
  - skalär dataset
  - eller vektordatauppsättning: som standard har varje vektordatauppsättning en skalärdatauppsättning som representerar dess magnitud automatiskt genererad.

  Klicka på ikonen bredvid datasetnamnet för att välja vilken grupp och typ av data som ska representeras.
- *Metadata för valda datasetgrupp(er)*, med detaljer om:
  - typ av nät: kanter eller ytor
  - datatyp: toppar, kanter, ytor eller volym
  - oavsett om den är av vektortyp eller inte
  - det ursprungliga namnet i mesh-lagret
  - enheten, om tillämpligt
- *blending mode* tillgänglig för de valda dataseten.

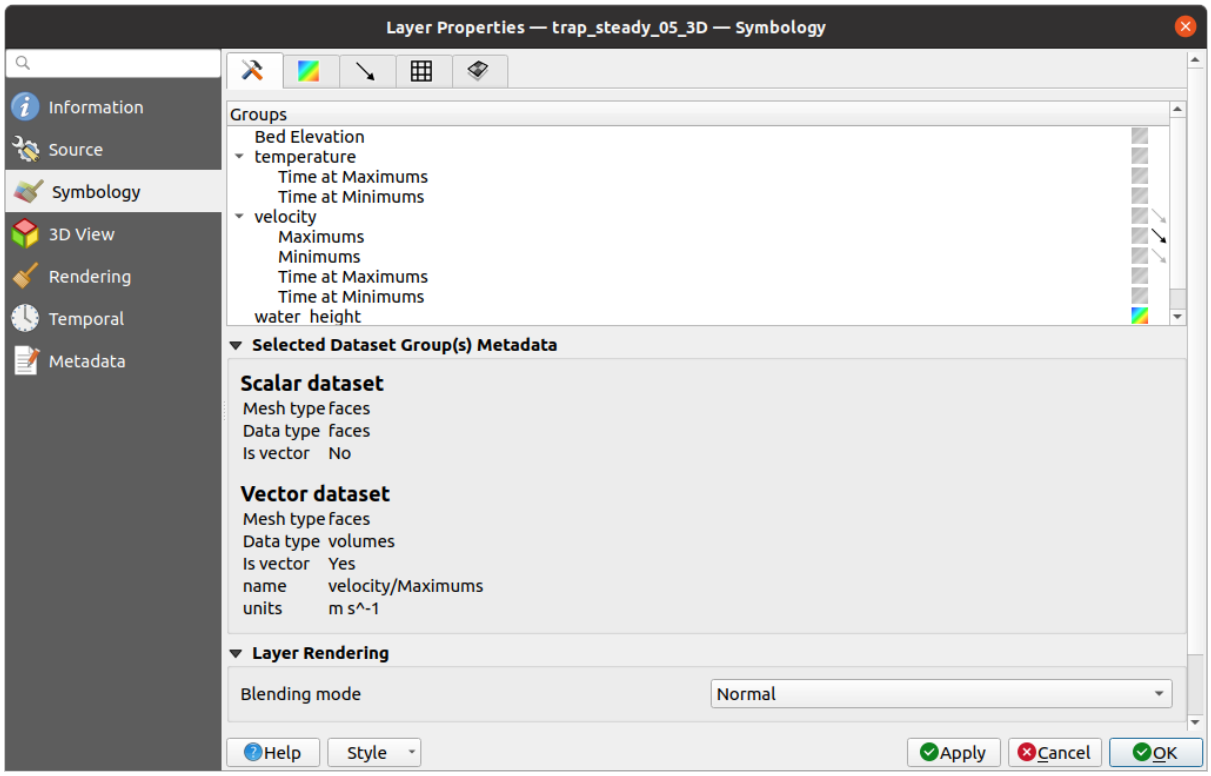


Fig. 14.6: Dataset för nätlager

Du kan tillämpa symbologi på den valda vektor- och/eller skalärgruppen med hjälp av följande flikar.

Konturer Symbologi

Observera

Fliken Contours kan endast aktiveras om en skalär dataset har valts i fliken *Datasets*.

I fliken Contours kan du se och ändra de aktuella visualiseringsalternativen för konturer för den valda gruppen, vilket visas i Fig. 14.7 nedan:

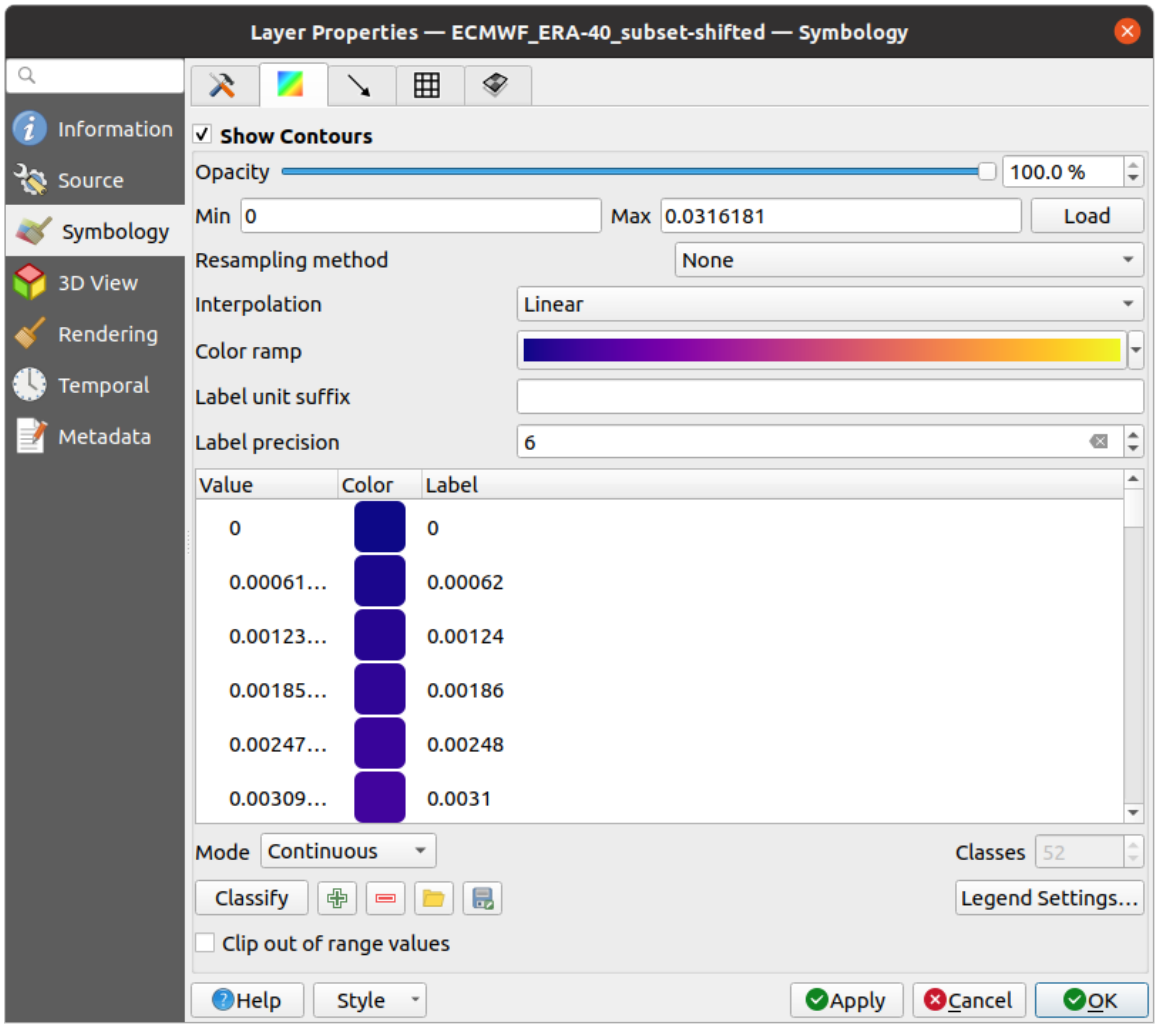


Fig. 14.7: Styling av konturer i ett mesh-lager

- För 1D-mesh, ställ in *Stroke width* för kanterna. Detta kan vara en fast storlek för hela datasetet, eller variera längs geometrin (mer detaljer med *interpolated line renderer*)
- Använd skjutreglaget eller snurrboxen för att ställa in *Opacity* för den aktuella gruppen, om den är av typen 2D mesh.
- Ange det värdeintervall som du vill representera i den aktuella gruppen: använd Load för att hämta min- och maxvärdena i den aktuella gruppen eller ange egna värden om du vill utesluta vissa.
- För 2D/3D-mesh, välj *Resampling method* för att interpolera värdena på de omgivande hörnen till ytorna (eller från de omgivande ytorna till hörnen) med hjälp av *Neighbour average*-metoden. Beroende på om datasetet är definierat på vertikalerna (respektive på ytorna), har QGIS denna inställning som standard till *None* (respektive *Neighbour average*) för att använda värden på vertikalerna och behålla standardrenderingen jämn.
- Klassificera datasetet med hjälp av *color ramp shader*-klassificeringen.

Vektorer Symbologi

Observera

Fliken Vectors kan endast aktiveras om ett vektordataset har valts på fliken *Datasets*. Klicka på ikonen Vectors på höger sida av fliken *Datasets*.

På fliken Vectors kan du se och ändra de aktuella visualiseringsalternativen för vektorer för den valda gruppen, som visas i Fig. 14.8:

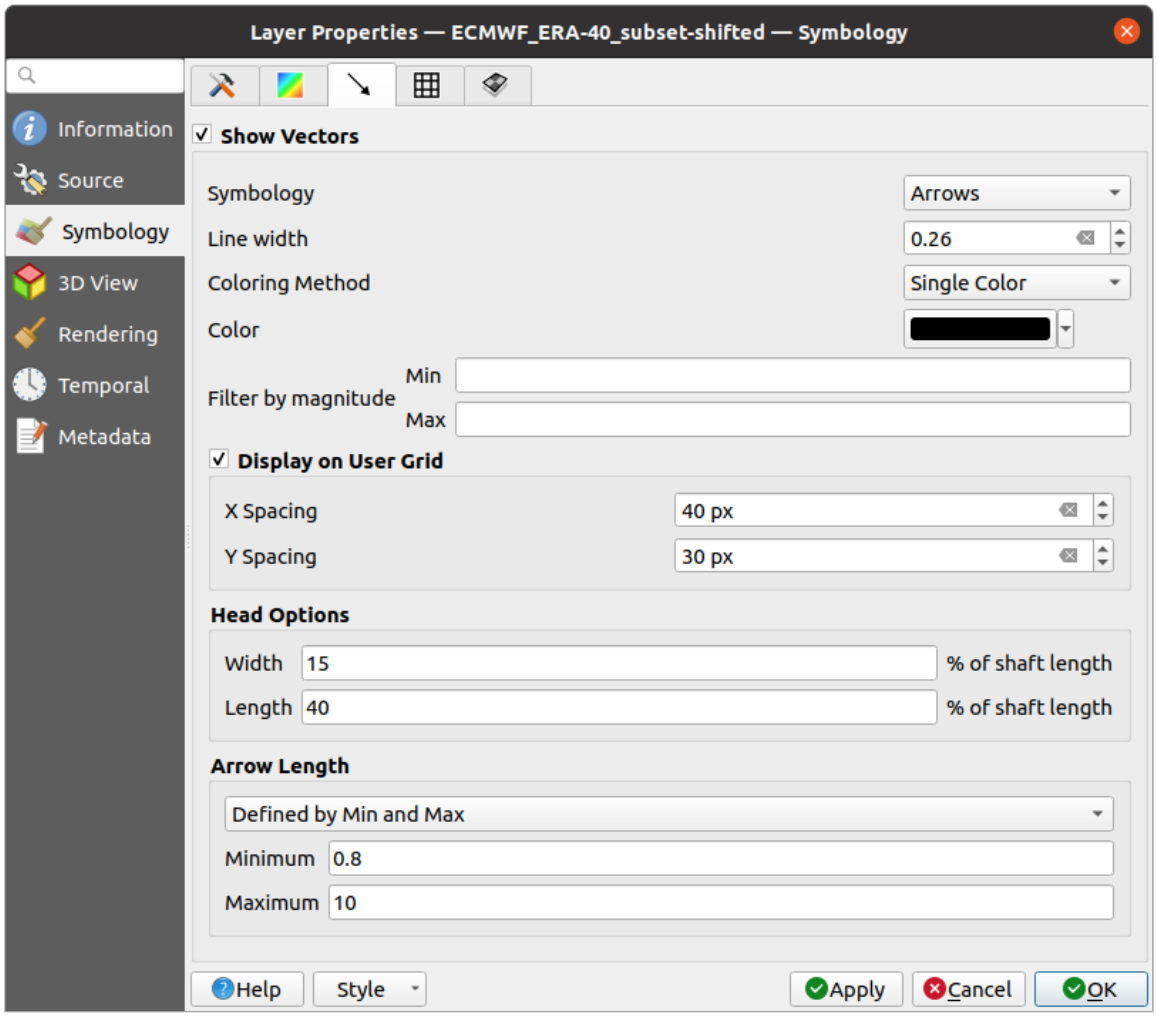


Fig. 14.8: Styling av vektorer i ett mesh-lager med pilar

Mesh-vektordataset kan utformas med hjälp av olika typer av *Symbologi*:

- **Pilar**: vektorer representeras med pilar på samma plats som de definieras i rådatasetet (dvs. på noderna eller i mitten av elementen) eller på ett användardefinierat rutnät (de är alltså jämnt fördelade). Pilens längd är proportionell mot pilens storlek enligt definitionen i rådata, men kan skalas med olika metoder.
- **Streamlines**: vektorer representeras med strömlinjer som utgår från startpunkter. Startpunkterna kan utgå från nätets hörn, från ett användarnät eller slumpmässigt.
- **Traces**: en snyggare animation av strömlinjerna, den typ av effekt du får när du slumpmässigt kastar sand i vattnet och ser var sandföremålen flyter.

- **Vindpinnar**: vektorer representeras med vindpinnar, ett vanligt sätt att representera vindhastighet och riktning.

Vilka egenskaper som är tillgängliga beror på den valda symbologin enligt följande tabell.

Table 14.3: Tillgänglighet och innebörd av egenskaperna för symbologin vektorer

Label	Beskrivning och egenskaper	Pil	Streamlines	Spår	Wind Barbs
Linjebredd	Bredd på vektorrepresentati	☑	☑	☑	☑
Färgningsmetod	• en *Enkel färg* som tilldelas alla vektorer • eller en variabel färg baserad på vektorstorle med hjälp av en *Color ramp shader*	☑	☑	☑	☑
Filtrera efter magnitud	Endast vektorer vars längd för den valda datauppsättningen ligger inom intervallen *Min* och *Max* visas	☑	☑		☑
Visa på användarens rutnät	Placerar vektorn på ett rutnät med anpassade *X spacing* och *Y spacing* och interpolerar deras längd baserat på grannar	☑	☑		☑
Huvudalternativ	*Längd* och *Bredd* på pilspetsen, som en procentandel av dess skaftlängd	☑			
Pilens längd	• **Definieras av Min och Max**: Du anger den minsta och största längden för pilarna, QGIS	☑			

Rendering

På fliken Rendering erbjuder QGIS möjligheter att visa och anpassa nätstrukturen. *Line width* och *Line color* kan ställas in för att representera:

- kanterna för 1D-mesh
- För 2D-mesh:
  - *Native mesh rendering*: visar originalytor och kanter från lagret
  - *Triangular mesh rendering*: lägger till fler kanter och visar ytorna som trianglar

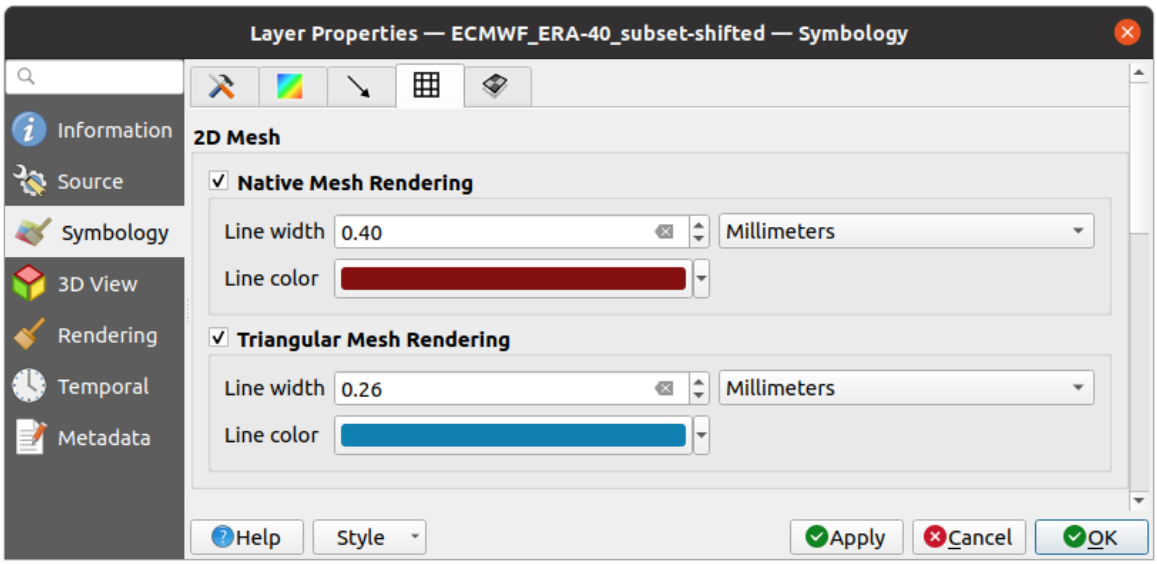


Fig. 14.9: rendering av 2D-nät

Metod för medelvärdesberäkning av staplade nät

3D-skiktade nät består av flera staplade ostrukturerade 2D-nät som var och en extruderas i vertikal riktning (`nivåer`) med hjälp av en vertikal koordinat. Topparna och ytorna har samma topologi i varje vertikal nivå. Värdena lagras vanligen på volymer som regelbundet staplas över basens 2D-maskor. För att kunna visualisera dem på en 2D-bild måste du konvertera värden på volymer (3d) till värden på ytor (2d) som kan visas i mesh-lagret. Metoden Stacked mesh averaging method tillhandahåller olika metoder för medelvärdesberäkning/interpolation för att hantera detta.

Du kan välja metod för att härleda 2D-dataset och motsvarande parametrar (nivåindex, djup- eller höjdvärden). För varje metod visas ett exempel på tillämpning i dialogrutan, men du kan läsa mer om metoderna på https://fvwiki.tuflow.com/Depth_Averaging_Results.

14.3.4 Etiketter Egenskaper

Fliken *Labels* ger dig dynamisk märkning av mesh-vertikaler och native mesh-ytor baserat på geometriska egenskaper och anpassade uttryck. Denna dialogruta kan också nås från panelen *Layer Styling*.

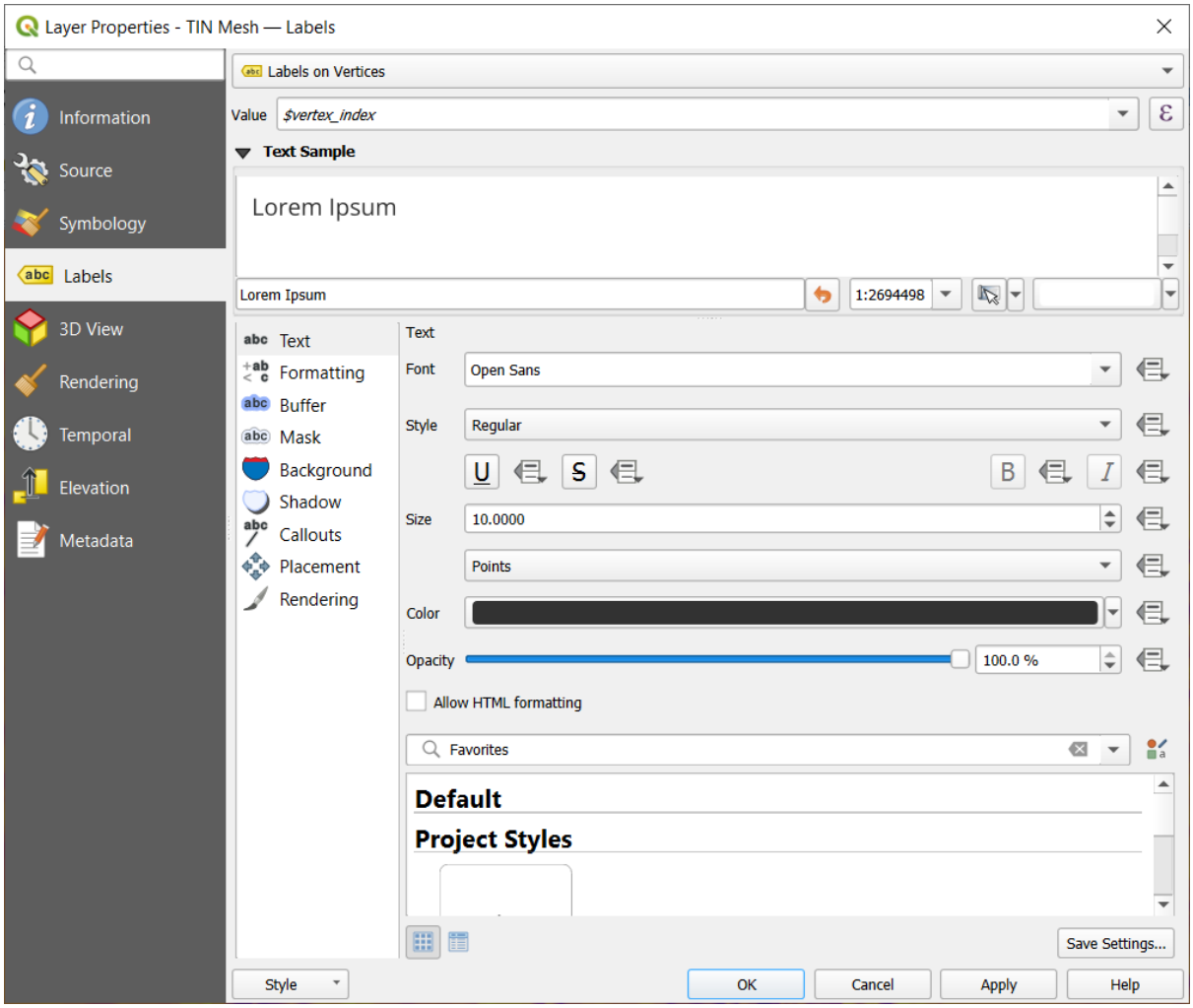


Fig. 14.10: Egenskaper för nätetiketter

Välj märkningsmetod från rullgardinsmenyn. Tillgängliga metoder är:

- *No labels*: standardvärdet, visar inga etiketter från lagret.
- *Märker på hörn*: märker nätets hörn som en punktfunktion.
- *Märker på ytor*: märker nätets ytor som en polygonfunktion.

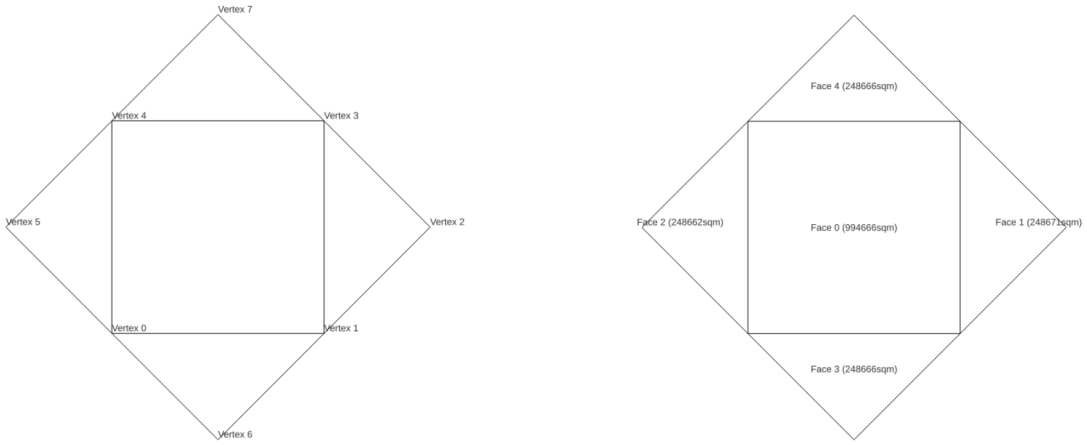


Fig. 14.11: Metoder för märkning av nät

När du har valt märkningsmetod öppnar du Expression dialog och väljer en *mesh-funktion*. Du kan kombinera meshes-funktioner med andra funktioner från *listan* för att ändra etiketten.

Nedan visas alternativ för att anpassa etiketterna, under olika flikar:

- *Text*
- *Formatering*
- *Buffert*
- *Mask*
- *Bakgrund*
- *Skugga*
- *Callouts*
- *Placering*
- *Rendering*

Beskrivning av hur man ställer in varje egenskap finns på *Ställa in en etikett*.

14.3.5 egenskaper för 3D-vy

Mesh-lager kan användas som *terrain i en 3D-kartvy* baserat på deras vertikala Z-värden. Från egenskapsfliken *3D View* är det också möjligt att rendera mesh-lagrets dataset i samma 3D-vy. Därför kan den vertikala komponenten i vertikalerna ställas in lika med datasetvärden (t.ex. vattenytans nivå) och mesh-strukturen kan ställas in för att återge andra datasetvärden med färgskuggning (t.ex. hastighet).

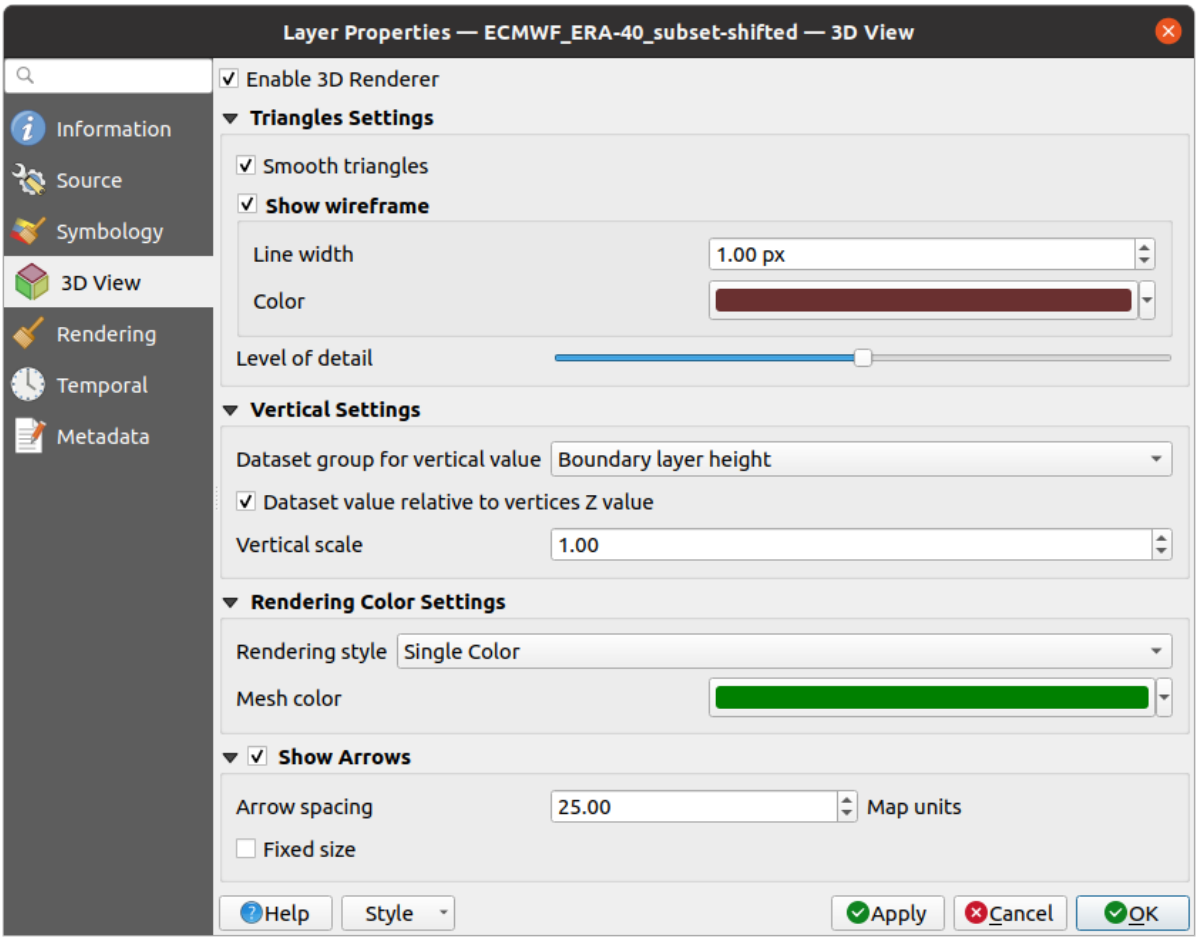


Fig. 14.12: Mesh-dataset 3D-egenskaper

Markera *Enable 3D Renderer* och du kan redigera följande egenskaper:

- Under *Triangelinställningar*
  - *Utjämnade trianglar*: Vinklar mellan på varandra följande trianglar jämnas ut för en bättre 3D-rendering
  - *Show wireframe* där du kan ställa in *Line width* och *Color*
  - *Detaljnivå*: Styr hur *förenklat* mesh-lagret som ska renderas ska vara. Längst till höger är det basmeshen, och ju mer du går åt vänster, desto mer förenklas lagret och återges med färre detaljer. Detta alternativ är endast tillgängligt om alternativet *Simplify mesh* under fliken *Rendering* är aktiverat.
- *Vertikala inställningar* för att styra beteendet för den vertikala komponenten i topparna i renderade trianglar:
  - *Datasetgrupp för vertikalt värde*: den datasetgrupp som ska användas för maskans vertikala komponent
  - *Dataset-värde i förhållande till hörnens Z-värde*: om dataset-värdena ska betraktas som absoluta Z-koordinater eller i förhållande till hörnens ursprungliga Z-värde
  - *Vertical scale*: den skalfaktor som ska tillämpas på datasetets Z-värden
- *Rendering color settings* med en *Rendering style* som kan baseras på den färgrampshader som anges i *Konturer Symbologi* (*2D contour color ramp shader*) eller som en *Single color* med en associerad *Mesh color*
- *Visa pilar*: visar pilar på nätlagrets dataset 3D-enhet, baserat på samma vektor-datasetgrupp som används i *vector 2D rendering*. De visas med hjälp av 2D-färginställningen. Det är också möjligt att definiera *Pilavstånd* och om det är en *Fixerad storlek* eller skalad på magnitud. Denna inställning för avstånd definierar också maxstorleken för pilar eftersom pilar inte kan överlappa varandra.

14.3.6 Rendering av egenskaper

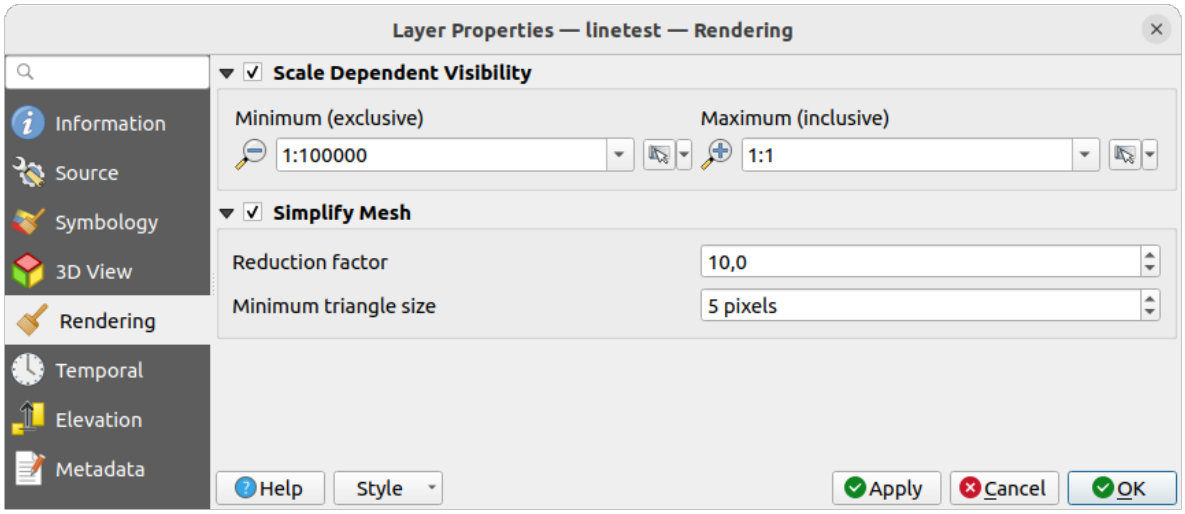


Fig. 14.13: Egenskaper för rendering av nät

I grupprutan *Scale dependent visibility* kan du ställa in *Maximum (inclusive)* och *Minimum (exclusive)* skala, vilket definierar ett skalområde inom vilket mesh-element kommer att vara synliga. Utanför detta intervall är de dolda. Knappen Set to current canvas scale hjälper dig att använda den aktuella kartans canvasskala som gräns för synlighetsområdet. Se *Väljare för synlighetsskala* för mer information.

Observera

Du kan också aktivera skalberoende synlighet på ett lager från panelen *Layers*: högerklicka på lagret och välj *Set Layer Scale Visibility* i snabbmenyn.

Eftersom mesh-lager kan ha miljontals ytor kan renderingen av dem ibland vara mycket långsam, särskilt när alla ytor visas i vyn trots att de är för små för att kunna ses. För att snabba upp renderingen kan du förenkla mesh-lagret, vilket resulterar i ett eller flera nät som representerar olika *detaljnivåer* och välja på vilken detaljnivå du vill att QGIS ska rendera mesh-lagret. Observera att det förenklade nätet endast innehåller triangulära ytor.

På fliken *Rendering*, markera *Simplify mesh* och ställ in:

- a *Reduktionsfaktor*: Styr genereringen av successiva nivåer av förenklade nät. Om t.ex. basnätet har 5 miljoner ytor och reduktionsfaktorn är 10, kommer det första förenklade nätet att ha ungefär 500 000 ytor, det andra 50 000 ytor, det tredje 5000, … Om en högre reduktionsfaktor snabbt leder till enklare nät (dvs. med trianglar av större storlek), ger den också färre detaljnivåer.
- *Minsta triangelstorlek*: den genomsnittliga storleken (i pixlar) på de trianglar som får visas. Om nätets genomsnittliga storlek är mindre än detta värde utlöses rendering av ett nät med lägre detaljnivå.

14.3.7 Temporala egenskaper

Fliken *Temporal* ger alternativ för att styra återgivningen av lagret över tid. Det gör det möjligt att dynamiskt visa temporala värden för de aktiverade datasetgrupperna. En sådan dynamisk rendering kräver att *temporal navigation* är aktiverad över kartans canvas.

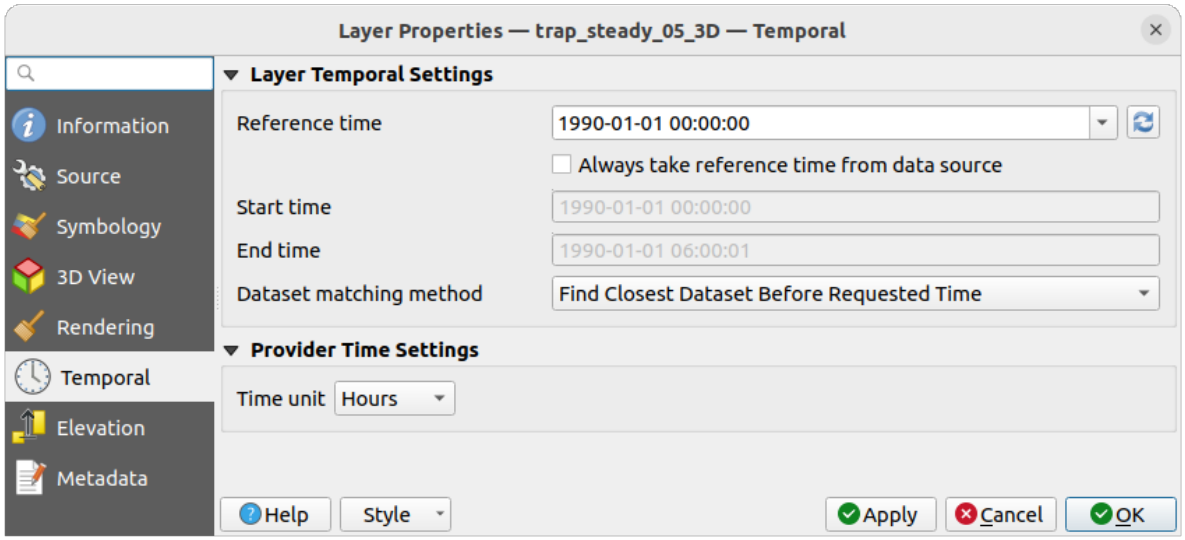


Fig. 14.14: Mesh Temporala egenskaper

Temporala inställningar för lager

- *Reference time* i datasetgruppen, som ett absolut tidsdatum. Som standard analyserar QGIS källagret och returnerar den första giltiga referenstiden i lagrets datasetgrupp. Om den inte är tillgänglig kommer värdet att ställas in av projektets tidsintervall eller falla tillbaka till aktuellt datum. De *Starttid* och *Sluttid* som ska beaktas beräknas sedan baserat på det interna tidsstämpelsteget i datasetet.

  Det är möjligt att ange en anpassad *Reference time* (och sedan tidsintervallet) och återgå till ändringarna med hjälp av knappen Reload from provider. Om du markerar kryssrutan *Alltid ta referenstid från datakällan* säkerställer du att tidsegenskaperna uppdateras från filen varje gång lagret laddas om eller projektet öppnas på nytt.

- *Dataset matching method*: avgör vilken dataset som ska visas vid den angivna tidpunkten. Alternativen är *Find closest dataset before requested time* eller *Find closest dataset from requested time (after or before)*.

Leverantörens tidsinställningar

- *Tidsenhet* extraherad från rådata, eller användardefinierad. Detta kan användas för att anpassa mesh-lagrets hastighet till andra lager i projektet under karttidsnavigering. Enheter som stöds är *Seconds*, *Minutes*, *Hours* och *Days*.

14.3.8 Elevation Egenskaper

På fliken *Elevation* finns alternativ för att styra lagrets höjdegenskaper i en *3D-kartvy* och *2D-kartvy* och dess utseende i *profile tool charts*. Du kan särskilt konfigurera hur höjder från din datauppsättning ska tolkas:

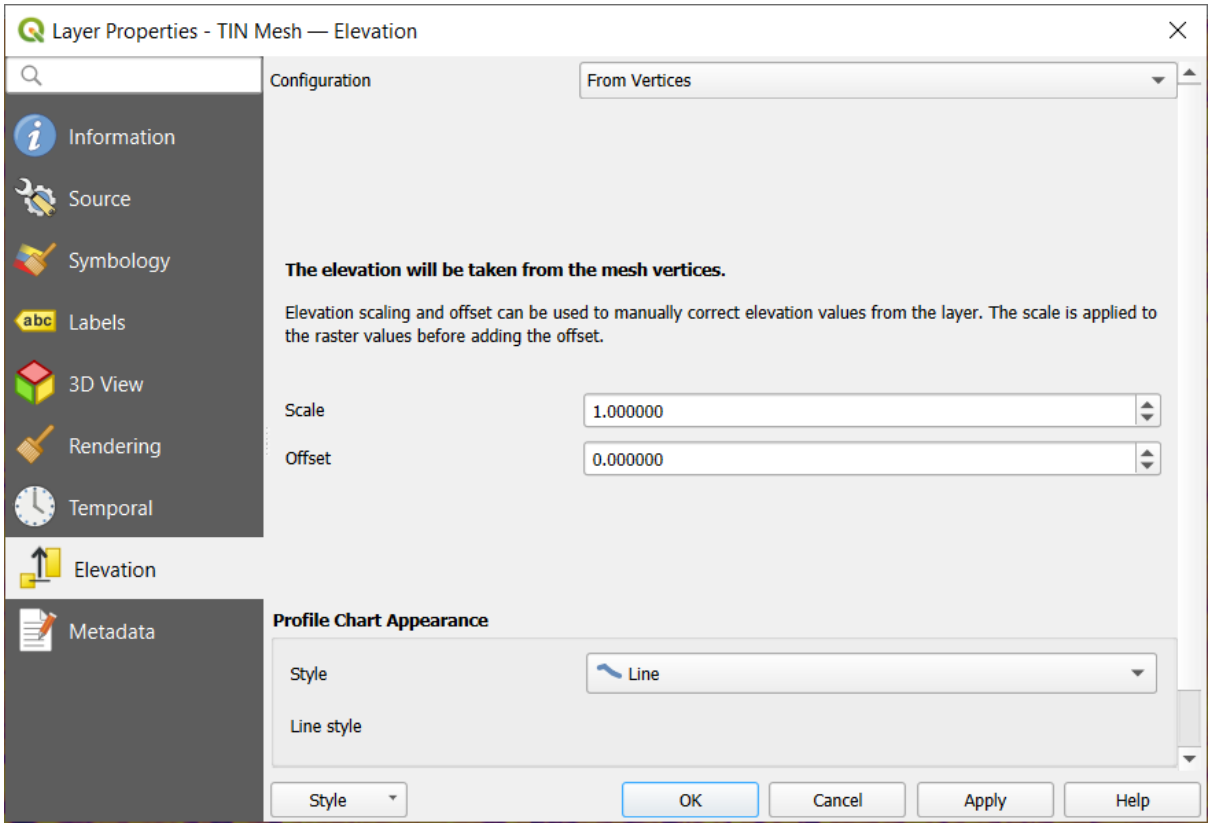


Fig. 14.15: Egenskaper för Mesh Elevation

- *From vertices*: hur Z-värdena för nätlagrets vertikaler ska tolkas som terränghöjd. Du kan använda en *Scale*-faktor och en *Offset*. Den här inställningen är tillgänglig för *3D map view* och *profile tool charts*.
- *Fixed Elevation Range*: nätlagret är kopplat till ett fast höjdintervall. Detta läge är tillämpligt när ett lager antingen har en enda fast höjd eller ett intervall (skiva) av höjdvärden. Om ett intervall anges kommer mesh-värdena att extruderas över detta intervall. Du kan ställa in *Lower* och *Upper* höjdintervallvärden för lagret och ange om de nedre eller övre *Limits* är inkluderande eller exklusiva. När lagret är aktiverat kommer det endast att vara synligt i *höjdfiltrerade 2D-kartor* när lagrets intervall ingår i kartans Z-intervall.
- *Fixed Elevation Range Per Group*: varje grupp i mesh-lagret är associerad med ett fast höjdintervall. Detta läge kan användas när ett lager har höjddata som exponeras genom olika datasetgrupper. Den här funktionen visas som en användarredigerbar tabell för datasetgrupper med lägre och högre värden. Du kan antingen fylla i de nedre och övre värdena manuellt eller använda ett ε *Expression* för att automatiskt fylla i alla gruppvärden baserat på ett uttryck. När detta är aktiverat filtreras lagret i *2D-kartvyn* och visar endast värden inom kartfiltrets intervall. I *3D map view* eller *profile tool charts* visas lagrets hela utsträckning, utan hänsyn till filtreringen.
- *Profile Chart Appearance*: styr återgivningen av nätelementens höjd i profildiagrammet. Profilen *Style* kan ställas in som:
  - en *Linje* med en specifik *Linjestil*
  - en höjdyta som återges med en fyllnadssymbol antingen ovanför (*Fill above*) eller nedanför (*Fill below*) höjdkurvans linje. Ytsymboliken representeras med hjälp av:
    - en *Fyllningsstil*
    - och en *Limit*: den maximala (respektive minimala) höjden som bestämmer hur hög fyllnadsytan kommer att vara

14.3.9 Egenskaper för metadata

Fliken *Metadata* ger dig alternativ för att skapa och redigera en metadatarapport om ditt lager. Se *Metadata* för mer information.

14.4 Redigera ett mesh-lager

QGIS gör det möjligt att *skapa ett mesh-lager* från grunden eller baserat på ett befintligt. Du kan skapa/ändra geometrierna i det nya lagret som du kan tilldela dataset efteråt. Det är också möjligt att redigera ett befintligt mesh-lager. Eftersom redigeringsoperationen kräver ett lager med endast ramar kommer du att bli ombedd att antingen ta bort alla associerade dataset först (se till att du har dem tillgängliga om de fortfarande behövs) eller skapa en kopia (endast geometrier) av lagret.

Observera

QGIS tillåter inte att kanter digitaliseras på mesh-lager. Endast hörn och ytor är mesh-element som kan skapas. Alla mesh-format som stöds kan inte heller redigeras i QGIS (se permissions).

14.4.1 Översikt över digitaliseringsverktygen för nät

Följande verktyg finns tillgängliga för att interagera med eller redigera ett element i basnätskiktet.

Table 14.4: Verktyg för digitalisering av nät

Label	Syfte	Plats
Aktuella redigeringar	Möjlighet att spara, rulla tillbaka eller avbryta ändringar i alla eller valda lager samtidigt	*Digitalisering* verktygsfält
Toggle för att redigera	Aktivera/avaktivera lagret i redigeringsläget	*Digitalisering* verktygsfält
Spara redigeringar	Spara ändringar som gjorts i lagret	*Digitalisering* verktygsfält
Ångra	Ångra den senaste ändringen (ändringarna) - `Ctrl+Z`	*Digitalisering* verktygsfält
Gör om	Gör om de senast ogjorda åtgärderna - `Ctrl+Shift+Z`	*Digitalisering* verktygsfält
Aktivera avancerade digitaliserings	Slå på/av *Advanced Digitizing Panel*	Verktygsfältet *Avancerad digitalisering*
Reindexera ansikten och vertikaler	Återskapa index och numrera om mesh-elementen för optimering	*Mesh* meny
Digitisera nätelement	Välj/skapa hörn och ytor	*Mesh* *Digitalisering* verktygsfält
Välj nätelement efter polygon	Välj hörnpunkter och ytor som överlappas av en ritad polygon	*Mesh* *Digitalisering* verktygsfält
Välj nätelement efter uttryck	Välj toppar och ytor med hjälp av ett uttryck	*Mesh* *Digitalisering* verktygsfält
Transformera vertikaler koordinate	Ändra koordinaterna för ett urval av hörnpunkter	*Mesh* *Digitalisering* verktygsfält
Force efter valda geometrier	Dela ytor och begränsa Z-värdet med hjälp av en linjär geometri	*Mesh* *Digitalisering* verktygsfält

14.4.2 Utforska logiken för tilldelning av Z-värde

När ett mesh-lager försätts i redigeringsläge öppnas en widget *Vertex Z-värde* uppe till höger på kartbilden. Som standard motsvarar dess värde det *Default Z value* som ställts in i :menuselection:` *Inställningar* –> *Alternativ* –> *Digitizing* fliken. När det finns valda hörnpunkter visar widgeten det genomsnittliga Z-värdet för de valda hörnpunkterna.

Under redigeringen tilldelas *Vertex Z value* till nya vertikaler. Det är också möjligt att ställa in ett eget värde: redigera widgeten, tryck på `Enter` så åsidosätts standardvärdet och det nya värdet används i digitaliseringsprocessen. Klicka på ikonen i widgeten för att återställa dess värde till Options standardvärde.

Regler för överlåtelse

När du **skapar** en ny vertex kan definitionen av dess Z-värde variera beroende på det aktiva valet i mesh-lagret och dess placering. I följande tabell visas de olika kombinationerna.

Table 14.5: Matris för tilldelning av Z-värde till ny toppunkt

Skapande av vertex	Finns det markerade vertikaler i mesh-lagret?	Källa till det tilldelade värdet	Tilldelat Z-värde
”Fri” vertex, inte kopplad till någon yta eller kant på en yta	Nej	*Vertex Z-värde*	Standard eller användardefinierad
		Advanced Digitizing Panel (om *z* widget är i Locked tillstånd)	*z* widget if in Låst tillstånd
	Ja	*Vertex Z-värde*	Medelvärde av de valda topparna
Toppunkt på en kant	—	Nätlager	Interpolerad från kantens hörnpunkter
Toppunkt på en yta	—	Nätlager	Interpolerad från ansiktets hörnpunkter
Vertex snäppt till en 2D-vektorfunktion	—	*Vertex Z-värde*	Standard eller användardefinierad
Vertex snappad till en 3D-vektorvertex	—	Ett vektorlager	Toppunkt
Vertex knäppt till ett 3D-vektorsegment	—	Ett vektorlager	Interpolerad längs vektorsegmentet

Observera

Widgeten *Vertex Z value* inaktiveras om *Advanced Digitizing Panel* är aktiverad och inget mesh-element är valt. Den senares widget *z* styr då tilldelningen av Z-värdet.

Ändra Z-värdet för befintliga hörn

Det enklaste sättet att ändra Z-värdet för toppar är:

1. Välj en eller flera hörn. Widgeten *Vertex Z value* visar den genomsnittliga höjden för urvalet.
2. Ändra värdet i widgeten.
3. Tryck på `Enter`. Det inmatade värdet tilldelas hörnpunkterna och blir standardvärde för nästa hörnpunkt.

Ett annat sätt att ändra Z-värdet för en vertex är att flytta och snäppa den på en vektorlagerfunktion med Z-värdesfunktion. Om mer än en vertex är markerad kan Z-värdet inte ändras på det här sättet.

I dialogrutan *Transform mesh vertices* kan du också ändra Z-värdet för ett urval av hörnpunkter (tillsammans med deras X- eller Y-koordinater).

14.4.3 Välja nätelement

Använda *Digitize Mesh Elements* (Digitalisera nätelement)

Aktivera verktyget Digitize Mesh Elements. Håll muspekaren över ett element så markeras det och du kan välja det.

- Klicka på en vertex så blir den markerad.
- Klicka på den lilla fyrkanten i mitten av en yta eller en kant så blir den markerad. Anslutna hörnpunkter markeras också. Omvänt gäller att om du markerar alla hörn på en kant eller en yta markeras även det elementet.
- Dra en rektangel för att markera överlappande element (en markerad yta har alla sina hörnpunkter). Tryck på `Alt` om du bara vill välja element som är helt inneslutna.
- Om du vill lägga till element i en markering trycker du på `Shift` medan du markerar dem.
- Om du vill ta bort ett element från markeringen trycker du på `Ctrl` och markerar det igen. En bortvald yta kommer också att avmarkera alla sina hörn.

Använda *Välj nätelement efter polygon*

Aktivera verktyget :sup:`Välj nätelement efter polygon` och:

- Rita en polygon (vänsterklicka för att lägga till vertex, `Backspace` för att ångra den sista vertexen, `Esc` för att avbryta polygonen och högerklicka för att validera den) över mesh-geometrierna. Alla delvis överlappande hörn och ytor kommer att väljas. Tryck på `Alt` medan du ritar om du bara vill välja helt inneslutna element.
- Högerklicka över geometrin för ett vektorlagers funktion, välj den i listan som dyker upp och alla delvis överlappande hörn och ytor i mesh-lagret kommer att väljas. Använd `Alt` medan du ritar för att endast välja helt inneslutna element.
- Om du vill lägga till element i en markering trycker du på `Shift` medan du markerar dem.
- Om du vill ta bort ett element från markeringen trycker du på `Ctrl` medan du ritar över markeringspolygonen.

Använda *Välj nätelement med uttryck*

Ett annat verktyg för val av nätelement är Välj nätelement med uttryck. När du trycker på verktyget öppnas dialogen mesh *expression selector dialog* där du kan välja:

1. Välj metod för urval:
   - *Select by vertices*: tillämpar det angivna uttrycket på vertikaler och returnerar matchande sådana och deras eventuellt associerade kanter/ytor
   - *Select by faces*: tillämpar det angivna uttrycket på ansikten och returnerar matchande ansikten och deras tillhörande kanter/vertikaler
2. Skriv uttrycket för valet. Beroende på den valda metoden kommer tillgängliga funktioner i *Meshes group* att filtreras i enlighet med detta.
3. Kör frågan genom att ställa in hur urvalet ska bete sig och trycka på :
   - *Select*: ersätter alla befintliga markeringar i lagret
   - *Lägg till i aktuellt urval*
   - *Ta bort från aktuellt urval*

14.4.4 Modifiera nätelement

Lägga till hörn

Lägga till hörnpunkter i ett mesh-lager:

1. Tryck på knappen Digitize mesh elements (Digitalisera nätelement)
2. En widget med *Vertex Z-värde* visas i det övre högra hörnet av kartbilden. Ställ in detta värde på den Z-koordinat som du vill tilldela de efterföljande vertikalerna
3. Dubbelklicka sedan:
   - outside a face: lägger till en ”fri vertex”, dvs. en vertex som inte är kopplad till någon face. Denna vertex representeras av en röd prick när lagret är i redigeringsläge.
   - på kanten av befintliga ytor: lägger till en vertex på kanten, delar upp den eller de ytor som berör den i trianglar som är anslutna till den nya vertexen.
   - inuti en yta: delar upp ytan i trianglar vars kanter förbinder de omgivande topparna med den nya toppen.

Lägga till ansikten

Så här lägger du till ytor i ett mesh-lager:

1. Tryck på knappen Digitize mesh elements (Digitalisera nätelement)
2. En widget med *Vertex Z-värde* visas i det övre högra hörnet av kartbilden. Ställ in detta värde på den Z-koordinat som du vill tilldela de efterföljande vertikalerna.
3. Håll muspekaren över en toppunkt och klicka på den lilla triangeln som visas bredvid.
4. Flytta markören till nästa toppunktsposition; du kan snäppa till befintligt toppunkt eller vänsterklicka för att lägga till ett nytt.
5. Fortsätt enligt ovan för att lägga till så många toppar som du vill för ansiktet. Tryck på `Backspace`-knappen för att ångra det sista toppunktet.

6. När du flyttar musen visas ett gummiband som visar ansiktets form. Om det visas i grönt är den förväntade ytan giltig och du kan högerklicka för att lägga till den i mesh. Om det är rött är ytan inte giltig (t.ex. för att den skär sig själv, överlappar en befintlig yta eller vertex, skapar ett hål, …) och kan inte läggas till. Du måste ångra några vertex och fixa geometrin.
7. Tryck på `Esc` för att avbryta digitaliseringen av ansiktet.
8. Högerklicka för att validera ansiktet.

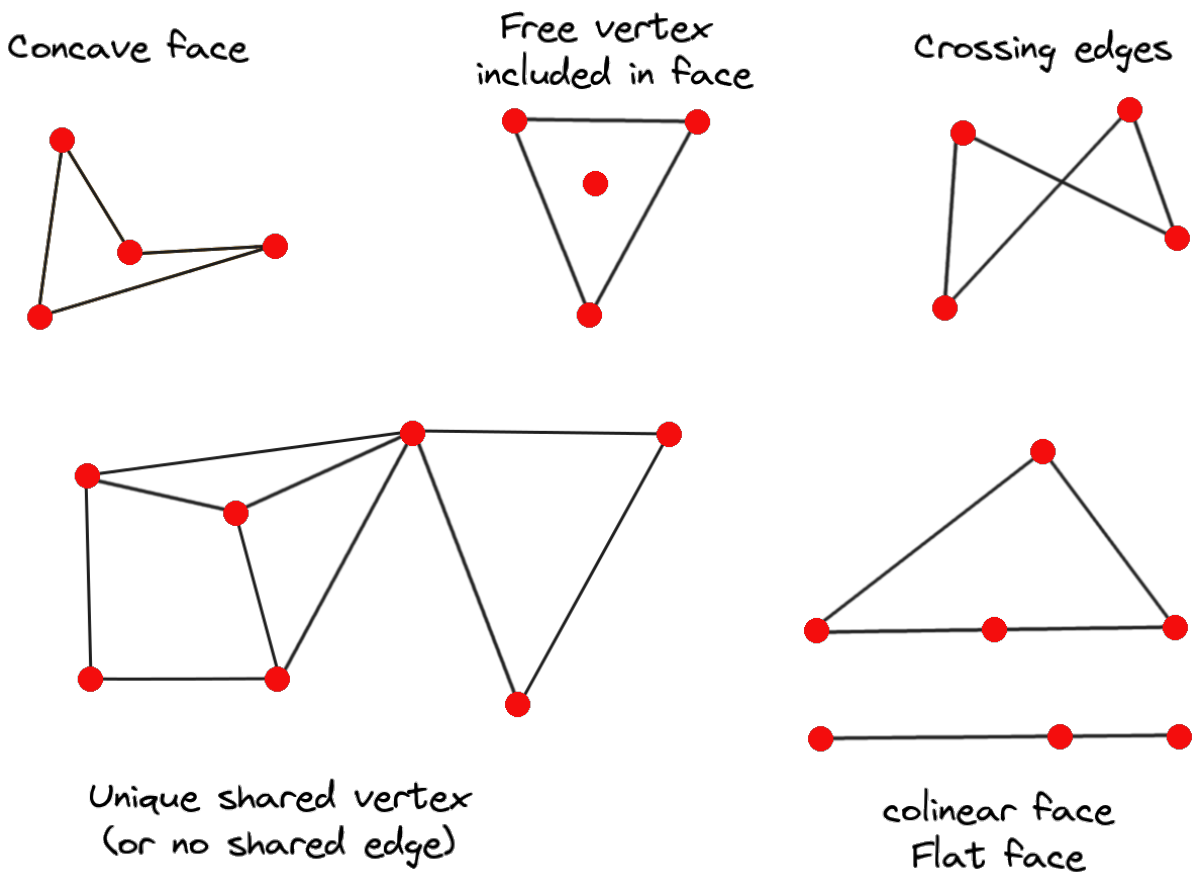


Fig. 14.16: Exempel på ogiltiga maskor

Ta bort nätelement

1. *Välj målelement*
2. Aktivera verktyget Digitize mesh elements (Digitalisera nätelement)
3. Högerklicka och välj:
   - *Remove Selected Vertices and Fill Hole(s)* eller tryck på `Ctrl+Del`: tar bort vertikaler och länkade ytor och fyller hålet/hålen genom triangulering från grannvertikalerna
   - *Remove Selected Vertices Without Filling Hole(s)* eller tryck på `Ctrl+Shift+Del`: tar bort vertikaler och länkade ytor och fyller inte i hål
   - *Remove Selected Face(s)* eller tryck på `Shift+Del`: tar bort ansikten men behåller hörnpunkterna

Dessa alternativ är också tillgängliga från den kontextuella menyn när du håller muspekaren över ett enskilt objekt utan att välja.

Flytta nätelement

För att flytta hörn och ytor i ett mesh-lager:

1. *Välj målelement*
2. Aktivera verktyget Digitize mesh elements (Digitalisera nätelement)
3. Börja flytta elementet genom att klicka på en vertex eller centroiden för en yta/kant
4. Flytta markören till målpositionen (snapping till vektorfunktioner stöds).
5. Om den nya platsen inte genererar en *invalid mesh* visas de flyttade elementen i grönt. Klicka igen för att släppa dem på den här platsen. Faces vars alla hörn är markerade translateras och deras grannar omformas i enlighet med detta.

Omvandling av nätets hörnpunkter

Verktyget Transform Vertices Coordinates ger ett mer avancerat sätt att flytta vertikaler genom att redigera deras X-, Y- och/eller Z-koordinater med hjälp av uttryck.

1. Välj de hörn som du vill redigera koordinaterna för
2. Tryck på Transform Vertices Coordinates. En dialogruta öppnas med uppgift om antalet valda vertikaler. Du kan fortfarande lägga till eller ta bort vertikaler från urvalet.
3. Beroende på vilka egenskaper du vill ändra måste du kontrollera *X coordinate*, *Y coordinate* och/eller *Z value*.
4. Ange sedan målpositionen i rutan, antingen som ett numeriskt värde eller som ett uttryck (med hjälp av dialogen :sup:`Expression`)
5. När Import Coordinates of the Selected Vertex är aktiverat fylls rutorna X, Y och Z automatiskt med koordinaterna när en enskild vertex väljs. Ett bekvämt och snabbt sätt att justera topparna individuellt.
6. Tryck på *Preview Transform* för att simulera topparnas nya placering och förhandsgranska nätet med transformationen.
   - Om förhandsgranskningen är grön är det transformerade nätet giltigt och du kan tillämpa transformationen.
   - Om förhandsgranskningen är röd är det transformerade nätet ogiltigt och du kan inte använda transformationen förrän det har korrigerats.
7. Tryck på *Apply Transform* för att ändra de valda koordinaterna för uppsättningen hörnpunkter.

Omformning av nätgeometri

Den kontextuella menyn

1. Aktivera Digitize mesh elements (Digitalisera nätelement)
2. Välj eller avstå från att välja maskpost(er)
3. Håll muspekaren över ett mesh-element så markeras det.
4. Högerklicka så kan du göra det:
   - *ta bort objektet/objekten*
   - *Split Selected Face(s)* (*Split Current Face*): delar upp ansiktet du håller muspekaren över eller varje valt quad mesh-ansikte i två trianglar
   - *Delaunay Triangulation with Selected vertices*: bygger triangulära ytor med hjälp av utvalda fria hörn.

- *Refine Selected Face(s)* (*Refine Current Face*): delar upp ytan i fyra ytor, baserat på hörn som läggs till i mitten av varje kant (en triangel resulterar i trianglar, en fyrkant i fyrkant). Triangulerar även intilliggande ytor som är anslutna till de nya hörnen.

Kantmarkeringarna

När funktionen Digitize mesh elements är aktiv och du håller muspekaren över en kant, markeras kanten och det är möjligt att interagera med den. Beroende på sammanhanget kan följande markörer vara tillgängliga:

- en **fyrkantsvy** i mitten av kanten: klicka på den för att välja yttersta hörn.
- ett **kors** om de två ansiktena på vardera sidan kan slås samman: klicka på det för att ta bort kanten och slå samman ansiktena.
- en **cirkel** om kanten ligger mellan två trianglar: Klicka på den för att vända på kanten, dvs. koppla den istället till de två andra ”fria” hörnen på ytorna

Verktyget *Force by Selected Geometries* (kraft genom valda geometrier)

Verktyget Force by Selected Geometries ger avancerade sätt att tillämpa brytlinjer med hjälp av linjegeometri. En brytlinje tvingar mesh att ha kanter längs linjen. Observera att brytlinjen inte betraktas som beständig när operationen är klar; resulterande kanter fungerar inte som begränsningar längre och kan modifieras som vilken annan kant som helst. Detta kan t.ex. användas för att lokalt modifiera ett mesh-lager med exakta linjer, som flodbankar eller vägbankar.

1. Aktivera verktyget Force genom valda geometrier
2. Ange den geometri som ska användas som ”forcing line”; det kan vara:
   - hämtas från en linje- eller polygonfunktion i kartbilden: högerklicka på vektorfunktionen och välj den från listan i snabbmenyn.
   - en virtuell linje som dras över meshramen: vänsterklicka för att lägga till vertices, högerklicka för validering. Vertex Z-värde ställs in via widgeten *Vertex Z-värde* eller widgeten *z* om widgeten *Avancerad digitaliseringspanel* är aktiverad. Om linjen knäpps till ett mesh-vertex eller ett vertex eller segment i en 3D-vektorfunktion, får det nya vertexet det knäppta elementets Z-värde.

Meshytor som överlappar linjegeometrin eller polygonens gräns påverkas på ett sätt som beror på de alternativ som du kan ställa in i verktygets rullgardinsmeny Force by Selected Geometries:

- *Lägg till ny vertex på korsande kanter*: med detta alternativ läggs en ny vertex till varje gång forceringslinjen korsar en kant. Detta alternativ leder till delning längs linjen varje gång den möter ansikten.

  Utan detta alternativ tas påträffade ytor bort och ersätts av ytor som kommer från en triangulering med endast de befintliga hörnen plus hörnen på forceringslinjerna (nya hörn läggs också till på gränskanten som skär forceringslinjerna).

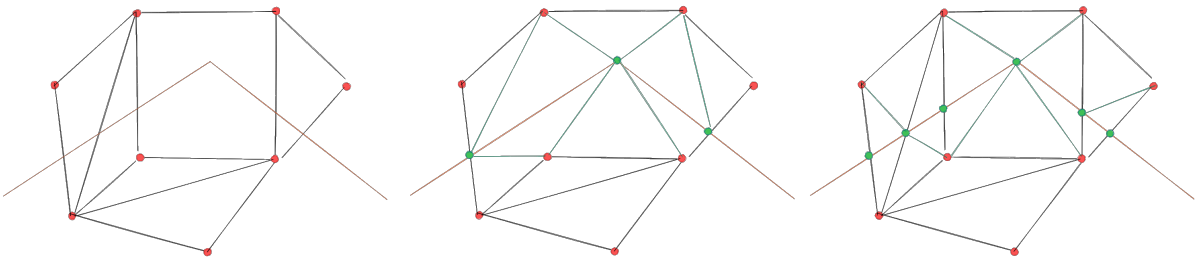


Fig. 14.17: Force Mesh med linjegeometri - Resultat utan (mitten) och med (höger) ny vertex på kanternas skärningspunkt

- *Interpolera Z-värde från*: anger hur den nya toppens Z-värde beräknas. Det kan vara från:
  - själva *Mesh*: de nya hörnens Z-värde interpoleras från hörnen i den yta de faller inom
  - eller *Forcing line*: om linjen definieras av en 3D-vektorfunktion eller en ritad linje härleds det nya Z-värdet för vertikalerna från dess geometri. När det gäller 2D-linjefunktioner är det nya Z-värdet för vertikalerna *Vertex Z value*.
- *Tolerance*: när ett befintligt mesh-vertex är närmare linjen än toleransvärdet, skapa inte ett nytt vertex på linjen utan använd det befintliga vertexet istället. Värdet kan ställas in i *Meters at Scale* eller i *Map Units* (mer information finns i *Väljare för enhet*).

14.4.5 Omindexering av maskor

Under redigering, och för att möjliggöra snabba ångra/återställa operationer, behåller QGIS tomma platser för borttagna element, vilket kan leda till växande minnesanvändning och ineffektiv mesh-strukturering. Verktyget *Mesh* ► *Reindex Faces and Vertices* är utformat för att ta bort dessa hål och omnumrera indexen för ytor och vertikaler så att de är kontinuerliga och någorlunda rimligt ordnade. Detta optimerar relationen mellan ytor och toppar och ökar effektiviteten i beräkningen.

Observera

Verktyget *Reindex Faces and Vertices* sparar lagret och rensar undo/redo-staplarna, vilket gör att ingen rollback är möjlig.

14.5 Mesh-kalkylator

Med verktyget *Mesh Calculator* från den övre menyn *Mesh* kan du utföra aritmetiska och logiska beräkningar på befintliga datasetgrupper för att generera en ny datasetgrupp (se Fig. 14.18).

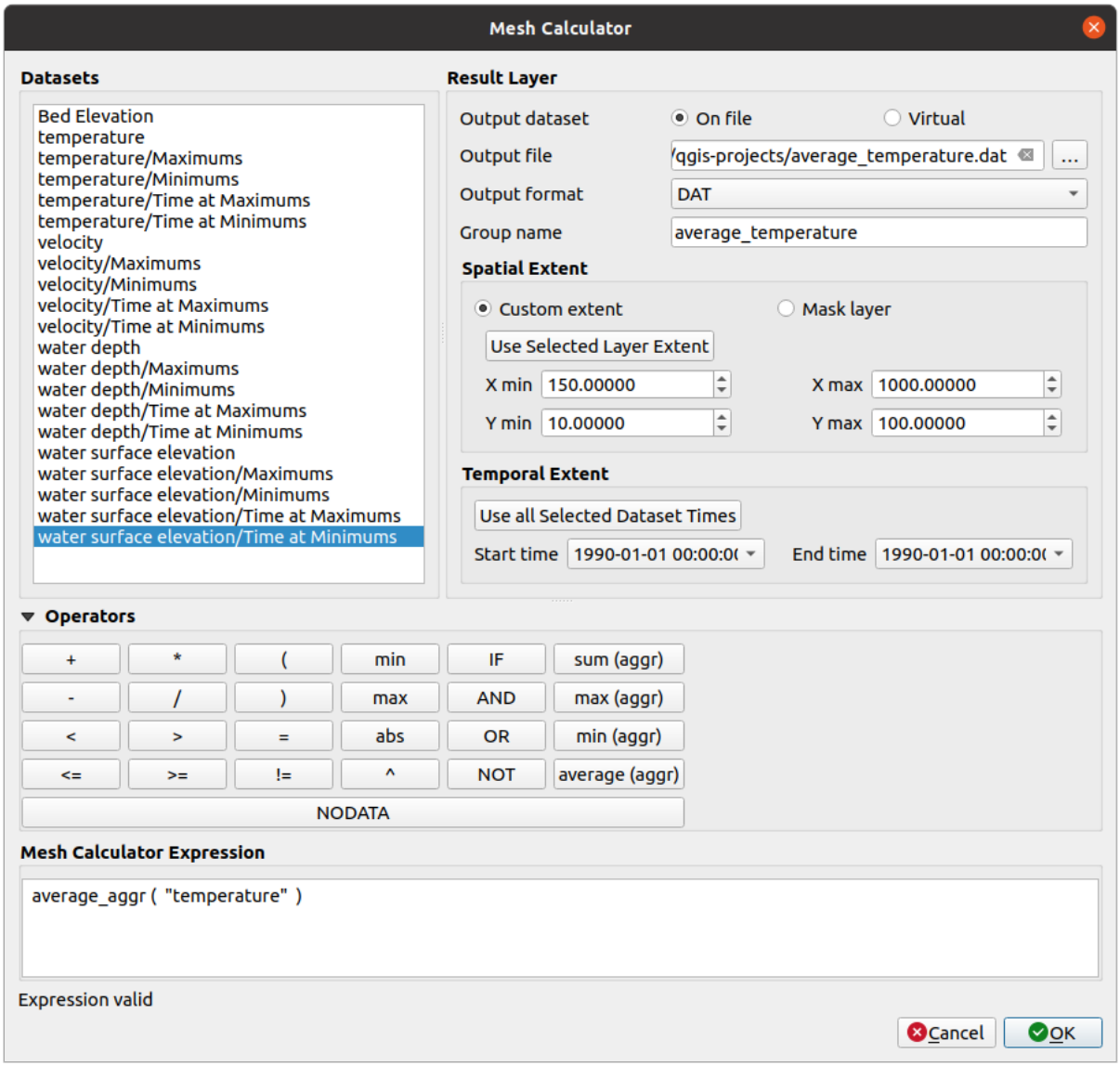


Fig. 14.18: Mesh-kalkylator

Listan *Datasets* innehåller alla datasetgrupper i det aktiva meshlagret. Om du vill använda en datasetgrupp i ett uttryck dubbelklickar du på dess namn i listan så läggs den till i fältet *Mesh calculator expression*. Du kan sedan använda operatörerna för att konstruera beräkningsuttryck eller bara skriva in dem i rutan.

Med *Result Layer* kan du konfigurera egenskaper för utdatalagret:

- *Skapa en datasetgrupp i farten istället för att skriva lagret till disken*:
  - Om den inte är markerad lagras utdata på disken som en ny vanlig fil. En sökväg till *Output File* och ett *Output Format* krävs.
  - Om du markerar detta läggs en ny datasetgrupp till i mesh-lagret. Värdena i datasetgruppen lagras inte i minnet, utan varje dataset beräknas vid behov med den formel som anges i meshkalkylatorn. Den virtuella datasetgruppen sparas med projektet och kan vid behov tas bort eller göras beständig i en fil på fliken Layer *Source* properties.

  I båda fallen bör du ange ett *Group Name* för gruppen i utdatasetet.
- Den *Rumliga utsträckning* som ska beaktas för beräkning kan vara:
  - en *Custom extent*, manuellt fylld med koordinaterna *X min*, *X max*, *Y min* och *Y max*, eller extraherad från en befintlig datasetgrupp (välj den i listan och tryck på *Use selected layer extent* för att fylla i de

ovannämnda koordinatfälten)

  - definieras av ett polygonlager (*Mask layer*) i projektet: polygonens geometri används för att klippa ut mesh-lagrets dataset

- Den *Temporal extent* som ska beaktas för dataset kan ställas in med alternativen *Start time* och *End time*, som väljs från de befintliga datasetgruppernas tidssteg. De kan också fyllas i med hjälp av knappen *Use all selected dataset times* för att ta med hela intervallet.

Avsnittet *Operators* innehåller alla tillgängliga operatorer. För att lägga till en operator i nätkalkylatorns uttrycksruta, klicka på lämplig knapp. Matematiska beräkningar (`+`, `-`, `*`, ...) och statistiska funktioner (`min`, `max`, `sum (aggr)`, `medel (aggr)`, ...) är tillgängliga. Villkorliga uttryck (`=`, `!=`, `<`, `>=`, `IF`, `AND`, `NOT`, ...) returnerar antingen 0 för falskt och 1 för sant, och kan därför användas med andra operatorer och funktioner. Värdet `NODATA` kan också användas i uttrycken.

Widgeten *Mesh Calculator Expression* visar och låter dig redigera det uttryck som ska köras.

KAPITEL 15

Arbeta med Vector Tiles

15.1 Vad är Vector Tiles?

Vector tiles är paket med geografiska data som paketeras i fördefinierade, ungefär kvadratiska ”tiles” för överföring via webben. De kombinerar förrenderade rasterkartor och vektorkartor. I stället för en förrenderad kartbild returnerar vector tile-servern vektorkartdata som har klippts till gränserna för varje kakel. De klippta brickorna representerar zoomnivåerna för vektorplattans tjänst, härledda från en pyramidmetod. Med hjälp av denna struktur minskas dataöverföringen jämfört med vektorkartor som inte är uppdelade i rutor. Endast data inom den aktuella kartvyn och på den aktuella zoomnivån behöver överföras. Jämfört med en kaklad rasterkarta minskar dataöverföringen också kraftigt, eftersom vektordata vanligtvis är mycket mindre än en renderad bitmapp. Vektorplattor har inte tilldelats någon stylinginformation, så QGIS måste tillämpa en kartografisk stil för att visa data.

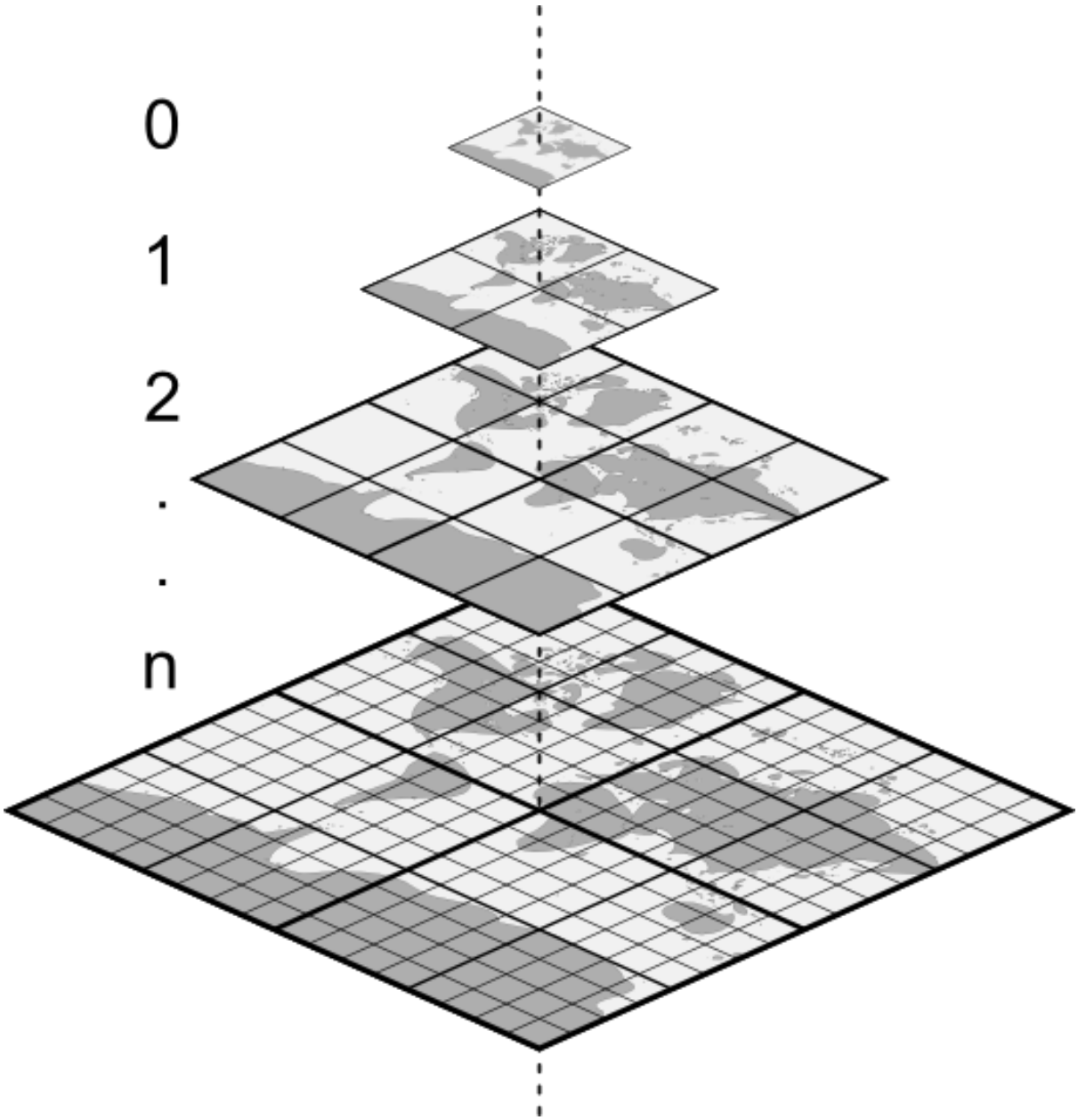


Fig. 15.1: Pyramidstruktur av vektorplattor med zoom-nivåer

15.2 Format som stöds

Det finns stöd för vektorplattor genom:

- fjärrkällor (HTTP/S) - med XYZ-mall - `type=xyz&url=http://example.com/{z}/{x}/{y}.pbf`
- lokala filer - med XYZ-mall - t.ex. `type=xyz&url=file:///path/to/tiles/{z}/{x}/{y}.pbf`
- lokal MBTiles-databas - t.ex. `type=mbtiles&url=file:///path/to/file.mbtiles`

För att ladda en vektorplattor-datauppsättning i QGIS, använd fliken *Vector Tile* i dialogrutan *Data Source Manager*. Läs *Använda Vector Tiles-tjänster* för mer information.

15.3 Vector Tiles Dataset Egenskaper

Dialogrutan vector tiles *Lageregenskaper* innehåller följande avsnitt:

[1] Finns även i *Layer stylingpanel*

15.3.1 Informationsegenskaper

Fliken *Information* är skrivskyddad och utgör en intressant plats för att snabbt hämta sammanfattad information och metadata om det aktuella lagret. Tillhandahållen information är:

- baserat på lagrets leverantör: namn, URI, källtyp och sökväg, antal zoomnivåer
- anpassade egenskaper, som används för att lagra ytterligare information om lagret i det aktiva projektet. Fler egenskaper kan skapas och hanteras med hjälp av PyQGIS, särskilt genom metoden setCustomProperty().
- det koordinatsatta referenssystemet: namn, enheter, metod, noggrannhet, referens (dvs. om det är statiskt eller dynamiskt)
- plockad från *fyllda metadata*: access, extents, länkar, kontakter, historia…

15.3.2 Källegenskaper

Fliken *Source* visar grundläggande information om den valda vektorplattan, inklusive:

- det *Layernamn* som ska visas i *Layers Panel*;
- *Koordinatreferenssystem*: Visar lagrets *Coordinate Reference System (CRS)*. Du kan ändra lagrets CRS genom att välja ett nyligen använt i listrutan eller klicka på knappen Välj CRS (se *Väljare för referenssystem för koordinater*). Använd denna process endast om lagrets CRS är fel eller inte har angetts.

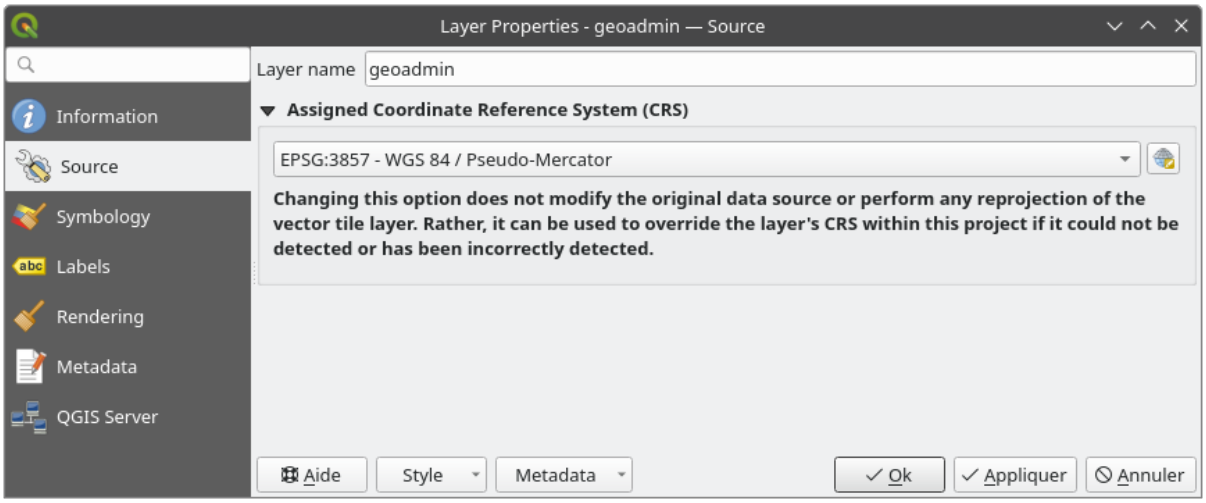


Fig. 15.2: Egenskaper för vektorplattor - dialogrutan Källa

15.3.3 Symbologi och etiketter Egenskaper

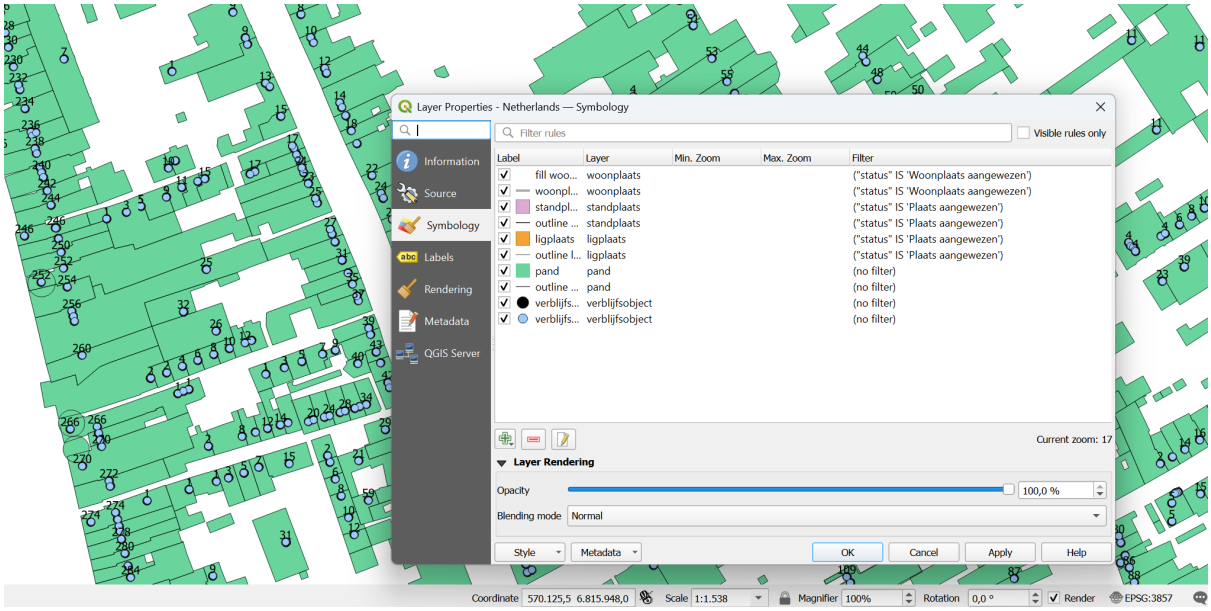


Fig. 15.3: Vektor Tile Layer Symbology

Fastställande av regler

Eftersom vektorplattor består av punkt-, linje- och polygongeometrier är respektive symboler tillgängliga. Om du vill använda en kartografisk stil (med symboler och/eller etiketter) kan du antingen:

- Använd en *Style URL* när du skapar *Vector Tiles Connection*. Symbologin visas omedelbart på fliken *Symbology* när lagret har laddats i QGIS.
- Du kan också skapa din egen symbologi och märkning på motsvarande flikar i lagrets egenskaper. Som standard tilldelar QGIS en identisk symbol till objekten baserat på deras geometrityp.

I båda fallen är inställningen av en stil för en vektorplatta beroende av en uppsättning *regler* som tillämpas på funktionerna, vilket indikerar:

- en vektor *symbol* eller *label*
- en *Label*, en titel för omfattande identifiering av regeln
- namnet på ett visst *Lager* som regeln ska tillämpas på, om den inte tillämpas på `(alla lager)`
- en *Min. Zoom* och en *Max. Zoom*, för visningsområdet. Symbologi och märkning kan vara beroende av zoomnivån.
- ett *Filter*, ett QGIS-uttryck för att identifiera de funktioner som stilen ska tillämpas på

Varje regel läggs till genom att trycka på knappen Add rule och välja den typ av symboler (*Marker*, *Line*, *Fill*) som motsvarar objektets geometrityp. Du kan också välja Remove selected rules eller Edit current rule.

Längst ner visas *Current Zoom*. Markera alternativet *Visible rules only* högst upp i dialogrutan för att filtrera listan med regler så att endast de som är synliga på den aktuella zoomnivån visas. Detta gör det lättare att arbeta med komplex vektorstyling och att lokalisera besvärliga regler. Textrutan *Filter rules* hjälper dig också att enkelt hitta en regel genom att söka i fälten *Label*, *Layer* och *Filter*.

I Fig. 15.3 ställer vi in stil för OpenStreetMap `landuse`-lagret. För bättre synlighet är de flesta reglerna bortvalda.

Rendering av lager

På fliken *Symbology* kan du också ställa in några alternativ som alltid påverkar alla funktioner i lagret:

- *Opacitet*: Med det här verktyget kan du göra det underliggande lagret i kartbilden synligt. Använd skjutreglaget för att anpassa synligheten för ditt vektorlager efter dina behov. Du kan också göra en exakt definition av den procentuella synligheten i menyn bredvid skjutreglaget.
- *Blandningsläge*: Med dessa verktyg kan du uppnå speciella renderingseffekter som du kanske tidigare bara kände till från grafikprogram. Pixlarna i de över- och underliggande lagren blandas genom de inställningar som beskrivs i *Blandningslägen*.

Stilar

Menyn *Styles* ► finns längst ned på de flesta flikar och innehåller genvägar för att spara, ladda, skapa och byta stilar som ska tillämpas på vektorplattorna. Vektorplattor kan ha sin stil sparad från QGIS som *QML*-filer och de kan importeras som:

- *QML*-filer (*QML - QGIS Style File Format*)
- Konfigurationsfiler i stil med *MapBox GL Json*

Mer information finns på *Spara och dela egenskaper för lager*.

15.3.4 Rendering av egenskaper

Under *Skalaberoende synlighet* kan du ställa in skalorna *Maximum (inklusive)* och *Minimum (exklusive)* och definiera ett skalområde där objekten ska vara synliga. Utanför detta intervall är de dolda. Knappen Set to current canvas scale hjälper dig att använda den aktuella kartans canvasskala som gräns för synlighetsintervallet. Se *Väljare för synlighetsskala* för mer information.

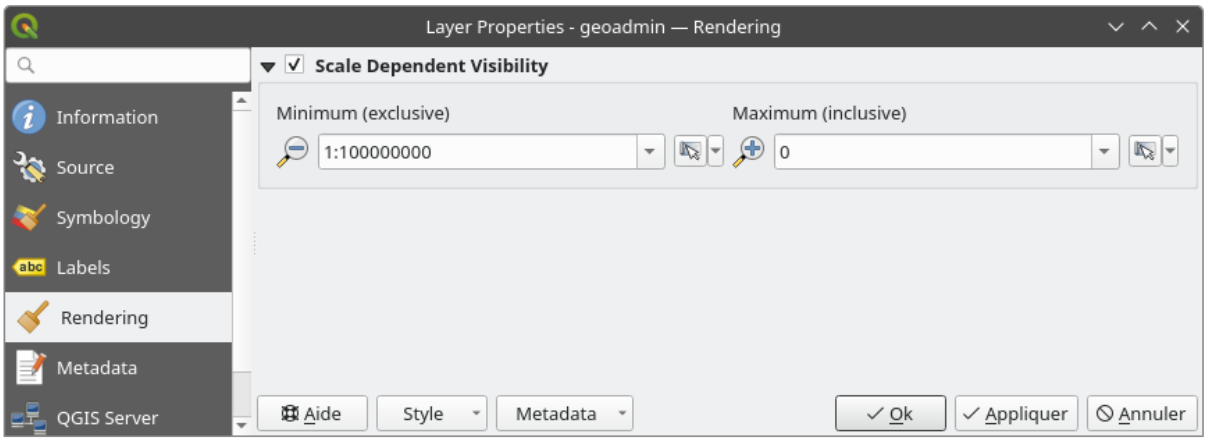


Fig. 15.4: Egenskaper för vektorplattor - dialogruta för rendering

15.3.5 Egenskaper för metadata

Fliken *Metadata* ger dig alternativ för att skapa och redigera en metadatarapport om ditt lager. Se *Metadata* för mer information.

KAPITEL 16

Arbeta med punktmoln

16.1 Introduktion till punktmoln

Vad är ett punktmoln?

Ett punktmoln är en tredimensionell bild av ett utrymme som består av många enskilda datapunkter (upp till miljarder, till och med biljoner). Var och en av punkterna har en x-, y- och z-koordinat. Beroende på insamlingsmetod har punktmoln vanligtvis också ytterligare attribut som kommer från insamlingen, t.ex. färgvärden eller intensitet. Dessa attribut kan t.ex. användas för att visa punktmoln i olika färger. I QGIS kan ett punktmoln användas för att generera en tredimensionell bild av landskapet (eller av ett annat utrymme).

Format som stöds

QGIS stöder dataformaten Entwine Point Tile (EPT) och LAS/LAZ. För att arbeta med punktmoln sparar QGIS alltid data i EPT. EPT är ett lagringsformat som består av flera filer som lagras i en gemensam mapp. För att ge snabb åtkomst till data använder EPT indexering. För mer information om EPT-formatet, se entwine homepage

Om data är i LAS- eller LAZ-format kommer QGIS att konvertera dem till EPT när de laddas för första gången. Beroende på filens storlek kan detta ta en viss tid. I den här processen skapas en undermapp i den mapp där LAS/LAZ-filen finns enligt schemat `ept_` + `name_LAS/LAZ_file`. Om en sådan undermapp redan finns laddar QGIS EPT:n omedelbart (vilket leder till en kortare laddningstid).

Värd att känna till

I QGIS är det (ännu) inte möjligt att redigera punktmoln. Om du vill manipulera ditt punktmoln kan du använda CloudCompare, ett bearbetningsverktyg för punktmoln med öppen källkod. Även Point Data Abstraction Library (PDAL - liknar GDAL) ger dig alternativ för att redigera punktmoln (PDAL är endast kommandorad).

På grund av det stora antalet datapunkter är det inte möjligt att visa en attributtabell för punktmoln i QGIS. Verktyget *Identify tool* stöder dock punktmoln, så du kan visa alla attribut, även för en enda datapunkt.

Om du vill skapa ett nytt lager, i samma eller ett annat format som stöds, från ett befintligt punktmolnslager, se *Skapa nya lager från ett befintligt lager*.

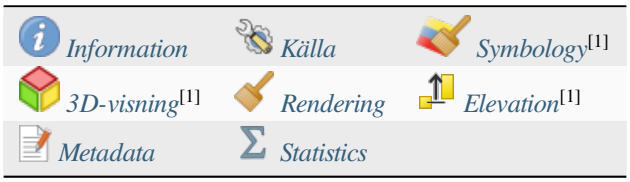
16.2 Punktmoln Egenskaper

Dialogrutan *Lageregenskaper* för ett punktmolnslager ger allmänna inställningar för lagret och dess rendering. Den innehåller också information om lagret.

För att komma till dialogrutan *Lageregenskaper*:

- Dubbelklicka på lagret i panelen *Layers* eller högerklicka och välj *Properties…* från snabbmenyn;
- Gå till menyn *Layer* ► *Lageregenskaper…* när lagret är markerat.

Dialogrutan för punktmoln *Lageregenskaper* innehåller följande avsnitt:



[1] Finns även i *Layer stylingpanel*

Observera

De flesta av egenskaperna för ett punktmolnslager kan sparas till eller laddas från en `.qml`-fil med hjälp av *Style*-menyn längst ned i dialogrutan för egenskaper. Mer information finns på *Spara och dela egenskaper för lager*

16.2.1 Informationsegenskaper

Fliken *Information* är skrivskyddad och utgör en intressant plats för att snabbt hämta sammanfattad information och metadata om det aktuella lagret. Tillhandahållen information är:

- Allmänna uppgifter som namn i projektet, sökväg till källan, senaste spartid och storlek, använd leverantör
- anpassade egenskaper som används för att lagra ytterligare information om lagret i det aktiva projektet. Standardanpassade egenskaper kan inkludera *layer notes*. Fler egenskaper kan skapas och hanteras med hjälp av PyQGIS, särskilt genom metoden setCustomProperty().
- Baserat på lagrets leverantör: omfattning och antal punkter
- Koordinatreferenssystemet: namn, enheter, metod, noggrannhet, referens (dvs. om det är statiskt eller dynamiskt)
- Metadata som levereras av leverantören: skapandedatum, version, dataformat, skala X/Y/Z, …
- Väljs från fliken *Metadata* (där de kan redigeras): access, extents, länkar, kontakter, historik…

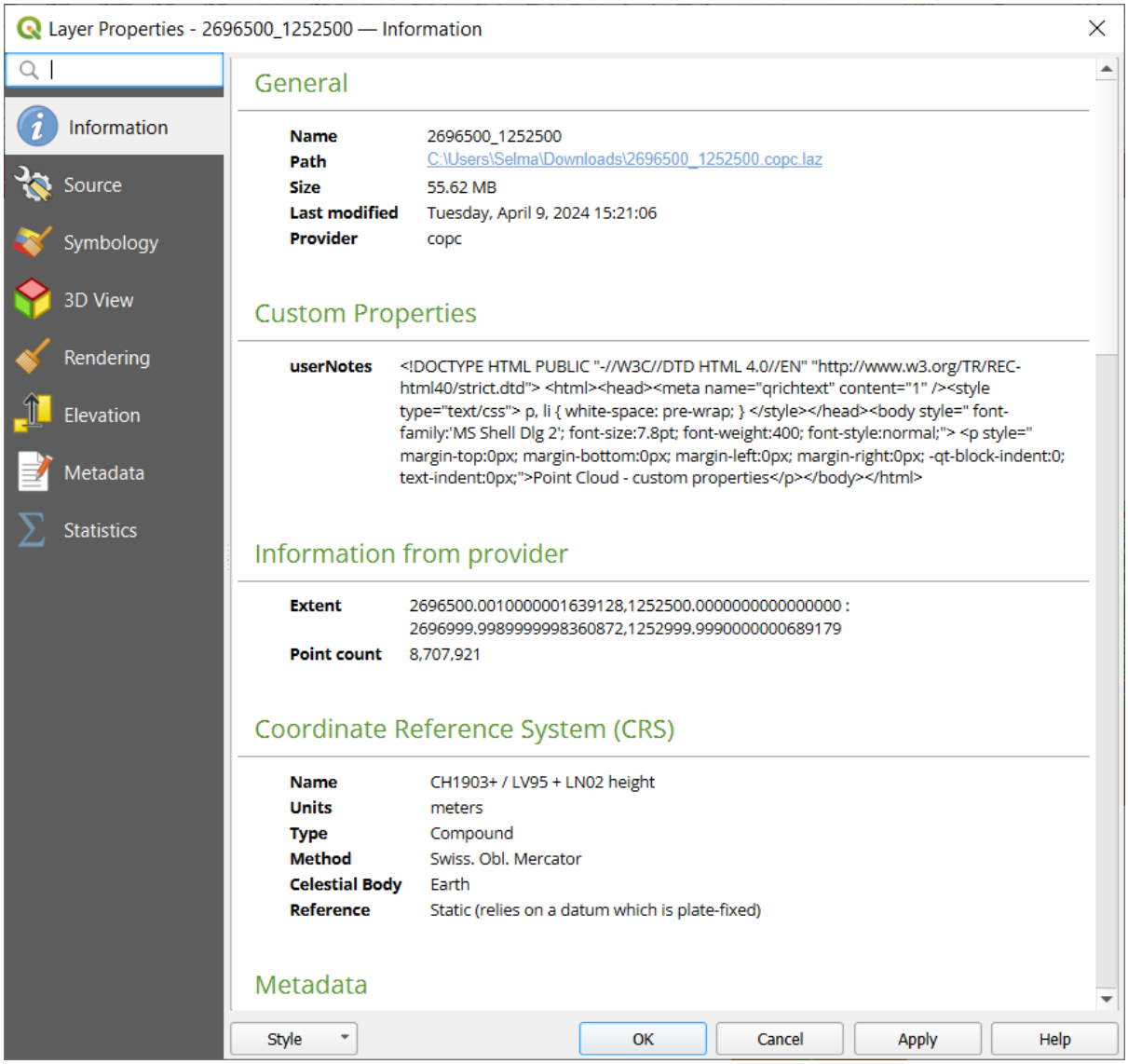


Fig. 16.1: Fliken Information om punktmoln

16.2.2 Källegenskaper

På fliken *Source* kan du se och redigera grundläggande information om punktmolnslagret:

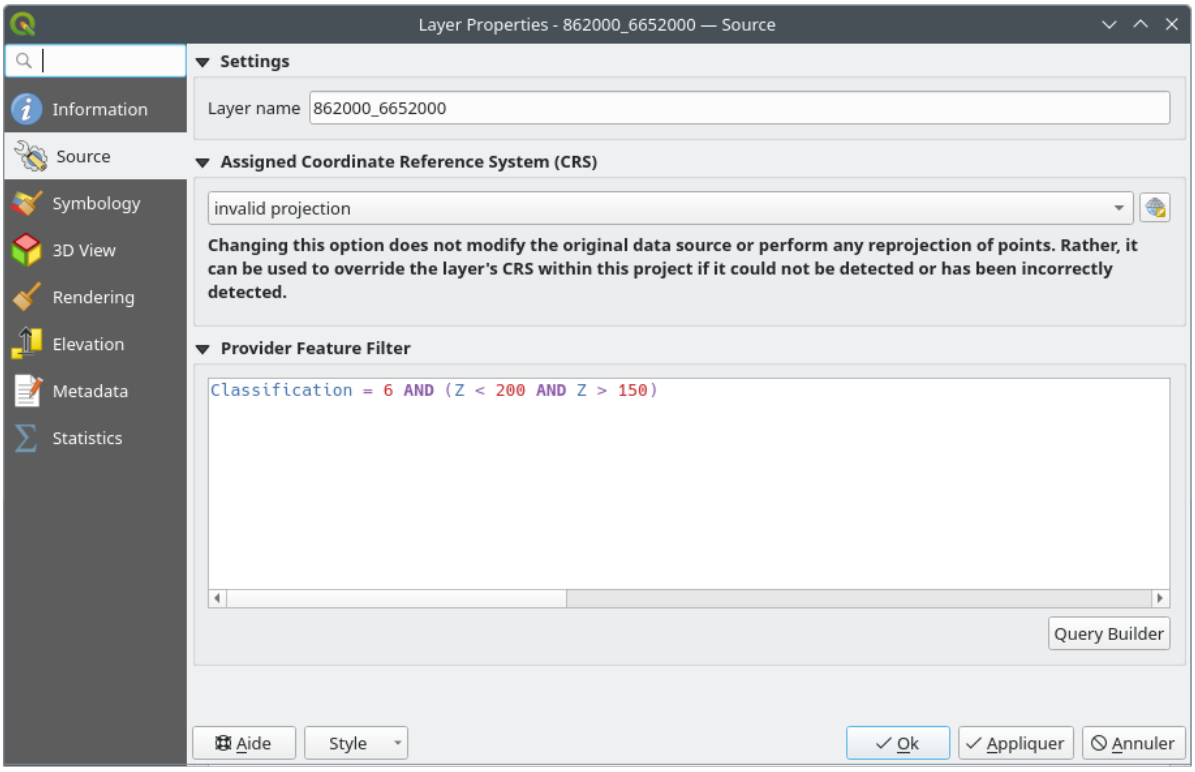


Fig. 16.2: Fliken Punktmolnskälla

- *Inställningar*: Ange ett lagernamn som skiljer sig från lagrets filnamn och som kommer att användas för att identifiera lagret i projektet (i lagerpanelen, med uttryck, i utskriftslayoutens teckenförklaring, …)
- *Tilldelat koordinatreferenssystem (CRS)*: Här kan du ändra lagrets tilldelade *Coordinate Reference System* genom att välja ett nyligen använt i listrutan eller klicka på knappen set Projection Select CRS (se *Väljare för referenssystem för koordinater*). Använd denna process endast om det CRS som tillämpas på lagret är felaktigt eller om inget CRS har tillämpats.
- *Provider Feature Filter*: gör det möjligt att begränsa data som ska användas i det aktuella projektet till en delmängd, med hjälp av funktioner och uttryck på PDAL-dataleverantörsnivå. Tryck på knappen *Query Builder* längst ned för att börja ställa in filtret.

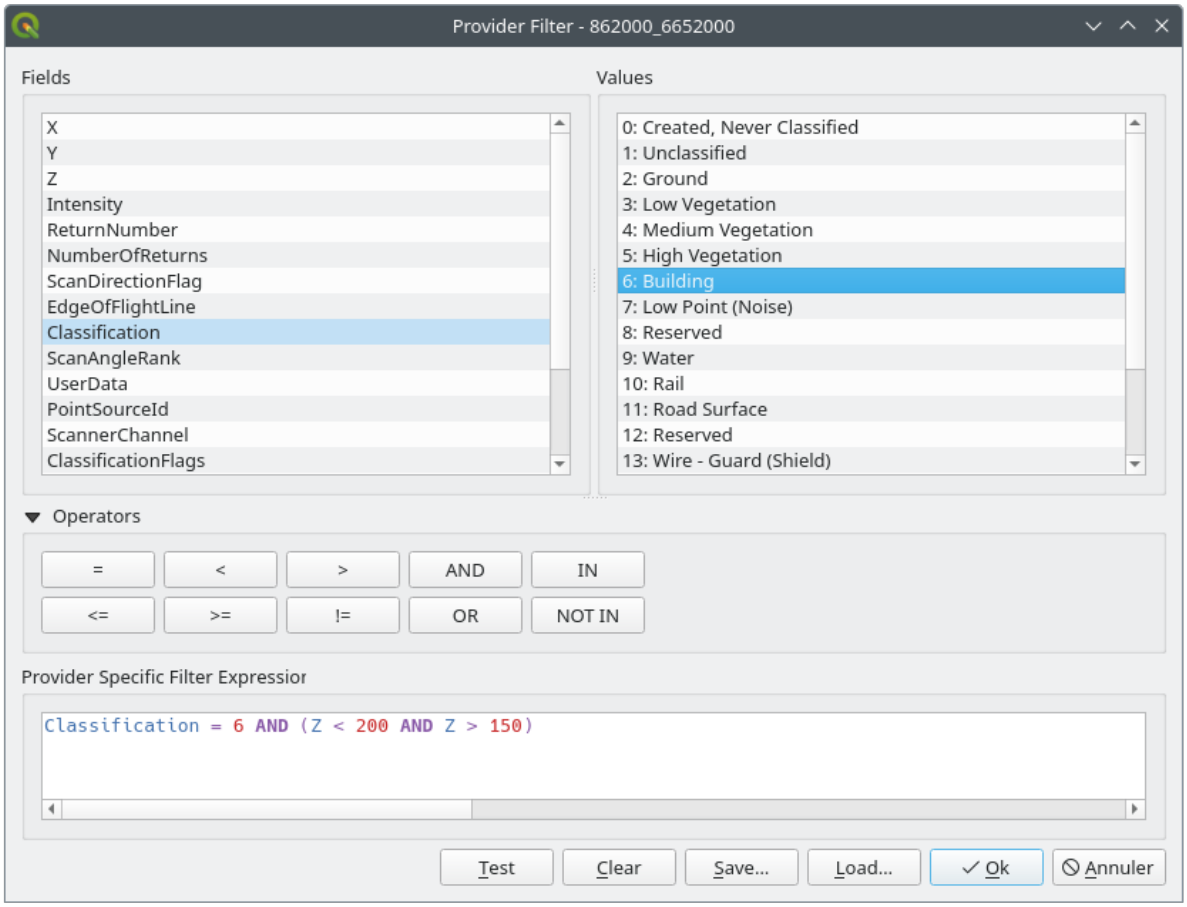


Fig. 16.3: Filtrering av punktmolnslager till byggnader på specifik höjd

I den nedre delen av dialogrutan kan du skriva in ett *Providerspecifikt filteruttryck*. Ett sådant uttryck kan byggas upp tack vare:

- *Fields*: listan innehåller alla attribut för punktmolnslagret. Om du vill lägga till ett attribut i uttrycket dubbelklickar du på dess namn eller skriver in det i textrutan.
- *Values*: -ramen visar värden eller statistik för det valda attributet, som ett par av typen nyckel:värde. För att lägga till ett värde i uttrycksfältet dubbelklickar du på dess namn i listan: beroende på typen av attribut läggs antingen nyckeln eller värdet till i uttrycket. Eller skriv in värdet i textrutan för uttrycket.
- *Operatorer*: detta verktygsfält innehåller alla operatorer som kan användas. För att lägga till en operator i uttrycksfältet klickar du på lämplig knapp. Relationella operatorer (= , > , …) och logiska operatorer (`AND`, `OR`, …) finns tillgängliga.

Knappen *Test* hjälper dig att kontrollera syntaxen i din fråga. Använd knappen *Clear* för att radera frågan och återställa lagret till dess ursprungliga tillstånd (dvs. ladda alla punkter i lagret). Det är möjligt att *Save…* spara frågan som en `.QQF`-fil eller *Load…* ladda frågan från en fil till dialogrutan.

När ett filter tillämpas behandlar QGIS den resulterande delmängden som om den vore hela lagret. Om du t.ex. har använt *filtret ovanför* för att filtrera byggnader kan du inte t.ex. visa, fråga, spara eller redigera punkter som har vegetationsklassificering eftersom de inte ingår i delmängden.

Tips

Filtrerade lager visas i panelen Lager

I panelen *Lager* listas ett filtrerat lager med en ikon Filter bredvid som anger vilken fråga som används

när musen förs över ikonen. Dubbelklicka på ikonen för att öppna dialogrutan *Query Builder* för redigering. Detta kan också göras via menyn *Layer* ► *Filter*….

16.2.3 Egenskaper för symbologi

På fliken *Symbology* görs inställningarna för rendering av punktmolnet. I den övre delen finns inställningarna för de olika funktionella renderingarna. I den nedre delen finns sektioner där allmänna inställningar för hela lagret kan göras och som gäller för alla funktionella renderingsprogram.

Renderingstyper för funktioner

Det finns olika alternativ för rendering av punktmoln som kan väljas med hjälp av rullgardinsmenyn längst upp på fliken *Symbology* (se Fig. 16.4):

- *Endast utsträckning*: Endast en avgränsande ruta av datautbredningen visas; praktiskt för att få en överblick över datautbredningen. Som vanligt hjälper *Symbol widget* dig att konfigurera alla egenskaper (färg, streck, opacitet, underlager, …) som du vill ha för rutan.
- *Attribut efter ramp*: Data ritas över en färggradient. Se *Attribut för rampåtergivaren*
- *RGB*: Rita data med hjälp av röda, gröna och blå färgvärden. Se *RGB-rendering*
- *Klassificering*: Data ritas med olika färger för olika klasser. Se *Klassificering Renderer*

När ett punktmoln laddas följer QGIS en logik för att välja den bästa renderingsenheten:

- om datasetet innehåller färginformation (röda, gröna, blå attribut) används RGB-renderingsprogrammet
- else om datasetet innehåller ett `Classification`-attribut används den klassificerade renderingstjänsten
- annars återgår den till rendering baserad på Z-attribut

Om du inte känner till punktmolnets attribut ger *Statistics tab* en bra översikt över vilka attribut som finns i punktmolnet och inom vilka intervall värdena ligger.

För varje renderingsprogram kan du förbättra datavisningen genom att justera *punktsymbolstorleken* eller aktivera *ytstriangulering*.

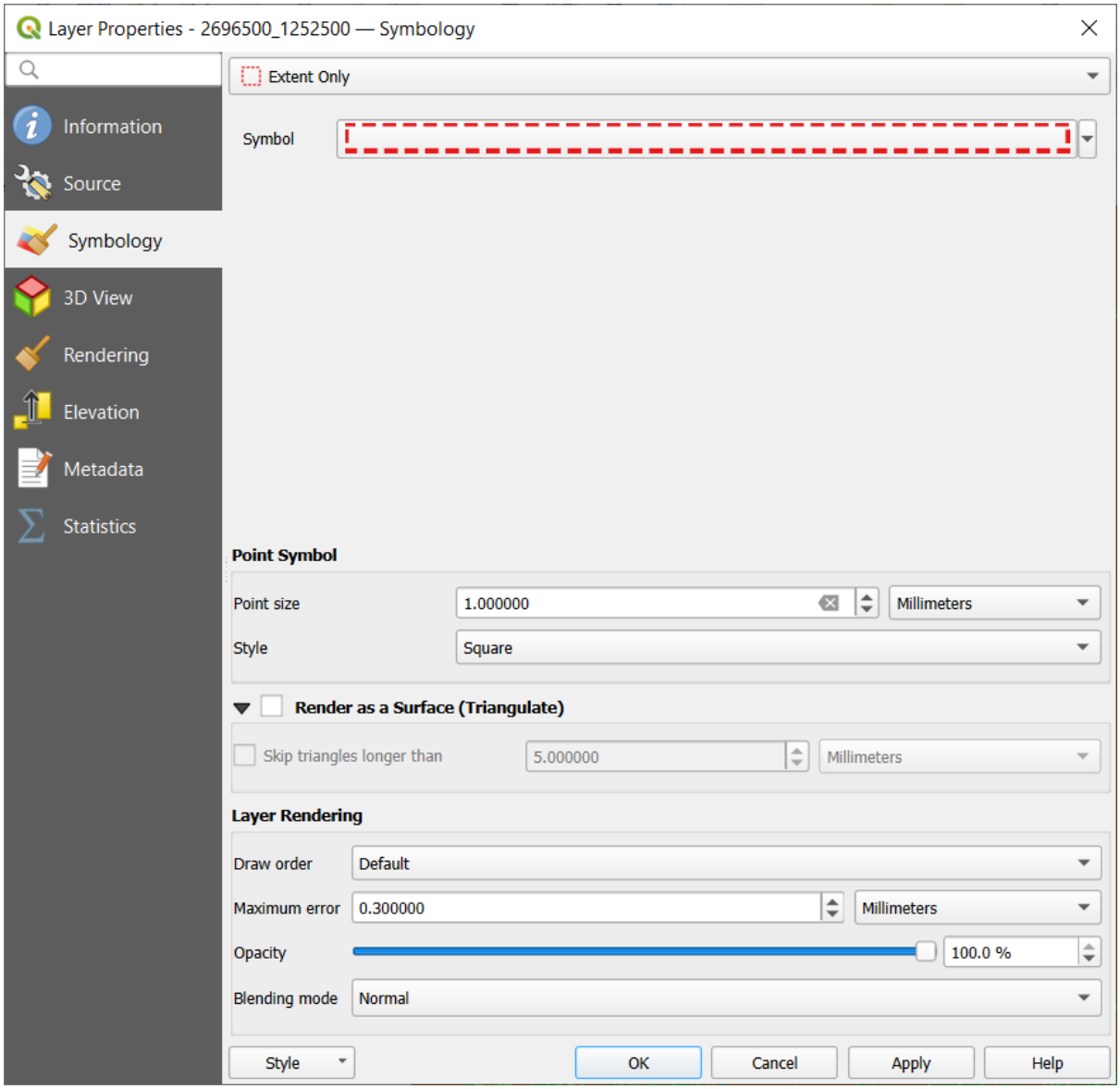


Fig. 16.4: Fliken Punktmolnssymbologi

Attribut för rampåtergivaren

Med *Attribute by Ramp* kan data visas som numeriska värden över en färggradient. Sådana numeriska värden kan t.ex. vara ett befintligt intensitetsattribut eller Z-värdet. Beroende på ett minimi- och ett maximivärde sprids de andra värdena till färggradienten via interpolation. De olika värdena och deras tilldelning till en viss färg kallas ”färgkarta” och visas i tabellen. Det finns olika inställningsmöjligheter som beskrivs under bilden.

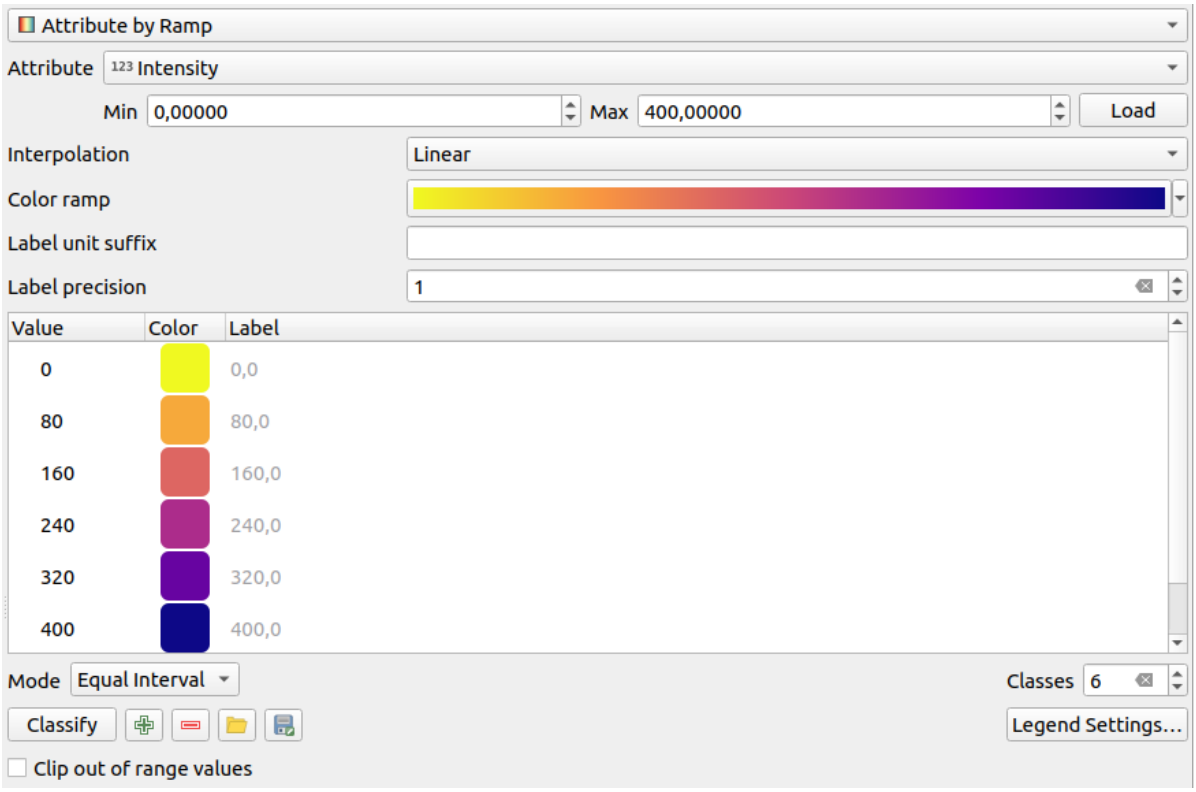


Fig. 16.5: Fliken Symbologi för punktmoln: Attribut per ramp

- *Min* och *Max* definierar det intervall som tillämpas på färgrampen: värdet *Min* representerar den vänstra och värdet *Max* den högra änden av färgrampen, värdena däremellan interpoleras. Som standard identifierar QGIS minimum och maximum från det valda attributet, men de kan ändras. När du har ändrat värdena kan du återställa standardvärdena genom att klicka på knappen *Load*.
- I *Interpolation* anges hur värden tilldelas färg:
  - *Discrete* (en <= symbol visas i rubriken på kolumnen *Value*): Färgen tas från den närmaste färgkartsposten med lika eller högre värde
  - *Linear* Färgen interpoleras linjärt från färgkartans poster över och under pixelvärdet, vilket innebär att varje värde i datasetet motsvarar en unik färg
  - *Exact* (en = symbol visas i rubriken på kolumnen *Value*): Endast pixlar med ett värde som är lika med en post i färgkartan får en färg; andra pixlar återges inte.
- Widgeten *Färgramp* hjälper dig att välja den färgramp som ska tilldelas datasetet. Som vanligt med *den här widgeten* kan du skapa en ny och redigera eller spara den som är vald för tillfället.
- Med *Label unit suffix* läggs en etikett till efter värdet i teckenförklaringen och med *Label precision* styrs antalet decimaler som ska visas.

Klassificeringen *Mode* hjälper dig att definiera hur värdena fördelas mellan klasserna:

- *Kontinuerlig*: Klasserna nummer och färg hämtas från färgrampens stopp; gränsvärdena ställs in efter stoppens fördelning i färgrampen (mer information om stopp finns i *Inställning av en färgramp*).
- *Equal interval*: Antalet klasser bestäms av fältet *Classes* i slutet av raden; gränsvärden definieras så att klasserna alla har samma storlek.

Klasserna bestäms automatiskt och visas i färgkartans tabell. Men du kan också redigera dessa klasser manuellt:

- Genom att dubbelklicka på ett *Value* i tabellen kan du ändra klassvärdet
- Dubbelklick i kolumnen *Color* öppnar widgeten *Färgväljare*, där du kan välja en färg som ska gälla för det värdet

- Dubbelklicka i kolumnen *Label* för att ändra etiketten för klassen
- Om du högerklickar på markerade rader i färgtabellen visas en kontextuell meny med *Change Color…* och *Change Opacity…* för markeringen

Under tabellen finns alternativen att återställa standardklasserna med *Classify* eller att manuellt Lägg till lägga till värden eller Ta bort valda värden från tabellen.

Eftersom en anpassad färgkarta kan vara mycket komplex finns det också möjlighet att Läs in en befintlig färgkarta eller att Spara den för användning i andra lager (som en `txt`-fil).

Om du har valt *Linear* för *Interpolation* kan du också konfigurera:

- *Clip out of range values* Som standard tilldelar den linjära metoden den första klassens (respektive den sista klassens) färg till värden i datasetet som är lägre än det inställda *Min*-värdet (respektive högre än det inställda *Max*-värdet). Markera denna inställning om du inte vill rendera dessa värden.
- *Legend settings*, för visning i panelen *Layers* och i *layout förklaring*. Anpassningen fungerar på samma sätt som med ett rasterlager (mer information finns i *Anpassa rasterförklaring*).

RGB-rendering

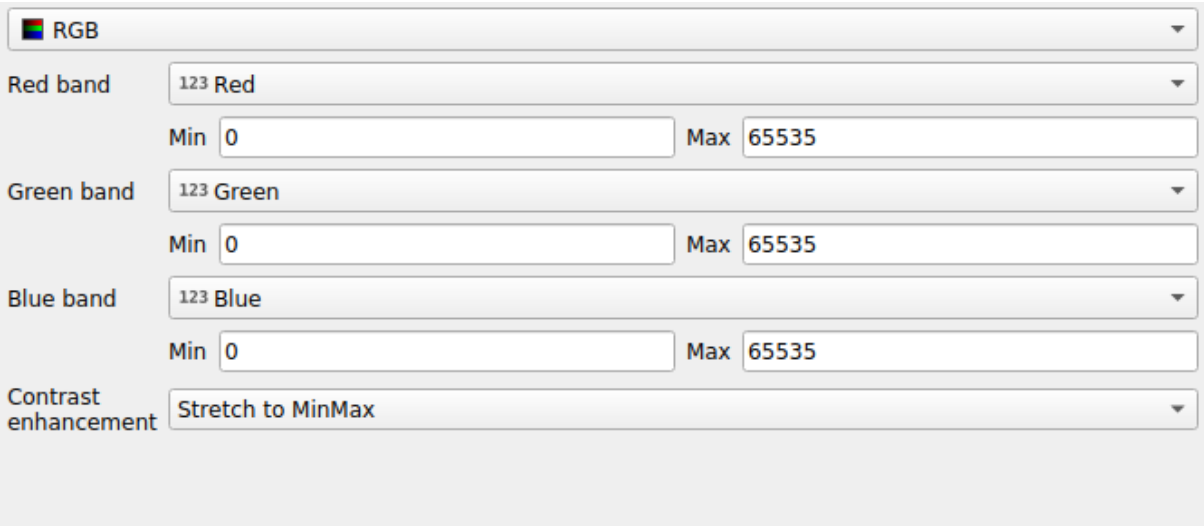


Fig. 16.6: RGB-rendering av punktmoln

Med återgivaren *RGB* kommer tre valda attribut från punktmolnet att användas som röd, grön och blå komponent. Om attributen har rätt namn väljer QGIS dem automatiskt och hämtar värdena *Min* och *Max* för varje band och skalar färgen i enlighet med detta. Det är dock också möjligt att ändra värdena manuellt.

En metod för *Kontrastförbättring* kan tillämpas på värdena: *Ingen förbättring*, *Stretch till MinMax*, *Stretch och Clip till MinMax* och *Clip till MinMax*

Observera

Verktyget *Kontrastförbättring* är fortfarande under utveckling. Om du har problem med det bör du använda standardinställningen *Stretch to MinMax*.

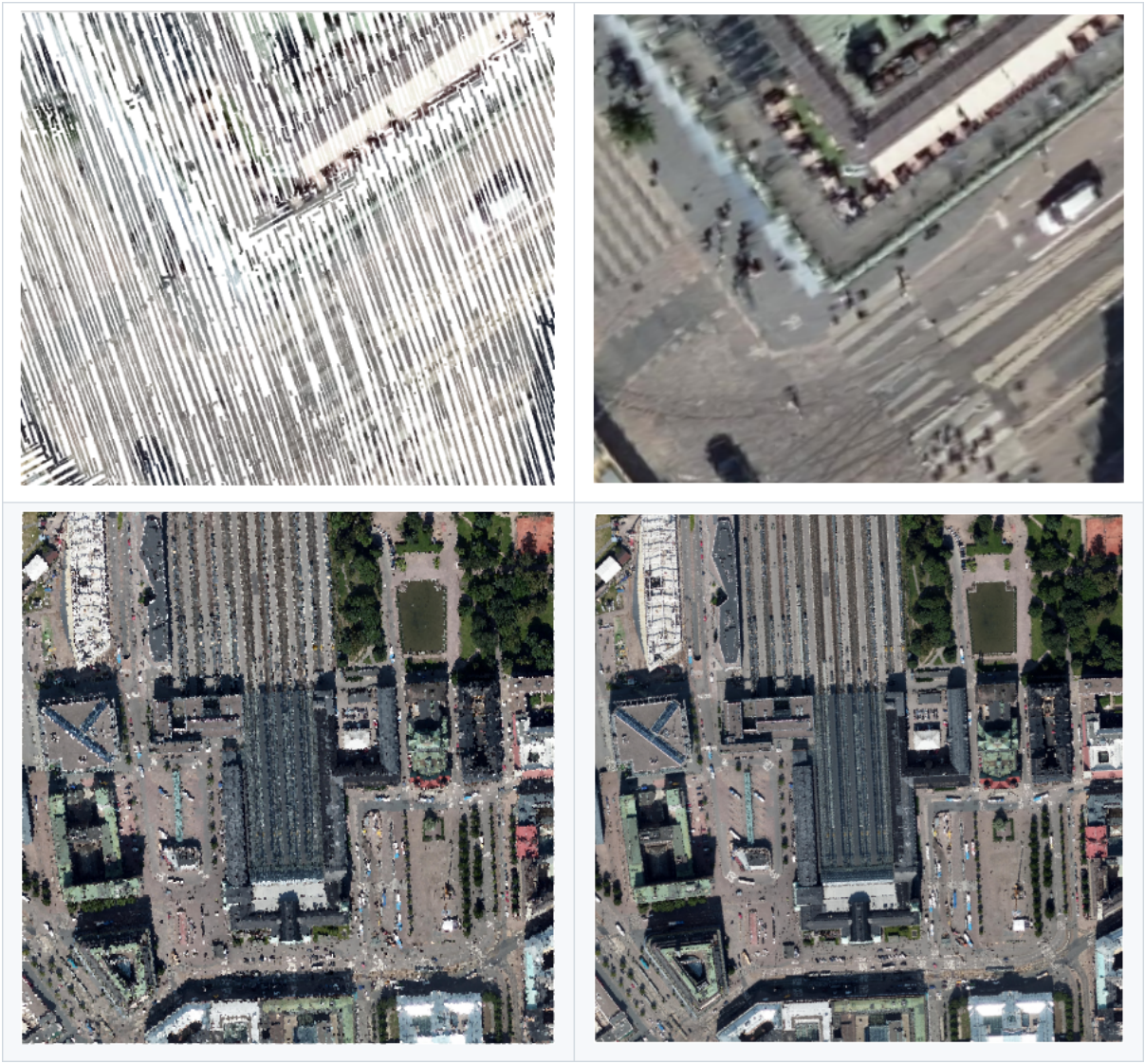


Fig. 16.7: Exempel på RGB-rendering (vänster) kombinerat med triangulering av ytan (höger)

Klassificering Renderer

I renderingen *Classification* visas punktmolnet differentierat med färg på grundval av ett attribut. Alla typer av attribut kan användas (numeriska, strängar, …). Punktmolnsdata innehåller ofta ett fält som kallas `Classification`. Detta innehåller vanligtvis data som bestäms automatiskt genom efterbearbetning, t.ex. om vegetation. Med *Attribute* kan du välja det fält från attributtabellen som ska användas för klassificeringen. Som standard använder QGIS definitionerna i LAS-specifikationen (se tabellen 'ASPRS Standard Point Classes' i PDF-filen på ASPRS hemsida). Uppgifterna kan dock avvika från detta schema; om du är osäker måste du fråga den person eller institution som du fick uppgifterna från om definitionerna.

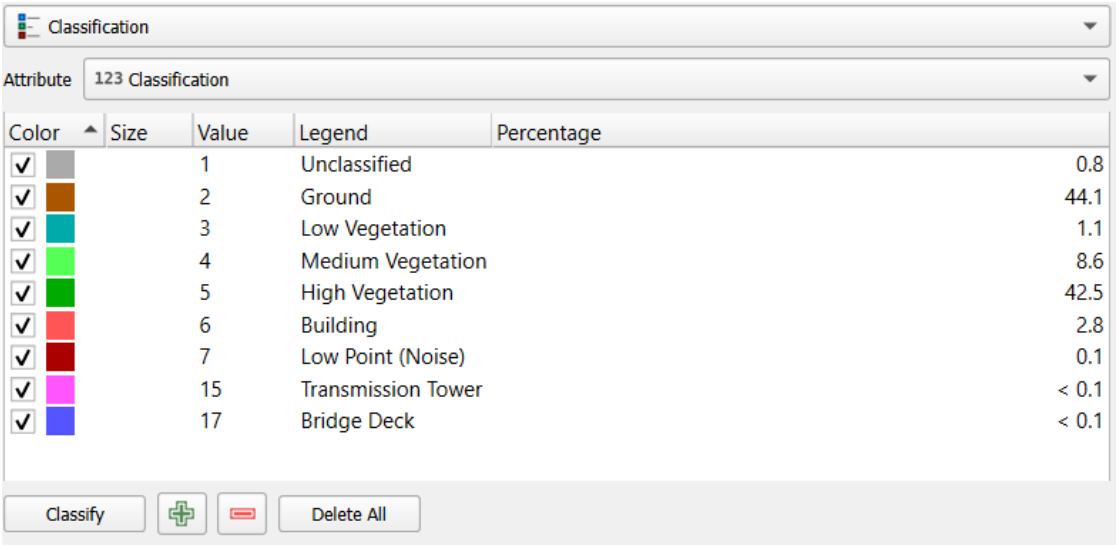


Fig. 16.8: Renderingsprogram för klassificering av punktmoln

I tabellen visas alla använda värden med motsvarande färg och teckenförklaring. I början av varje rad finns en kryssruta (); om den är avmarkerad visas inte längre detta värde på kartan. Genom att dubbelklicka i tabellen kan du ändra:

- *Color*: widgeten *Färgväljare* öppnas
- *Size*: om du tilldelar en storlek på 0 till en kategori kommer den att återgå till att använda standard *punktstorlek* som ställts in för lagret
- den *Värde*
- *Legend*
- kolumnen *Percentage* visar hur kategorin är representerad i lagret.

Du kan också högerklicka på en eller flera rader för att öppna en snabbmeny med alternativ för att ändra färg, opacitet eller punktstorlek.

Under tabellen finns knappar med vilka du kan ändra de standardklasser som genereras av QGIS:

- Med knappen *Classify* kan data klassificeras automatiskt: alla värden som förekommer i attributen och som ännu inte finns i tabellen läggs till
- Med Lägg till och Ta bort kan värden läggas till eller tas bort manuellt
- *Delete All* tar bort alla värden från tabellen

Råd

I panelen *Layers* kan du högerklicka på en klassbladspost i ett lager för att snabbt konfigurera synligheten för motsvarande funktioner.

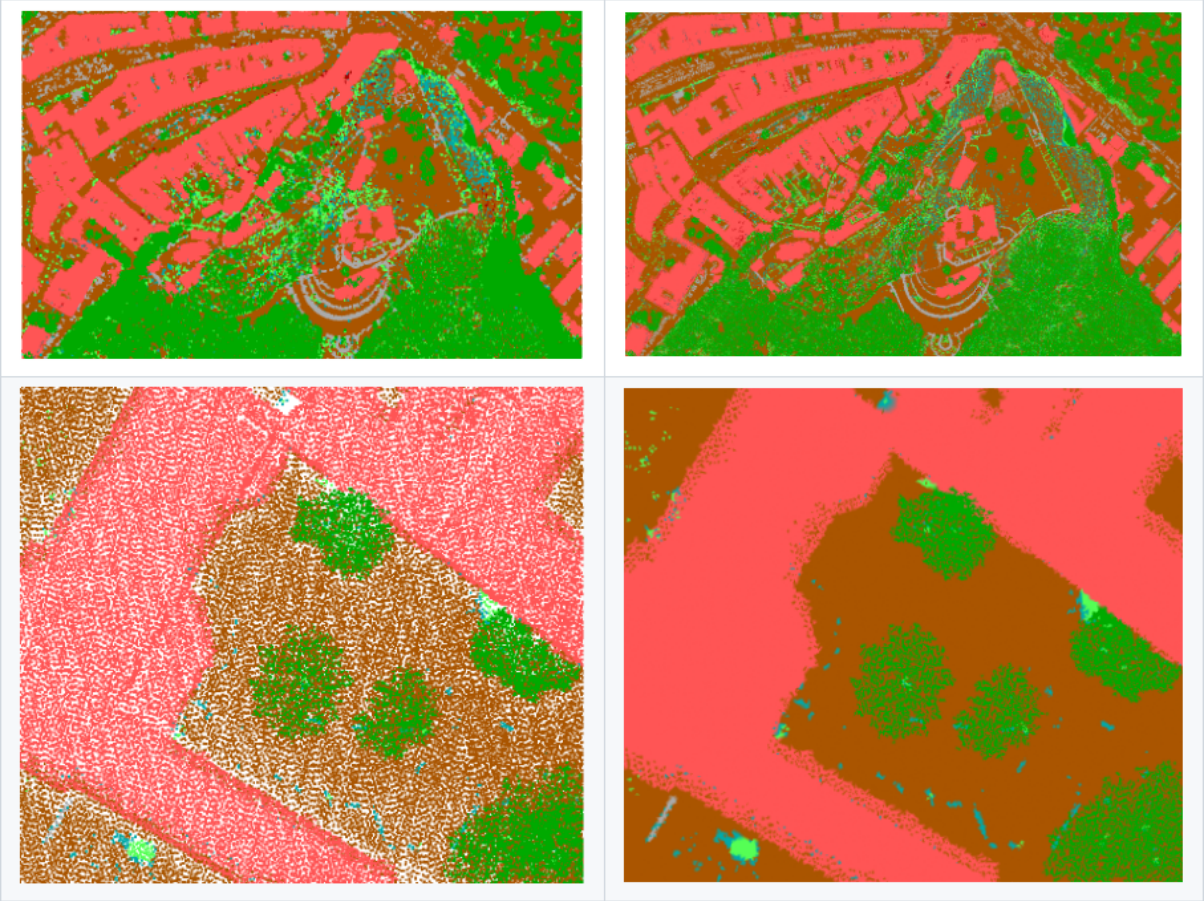


Fig. 16.9: Exempel på klassificeringsrendering (vänster) kombinerat med triangulering av ytan (höger)

Punktsymbol

Under *Point Symbol* kan du ställa in storleken och enheten (t.ex. millimeter, pixlar, tum) som varje datapunkt ska visas med. Antingen *Circle* eller *Square* kan väljas som stil för punkterna.

Rendera som en yta (triangulera)

Markera *Rendera som yta (Triangulera)* för att aktivera triangulering av punktmolnslagret i 2D-vyn. Detta alternativ gör det möjligt att rendera trianglar istället för punkter. Varje punkt behåller sin färg för interpolering i triangeln. Du kan styra den horisontella längden på de beräknade trianglarna:

Genom att markera alternativet *Skippa trianglar längre än* och ställa in tröskelvärdet kan du styra den maximala längden på en sida av trianglarna som ska beaktas i den horisontella planen. Detta kan vara särskilt användbart om du vill identifiera faktiska hål i data.

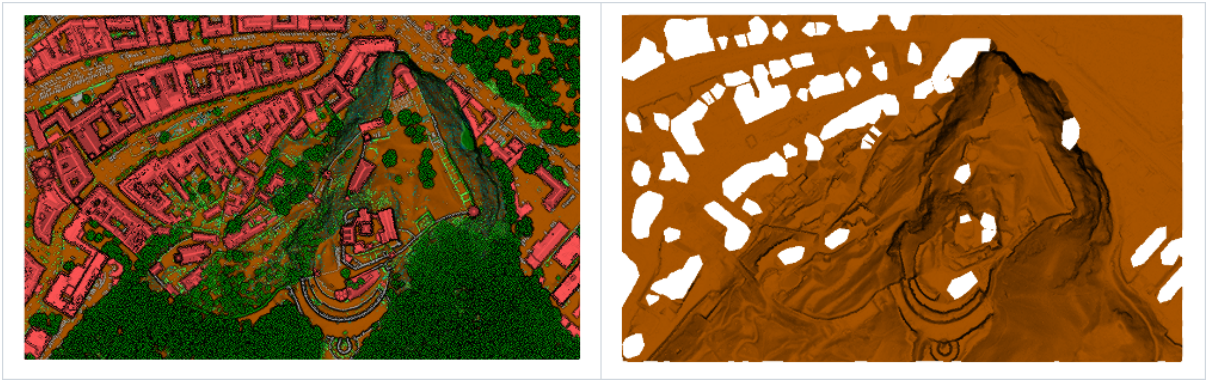


Fig. 16.10: Rendering av data som en yta med kartskuggning (vänster) och med kartskuggning, filtrering av stora trianglar (höger)

Rendering av lager

I avsnittet *Layer Rendering* har du följande alternativ för att ändra renderingen av lagret:

- *Draw order*: gör det möjligt att styra om punktmolnens renderingsordning på 2d-kartans duk ska bero på deras Z-värde. Det är möjligt att rendera :
  - med *Default*-ordningen i vilken punkterna lagras i lagret,
  - från *Underst till överst* (punkter med större Z-värden täcker lägre punkter vilket ger utseendet på ett äkta ortofoto),
  - eller från *Top to bottom* där scenen visas sedd underifrån.
- *Maximalt fel*: Punktmoln innehåller vanligtvis fler punkter än vad som behövs för visningen. Med detta alternativ ställer du in hur tät eller gles visningen av punktmolnet ska vara (detta kan också förstås som ”maximalt tillåtet mellanrum mellan punkter”). Om du ställer in ett stort värde (t.ex. 5 mm) kommer det att finnas synliga mellanrum mellan punkterna. Ett lågt värde (t.ex. 0,1 mm) kan tvinga fram rendering av onödigt många punkter, vilket gör renderingen långsammare (olika enheter kan väljas).
- *Opacitet*: Med det här verktyget kan du göra det underliggande lagret i kartbilden synligt. Använd skjutreglaget för att anpassa synligheten för ditt lager efter dina behov. Du kan också göra en exakt definition av den procentuella synligheten i menyn bredvid skjutreglaget.
- *Blandningsläge*: Med detta verktyg kan du uppnå speciella renderingseffekter. Pixlarna i de överliggande och underliggande lagren blandas med hjälp av de inställningar som beskrivs i *Blandningslägen*.
- *Eye dome lighting*: detta tillämpar skuggningseffekter på kartans canvas för en bättre djupåtergivning. Renderingskvaliteten beror på egenskapen *draw order*; *Default* draw order kan ge suboptimala resultat. Följande parametrar kan kontrolleras:
  - *Strength*: ökar kontrasten, vilket ger bättre djupseende
  - *Distance*: representerar avståndet mellan de använda pixlarna och mittpixeln och gör att kanterna blir tjockare.

16.2.4 egenskaper för 3D-vy

På fliken *3D View* kan du göra inställningar för hur punktmolnet ska återges i 3D-kartor.

3D-renderingslägen

Följande alternativ kan väljas från rullgardinsmenyn längst upp på fliken:

- *Ingen rendering*: Data visas inte
- *Följ 2D-symbologi*: Synkroniserar funktioner som renderas i 3D med *symbologi som tilldelats i 2D*
- *Single Color*: Alla punkter visas i samma *färg* oavsett attribut
- *Attribut efter ramp*: Interpolerar ett givet attribut över en färgramp och tilldelar funktioner deras matchande färg. Se *Attribut för rampåtergivaren*.
- *RGB*: Använd olika attribut för funktionerna för att ställa in de röda, gröna och blå färgkomponenterna som ska tilldelas dem. Se *RGB-rendering*.
- *Classification*: särskiljer punkter med färg på grundval av ett attribut. Se *Klassificering Renderer*.

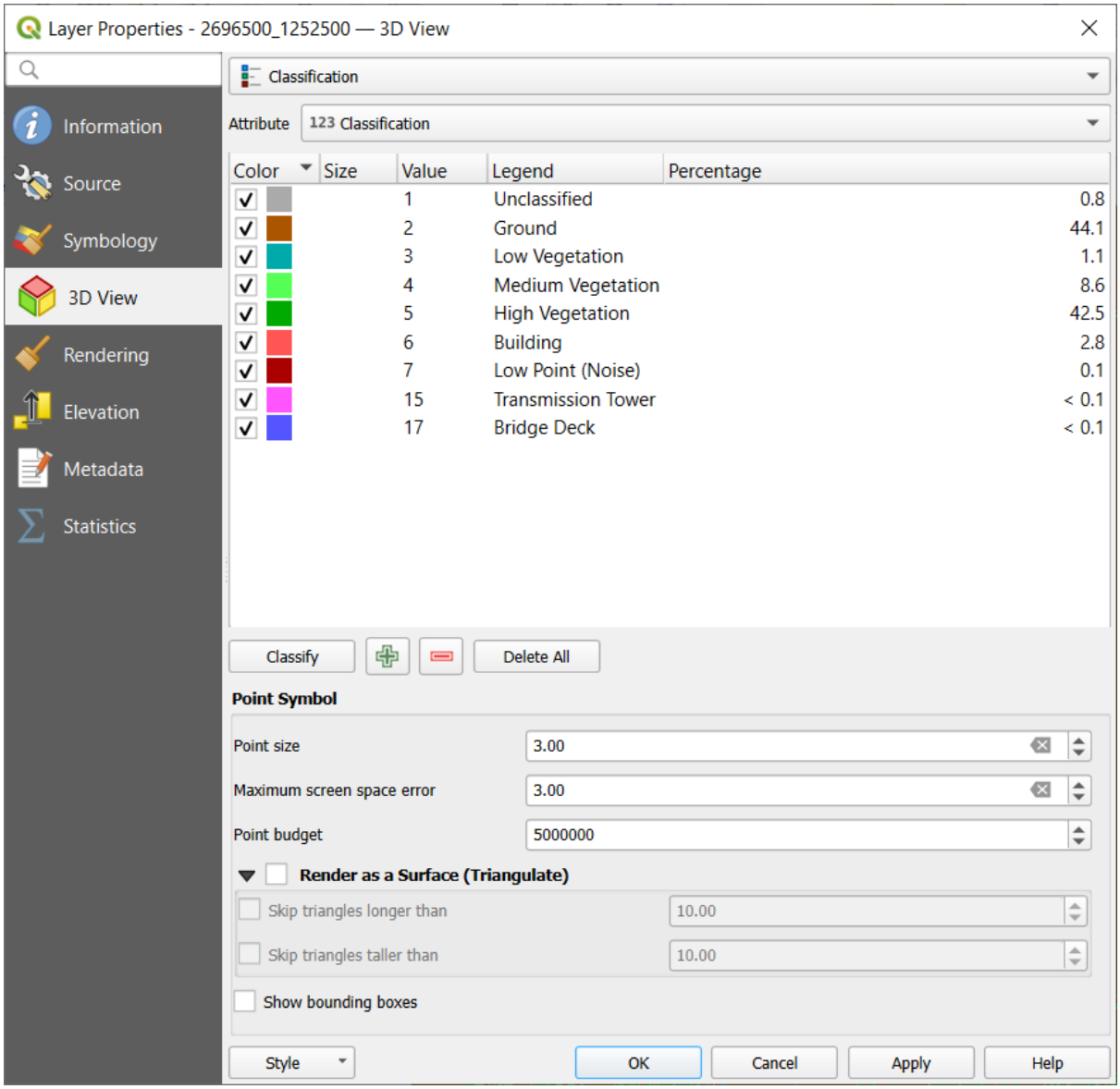


Fig. 16.11: Punktmolnets 3D-vyflik med klassificeringsrenderaren

3D-punktsymbol

I den nedre delen av fliken *3D View* hittar du avsnittet *Point Symbol*. Här kan du göra allmänna inställningar för hela lagret som är desamma för alla renderare. Det finns följande alternativ:

- *Punktstorlek*: Den storlek (i pixlar) som varje datapunkt visas med kan ställas in
- *Maximalt fel på skärmutrymmet*: Med detta alternativ ställer du in hur tät eller gles visningen av punktmolnet ska vara (i pixlar). Om du anger ett stort värde (t.ex. 10) kommer det att finnas synliga luckor mellan punkterna; ett lågt värde (t.ex. 0) kan tvinga fram rendering av onödigt många punkter, vilket gör renderingen långsammare (du hittar mer information i *Symbology Maximum error*).
- *Punktbudget*: För att undvika långa renderingar kan du ställa in det maximala antalet punkter som ska renderas
- Markera *Render as surface (Triangulate)* för att rendera punktmolnslagret i 3D-vyn med en solid yta som erhålls genom triangulering. Du kan styra dimensionerna på de beräknade trianglarna:
  - *Hoppa över trianglar som är längre än* ett tröskelvärde: anger i den horisontella planen den maximala

längden på en sida av trianglarna som ska beaktas

  - *Hoppa över trianglar som är högre än* ett tröskelvärde: anger i den vertikala planen den maximala höjden på en sida av trianglarna som ska beaktas

- *Visa begränsningsrutor*: Speciellt användbart för felsökning, visar begränsningsrutor för noder i hierarkin

16.2.5 Rendering av egenskaper

I grupprutan *Scale dependent visibility* kan du ställa in skalan *Maximum (inclusive)* och *Minimum (exclusive)* och definiera ett skalområde inom vilket objekten ska vara synliga. Utanför detta intervall är de dolda. Knappen Set to current canvas scale hjälper dig att använda den aktuella kartans skala som gräns för synlighetsintervallet. Se *Väljare för synlighetsskala* för mer information.

Observera

Du kan också aktivera skalberoende synlighet på ett lager från panelen *Layers*: högerklicka på lagret och välj *Set Layer Scale Visibility* i snabbmenyn.

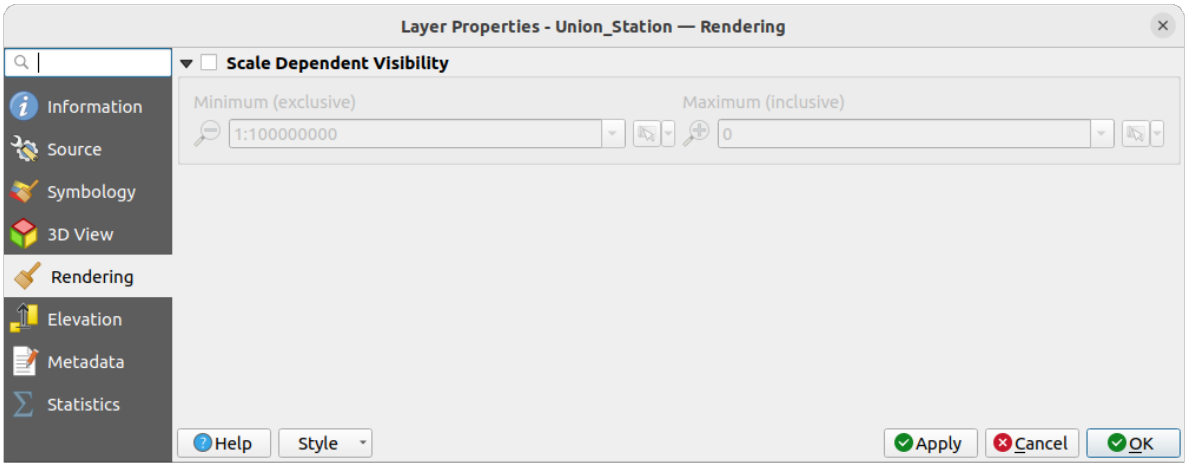


Fig. 16.12: Fliken Rendering av punktmoln

16.2.6 Elevation Egenskaper

På fliken *Elevation* kan du ställa in korrigeringar för Z-värdena i data. Detta kan vara nödvändigt för att justera höjden på data i 3D-kartor och dess utseende i *profile tool charts*. Det finns följande alternativ:

- *Vertikalt referenssystem*: Om CRS för punktmolnslagret är ett sammansatt CRS (inklusive en Z-dimension), kommer det vertikala CRS som används för lagret automatiskt att härledas från den vertikala komponenten i lagrets CRS. I det här fallet kan du inte manuellt ställa in en annan vertikal CRS och alternativet att ändra den kommer att inaktiveras. Om punktmolnslagret använder ett horisontellt (2D) CRS (vilket dock är ovanligt) kan du manuellt välja ett specifikt vertikalt CRS genom att klicka på Välj CRS. Vertikala referenssystem stöds för punktmolnlager i:

  - *Höjningsprofiler*
  - *Identifiera verktygsresultat*
  - *3D-kartvyer*

- Under *Elevation*-gruppen:

  - Du kan ställa in en *Scale*: Om `10` anges här visas en punkt som har värdet Z = `5` på en höjd av `50`.
  - En *förskjutning* till z-nivån kan anges. Detta är användbart för att matcha olika datakällor i höjdled med varandra. Som standard används det lägsta z-värdet i data som detta värde. Detta värde kan också återställas med knappen Uppdatera i slutet av raden.
- Under *Profile Chart Accuracy* hjälper *Maximum error* dig att kontrollera hur tätt eller glest punkterna ska återges i höjdprofilen. Större värden resulterar i en snabbare generering med färre punkter inkluderade.
- Under *Profile Chart Appearance* kan du styra poängvisningen:
  - *Point size*: den storlek som punkterna ska återges med, i enheter som stöds (millimeter, kartenheter, pixlar, …)
  - *Style*: om punkterna ska återges som *Circle* eller *Square*
  - Använd en enda *Color* på alla punkter som syns i profilvyn
  - Markera *Respect layer's coloring* för att istället visa punkterna med den färg som tilldelats via deras *2D symbology*
  - *Apply opacity by distance from curve effect*, minskar opaciteten för punkter som ligger längre bort från profilkurvan

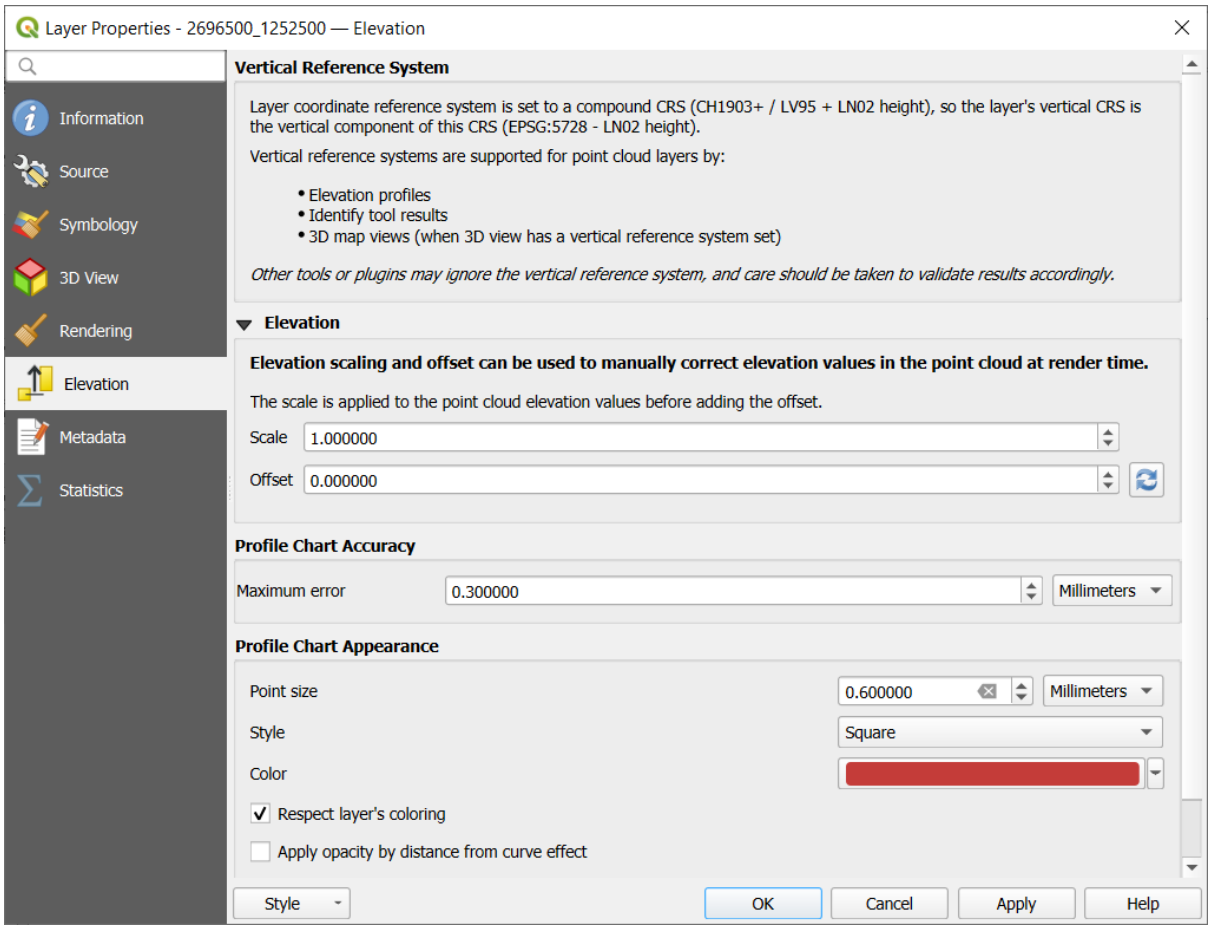


Fig. 16.13: Fliken för punktmolnets höjd

16.2.7 Egenskaper för metadata

Fliken *Metadata* ger dig alternativ för att skapa och redigera en metadatarapport om ditt lager. Se *Metadata* för mer information.

16.2.8 Statistik Egenskaper

På fliken *Statistics* kan du få en översikt över attributen i ditt punktmoln och deras fördelning.

Längst upp hittar du avsnittet *Attribute Statistics*. Här listas alla attribut som finns i punktmolnet, samt några av deras statistiska värden: *Minimum*, *Maximum*, *Mean*, *Standard Deviation*

Om det finns ett attribut *Classification*, så finns det en annan tabell i den nedre delen. Här listas alla värden som ingår i attributet, samt deras absoluta *Count* och relativa *%* abundans.

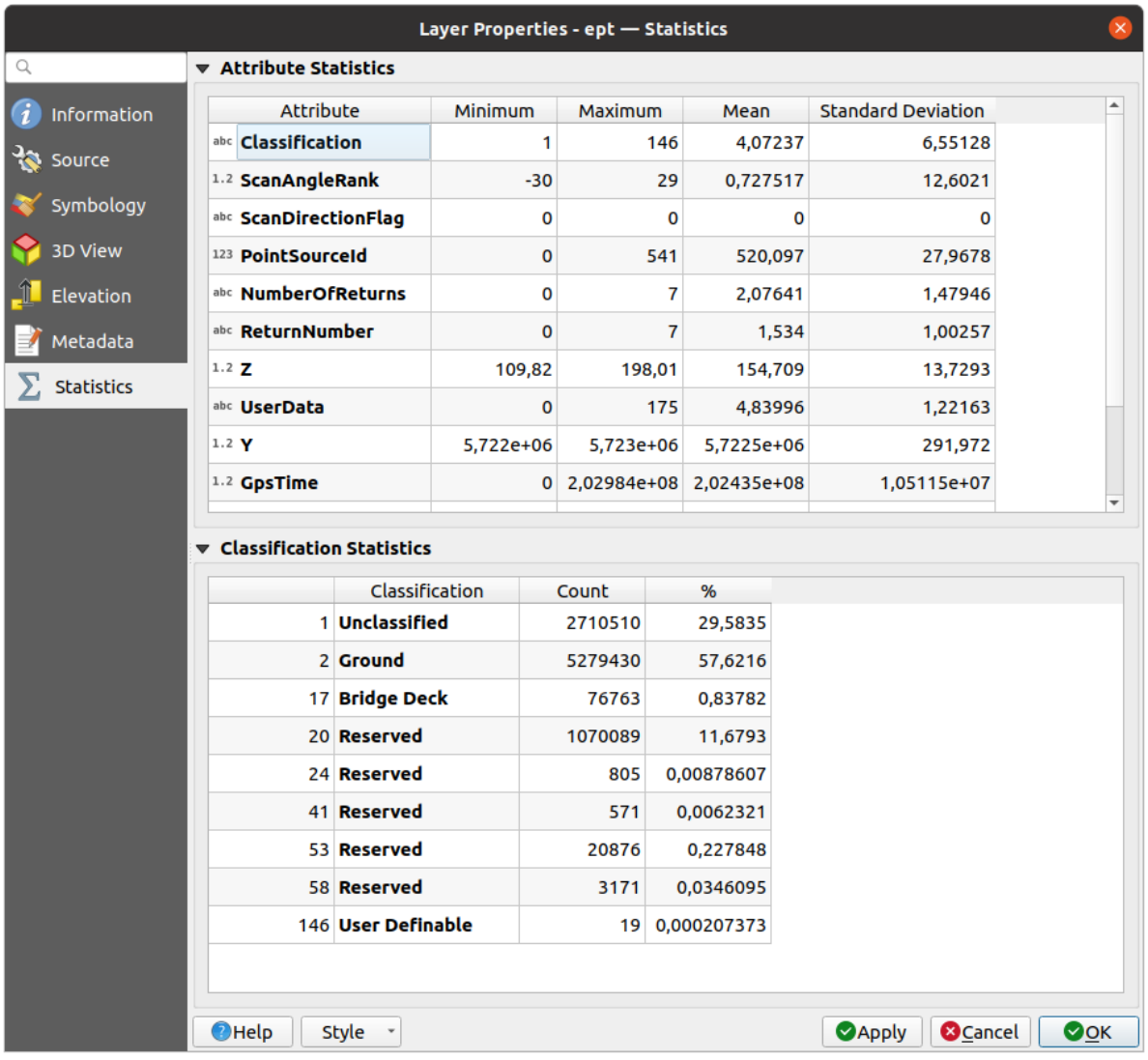


Fig. 16.14: Fliken Statistik för punktmoln

16.3 Virtuellt punktmoln

Lidarundersökningar av större områden är ofta dataset på flera terabyte med många miljarder punkter. Att ha så stora datamängder representerade som en enda punktmolnsfil är inte praktiskt på grund av svårigheterna med lagring, överföring, visning och analys. Punktmolnsdata lagras och distribueras därför vanligtvis uppdelade i kvadratiska rutor (t.ex. ”1 km x 1 km”), där varje ruta har en mer hanterbar filstorlek (t.ex. ~ 200 MB när den komprimeras).

Tiling av data löser problemen med datastorlek, men det medför problem vid bearbetning eller visning av ett intresseområde som inte helt ryms inom en enda tile. Användarna måste utveckla arbetsflöden som tar hänsyn till flera tiles och särskild omsorg måste ägnas åt att hantera data nära tiles kanter för att undvika oönskade artefakter i utdata. På samma sätt blir det besvärligt att ladda många enskilda filer och använda samma symbologi när man tittar på punktmolnsdata.

Här är ett exempel på flera punktmoln som laddats i QGIS. Varje kakelplatta stylas baserat på kakelplattans min/max Z-värden, vilket skapar synliga artefakter på kakelplattans kanter. Stylingen måste justeras för varje lager separat:

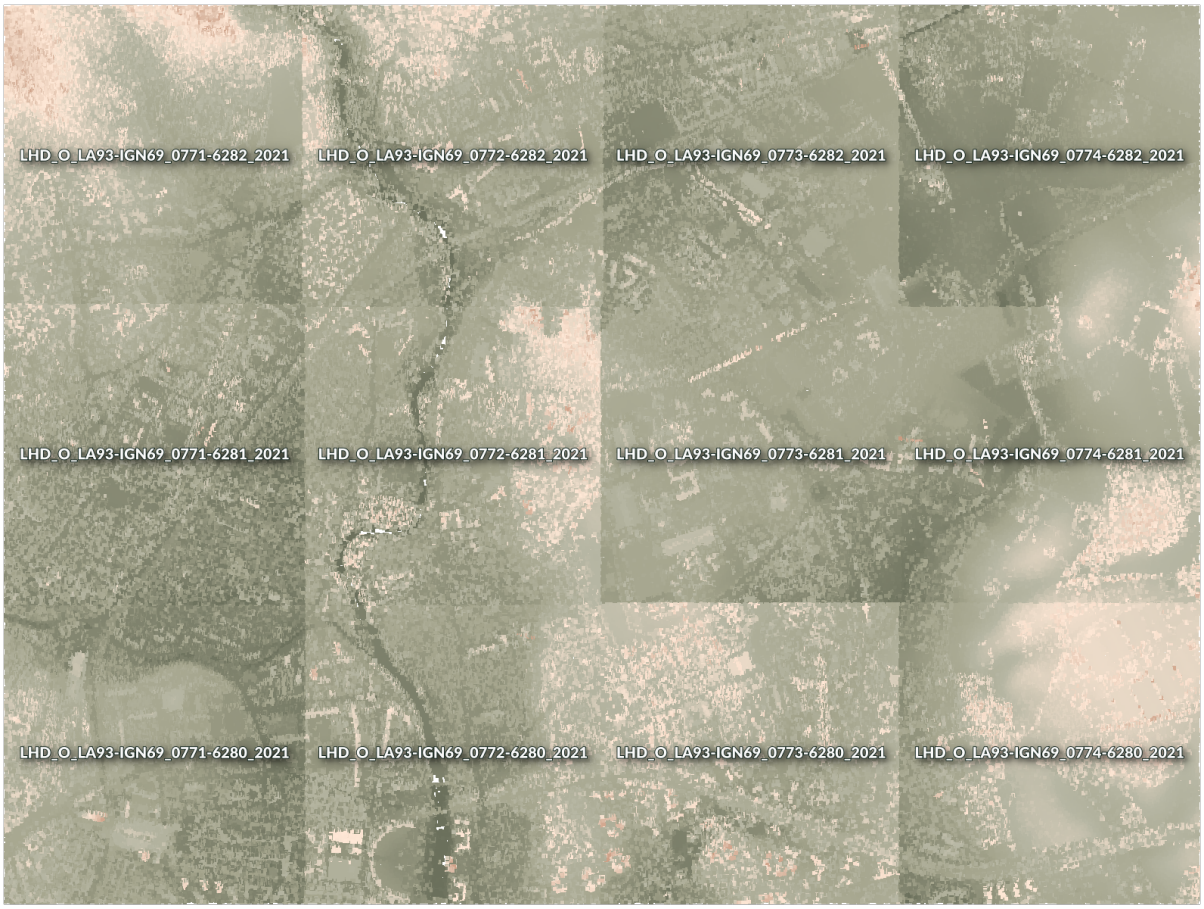


Fig. 16.15: Enskilda punktmoln laddas, med artefakter på kanterna

I GIS-världen är många användare bekanta med begreppet virtuella raster. Ett virtuellt raster är en fil som helt enkelt refererar till andra rasterfiler med faktiska data. På så sätt behandlar GIS-programvaran hela datasetet som består av många filer som ett enda rasterlager, vilket gör det mycket enklare att visa och analysera alla raster som listas i den virtuella filen.

Med utgångspunkt i konceptet virtuella raster från GDAL är **virtuella punktmoln (VPC)** ett filformat som refererar till andra punktmolnsfiler. Programvara som stöder virtuella punktmoln hanterar hela den kaklade datamängden som en enda datakälla.

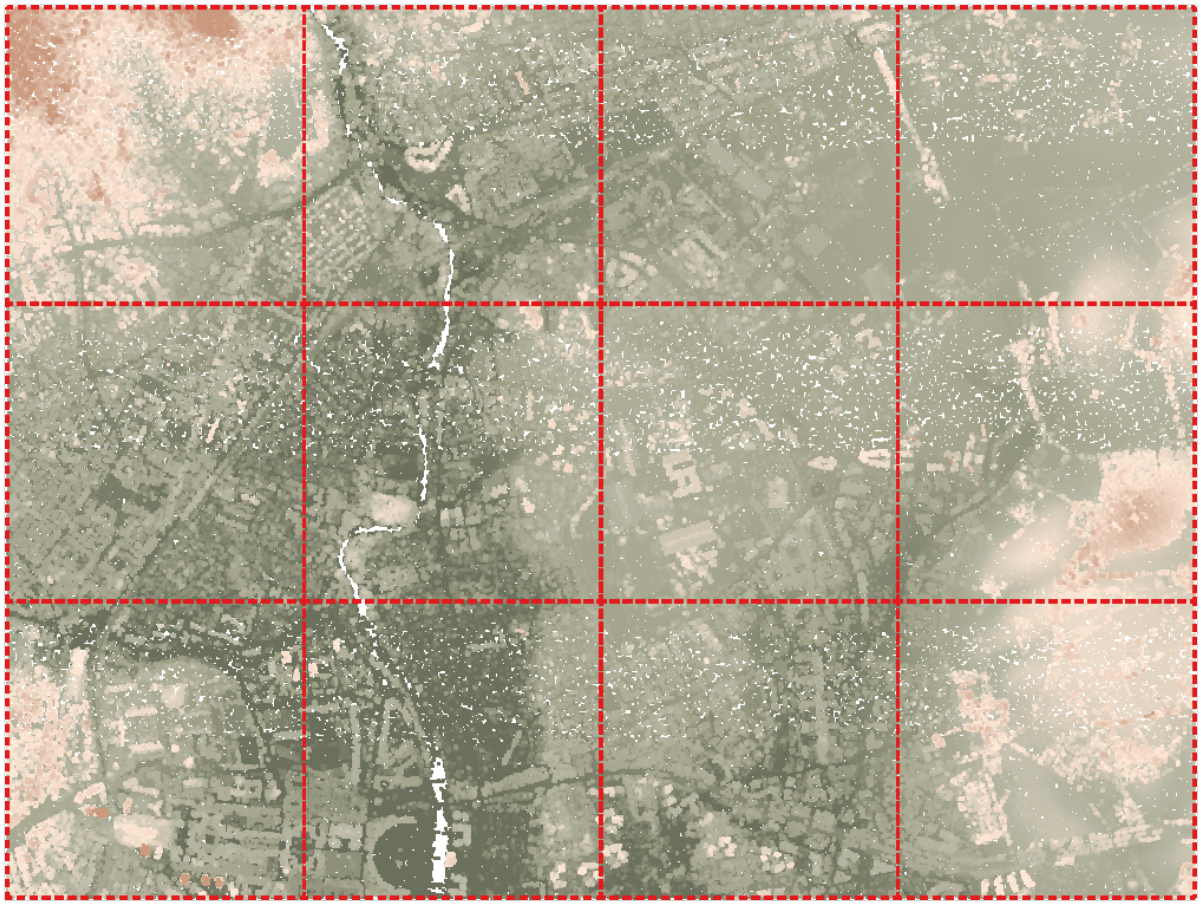


Fig. 16.16: Det virtuella punktmolnet

Att visa och manipulera virtuella punktmoln är mycket smidigare och enklare.

I grund och botten är en virtuell punktmolnsfil en enkel JSON-fil med tillägget `.vpc`, som innehåller referenser till faktiska datafiler (t.ex. `.LAS`, `.LAZ` eller `.COPC`-filer) och ytterligare metadata som extraherats från filerna. Även om det är möjligt att skriva VPC-filer för hand, rekommenderas det starkt att skapa dem med hjälp av ett automatiserat verktyg, t.ex:

- Algoritmen Processing *Skapa ett virtuellt punktmoln (VPC)*
- Kommandot `build_vpc` i verktyget PDAL wrench

För mer information hänvisas till ”VPC-specifikationen” <https://github.com/PDAL/wrench/blob/main/vpc-spec.md>`_ som också innehåller bästa praxis och valfria tillägg (t.ex. översikter).

KAPITEL 17

Arbeta med 3D-plattor

17.1 Vad är 3D-plattor?

3D tiles är en specifikation för streaming och rendering av storskaliga geospatiala 3D-dataset. De använder en hierarkisk struktur för att effektivt hantera och visa 3D-innehåll och optimera prestanda genom att dynamiskt ladda lämpliga detaljnivåer. Denna teknik används ofta inom stadsplanering, arkitektur, simulering, spel och virtuell verklighet och erbjuder en standardiserad och interoperabel lösning för visualisering av komplexa geografiska data.

Fig. 17.1: Exempel på 3D-plattor

För närvarande stöder QGIS två format av 3D-rutor:

- ”Cesium 3D tiles <https://cesium.com/why-cesium/3d-tiles/>`_, används främst för komplexa 3D-modeller av byggnader eller hela städer. Sådana datamängder kan tillhandahållas av molnbaserade plattformar som Cesium Ion eller Google (*Photorealistic 3D Tiles*).
- Quantized Mesh tiles, används för terränghöjdsdata.

Se *Använda 3D-tjänster för kaklade scener* för instruktioner om hur du lägger till dessa datakällor i QGIS.

17.2 3D Tiles Egenskaper

Dialogrutan 3D tiles *Lageregenskaper* innehåller följande avsnitt:

[1] Finns även i *Layer stylingpanel*

17.2.1 Informationsegenskaper

Fliken *Information* är skrivskyddad och utgör en intressant plats för att snabbt hämta sammanfattad information och metadata om det aktuella lagret. Tillhandahållen information är:

- baserat på lagrets leverantör: namn, URL, källtyp och sökväg, antal zoomnivåer
- anpassade egenskaper, som används för att lagra ytterligare information om lagret i det aktiva projektet. Fler egenskaper kan skapas och hanteras med hjälp av PyQGIS, särskilt genom metoden setCustomProperty().
- det koordinatsatta referenssystemet: namn, enheter, metod, noggrannhet, referens (dvs. om det är statiskt eller dynamiskt)
- plockad från *fyllda metadata*: access, extents, länkar, kontakter, historia…

17.2.2 Källegenskaper

Fliken *Källa* visar grundläggande information om den valda 3D-plattan, inklusive:

- det *Layernamn* som ska visas i *Layers Panel*;
- *Koordinatreferenssystem*: Visar lagrets *Coordinate Reference System (CRS)*. Du kan ändra lagrets CRS genom att välja ett nyligen använt i listrutan eller klicka på knappen Välj CRS (se *Väljare för referenssystem för koordinater*). Använd denna process endast om lagrets CRS är fel eller inte har angetts.

17.2.3 Egenskaper för symbologi

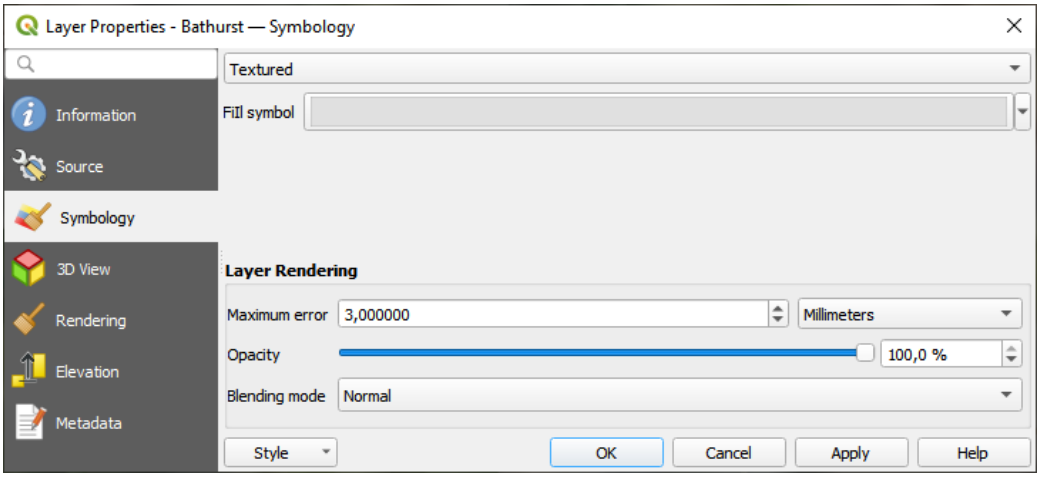


Fig. 17.2: symbologi för lager av 3D-kakelplattor

Som standard är lagret utformat med hjälp av textur, men du kan ändra det så att du ser wireframe-nätet bakom scenen genom att välja *Wireframe* i rullgardinsmenyn. Du kan också ändra nätets fyllnings- och linjesymboler på samma sätt som för vektorpolygoner. Om du markerar *Use texture colors* kommer varje mesh-element att återges med medelvärdet för hela texturen. Det här är ett bra alternativ när man arbetar med ett stort dataset och vill få en snabb överblick över datan.

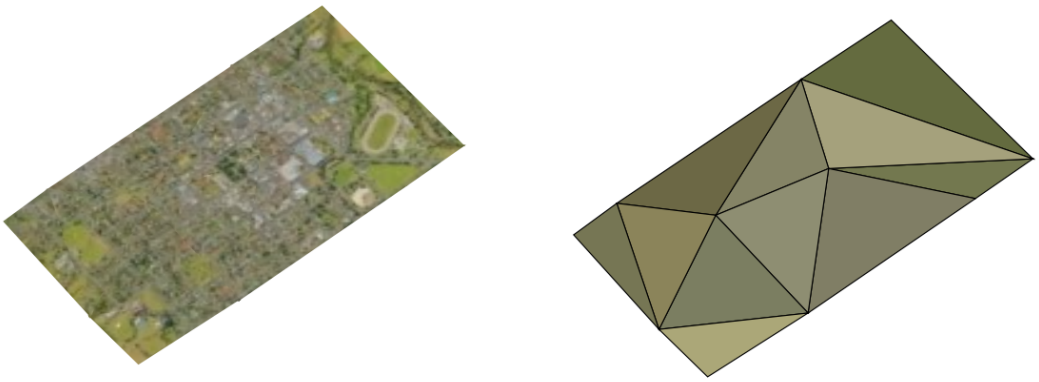


Fig. 17.3: 3D-plattor - strukturerade och trådformade

För att visa data kan du öppna *Ny 3D-kartvy*.

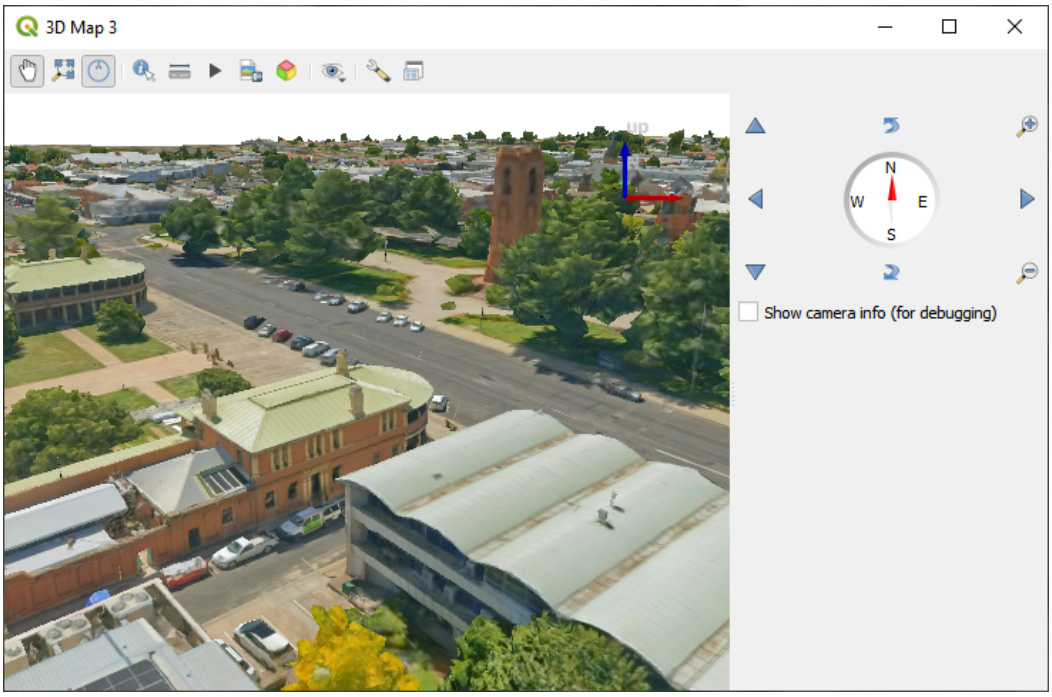


Fig. 17.4: 3D-kartvy

På fliken *Symbology* kan du också ställa in några alternativ som alltid påverkar alla funktioner i lagret:

- *Maximalt fel*: Denna parameter bestämmer detaljnivån som visas i 3D-modellen. I likhet med punktmoln innehåller 3D-plattor ofta mer information än vad som är nödvändigt för visuell representation. Genom att justera den här inställningen styr du balansen mellan visningstäthet och renderingshastighet. Ett större värde (t.ex. 5 mm) kan ge märkbara luckor mellan elementen, medan ett mindre värde (t.ex. 0,1 mm) kan leda till att ett alltför stort antal detaljer återges, vilket kan göra renderingsprocessen långsammare. Olika enheter kan väljas för att skräddarsy inställningen efter dina specifika behov.
- *Opacitet*: Justera synligheten för det underliggande lagret på kartbilden med hjälp av det här verktyget. Använd

skjutreglaget för att skräddarsy synligheten för ditt scenlager enligt dina önskemål. Alternativt kan du ange den exakta procentandelen synlighet via menyn bredvid skjutreglaget.

- *Blandningsläge*: Med dessa verktyg kan du uppnå speciella renderingseffekter som du kanske tidigare bara kände till från grafikprogram. Pixlarna i de över- och underliggande lagren blandas genom de inställningar som beskrivs i *Blandningslägen*.

17.2.4 egenskaper för 3D-vy

- *Maximum screen space error* bestämmer tröskeln för att byta ut terrängplattor mot mer detaljerade plattor (och vice versa) - dvs. hur snart 3D-vyn ska använda plattor av högre kvalitet. Lägre siffror innebär fler detaljer i scenen på bekostnad av ökad renderingskomplexitet.
- *Visa begränsningsrutor* Visa 3D-begränsningsrutor för terrängplattorna (användbart för felsökning av terrängproblem).

17.2.5 Rendering av egenskaper

Under *Skalaberoende synlighet* kan du ställa in skalorna *Maximum (inklusive)* och *Minimum (exklusive)* och definiera ett skalområde där objekten ska vara synliga. Utanför detta intervall är de dolda. Knappen Set to current canvas scale hjälper dig att använda den aktuella kartans canvasskala som gräns för synlighetsintervallet. Se *Väljare för synlighetsskala* för mer information.

17.2.6 Elevation Egenskaper

På fliken *Elevation* finns alternativ för att styra lagrets höjdegenskaper i en *3D-kartvy*. Specifikt kan du ställa in:

- *Elevation Surface*: hur Z-värdena för 3D-lagrets vertikaler ska tolkas som terränghöjd. Du kan använda en *Scale*-faktor och en *Offset*.

17.2.7 Egenskaper för metadata

Fliken *Metadata* ger dig alternativ för att skapa och redigera en metadatarapport om ditt lager. Se *Metadata* för mer information.

KAPITEL 18

Lägga ut kartorna

Med Print Layouts and Reports kan du skapa kartor och atlaser och skriva ut dem eller spara dem som bild-, PDF- eller SVG-filer.

18.1 Översikt över utskriftslayouten

Utskriftslayouten ger växande layout- och utskriftsfunktioner. Du kan lägga till element som QGIS 2D- eller 3D-karta, textetiketter, bilder, legender, skalstreck, grundläggande former, pilar, attribut och enkla tabeller, höjdprofiler och HTML-ramar… Du kan ändra storlek, gruppera, rikta in, positionera och rotera varje element och justera deras egenskaper för att skapa din layout. Layouten kan skrivas ut eller exporteras till bildformat, PostScript, PDF eller till SVG. Du kan spara layouten som en mall och ladda den igen i en annan session. Slutligen kan du generera flera kartor baserat på en mall med hjälp av atlasgeneratorn.

18.1.1 Exempelsession för nybörjare

Innan du börjar arbeta med utskriftslayouten måste du ladda några raster- eller vektorlager i QGIS kartkanvas och anpassa deras egenskaper så att de passar dina egna önskemål. När allt är renderat och symboliserat som du vill, klicka på ikonen New Print Layout i verktygsfältet *Project* eller välj *Projekt* ► *New Print Layout*. Du kommer att bli ombedd att välja en titel för den nya layouten.

För att visa hur du skapar en karta följer du följande instruktioner.

1. På vänster sida väljer du knappen Lägg till karta i verktygsfältet och ritar en rektangel på duken genom att hålla ned vänster musknapp. Inuti den ritade rektangeln läggs QGIS-kartvyn till på duken.

2. Välj knappen Add scalebar i verktygsfältet och klicka med vänster musknapp på utskriftslayoutens duk. En scalebar läggs till på duken.

3. Välj knappen Lägg till förklaring i verktygsfältet och rita en rektangel på duken genom att hålla ned vänster musknapp. Legenden kommer att ritas inuti den ritade rektangeln.

4. Välj ikonen Select/Move item för att markera kartan på duken och flytta den lite.

5. Medan kartobjektet fortfarande är markerat kan du också ändra storleken på kartobjektet. Klicka med vänster musknapp nedtryckt i en vit liten rektangel i ett av hörnen på kartobjektet och dra den till en ny plats för att ändra storleken.
6. Klicka på panelen *Item Properties* till vänster och leta upp inställningen för orientering. Ändra värdet för inställningen *Map orientation* till ’15.00°’. Du bör se att kartobjektets orientering ändras.
7. Nu kan du skriva ut eller exportera din utskriftslayout till bildformat, PDF eller SVG med exportverktygen i menyn *Layout*.
8. Slutligen kan du spara din utskriftslayout i projektfilen med hjälp av knappen Save Project.

Du kan lägga till flera element i utskriftslayouten. Det är också möjligt att ha mer än en kartvy eller teckenförklaring eller skalstreck i utskriftslayoutens canvas, på en eller flera sidor. Varje element har sina egna egenskaper och, när det gäller kartan, sin egen utsträckning. Om du vill ta bort något element från layoutkanvasen kan du göra det med `Delete` eller `Backspace`.

18.1.2 Layoutansvarig

Layout Manager är huvudfönstret för att hantera utskriftslayouter i projektet. Det ger dig en översikt över befintliga utskriftslayouter och rapporter i projektet och erbjuder verktyg för att:

- sök efter en layout;
- lägga till ny utskriftslayout eller ny rapport från början, mall eller duplicering av en befintlig;
- byta namn på eller radera någon av dem;
- öppna dem i projektet.

För att öppna dialogrutan för layouthanteraren:

- från QGIS huvuddialog, välj *Project* ► *Layout Manager…* menyn eller klicka på Layout Manager knappen i *Project Toolbar*;
- från en utskriftslayout eller rapportdialog, välj menyn *Layout* ► *Layout Manager…* eller klicka på knappen Layout Manager i verktygsfältet *Layout Toolbar*.

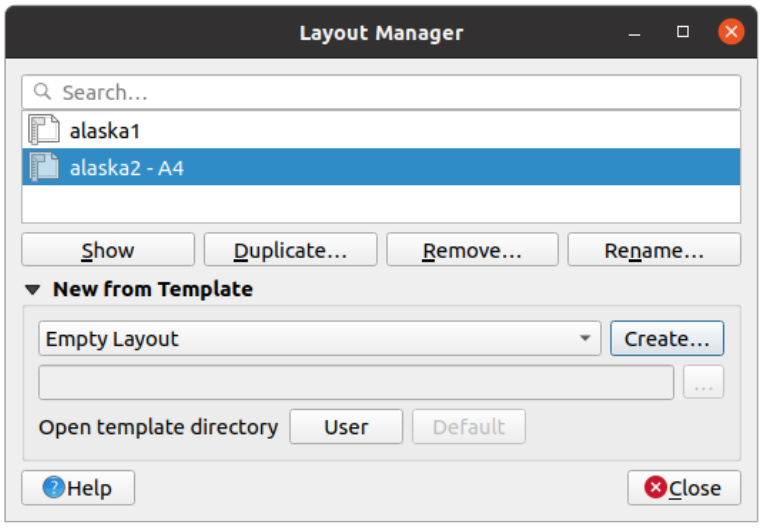


Fig. 18.1: Hanteraren för utskriftslayout

Layouthanteraren visar i sin övre del alla tillgängliga utskriftslayouter eller rapporter i projektet med verktyg för att:

- visa urvalet: du kan välja flera rapporter och/eller utskriftslayouter och öppna dem med ett enda klick. Dubbelklicka på ett namn för att öppna det;

- duplicera den valda utskriftslayouten eller rapporten (endast tillgängligt om ett objekt är markerat): en ny dialog skapas med den valda som mall. Du kommer att uppmanas att välja en ny titel för den nya layouten;
- byta namn på rapporten eller layouten (endast tillgängligt om ett objekt är markerat): du uppmanas att välja en ny titel för layouten;
- ta bort layouten: den eller de valda utskriftslayouterna tas bort från projektet.

I den nedre delen är det möjligt att skapa nya utskriftslayouter eller rapporter från grunden eller från en mall. Som standard kommer QGIS att leta efter mallar i mappen *användarprofil* och applikationsmallkatalogerna (nås med de två knapparna längst ner i ramen) men även i alla mappar som deklarerats som *Sökväg(ar) för att söka efter extra utskriftsmallar* i *Inställningar* ► *Alternativ* ► *Layouter*. De mallar som hittas listas i kombinationsrutan. Välj ett objekt och tryck på knappen *Create* för att generera en ny rapport eller utskriftslayout.

Du kan också använda layoutmallar från en egen mapp; i så fall väljer du *specifik* i rullgardinsmenyn för mallar, bläddrar till mallen och trycker på *Create*.

Tips

Skapa mallbaserade utskriftslayouter från Browser-panelen

Dra och släpp en utskriftslayoutmall `.qpt` från valfri filbläddrare på kartbilden eller dubbelklicka på den i *Browser panel* genererar en ny utskriftslayout från mallen.

18.1.3 Menyer, verktyg och paneler i utskriftslayouten

När du öppnar utskriftslayouten får du en tom duk som representerar pappersytan när du använder utskriftsalternativet. Inledningsvis hittar du knappar till vänster bredvid duken för att lägga till utskriftslayoutobjekt: 2D- eller 3D-kartor, textetiketter, bilder, legender, skalstreck, grundläggande former, pilar, attribut eller enkla tabeller, HTML-ramar, höjdprofiler, … I det här verktygsfältet finns också knappar för att navigera, zooma in på ett område och panorera vyn på layouten samt knappar för att välja ett layoutobjekt och flytta kartobjektens utbredning.

Fig. 18.2 visar den första vyn av utskriftslayouten innan några element läggs till.

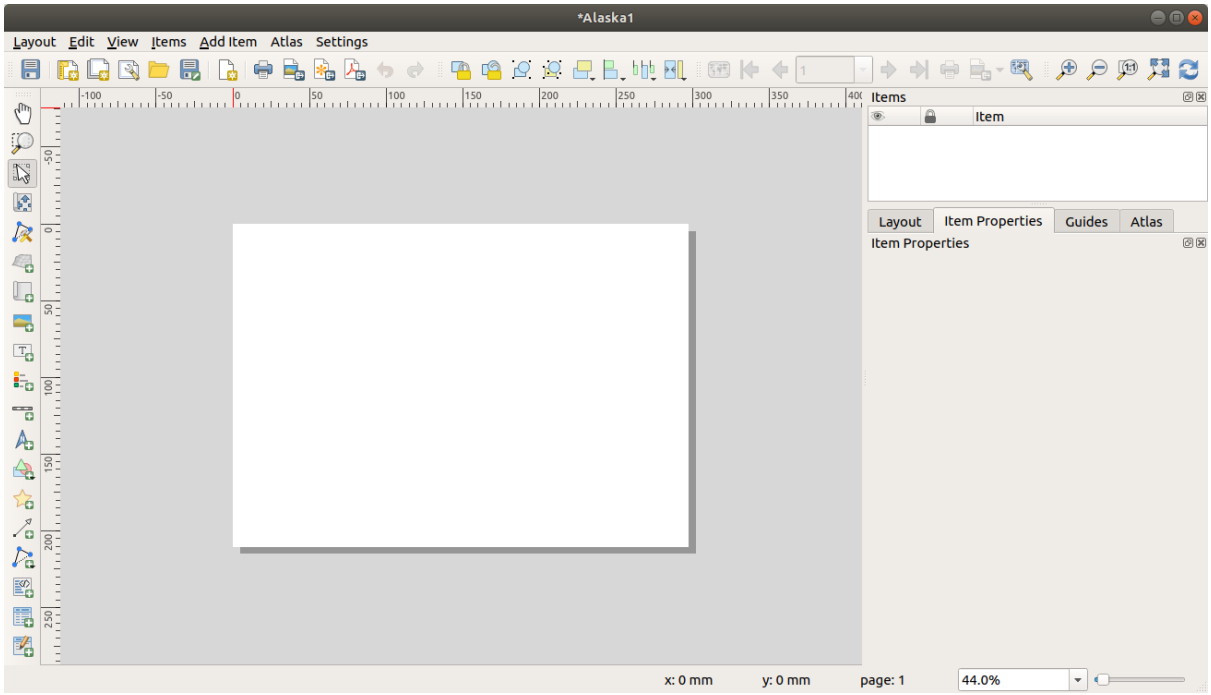


Fig. 18.2: Layout för utskrift

Till höger om duken finns två uppsättningar paneler. Den övre innehåller panelerna *Items* och *Undo History* och den nedre innehåller panelerna *Layout*, *Item properties* och *Atlas generation*.

- Panelen *Items* innehåller en lista över alla utskriftslayoutobjekt som lagts till i Canvas och sätt att interagera globalt med dem (se *Panelen för objekt* för mer information).
- Panelen *Undo History* visar historiken för alla ändringar som har gjorts i layouten. Med ett musklick är det möjligt att ångra och göra om layoutstegen fram och tillbaka till en viss status.
- I panelen *Layout* kan du ställa in allmänna parametrar som ska gälla för layouten när du exporterar eller arbetar i den (se *Layoutpanelen* för mer information);
- Panelen *Item Properties* visar egenskaperna för det valda objektet. Klicka på ikonen Select/Move item för att välja ett objekt (t.ex. en förklaring, skalstapel eller etikett) på duken. Klicka sedan på panelen *Item Properties* och anpassa inställningarna för det valda objektet (se *Layoutobjekt* för detaljerad information om varje objekts inställningar).
- Med panelen *Atlas* kan du aktivera generering av en atlas för den aktuella layouten och få tillgång till dess parametrar (se *Skapa en Atlas* för detaljerad information om hur atlas genereras).

I den nedre delen av utskriftslayoutfönstret finns ett statusfält med musposition, aktuellt sidnummer, en kombinationsruta för att ställa in zoomnivån, antalet valda objekt om tillämpligt och, när det gäller atlasgenerering, antalet funktioner.

I den övre delen av utskriftslayoutfönstret finns menyer och andra verktygsfält. Alla verktyg för utskriftslayout finns tillgängliga i menyer och som ikoner i ett verktygsfält.

Verktygsfälten och panelerna kan slås av och på med hjälp av höger musknapp över valfritt verktygsfält, eller genom *View* ► *Toolbars* ► eller *View* ► *Panels* ►.

Menyer och verktyg

Layoutmeny

I *Layout* finns åtgärder för att hantera layouten:

- Spara projektfilen direkt från fönstret för utskriftslayout.
- Skapa en ny och tom utskriftslayout med *New Layout…*.
- *Duplicate Layout…* : Skapa en ny utskriftslayout genom att duplicera den nuvarande.
- Ta bort den aktuella layouten med *Ta bort layout…*.
- Öppna *Layouthanterare…*.
- *Layouts* ► : Öppna en befintlig utskriftslayout.

När layouten har utformats kan du med ikonerna *Spara som mall* och *Add Items from Template* spara det aktuella läget i en utskriftslayoutsession som en `.qpt`-mallfil och ladda dess objekt igen i en annan session/utskriftslayout.

I menyn *Layout* finns det också kraftfulla sätt att dela geografisk information som producerats med QGIS och som kan inkluderas i rapporter eller publiceras. Dessa verktyg är *Export as Image…*, *Export as PDF…*, *Export as SVG…* och *Print…*.

Nedan finns en lista över alla tillgängliga verktyg i denna meny med lite praktisk information.

Verktyg	Genväg	Verktygsrad	Referens
Spara projekt	`Ctrl+S`	*Layout*	*Introduktion till QGIS-projekt*
Ny layout	`Ctrl+N`	*Layout*	*Layoutansvarig*
Duplicera layout		*Layout*	*Layoutansvarig*
Delete Layout			
Layout Manager…		*Layout*	*Layoutansvarig*
Layouter ►			
Layout-egenskaper…			*Layoutpanelen*
Rename Layout…			
Sidans egenskaper…			*Arbeta med sidans egenskaper*
Lägg till sidor…		*Layout*	*Arbeta med sidans egenskaper*
Lägg till objekt från mall		*Layout*	*Skapa ett layoutobjekt*
Spara som mall…		*Layout*	*Layoutansvarig*
Exportera som bild…		*Layout*	*Exportera som bild*
Exportera som SVG…		*Layout*	*Exportera som SVG*
Exportera som PDF… `		*Layout*	*Exportera som PDF*
Printer Page Setup…	`Ctrl+Shift+P`		
Skriv ut…	`Ctrl+P`	*Layout*	*Skapa en utdata*
Stäng	`Ctrl+Q`		

Redigera meny

Menyn *Edit* erbjuder verktyg för att manipulera objekt i utskriftslayouten. Den innehåller vanliga åtgärder som markeringsverktyg, kopiera/klippa/klistra in och ångra/redo (se *Panelen Ångra historik: Återgå och återställ åtgärder*) för objekten i layouten.

När du använder åtgärden Klistra in kommer elementen att klistras in i enlighet med musens aktuella position. Om du använder åtgärden *Edit* ► *Paste in Place* eller trycker på `Ctrl+Shift+V` klistras objekten in på den aktuella sidan, på samma plats som de befann sig på den ursprungliga sidan. Det säkerställer att kopiera/klistra in objekt på samma plats, från sida till sida.

Nedan finns en lista över alla tillgängliga verktyg i denna meny med lite praktisk information.

Table 18.1: Tillgängliga verktyg

Verktyg	Genväg	Verktygsra	Referens
Ångra (sista ändringen)	`Ctrl+Z`	*Layout*	*Panelen Ångra historik: Återgå och återställ åtgärder*
Gör om (senast återkallade ändring)	`Ctrl+Y`	*Layout*	*Panelen Ångra historik: Återgå och återställ åtgärder*
Ta bort	`Del`		
Klipp ut	`Ctrl+X`		
Kopiera	`Ctrl+C`		
Klistra in	`Ctrl+V`		
Pasta in på plats	`Ctrl+Shift+V`		
Markera alla	`Ctrl+A`		
Avmarkera alla	`Ctrl+Shift+A`		
Invertera markering			
Välj nästa artikel nedan	`Ctrl+Alt+[`		
Välj nästa artikel ovan	`Ctrl+Alt+]`		
Panorera layout	`P`	*Verktygslåa*	
Zooma	`Z`	*Verktygslåa*	
Välj/flytta objekt	`V`	*Verktygslåa*	*Interagera med layoutobjekt*
Flytta innehåll	`C`	*Verktygslåa*	*Kartobjektet*
Edit Nodes Item		*Verktygslåa*	*De nodbaserade formobjekten*

Visa meny

Menyn *View* ger tillgång till navigeringsverktyg och hjälper till att konfigurera utskriftslayoutens allmänna beteende. Förutom de vanliga zoomverktygen har du möjlighet att:

- Friskställ vyn (om du upptäcker att vyn är inkonsekvent);
- aktivera ett *grid* som du kan snäppa objekt till när du flyttar eller skapar dem. Inställning av rutnät görs i *Inställningar* ► *Layoutalternativ…* eller i *Layoutpanel*;
- aktivera *guides* som du kan fästa objekt vid när du flyttar eller skapar dem. Guider är röda linjer som du kan skapa genom att klicka i linjalen (ovanför eller till vänster i layouten) och dra och släppa till önskad plats;
- *Smart Guides*: använder andra layoutobjekt som guider att dynamiskt fästa vid när du flyttar eller omformar ett objekt;
- *Clear Guides* för att ta bort alla nuvarande guider;
- *Visa avgränsningsruta* runt objekten för att bättre identifiera ditt val;
- *Visa regler* runt layouten;
- *Visa sidor* eller ställ in sidor så att de är transparenta. Ofta används layout för att skapa layouter som inte är tryckta, t.ex. för att inkluderas i presentationer eller andra dokument, och det är önskvärt att exportera kompositionen med en helt transparent bakgrund. Det kallas ibland ”infinite canvas” i andra redigeringspaket.

I utskriftslayouten kan du ändra zoomnivån med hjälp av mushjulet eller skjutreglaget och kombinationsrutan i statusfältet. Om du behöver växla till panoreringsläge medan du arbetar i layoutområdet kan du hålla ned `Spacebar`

eller mushjulet. Med `Ctrl+Spacebar` kan du tillfälligt växla till Zoom In-läge och med `Ctrl+Alt+Spacebar` till Zoom Out-läge.

Paneler och verktygsfält kan aktiveras från menyn *View* ►. För att maximera det tillgängliga utrymmet för att interagera med en komposition kan du markera alternativet *View* ► *Toggle Panel Visibility* eller trycka på `Ctrl+Tab`; alla paneler döljs och endast tidigare synliga paneler återställs när du avmarkerar dem.

Det är också möjligt att växla till helskärmsläge för att få mer utrymme att interagera med genom att trycka på `F11` eller använda *View* ► *Toggle Full Screen*.

Verktyg	Genväg	Verktygsrad	Referens
Uppdatera	`F5`	*Navigation*	
Förhandsvisning ►			
Zooma in	`Ctrl++`	*Navigation*	
Zooma ut	`Ctrl+-`	*Navigation*	
Zooma till 100%	`Ctrl+1`	*Navigation*	
Zoom Full	`Ctrl+0`	*Navigation*	
Zoom till bredd			
Visa rutnät	`Ctrl+'`		*Guider och rutnät*
Fäst till rutnät	`Ctrl+Shift+'`		*Guider och rutnät*
Visa guider	`Ctrl+;`		*Guider och rutnät*
Snap to Guides	`Ctrl+Shift+;`		*Guider och rutnät*
Smarta guider	`Ctrl+Alt+;`		
Hantera guider…			*Panelen för guider*
Clear Guides			*Panelen för guider*
Visa linjaler	`Ctrl+R`		
Visa avgränsningsrutor	`Ctrl+Shift+B`		
Visa sidor			
Verktygsfält ►			*Paneler och verktygsfält*
Paneler ►			*Paneler och verktygsfält*
Växla helskärm	`F11`		*Vy*
Växla panelsynlighet	`Ctrl+Tab`		*Vy*

Meny för objekt

Med hjälp av *Items* kan du konfigurera objektens position i layouten och relationerna mellan dem (se *Interagera med layoutobjekt*).

Verktyg	Genväg	Verktygsrad	Referens
Gruppera	`Ctrl+G`	*Åtgärder*	*Gruppera objekt*
Avgruppera	`Ctrl+Shift+G`	*Åtgärder*	*Gruppera objekt*
Höj	`Ctrl+]`	*Åtgärder*	*Justering och distribution*
Sänk	`Ctrl+[`	*Åtgärder*	*Justering och distribution*
Lägg längst fram	`Ctrl+Shift+]`	*Åtgärder*	*Justering och distribution*
Lägg längst bak	`Ctrl+Shift+[`	*Åtgärder*	*Justering och distribution*
Lås valda objekt	`Ctrl+L`	*Åtgärder*	*Låsning av föremål*
Lås upp alla	`Ctrl+Shift+L`	*Åtgärder*	*Låsning av föremål*
Justera objekt ►		*Åtgärder*	*Justering och distribution*
Distribuera föremål ►		*Åtgärder*	*Flytta och ändra storlek på objekt*
Återställ storlek ►		*Åtgärder*	*Flytta och ändra storlek på objekt*

Meny Lägg till objekt

Dessa är verktyg för att skapa layoutobjekt. Var och en av dem beskrivs ingående i kapitlet *Layoutobjekt*.

Verktyg	Verktygsrad	Referens
Lägg till karta	*Verktygslåda*	*Kartobjektet*
Lägg till 3D-karta	*Verktygslåda*	*Objektet 3D-karta*
Lägg till bild	*Verktygslåda*	*Bildobjektet*
Lägg till etikett	*Verktygslåda*	*Etikettobjektet*
Lägg till dynamisk text ►		*Etikettobjektet*
Lägg till förklaring	*Verktygslåda*	*Det legendariska föremålet*
Lägg till skalfält	*Verktygslåda*	*Objektet för skalfältet*
Lägg till norrpil	*Verktygslåda*	*North Arrow-objektet*
Lägg till form ►	*Verktygslåda*	*Det vanliga formobjektet*
► *Lägg till rektangel*	*Verktygslåda*	*Det vanliga formobjektet*
► *Lägg till ellips*	*Verktygslåda*	*Det vanliga formobjektet*
► *Lägg till triangel*	*Verktygslåda*	*Det vanliga formobjektet*
Lägg till markör	*Verktygslåda*	*Markeringsobjektet*
Lägg till pil	*Verktygslåda*	*Föremålet Arrow*
Lägg till nodobjekt ►	*Verktygslåda*	*De nodbaserade formobjekten*
► *Lägg till polygon*	*Verktygslåda*	*De nodbaserade formobjekten*
► *Lägg till polylinje*	*Verktygslåda*	*De nodbaserade formobjekten*
Lägg till HTML	*Verktygslåda*	*HTML-ramobjektet*
Lägg till attributtabell	*Verktygslåda*	*Attributtabellens objekt*
Lägg till fast tabell	*Verktygslåda*	*Den fasta tabellposten*
Lägg till höjdprofil	*Verktygslåda*	*Höjdprofilens objekt*

Atlas meny

Verktyg	Genväg	Verktygsrad	Referens
Preview Atlas	`Ctrl+ALt+/`	*Atlas*	*Förhandsgranska och skapa en atlas*
Första funktionen	`Ctrl+<`	*Atlas*	*Förhandsgranska och skapa en atlas*
Föregående funktion	`Ctrl+,`	*Atlas*	*Förhandsgranska och skapa en atlas*
Nästa funktion	`Ctrl+.`	*Atlas*	*Förhandsgranska och skapa en atlas*
Sista funktionen	`Ctrl+>`	*Atlas*	*Förhandsgranska och skapa en atlas*
Skriv ut Atlas…		*Atlas*	*Förhandsgranska och skapa en atlas*
Exportera Atlas som bilder…		*Atlas*	*Förhandsgranska och skapa en atlas*
Exportera Atlas som SVG…		*Atlas*	*Förhandsgranska och skapa en atlas*
Exportera Atlas som PDF…		*Atlas*	*Förhandsgranska och skapa en atlas*
Atlasinställningar		*Atlas*	*Skapa en Atlas*

Meny för inställningar

Menyn *Settings* ► *Layout Options…* är en genväg till menyn *Inställningar* ► *Alternativ* ► *Layouts* i QGIS huvudmeny. Här kan du ställa in några alternativ som kommer att användas som standard på alla nya utskriftslayouter:

- *Layout defaults* låter dig ange vilket typsnitt som ska användas som standard;
- Med *Grid appearance* kan du ställa in rutnätets stil och färg. Det finns tre typer av rutnät: **Punkter**, **Solid** linjer och **Kors**;
- *Grid and guide defaults* definierar avstånd, förskjutning och tolerans för rutnätet (se *Guider och rutnät* för mer information);
- *Layout Paths*: för att hantera en lista med anpassade sökvägar för utskriftsmallar.

Med menyn *Inställningar* ► *Tangentbordsgenvägar…* kan du använda *shortcuts manager* i utskriftslayoutgränssnittet.

Kontextuella menyer

Beroende på var du högerklickar i dialogrutan för utskriftslayout öppnas en kontextuell meny med olika funktioner:

- Högerklicka på menyraden eller ett verktygsfält och du får en lista över layoutpaneler och verktygsfält som du kan aktivera eller inaktivera med ett klick.
- Högerklicka över en linjal så kan du *Show Guides*, *Snap to Guides*, *Manage Guides…* öppna *Guides panel* eller *Clear Guides*. Det är också möjligt att dölja linjalerna.
- Högerklicka i utskriftslayoutens arbetsyta och:
  - Du kommer att kunna *Undo* och *Redo* de senaste ändringarna, eller *Paste* ett kopierat objekt (endast tillgängligt om inget objekt är markerat).
  - Om du klickar på en sida kan du dessutom komma åt den aktuella *Page Properties*-panelen eller *Remove Page*.
  - Om du klickar på ett markerat objekt kan du klippa ut eller kopiera det samt öppna panelen *Item Properties*.
  - Om mer än ett objekt är markerat kan du antingen gruppera dem och/eller avgruppera dem om minst en grupp redan finns i markeringen.
- Om du högerklickar i en textruta eller spinbox-widget i en layoutpanel visas redigeringsalternativ för att manipulera dess innehåll.

Layoutpanelen

I panelen *Layout* kan du definiera de globala inställningarna för din utskriftslayout.

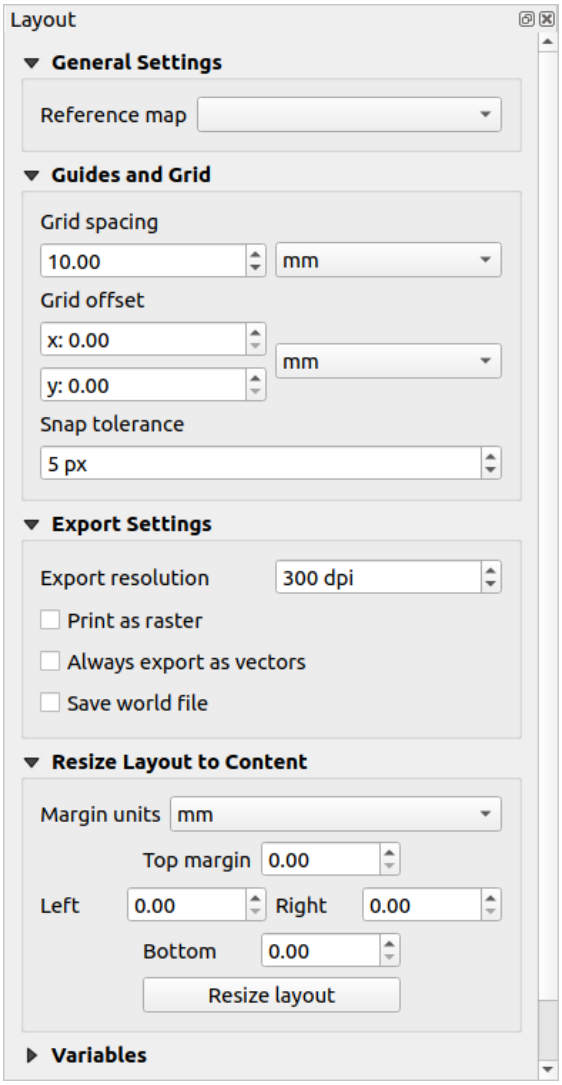


Fig. 18.3: Layoutinställningar i utskriftslayouten

Allmänna inställningar

I en utskriftslayout kan du använda mer än ett kartobjekt. *Reference map* representerar det kartobjekt som ska användas som layoutens huvudkarta. Den tilldelas så länge som det finns ett kartobjekt i layouten. Layouten kommer att använda denna karta i alla sina egenskaper och variabler som beräknar enheter eller skala. Detta inkluderar export av utskriftslayouten till georefererade format.

Dessutom har nya layoutobjekt som skalstreck, teckenförklaring eller norrpil som standard inställningar (orientering, visade lager, skala, …) som är bundna till det kartobjekt som de ritas över, och faller tillbaka till referenskartan om det inte finns någon överlappande karta.

Guider och rutnät

Du kan sätta några referensmärken på pappersarket för att hjälpa dig att placera vissa föremål korrekt. Dessa märken kan vara:

- enkla horisontella eller vertikala linjer (kallade **Guides**) som placeras på den position du vill ha (se *Panelen för guider* för skapande av guider).
- eller vanlig **Grid**: ett nätverk av horisontella och vertikala linjer som läggs över layouten.

Inställningar som *Grid spacing* eller *Grid offset* kan justeras i den här gruppen, liksom *Snap tolerance* som ska användas för objekt. Toleransen är det maximala avstånd under vilket muspekaren snäpps till ett rutnät eller en guide när du flyttar, ändrar storlek på eller skapar ett objekt.

Huruvida rutnät eller guider ska visas ställs in i menyn *View*. Där kan du också bestämma om de ska användas för att fästa layoutobjekt. När både en rutnätslinje och en styrlinje ligger inom toleransområdet för en punkt, kommer styrlinjerna alltid att ha företräde - eftersom de har ställts in manuellt (därav antagandet att de uttryckligen har placerats på mycket önskvärda snäpplägen och bör väljas framför det allmänna rutnätet).

Observera

I menyn :menuselection:`Settings --> Layout Options` kan du också ställa in de parametrar för rutnät och guider som beskrivs ovan. Dessa alternativ kommer dock endast att gälla som standard för nya utskriftslayouter.

Inställningar för export

Du kan definiera en upplösning som ska användas för alla exporterade kartor i *Export resolution*. Den här inställningen kan sedan åsidosättas varje gång du exporterar en karta.

På grund av vissa avancerade renderingsalternativ (*blending mode*, *effects*…) kan ett layoutobjekt behöva rastreras för att kunna exporteras korrekt. QGIS kommer att rasterisera det individuellt utan att tvinga alla andra objekt att också rasteriseras. Detta gör det möjligt att skriva ut eller spara som PostScript eller PDF för att hålla objekten så mycket som möjligt som vektorer, t.ex. kommer ett kartobjekt med lageropacitet inte att tvinga etiketter, skalstreck etc. att också rasteriseras. Du kan dock göra det:

- tvinga alla objekt att rastreras genom att markera rutan *Print as raster*;
- eller använd det motsatta alternativet, d.v.s. *Alltid exportera som vektorer*, för att tvinga exporten att behålla objekt som vektorer när de exporteras till ett kompatibelt format. Observera att detta i vissa fall kan leda till att utdata ser annorlunda ut än i layouten.

Om formatet gör det möjligt (t.ex. `.TIF`, `.PDF`) resulterar export av en utskriftslayout som standard i en georefererad fil (baserad på *Referenskarta* i gruppen *Allmänna inställningar*). För andra format kräver georefererade utdata att du genererar en världsfil genom att markera kryssrutan *Spara världsfil*. Världsfilen skapas bredvid de exporterade kartorna, har namnet på den sida som visas med referenskartobjektet och innehåller information som gör det enkelt att georeferera den.

Ändra storlek på layout till innehåll

Med hjälp av verktyget *Resize page* i den här gruppen skapar du en unik sidkomposition vars omfattning täcker det aktuella innehållet i utskriftslayouten (med några valfria *marginaler* runt de beskurna gränserna).

Observera att det här beteendet skiljer sig från alternativet *crop to content* genom att alla objekt placeras på en riktig och unik sida som ersätter alla befintliga sidor.

Variabler

I *Variables* listas alla variabler som är tillgängliga på layoutnivå (vilket inkluderar alla globala variabler och projektvariabler).

Det gör det också möjligt för användaren att hantera variabler på layoutnivå. Klicka på knappen för att lägga till en ny anpassad variabel på layoutnivå. På samma sätt kan du välja en anpassad variabel på layoutnivå i listan och klicka på knappen för att ta bort den.

Mer information om hur variabler används finns i avsnittet *General Tools*.

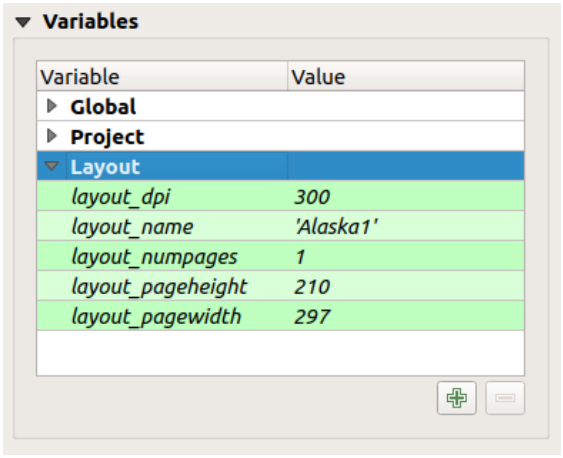


Fig. 18.4: Variabeleditorn i utskriftslayouten

Arbeta med sidans egenskaper

En layout kan bestå av flera sidor. En första sida kan t.ex. visa en kartbild, en andra sida kan visa attributtabellen som är kopplad till ett lager, medan en tredje sida visar en HTML-ram som länkar till din organisations webbplats. Du kan också lägga till många olika typer av objekt på varje sida.

Lägga till en ny sida

Dessutom kan en layout göras med olika storlek och/eller orientering på sidorna. För att lägga till en sida, välj verktyget *Lägg till sidor…* från menyn *Layout* eller *Layout Toolbar*. Dialogrutan *Infoga sidor* öppnas och du ombeds fylla i:

- antalet sidor som ska infogas;
- sidans/sidornas position: före eller efter en viss sida eller i slutet av utskriftslayouten;
- *Sidstorlek*: den kan vara en sida i ett förinställt format (`A4`, `B0`, `Legal`, `Letter`, `ANSI A`, `Arch A` och deras derivat samt en upplösningstyp, t.ex. `1920x1080` eller `1024x768`) med tillhörande *Orientation* (Porträtt eller Landskap).

Sidstorleken kan också vara av ett ”anpassat” format; i så fall måste du ange dess *Bredd* och *Höjd* (med låst storleksförhållande om det behövs) och välja den enhet som ska användas bland `mm`, `cm`, `px`, `pt`, `in`, `ft`... Omvandling av inmatade värden sker automatiskt när man byter från en enhet till en annan.

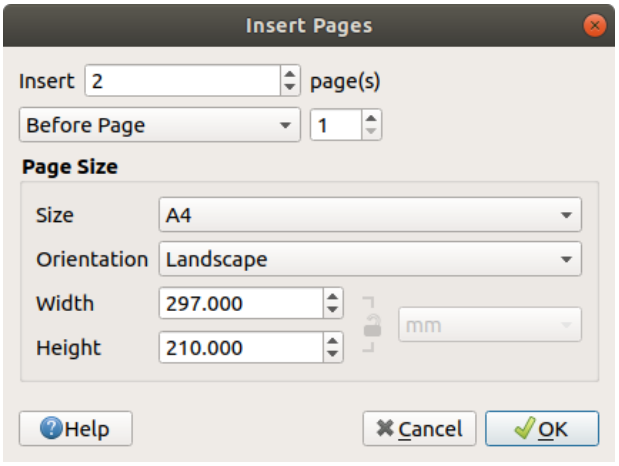


Fig. 18.5: Skapa en ny sida i utskriftslayouten

Uppdatering av sidans egenskaper

Varje sida kan senare anpassas genom panelen Page *Item Properties*. För att komma åt en sidas egenskaper, antingen:

- Vänsterklicka på en tom del av sidan
- Högerklicka på en sida och välj *Page Properties*...
- Gå till *Layout* ► *Sidans egenskaper*... när du håller muspekaren över sidan

Panelen *Item Properties* öppnas med inställningar som t.ex:

- för layouter med flera sidor visas numret på den aktiva sidan och antalet tillgängliga sidor
- ramen *Sidstorlek* som beskrivs ovan. Du kan ändra varje egenskap med hjälp av de datadefinierade åsidosättningsalternativen (se *Utforska datadefinierade åsidosättningsknappar med atlas* för ett användningsfall);
- *Exkludera sida från export* för att styra om den aktuella sidan med dess innehåll ska inkluderas i *layoututdata*;
- *Bakgrund* på den aktuella sidan med den *färg* eller *symbol* du vill ha.

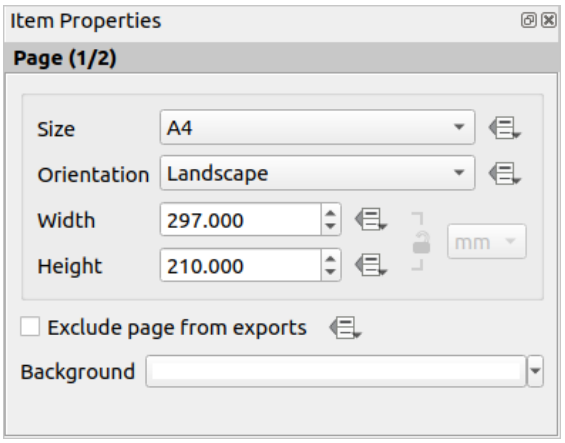


Fig. 18.6: Dialog för sidans egenskaper

Panelen för guider

Guider är vertikala eller horisontella linjereferenser som du kan placera på en layoutsida för att hjälpa dig att placera objekt när du skapar, flyttar eller ändrar storlek på dem. För att guider ska vara aktiva krävs att alternativen *View* ► *Show Guides* och *View* ► *Snap to Guides* är markerade. För att skapa en guide finns det två olika metoder:

- om alternativet *View* ► *Show Rulers* är inställt, dra ut en linjal och släpp musknappen inom sidområdet, på önskad position.
- för mer precision, använd panelen *Guides* från *View* ► *Toolbox* ► eller genom att välja *Manage guides for page…* från sidans kontextuella meny.

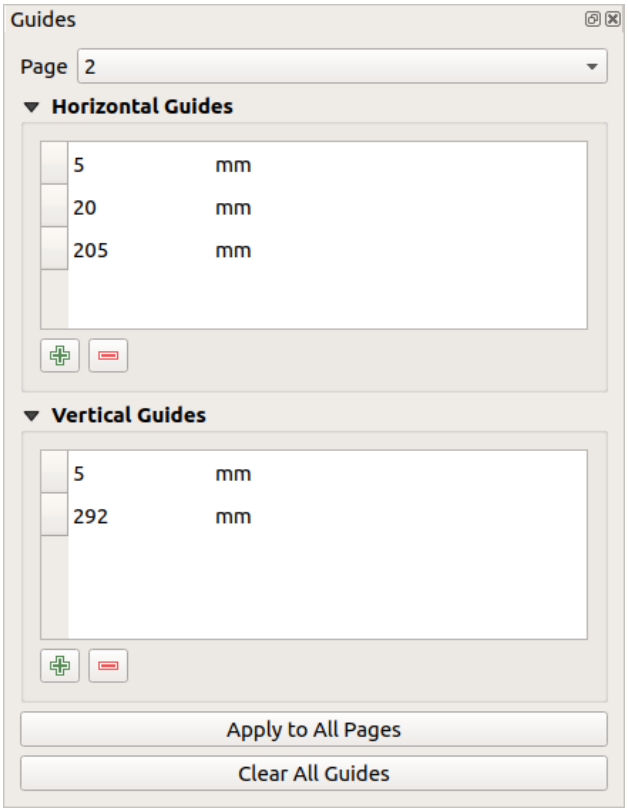


Fig. 18.7: Panelen Guider

Panelen *Guides* gör det möjligt att skapa snäpplinjer på specifika platser:

1. Välj den *Sida* som du vill lägga till guiderna på
2. Klicka på knappen :sup:`Add new guide` och ange koordinaterna för den horisontella eller vertikala linjen. Ursprunget är i det övre vänstra hörnet. Olika enheter finns tillgängliga för detta.

   Panelen gör det också möjligt att justera positionen för befintliga guider till exakta koordinater: dubbelklicka och ersätt värdet.
3. Panelen *Guides* listar endast objekten för den aktuella sidan. Det gör det möjligt att skapa eller ta bort guider endast på den aktuella sidan. Du kan dock använda knappen *Apply to All Pages* för att kopiera guidekonfigurationen för den aktuella sidan till de andra sidorna i layouten.
4. Om du vill ta bort en guide markerar du den och trycker på knappen Remove selected guide. Använd *Clear All Guides* för att ta bort alla guider på den aktuella sidan.

Tips

Snabbkoppling till befintliga layoutobjekt

Förutom guider och rutnät kan du använda befintliga objekt som snäppreferenser när du flyttar, ändrar storlek på eller skapar nya objekt; dessa kallas **smarta guider** och kräver att alternativet *View* ► *Smart Guides* är markerat. Varje gång muspekaren är nära ett objekts gräns visas ett snäppkors.

Panelen för objekt

Panelen *Items* erbjuder några alternativ för att hantera val och synlighet av objekt. Alla objekt som har lagts till i utskriftslayouten (inklusive *items group*) visas i en lista och om du väljer ett objekt markeras motsvarande rad i listan, och om du väljer en rad markeras motsvarande objekt i utskriftslayouten. Detta är alltså ett praktiskt sätt att välja ett objekt som är placerat bakom ett annat. Observera att en markerad rad visas med fetstil. Det är också möjligt att välja flera objekt genom att hålla in `Shift` för sammanhängande objekt och `Ctrl` för icke sammanhängande objekt.

För alla tillgängliga objekt kan du göra det:

- gör den synlig eller inte;
- låsa eller låsa upp sin position;
- sortera dess Z-position. Du kan flytta upp och ner varje objekt i listan genom att klicka och dra. Det översta objektet i listan kommer att placeras i förgrunden i utskriftslayoutens canvas. Som standard placeras ett nyskapat objekt i förgrunden.
- ändra artikel-ID genom att dubbelklicka på texten;
- högerklicka på ett objekt och välj om du vill kopiera eller ta bort det eller öppna dess *egenskapspanel*.

När du har hittat rätt position för ett objekt kan du låsa det genom att kryssa i rutan i kolumnen . Låsta objekt är **inte** valbara på duken. Låsta objekt kan låsas upp genom att markera objektet i panelen *Items* och avmarkera kryssrutan eller så kan du använda ikonerna i verktygsfältet.

Panelen Ångra historik: Återgå och återställ åtgärder

Under layoutprocessen är det möjligt att återgå till och återställa ändringar. Detta kan göras med verktygen revert och restore som finns i menyn *Edit*, verktygsfältet *Layout* eller i snabbmenyn när du högerklickar i utskriftslayoutområdet:

- Återställ senaste ändring
- Återskapa senaste ändring

Detta kan också göras genom att klicka med musen i panelen *Undo history* (se Fig. 18.8). Panelen Historik visar de senaste åtgärderna som gjorts i utskriftslayouten. Välj den punkt som du vill återgå till och när du gör en ny åtgärd kommer alla åtgärder som gjorts efter den valda att tas bort.

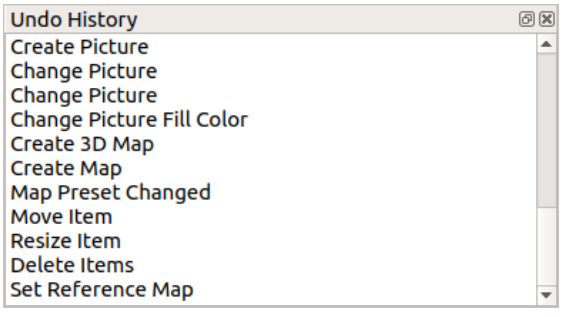


Fig. 18.8: Ångra historik i utskriftslayouten

18.2 Layoutobjekt

18.2.1 Layoutobjekt Gemensamma alternativ

QGIS tillhandahåller en stor uppsättning objekt för att layouta en karta. De kan vara av typen karta, teckenförklaring, skalstreck, bild, tabell, norrpil, bildtyp… De delar dock några gemensamma alternativ och beteenden som beskrivs nedan.

Skapa ett layoutobjekt

Artiklar kan skapas med hjälp av olika verktyg, antingen från grunden eller baserat på befintliga artiklar.

Så här skapar du ett layoutobjekt från början:

1. Välj motsvarande verktyg antingen från menyn *Add Item* eller från fältet *Toolbox*.
2. Då så:
   - Klicka på sidan och fyll i den information om storlek och placering som efterfrågas i dialogrutan *New Item Properties* som dyker upp (för mer information, se *Position och storlek*);

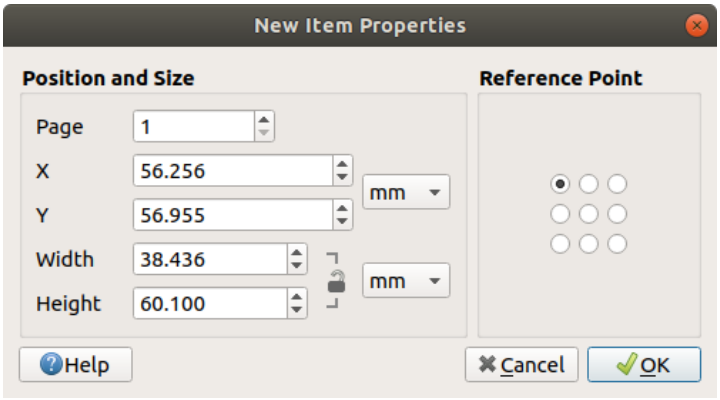


Fig. 18.9: Dialog för egenskaper för nytt objekt

   - Eller klicka och dra för att definiera den ursprungliga storleken och placeringen av objektet. Du kan förlita dig på att *grids and guides* snappar för en bättre position.

> **Observera**
>
> Eftersom de kan ha särskilda former fungerar det inte att rita nod- eller pilobjekt med enklicks- eller klick-och-drag-metoder; du måste klicka och placera varje nod i objektet. Se *De nodbaserade formobjekten* för mer information.

Det kan du också göra:

1. Välj ett befintligt objekt med knappen Välj/flytta objekt från verktygsfältet *Toolbox*

2. Använd snabbmenyn eller menyverktygen *Edit* för att kopiera/skära ut objektet och klistra in det på muspositionen som ett nytt objekt.

   Du kan också använda kommandot *Paste in Place* (`Ctrl+Shift+V`) för att duplicera ett objekt från en sida till en annan och placera det på den nya sidan med samma koordinater som originalet.

Dessutom kan du skapa objekt med hjälp av en mall för utskriftslayout (för detaljer, se *Layoutansvarig*) genom kommandot *Layout* ► *Add Items from Template…*.

Tips

Lägga till layoutobjekt med hjälp av filbläddraren

Dra och släpp en utskriftslayoutmall (`.qpt`-fil) från filbläddraren eller panelen *Browser* till en dialogruta för utskriftslayout, så lägger QGIS automatiskt till alla objekt från mallen i layouten.

Interagera med layoutobjekt

Varje objekt i utskriftslayouten kan flyttas och storleksändras för att skapa en perfekt layout. För båda operationerna är det första steget att aktivera verktyget :sup:`Select/Move item` och klicka på objektet.

Du kan välja flera objekt med knappen Select/Move item: klicka och dra över objekten eller håll ned `Shift`-knappen och klicka på vart och ett av de objekt du vill ha. Om du vill avmarkera ett objekt klickar du på det genom att hålla in `Shift`-knappen.

Varje gång ett val görs visas antalet valda objekt i statusfältet. I menyn *Edit* kan du hitta åtgärder för att välja alla objekt, rensa alla val, invertera det aktuella valet och mycket mer…

Flytta och ändra storlek på objekt

Om inte *View* ► *Show Bounding Boxes* är avmarkerat, kommer ett markerat objekt att visa rutor på dess gränser; om du flyttar en av dem med musen ändras objektets storlek i motsvarande riktning. Håll in `Shift` för att bibehålla bildförhållandet när du ändrar storlek. Om du håller in `Alt` ändras storleken från objektets mitt.

Om du vill flytta ett layoutobjekt markerar du det med musen och flyttar det medan du håller vänsterknappen intryckt. Om du behöver begränsa rörelserna till den horisontella eller vertikala axeln håller du ned `Shift`-knappen på tangentbordet medan du flyttar musen. Du kan också flytta ett markerat objekt med hjälp av `Piltangenterna` på tangentbordet; om rörelsen är för långsam kan du snabba upp den genom att hålla in `Shift`. Om du behöver bättre precision kan du använda egenskaperna *Position och storlek*, eller grid/guides snapping enligt förklaringen ovan för att skapa objekt.

Storleksändring eller flyttning av flera objekt samtidigt görs på samma sätt som för ett enda objekt. QGIS tillhandahåller dock några avancerade verktyg för att automatiskt ändra storlek på ett urval av objekt enligt olika regler:

- varje objekts höjd matchar det högsta eller det kortaste valda objektet;
- varje objektbredd matchar det bredaste eller smalaste valda objektet ( );
- ändrar storlek på objekt till kvadrater: varje objekt förstoras så att det bildar en kvadrat.

På samma sätt finns det *verktyg* tillgängliga för att organisera flera objektpositioner genom att fördela dem jämnt:

- kanterna (vänster, höger, överst eller nederst) på objekten;
- centrering av objekt horisontellt eller vertikalt;
- avstånd mellan objekt horisontellt eller vertikalt.

Gruppera objekt

Genom att gruppera objekt kan du hantera en uppsättning objekt som ett enda: du kan enkelt ändra storlek, flytta, radera och kopiera objekten som en helhet.

För att skapa en grupp av objekt, markera mer än ett och tryck på knappen . *Group* på menyn *Items* eller verktygsfältet *Actions*, eller från högerklicksmenyn. En rad med namnet `Group` läggs till i panelen *Items* och kan låsas eller döljas på samma sätt som alla andra *Items-panelens objekt*. Grupperade objekt är **inte** individuellt** valbara på duken; använd panelen Items för direktval och få tillgång till objektets egenskapspanel.

Låsning av föremål

När du har hittat rätt position för ett objekt kan du låsa det med hjälp av knappen *Lock selected items* i menyn *Items* eller verktygsfältet *Actions*, eller genom att kryssa i rutan bredvid objektet i panelen *Items*. Låsta objekt är **inte** valbara på duken.

Låsta objekt kan låsas upp genom att markera objektet i panelen *Items* och avmarkera kryssrutan eller så kan du använda ikonerna i verktygsfältet.

Justering och distribution

Verktyg för att höja eller sänka Z-positionen för element i layouten finns i rullgardinsmenyn :sup:`Höj valda element`. Välj ett element i utskriftslayouten och välj den matchande funktionen för att höja eller sänka det valda elementet över de andra elementen. Denna ordning visas i panelen *Items*. Du kan också höja eller sänka objekt i panelen *Objekt* genom att klicka och dra i objektets etikett i listan.

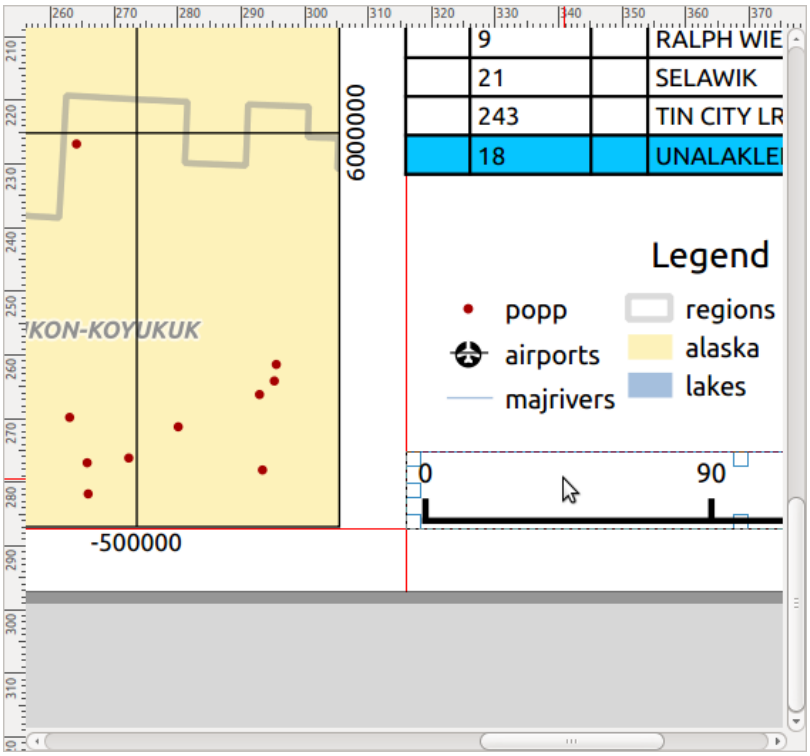


Fig. 18.10: Hjälplinjer för justering i utskriftslayouten

Det finns flera inriktningsalternativ tillgängliga i rullgardinsmenyn Align selected items (se Fig. 18.10). För att använda en justeringsfunktion väljer du först elementen och klickar sedan på en av justeringsikonerna:

- *Justera åt vänster* eller *Justera åt höger*;
- *Align Top* eller *Align Bottom*;
- *Align Center* horisontellt eller *Align Center Vertical*.

Alla markerade element kommer då att justeras till sin gemensamma begränsningsbox. När du flyttar objekt på layoutkanvasen visas hjälplinjer för justering när kanter, mittpunkter eller hörn justeras.

Ett annat sätt att förbättra placeringen av layoutobjekt är att justera avståndet mellan dem över hela layoutsidan. Detta kan göras genom att markera objekten och trycka på rullgardinsmenyn *Distribute Left Edges* till:

- *Distribuera vänsterkanter* eller *Distribuera högerkanter* av objekt på samma sätt
- *Fördela övre kanter* eller *Fördela nedre kanter* av objekt på ett likvärdigt sätt
- *Distribute Horizontal Centers* eller *Distribute Vertical Centers* av föremål på samma sätt
- Lägg till lika stort utrymme mellan objekt: *Fördela horisontellt avstånd lika* eller *Fördela vertikalt avstånd lika*

Posters vanliga egenskaper

Layoutobjekt har en uppsättning gemensamma egenskaper som du hittar längst ner i panelen *Item Properties*: Position och storlek, Rotation, Ram, Bakgrund, Objekt-ID, Variabler och Rendering (se Fig. 18.11).

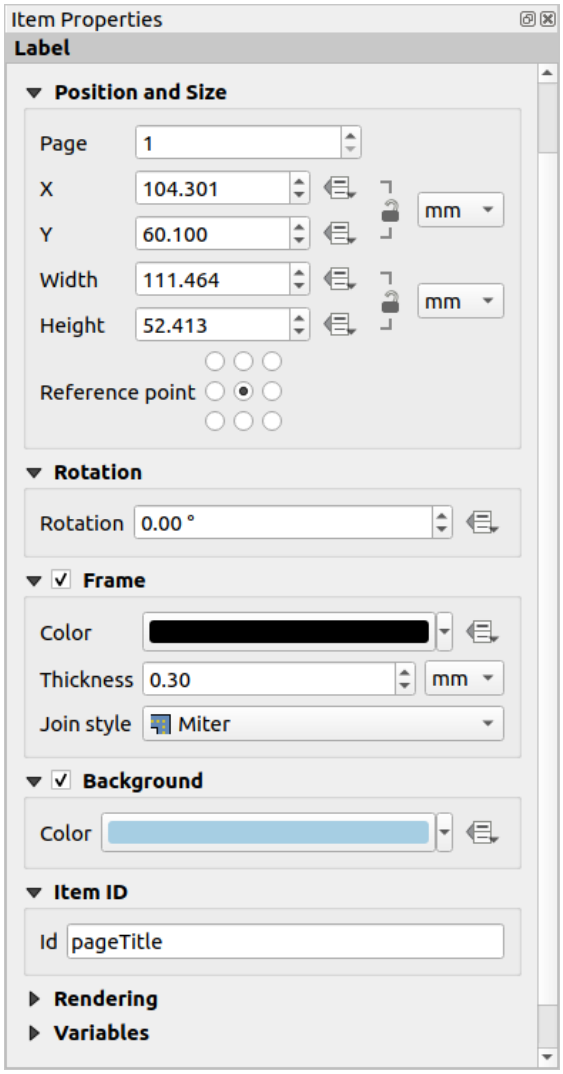


Fig. 18.11: Grupper för gemensamma artikelegenskaper

Observera

Ikonen Data defined override bredvid de flesta alternativen innebär att du kan associera egenskapen med ett lager, funktionsattribut, geometri eller med någon annan egenskap för layoutobjektet med hjälp av *expressions* eller *variables*. För mer information se *Inställning av datadefinierad åsidosättning*.

- I gruppen *Position and size* kan du definiera storlek och position för den ram som innehåller objektet (se *Position och storlek* för mer information).
- Med *Rotation* anges elementets rotation (i grader).
- Checkboxen *Frame* visar eller döljer ramen runt objektet. Använd widgetarna *Color*, *Thickness* och *Join style* för att justera dessa egenskaper.
- I gruppen *Background* kan du skapa eller välja en bakgrundsfärg *Color*. Transparensen kan justeras genom att ändra inställningarna för alfafältet.
- Använd *Item ID* för att skapa en relation till andra utskriftslayoutobjekt. Detta används med QGIS-servern och andra potentiella webbklienter. Du kan ange ett ID för ett objekt (t.ex. en karta eller en etikett), och sedan kan webbklienten skicka data för att ange en egenskap (t.ex. etikettext) för det specifika objektet. Kommandot GetProjectSettings listar de objekt och ID:n som finns tillgängliga i en layout.

- *Rendering* mode hjälper dig att ställa in om och hur objektet kan visas: du kan till exempel använda *blending mode*, justera objektets opacitet eller *Exclude item from exports*.

Position och storlek

Denna grupp utökar funktionerna i dialogen *New Item Properties* med datadefinierade funktioner och gör att du kan placera objekten exakt.

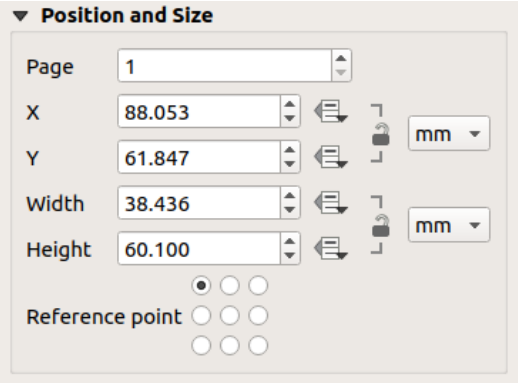


Fig. 18.12: Position och storlek

- det faktiska numret på den sida som objektet ska placeras på;
- referenspunkten för objektet;
- koordinaterna *X* och *Y* för *Reference point* för objektet på den valda sidan. Förhållandet mellan dessa värden kan låsas genom att klicka på knappen . Ändringar som görs i ett värde med hjälp av widgeten eller verktyget Select/Move item kommer att återspeglas i dem båda;
- *Width* och *Height* för objektets begränsningsbox. Precis som för koordinater kan förhållandet mellan bredd och höjd låsas.

Renderingsläge

QGIS tillåter avancerad rendering för layoutobjekt precis som för vektor- och rasterlager.

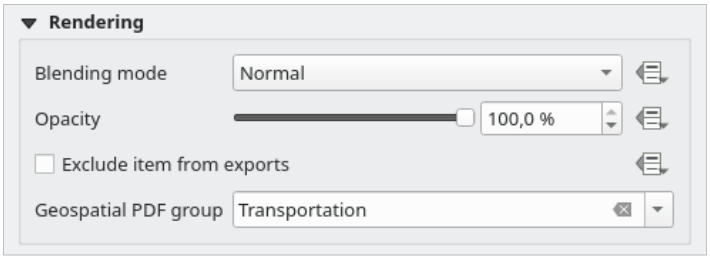


Fig. 18.13: Renderingsläge

- *Blandningsläge*: Med det här verktyget kan du uppnå effekter som annars bara kan uppnås med hjälp av grafiska renderingsprogram. Pixlarna i de över- och underliggande objekten kan blandas i enlighet med det inställda läget (se *Blandningslägen* för beskrivning av varje effekt).
- *Opacitet*: Med det här verktyget kan du göra det underliggande objektet i layouten synligt. Använd skjutreglaget för att anpassa synligheten för ditt objekt efter dina behov. Du kan också göra en exakt definition av den procentuella synligheten i menyn bredvid skjutreglaget.

- *Exkludera objekt från export*: I kombination med datadefinierade uttryck kan du bestämma att ett objekt ska vara osynligt i vissa exporter. När du har aktiverat den här kryssrutan kommer objektet inte att inkluderas i export till PDF, utskrift etc.
- *Geospatial PDF-grupp*: När den är inställd skapas en matchande lagerträdsgrupp i den exporterade geospatiala PDF-filen och objektet är endast synligt när den här gruppen är markerad. Detta gör att innehåll selektivt kan visas som en grupp av dem som tittar på Geospatial PDF. Det kan t.ex. innebära att extra layoutinnehåll som beskrivande etiketter eller legender endast visas när lager från gruppen är synliga, vilket gör filexporten mycket mer flexibel.

Variabler

I *Variables* listas alla variabler som är tillgängliga på layoutobjektets nivå (vilket inkluderar alla globala variabler, projektvariabler och layoutvariabler). Layoutkartobjekt har ett extra avsnitt *Kartinställningar* för variabler som ger enkel åtkomst till värden som kartans skala, utsträckning och så vidare.

I *Variabler* är det också möjligt att hantera variabler på layoutobjektnivå. Klicka på knappen för att lägga till en ny anpassad variabel. På samma sätt kan du välja en anpassad variabel på objektnivå i listan och klicka på knappen för att ta bort den.

Mer information om användning av variabler finns i avsnittet *Lagring av värden i variabler*.

18.2.2 Kartobjektet

Kartobjektet är den huvudram som visar den karta du har utformat i kartbilden. Använd verktyget *Add Map* efter *instruktioner för skapande av objekt* för att lägga till ett nytt kartobjekt som du senare kan manipulera på samma sätt som i *Interagera med layoutobjekt*.

Som standard visar ett nytt kartobjekt den aktuella statusen för *map canvas* med dess utsträckning och synliga lager. Du kan anpassa den med hjälp av panelen *Item Properties*. Förutom *objektens gemensamma egenskaper* har den här funktionen följande funktioner:

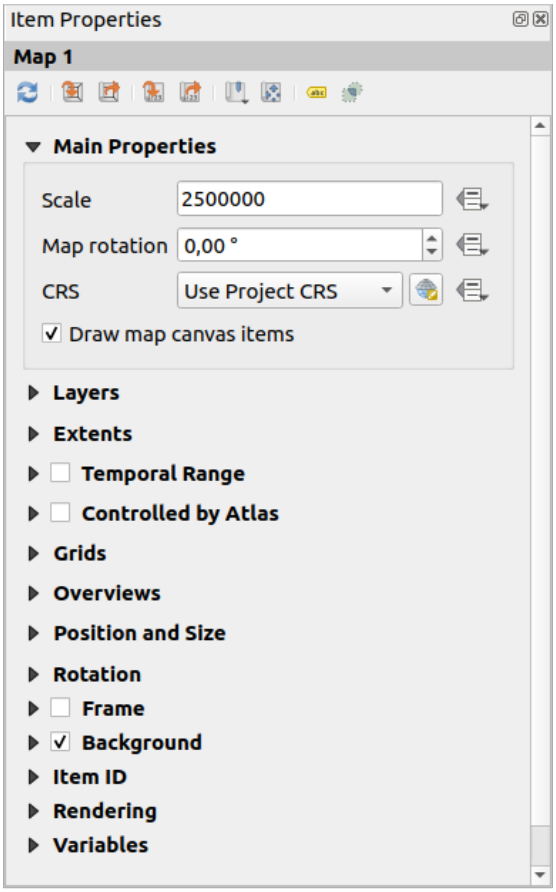


Fig. 18.14: Kartobjektets egenskapspanel

Verktygsrad

Panelen Map *Item Properties* innehåller ett verktygsfält med följande funktioner:

- Aktualisera förhandsgranskning av karta
- Ställ in kartans canvas så att den matchar huvudcanvasens utsträckning
- Visa aktuell kartutbredning i huvudkanvas
- Ställ in kartans skala så att den matchar huvudkartans skala
- Ställ in huvudkartan så att den matchar aktuell kartskala
- Bookmarks: ställa in kartobjektets omfattning så att det matchar ett befintligt spatialt bokmärke
- :sup:`Interaktivt redigera kartans omfattning`: panorera och zooma interaktivt inom kartobjektet
- Märkningsinställningar: styr funktionens märkningsbeteende (placering, synlighet…) i layoutkartans objektutsträckning:
  - ange en *Marginal från kartans kanter*, ett datadefinierbart avstånd från kartobjektets gränser inom vilket ingen etikett ska visas
  - *Tillåt avkortade etiketter på kartans kanter*: styr om etiketter som delvis faller utanför kartobjektets tillåtna utsträckning ska återges. Om det är markerat visas dessa etiketter (när det inte finns något sätt att

placera dem helt inom det synliga området). Om den inte är markerad kommer delvis synliga etiketter att hoppas över.

- *Objekt som blockerar etiketter*: gör att andra layoutobjekt (t.ex. skalstreck, norrpilar, infällda kartor etc.) kan markeras som blockerare för kartetiketterna i det **aktiva** kartobjektet. Detta förhindrar att några kartetiketter placeras under dessa objekt - vilket gör att etiketteringsmotorn antingen försöker placera dessa etiketter på alternativa platser eller ignorerar dem helt och hållet.

  Om en *Marginal från kartans kanter* är inställd placeras inte kartans etiketter närmare än det angivna avståndet från de markerade layoutobjekten.

- *Visa ej placerade etiketter*: kan användas för att avgöra om etiketter saknas på layoutkartan (t.ex. på grund av konflikter med andra kartetiketter eller på grund av otillräckligt utrymme för att placera etiketten) genom att markera dem i en *fördefinierad färg*.

- Clipping settings: gör det möjligt att klippa kartobjektet till atlasobjektet och till form- och polygonobjekt:

  - *Clip to atlas feature*: du kan bestämma att layoutkartobjektet automatiskt ska klippas till den aktuella *atlasfunktionen*.

    Det finns olika klippningslägen tillgängliga:

    * *Clip During Render Only*: tillämpar ett målarbaserat klipp, så att delar av vektorfunktioner som ligger utanför atlasfunktionen blir osynliga
    * *Clip Feature Before Render*: tillämpar klippning före rendering av funktioner, så att gränser för funktioner som delvis faller utanför atlasfunktionen fortfarande kommer att vara synliga på atlasfunktionens gräns
    * *Render Intersecting Features Unchanged*: Renderar alla funktioner som skär den aktuella atlasfunktionen, men utan att klippa deras geometri.

    Du kan *Tvinga in etiketter i atlasfunktionen*. Om du inte vill *Clip all layers* till atlasfunktionen kan du använda alternativet *Clip selected layers*.

  - *Clip to item*: det är möjligt att ändra formen på kartobjektet genom att använda ett *shape* eller *polygon* objekt från utskriftslayouten. När du aktiverar det här alternativet kommer kartan automatiskt att klippas till den valda formen i kombinationsrutan. Återigen är de ovan nämnda klippningslägena tillgängliga och etiketter kan tvingas att endast visas inuti klippningsformen.

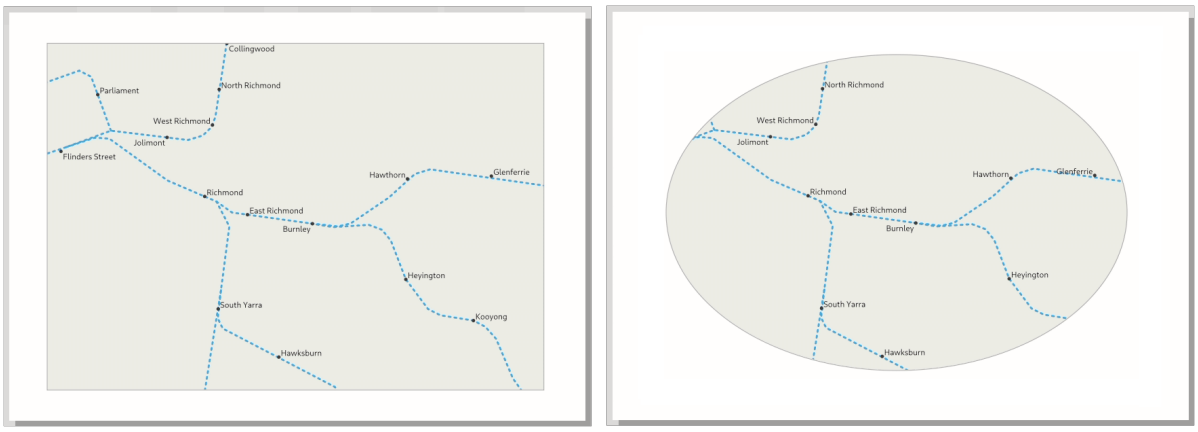


Fig. 18.15: Klippning av ett layoutmappobjekt till former

Huvudsakliga egenskaper

I gruppen *Main properties* (se Fig. 18.14) i kartans panel *Item Properties* finns följande alternativ:

- Knappen *Update Preview* för att uppdatera renderingen av kartobjektet om vyn i kartkanvasen har ändrats. Observera att uppdateringen av kartobjektet för det mesta utlöses automatiskt av ändringarna;
- *Scale* för att manuellt ställa in kartobjektets skala;
- Med *Map rotation* kan du rotera kartobjektets innehåll medurs i grader. Rotationen av kartans canvas kan imiteras här;
- Med *CRS* kan du visa innehållet i kartobjektet i valfri *CRS*. Standardvärdet är `Use project CRS`;
- *Draw map canvas items* låter dig visa i utskriftslayouten *annotationer* som är placerade på huvudkartans canvas.

Sikt

Som standard är kartobjektets utseende synkroniserat med renderingen av kartans canvas, vilket innebär att om du växlar synlighet för lagren eller ändrar deras stil i *Layers Panel* tillämpas detta automatiskt på kartobjektet. Eftersom du, precis som med alla andra objekt, kanske vill lägga till flera kartobjekt i en utskriftslayout, finns det ett behov av att bryta denna synkronisering för att kunna visa olika områden, lagerkombinationer, i olika skalor… Egenskapsgruppen *Layers* (se Fig. 18.16) hjälper dig att göra det.

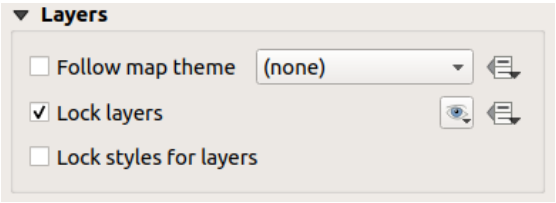


Fig. 18.16: Gruppen Kartlager

Om du vill att kartobjektet ska överensstämma med ett befintligt *karttema* markerar du *Follow map theme* och väljer önskat tema i rullgardinsmenyn. Alla ändringar som görs i temat i QGIS huvudfönster (med hjälp av funktionen replace theme) kommer automatiskt att påverka kartobjektet. Om ett karttema väljs inaktiveras alternativet *Lock styles for layers* eftersom *Follow map theme* även uppdaterar stilen (symbologi, etiketter, diagram) för lagren.

Om du vill låsa de lager som visas i ett kartobjekt till den aktuella kartkanvasens synlighet markerar du *Lock layers*. När det här alternativet är aktiverat kommer eventuella ändringar av lagrens synlighet i QGIS huvudfönster inte att påverka layoutens kartobjekt. Stil och etiketter för låsta lager uppdateras dock fortfarande i enlighet med QGIS huvudfönster. Du kan förhindra detta genom att använda *Lock styles for layers*.

I stället för att använda den aktuella kartbilden kan du också låsa kartobjektets lager till lagren i ett befintligt karttema: välj ett karttema från rullgardinsmenyn Set layer list from a map theme och kryssrutan | *Lock layers* aktiveras. Den uppsättning synliga lager som finns i karttemat används från och med nu för kartobjektet tills du väljer ett annat karttema eller avmarkerar alternativet *Lock layers*. Du kan då behöva uppdatera vyn med hjälp av knappen Refresh view i verktygsfältet *Navigation* eller knappen *Update Preview* som du ser ovan.

Observera att om alternativet *Lock layers* är aktiverat och inställt på ett karttema, till skillnad från alternativet *Follow map theme*, kommer lagren i kartobjektet inte att uppdateras även om karttemat uppdateras (med funktionen replace theme) i QGIS huvudfönster.

Låsta lager i kartobjektet kan också *data-definieras* med hjälp av ikonen bredvid alternativet. När detta används åsidosätter det valet i rullgardinsmenyn. Du måste skicka en lista med lager åtskilda med tecknet `|`. I följande exempel låses kartobjektet så att endast lagren `lager 1` och `lager 2` används:

```
concat ('layer 1', '|', 'layer 2')
```

Utsträckningar

Gruppen *Extents* i egenskapspanelen för kartobjekt ger följande funktioner (se Fig. 18.17):

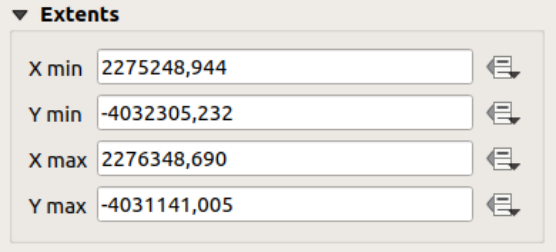


Fig. 18.17: Map Extents-grupp

Området **Extents** visar koordinaterna `X` och `Y` för det område som visas i kartobjektet. Vart och ett av dessa värden kan ersättas manuellt, vilket ändrar det område som visas på kartans canvas och/eller storleken på kartobjektet. Omfattningen kan också ändras med hjälp av verktyg högst upp på kartobjektets panel, t.ex:

- Ställ in kartans canvas så att den matchar huvudcanvasens utsträckning
- Ställ in kartans skala så att den matchar huvudkartans skala

Du kan också ändra ett kartobjekts utsträckning med hjälp av verktyget :sup:`Move item content`: klicka och dra i kartobjektet för att ändra dess aktuella vy, med bibehållen skala. När verktyget är aktiverat kan du använda mushjulet för att zooma in eller ut och ändra skalan på den visade kartan. Kombinera rörelsen med `Ctrl`-tangenten nedtryckt för att få en mindre zoom.

Höjdintervall

Med inställningen *Elevation range* i kartobjektets egenskaper kan du filtrera innehållet i vissa lager baserat på ett specifikt höjdintervall. Alla lager förblir synliga, men de som stöder höjdfiltrering (för närvarande punktmoln och raster-DEM) kommer att få sina data filtrerade. Endast de delar av dessa lager som faller inom det höjdintervall som anges av värdena *Lower* och *Upper* visas.

Elevation range kan vara datadefinierad. Detta innebär att du kan ange olika höjdintervall för olika funktioner i *atlas* eller *report*.

Temporalt intervall

I gruppen *Temporal range* i egenskapspanelen för kartobjektet kan du styra vilka lager som ska visas i kartobjektet baserat på ett tidsintervall. Endast lager vars tidsmässiga egenskaper överlappar med det tidsintervall som anges av datumen *Start* och *End* visas i kartobjektet.

De tillhörande datadefinierade widgetarna hjälper till att göra tidsintervallet dynamiskt och gör det möjligt att mata ut temporala *atlas*, dvs. automatiserade kartor med fast spatial utsträckning och vars innehåll varierar baserat på tid. Om du till exempel använder en csv-fil som täckningslager med ett start- och slutpar av fält och ett antal rader som representerar datumintervall, aktiverar du både det temporala intervallet och kontrollen av atlas i kartobjektets egenskaper och trycker på atlasexport.

Kontrolleras av atlas

Gruppegenskaperna *Kontrollerad av atlas* är endast tillgängliga om en *atlas* är aktiv i utskriftslayouten. Markera det här alternativet om du vill att kartobjektet ska styras av atlasen; när du itererar över täckningslagret panoreras/zoomas kartobjektets utsträckning till den atlasfunktion som följer:

- *Marginal runt objekt*: zoomar till objektet i den bästa skalan och håller runt varje objekt en marginal som representerar en procentandel av kartobjektets bredd eller höjd. Marginalen kan vara densamma för alla objekt eller *set variable*, t.ex. beroende på kartans skala;
- *Predefined scale (best fit)*: zoomar till objektet i projektet *fördefinierad skala* där atlasobjektet passar bäst;
- *Fixed scale*: atlasfunktioner panoreras från en till en annan och behåller samma skala som kartobjektet. Perfekt när du arbetar med objekt av samma storlek (t.ex. ett rutnät) eller vill markera storleksskillnader mellan atlasobjekt.

Rutnät

Med hjälp av rutnät kan du lägga till information om kartans utsträckning eller koordinater på kartan, antingen i kartobjektets projektion eller i en annan projektion. Gruppen *Grids* ger möjlighet att lägga till flera rutnät till ett kartobjekt.

- Med knapparna och kan du lägga till eller ta bort ett markerat rutnät;
- Med knapparna och kan du flytta ett rutnät uppåt och nedåt i listan, och därmed flytta det ovanpå eller under ett annat, över kartobjektet.

Dubbelklicka på det tillagda rutnätet för att byta namn på det.

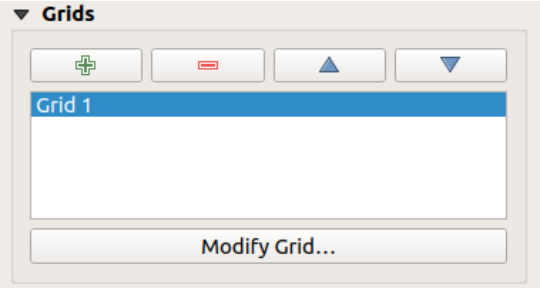


Fig. 18.18: Dialogrutor för kartor

Om du vill ändra ett rutnät markerar du det och trycker på knappen *Modify Grid…* för att öppna panelen *Map Grid Properties* och komma åt dess konfigurationsalternativ.

Utseende för rutnät

I panelen *Map Grid Properties*, kryssa i *Grid enabled* för att visa rutnätet på kartobjektet.

Som gridtyp kan du ange att du vill använda a:

- **Solid**: visar en linje över rutnätsramen. *Line style* kan anpassas med hjälp av *color* och *symbol* väljarmodulerna;
- **Cross**: visar segmentet vid rutnätslinjernas skärningspunkt för vilket du kan ställa in *Line style* och *Cross width*;
- **Markers**: visar endast anpassningsbar markörsymbol vid korsning av rutnätslinjer;
- eller **Enbart ram och anteckningar**.

Du kan även definiera andra typer av rutnät än rutnätstypen:

- rutnätets *CRS*: som standard kommer det att följa kartobjektets CRS. Tryck på Välj CRS-knappen för att ställa in den på en annan CRS.
- den *Interval*-typ som ska användas för rutnätsreferenserna:
  - **Map Units**: du ställer in ett avstånd på kartan (i enheten för rutnätets CRS) mellan på varandra följande rutnätsreferenser i riktningarna *X* och *Y*. Antalet rutnätsmarkeringar varierar beroende på kartans skala.
  - om du väljer **Fit Segment Width** väljs rutnätsintervallet dynamiskt baserat på kartans utsträckning till ett ”vackert” intervall. Det optimala intervallet beräknas inom ett intervall av avstånd vars värden *Minimum* och *Maximum* kan anpassas.
  - Med **Millimeter** eller **Centimeter** anger du ett avstånd på papperet mellan på varandra följande rutnätsreferenser i riktningarna *X* och *Y*. Antalet rutnätsmarkeringar är detsamma oavsett kartans skala.
- *Offset* från kartobjektets kanter, i *X* och/eller *Y* riktning
- och *Blend mode* för rutnätet (se *Blandningslägen*) när det är kompatibelt.

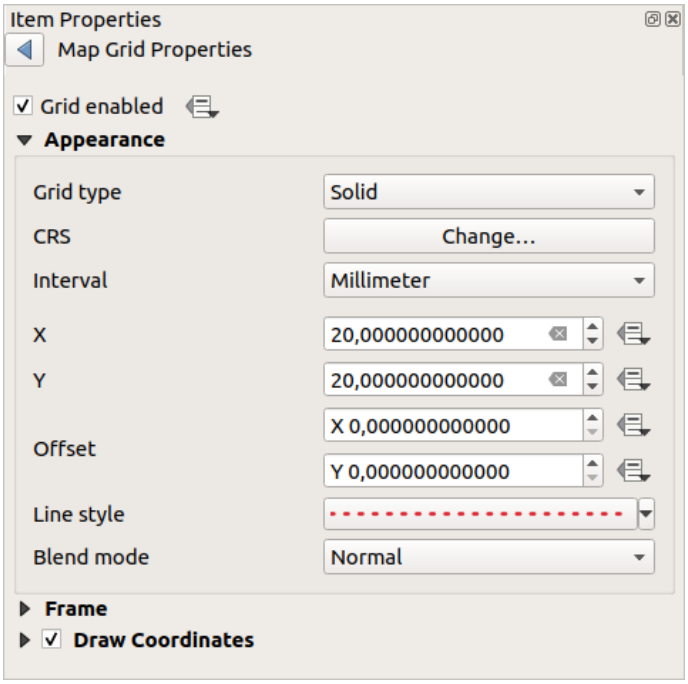


Fig. 18.19: Dialog för rutnätsutseende

Rutnätsram

Det finns olika alternativ för att utforma den ram som håller kartan. Följande alternativ finns tillgängliga: `Ingen ram`, `Zebra`, `Zebra (nautisk)`, `Inre kryss`, `Externa kryss`, `Inre och yttre kryss`, `Linjerand` och `Linjerand (nautisk)`.

När det är kompatibelt är det möjligt att ställa in *Frame size*, en *Frame margin*, *Frame line thickness* med tillhörande färg och *Frame fill colors*.

Med hjälp av värdena `Latitud/Y only` och `Longitud/X only` i avsnittet Divisions kan du förhindra att en blandning av latitud/Y- och longitud/X-koordinater visas på varje sida när du arbetar med roterade kartor eller reprojicerade rutnät. Du kan också välja att ställa in synlig eller inte varje sida av rutnätsramen.

När kartobjektets utsträckning roteras (från gruppen *Huvudegenskaper*) eller om rutnätet har en annan CRS tillämpas, är det inte säkert att rutnätets linjer är ortogonala mot kartobjektets sidor. Detta kan leda till att rutnätet ser illa ut när

det stylas med inre och/eller yttre fästingar. Om du markerar *Följ rutnätsrotation* kommer fästingarna att anpassas till rutnätslinjerna. Dessutom kan du justera några fler egenskaper:

- *Ticks inriktning*: De inre och/eller yttre fästpunkterna kommer att vara parallella med motsvarande rutnätslinje. Deras inriktning kan vara:
  - **Orthogonal**: ticks på samma sida slutar på en linje, parallellt med sidan. Detta kan t.ex. leda till att vissa ticks blir längre när de har en låg vinkel mot ramen.
  - **Fixerad längd**: alla fästingar har samma längd, så de kanske inte är i linje
- *Skippa under vinkeln*: förhindrar visning av ticks för rutnätslinjer som skär ramens kant under ett angivet tröskelvärde
- *Marginal från kartans hörn*: förhindrar att fästingar visas för nära kartans hörn, eftersom de kan överlappa varandra och/eller vara utanför gränserna.

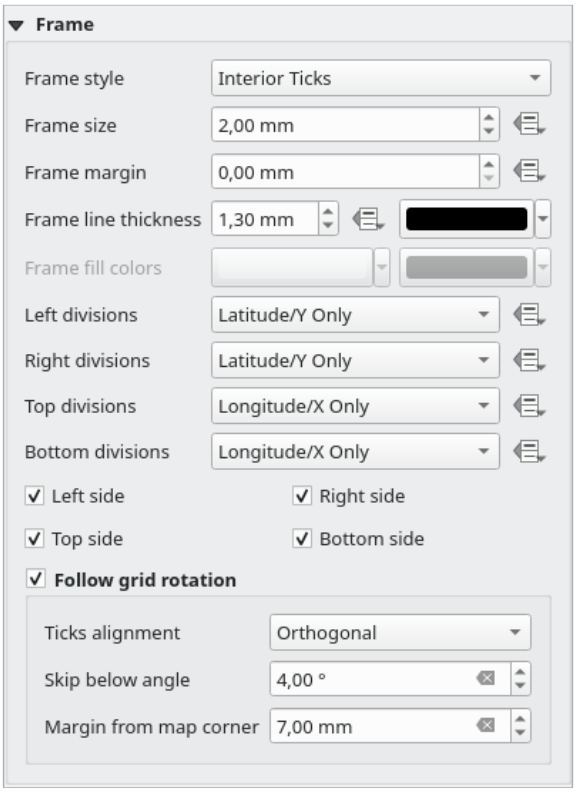


Fig. 18.20: Dialog för rutnätsram

Koordinater

Med kryssrutan *Draw coordinates* kan du lägga till koordinater i kartramen. Visade värden relaterar till den valda enheten *grid interval*. Du kan välja annotationens numeriska format, alternativen sträcker sig från decimal till grader, minuter och sekunder, med eller utan suffix, justerade eller inte och ett anpassat format med hjälp av uttrycksdialogen.

För var och en av sidorna *Left*, *Right*, *Top* och *Bottom* i rutnätsramen kan du ange:

- om koordinaterna ska återges: **Visa alla**, **Visa endast latitud/Y**, **Visa endast longitud/X**, **Avaktiverad**. Om du bara visar latitud/Y- eller longitud/X-värden i divisionerna kan du undvika att en blandning av latitud/Y- och longitud/X-koordinater visas på varje sida när du arbetar med roterade kartor eller omprojicerade rutnät.
- textens relativa position i förhållande till rutnätsramen: **Utvändig ram** eller **Invändig ram**
- placering och orientering av anteckningen:

- Horisontell bollhastighet
- **Vertikalt stigande**, **Vertikalt fallande**
- **Gränsens riktning**
- **Above tick**, **On tick**, **Under tick** när en tickbaserad ram används

Du kan också definiera *Font properties* (font, storlek, färg, buffert,…) *Distance to the map frame* och *Coordinate precision* (antal decimaler) för de ritade anteckningarna.

Följ rutnätsrotation: tillgänglig när kartans omfattning roteras eller rutnätet återprojiceras, hjälper den dig att justera annoteringarnas placering. Beroende på det valda placeringsläget roteras även anteckningarna:

- *Annotations alignment*: den kan vara **Orthogonal** eller av **Fixerad längd**
- *Skippa under vinkeln*: förhindrar visning av anteckningar för rutnätslinjer som skär ramens kant under ett angivet tröskelvärde
- *Marginal från kartans hörn*: förhindrar att anteckningar visas för nära kartans hörn, eftersom de kan överlappa och/eller ligga utanför gränserna.

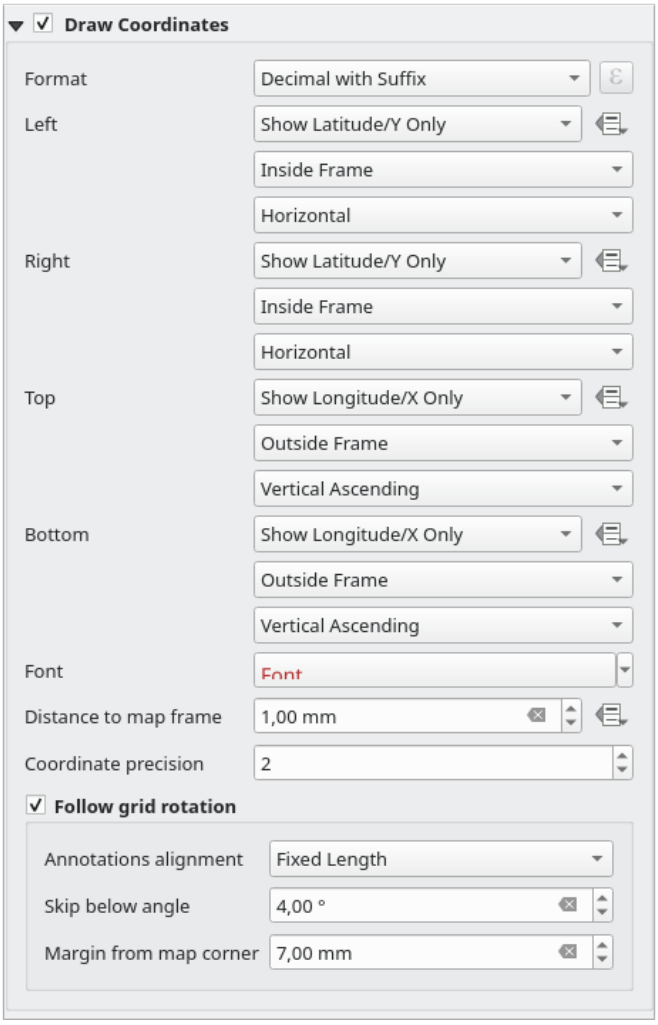


Fig. 18.21: Dialogrutan Grid Draw Coordinates

Översikter

Ibland har du kanske mer än en karta i den tryckta layouten och vill lokalisera studieområdet för ett kartobjekt på en annan karta. Det kan t.ex. vara för att hjälpa kartläsarna att identifiera området i förhållande till dess större geografiska sammanhang som visas på den andra kartan.

Gruppen *Oversikter* i kartpanelen hjälper dig att skapa en länk mellan två olika kartor och erbjuder följande funktioner:

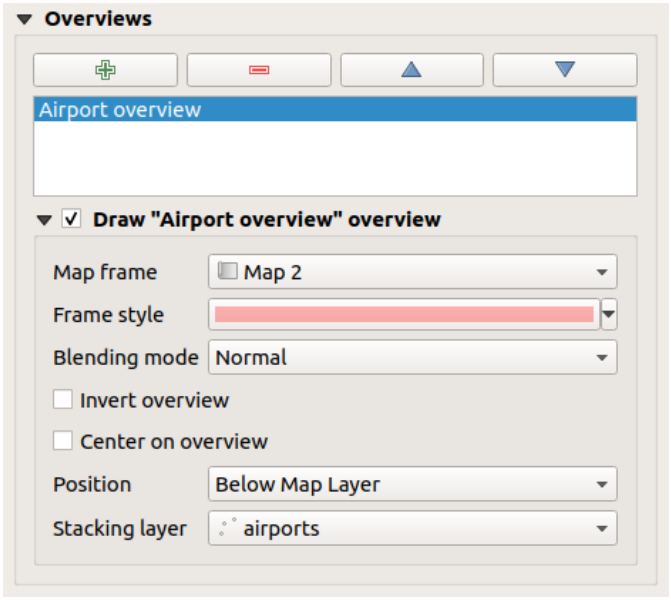


Fig. 18.22: Grupp för kartöversikter

Du skapar en översikt genom att markera det kartobjekt där du vill visa det andra kartobjektets utsträckning och expandera alternativet *Oversikter* i panelen *Objektegenskaper*. Tryck sedan på knappen för att lägga till en översikt. Ursprungligen heter denna översikt ”Översikt 1” (se Fig. 18.22). Det kan du göra:

- Byt namn på den med en dubbelklick
- Med knapparna och kan du lägga till eller ta bort översikter
- Med knapparna och flyttar du en översikt uppåt och nedåt i listan och placerar den över eller under andra översikter i kartobjektet (när de har samma *stapelposition*).

Välj sedan översiktsobjektet i listan och markera kryssrutan | *Rita ”<namn_översikt>” översikt* för att aktivera ritningen av översikten på den valda kartramen. Du kan anpassa den med:

- Med *Map frame* väljs det kartobjekt vars utbredning visas på det aktuella kartobjektet.
- *Frame Style* använder *symbol properties* för att rendera översiktsramen.
- Med *Blending mode* kan du ställa in olika blandningslägen för transparens.
- *Invert overview* skapar en mask runt utdragen när den aktiveras: de refererade kartutdragen visas tydligt, medan resten av kartobjektet blandas med ramens fyllnadsfärg (om en fyllnadsfärg används).
- *Center on overview* panorerar kartobjektets innehåll så att översiktsramen visas i mitten av kartan. Du kan bara använda ett översiktsobjekt för att centrera när du har flera översikter.
- *Position* styr exakt var i kartobjektets lagerstapel som översikten ska placeras, t.ex. så att en översiktsutbredning kan ritas under vissa funktionslager som vägar medan den ritas över andra bakgrundslager. Tillgängliga alternativ är:
  - *Nedanstående karta*

- *Below map layer* och *Above map layer*: placerar översiktsramen under respektive över geometrierna i ett lager. Lagret väljs i alternativet *Stacking layer*.
- *Below map labels*: eftersom etiketter alltid återges ovanför alla geometrier i ett kartobjekt, placeras översiktsramen ovanför alla geometrier och under alla etiketter.
- *Above map labels*: placerar översiktsramen ovanför alla geometrier och etiketter i kartobjektet.

18.2.3 Objektet 3D-karta

Objektet 3D-karta används för att visa en *3D-kartvy*. Använd knappen *Add 3D Map* och följ *instruktioner för skapande av objekt* för att lägga till ett nytt 3D Map-objekt som du senare kan manipulera på samma sätt som demonstreras i *Interagera med layoutobjekt*.

Som standard är ett nytt 3D Map-objekt tomt. Du kan ställa in egenskaperna för 3D-vyn och anpassa den i panelen *Item Properties*. Utöver *common properties* har den här funktionen följande funktioner (Fig. 18.23):

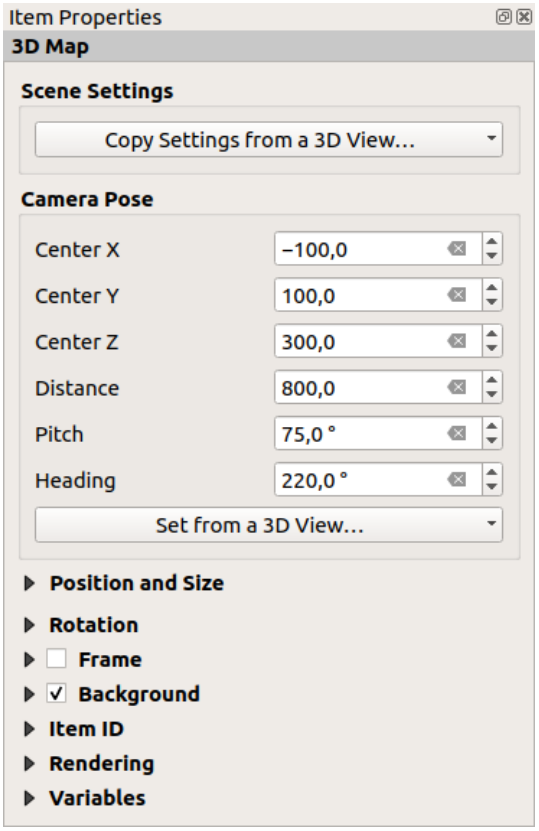


Fig. 18.23: egenskaper för 3D-kartobjekt

Sceninställningar

Tryck på *Kopiera inställningar från en 3D-vy…* för att välja vilken 3D-kartvy som ska visas.

3D-kartvyn återges med aktuell konfiguration (lager, terräng, belysning, kameraposition och -vinkel…).

Kamerans position

- *Center X* anger X-koordinaten för den punkt som kameran pekar mot
- *Center Y* anger Y-koordinaten för den punkt som kameran pekar mot
- *Center Z* anger Z-koordinaten för den punkt som kameran pekar mot
- *Distance* anger avståndet från kamerans mittpunkt till den punkt som kameran pekar mot
- *Pitch* ställer in kamerans rotation runt X-axeln (vertikal rotation). Värden från 0 till 360 (grader). 0°: terrängen sedd rakt uppifrån; 90°: horisontellt (från sidan); 180°: rakt nerifrån; 270°: horisontellt, upp och ner; 360°: rakt uppifrån.
- *Heading* ställer in kamerans rotation runt Y-axeln (horisontell rotation - 0 till 360 grader). 0°/360°: norr; 90°: väst; 180°: syd; 270°: öst.

I rullgardinsmenyn *Set from a 3D View…* kan du fylla i objekten med parametrarna för en 3D-vy.

18.2.4 Etikettobjektet

Objektet *Label* är ett verktyg som hjälper dig att dekorera din karta med texter som hjälper dig att förstå den; det kan vara titel, författare, datakällor eller annan information… Du kan lägga till en etikett med verktyget *Add Label* efter *instruktioner för skapande av objekt* och manipulera den på samma sätt som i *Interagera med layoutobjekt*.

Som standard har etikettobjektet en standardtext som du kan anpassa med hjälp av dess *Item Properties*-panel. Förutom *objektens gemensamma egenskaper* har den här funktionen följande funktioner (se Fig. 18.24):

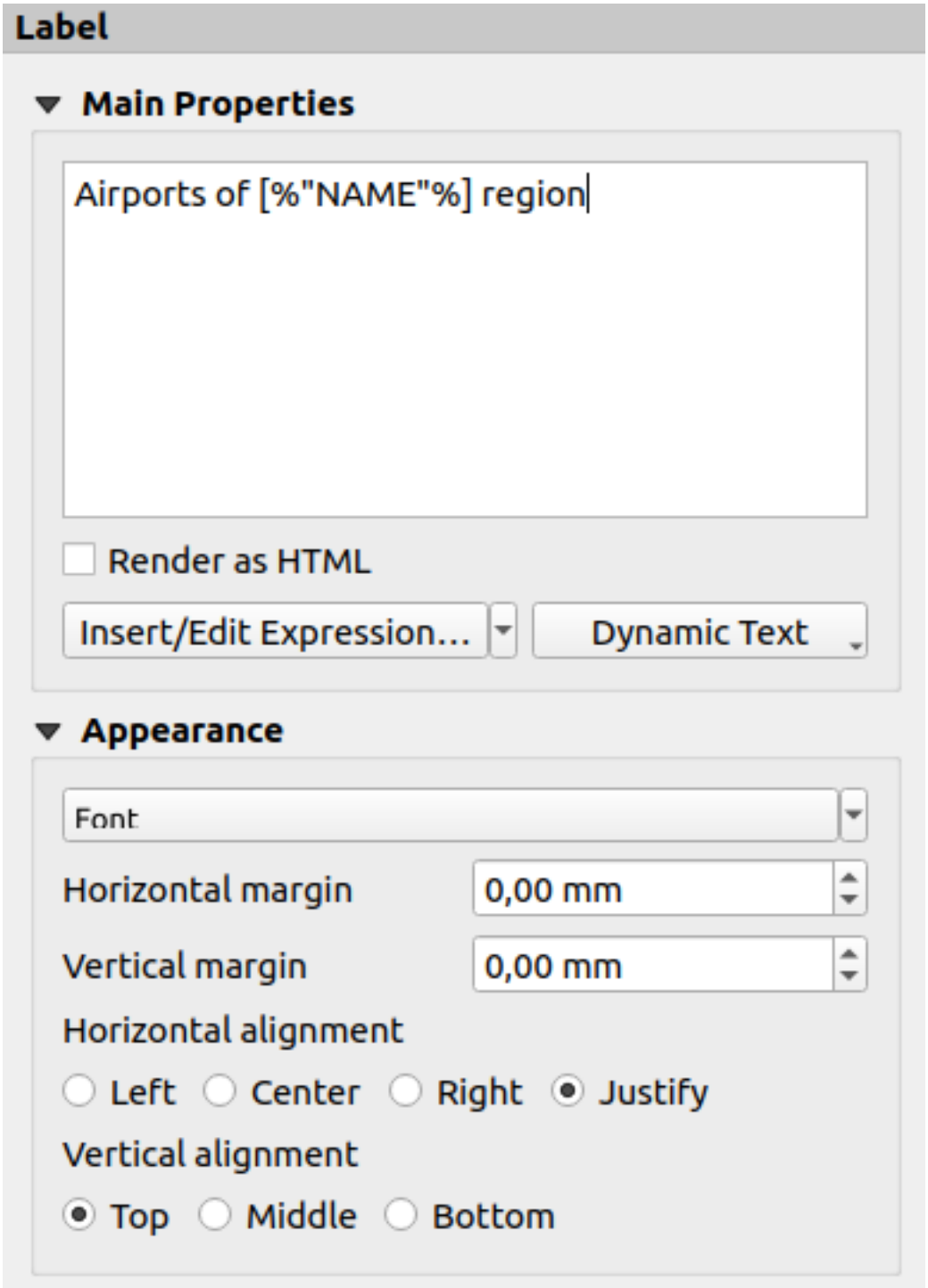


Fig. 18.24: Etikettobjektets egenskapspanel

Huvudsakliga egenskaper

Gruppen *Main properties* är platsen för att ange etikettens text. Texten kan vara statisk, dynamisk med *expression* funktioner och variabler, och/eller formaterad med HTML. Dynamiska delar av en etikett måste omges av `[%` och `%]` för att kunna tolkas och utvärderas som sådana.

- För att använda uttryck i etiketter kan du klicka på knappen *Infoga/redigera uttryck…*, skriva din formel som vanligt och när dialogen tillämpas lägger QGIS automatiskt till de omgivande tecknen.

  Råd

  När du klickar på knappen *Infoga/redigera uttryck…* utan att ha gjort något val i textrutan, kommer det nya uttrycket att läggas till den befintliga texten. Om du däremot vill ändra ett befintligt uttryck ändras beteendet beroende på vad du har markerat: Om någon text är markerad och markeringen är inom ett uttryck (mellan '[%' och '%]'), eller om ingen text är markerad men markören är inom ett uttryck, kommer hela uttrycket att markeras. I alla andra fall förblir markeringen som den är.

  Eftersom kartor vanligtvis fylls med vanlig textinformation (datum, författare, titel, sidnummer, …) ger QGIS direkt tillgång till motsvarande uttryck eller variabler: tryck på knappen *Dynamisk text* för att välja och infoga dem i din etikett.

  Tips

  Toppmenyn *Add Item* ► *Add Dynamic Text* ► kan användas för att skapa ett nytt etikettobjekt som fylls med det valda fördefinierade uttrycket.

  Det är möjligt att göra en dynamisk etikett statisk: tryck på rullgardinsmenyn bredvid knappen *Infoga/redigera uttryck…* och välj *Konvertera till statisk*. Alla dynamiska delar av etikettens innehåll kommer att utvärderas och ersättas med sina aktuella värden. Du kan sedan manuellt justera den resulterande texten när det behövs.

- Etiketter kan tolkas som HTML-kod: check *Render as HTML*. Du kan nu infoga HTML-taggar eller stilar, URL, en klickbar bild som länkar till en webbsida, eller något mer komplext…

Följande kod kombinerar HTML-rendering med uttryck, för en avancerad märkning och kommer att mata ut 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 -->
```

(continues on next page)

(fortsättning från föregående sida)

```
   <p align=center><img src="path/to/logos/qgis-logo-made-with-color.svg" alt=
↪"QGIS icon" style="width:80px;height:50px;"</p>
 </body>
</html>
```

Fig. 18.25: Utnyttja en etikett med HTML-styling

Utseende

- Definiera typsnitt och stil för texten genom att klicka på knappen *Font*. I menyn *Label Font* kan du använda några av alternativen för *Formatering av etikettexten*.
- Du kan ange olika horisontella och vertikala marginaler i `mm`. Detta är marginalen från kanten av layoutobjektet. Etiketten kan placeras utanför etikettgränserna, t.ex. för att anpassa etikettobjekt till andra objekt. I så fall måste du använda negativa värden för marginalen.
- Ett annat sätt att placera etiketten är att använda textjusteringen. Den kan vara:
  - *Left*, *Center*, *Right* eller *Justify* för *Horizontal alignment*
  - och *Top*, *Vertical Center*, *Bottom* för *Vertical alignment*.

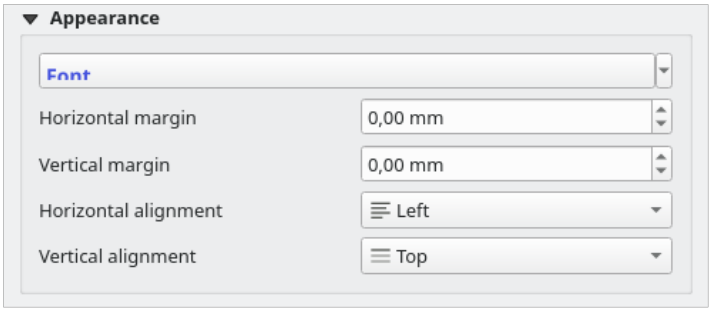


Fig. 18.26: Egenskaper för etikettobjektets utseende

Utforska uttryck i ett etikettobjekt

Nedan följer några exempel på uttryck som du kan använda för att fylla etikettobjektet med intressant information - kom ihåg att koden, eller åtminstone den beräknade delen, ska omges av `[%` och `%]` i ramen *Main properties*:

- Visa en titel med det aktuella atlasvärdet i ”field1”:

```
'This is the map for ' || "field1"
```

  eller, skrivet i avsnittet *Main properties*:

```
This is the map for [% "field1" %]
```

- Lägg till en paginering för bearbetade atlasfunktioner (t.ex. `Sida 1/10`):

```
concat( 'Page ', @atlas_featurenumber, '/', @atlas_totalfeatures )
```

- Returnerar namnet på flygplatserna i den aktuella atlasregionfunktionen, baserat på deras gemensamma attribut:

```
aggregate( layer := 'airports',
           aggregate := 'concatenate',
           expression := "NAME",
           filter := fk_regionId = attribute( @atlas_feature, 'ID' ),
           concatenator := ', '
         )
```

  Eller om en *attributsrelation* är inställd:

```
relation_aggregate( relation := 'airports_in_region_relation',
                    aggregate := 'concatenate',
                    expression := "NAME",
                    concatenator := ', '
                  )
```

- Returnerar namnet på de flygplatser som ingår i den aktuella atlasregionfunktionen, baserat på deras spatiala relation:

```
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:= ', '
               )
```

- Returnerar den nedre X-koordinaten för `Map 1`-objektets utsträckning:

```
x_min( map_get( item_variables( 'Map 1' ), 'map_extent' ) )
```

- Hämta namnet på lagren i den aktuella layouten `Map 1` och formatera dem i ett namn per rad:

```
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
 ),
 '\n' -- converts the list to string separated by breaklines
)
```

- Visa listan över lager med deras licenssträngar (användningsrättigheter) i ett objekt i layouten `Map 1`. Du måste fylla i lagrens *Åtkomstmetadata* egenskaper först.

```
array_to_string( map_credits( 'Map 1', true ) )
```

18.2.5 Det legendariska föremålet

Objektet *Legend* är en ruta eller en tabell som förklarar betydelsen av de symboler som används på kartan. En förklaring är sedan bunden till ett kartobjekt. Du kan lägga till ett legendobjekt med verktyget *Add Legend* enligt *instruktioner för skapande av objekt* och manipulera det på samma sätt som i *Interagera med layoutobjekt*.

Som standard visar legendobjektet alla tillgängliga lager och kan förfinas med hjälp av dess *Item Properties*-panel. Förutom *items common properties*, har denna funktion följande funktioner (se Fig. 18.27):

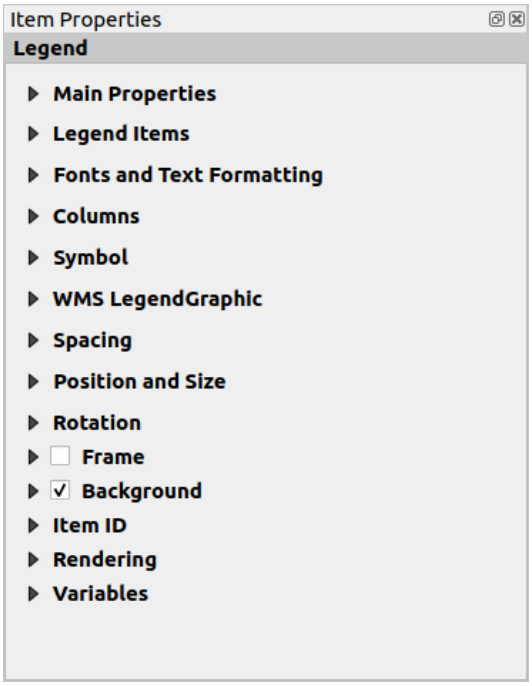


Fig. 18.27: Panel för egenskaper för legendobjekt

Huvudsakliga egenskaper

Gruppen *Main properties* i panelen *Item Properties* för förklaringen ger följande funktioner (se Fig. 18.28):

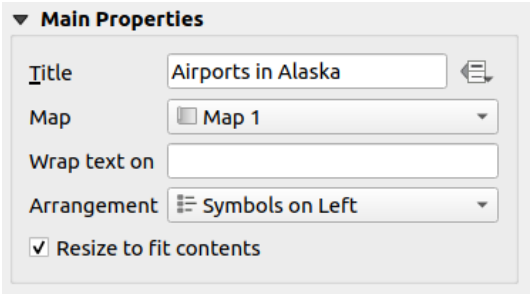


Fig. 18.28: Teckenförklaring Huvudegenskapsgrupp

I huvudegenskaperna kan du:

- Ändra *Title* för teckenförklaringen. Den kan göras dynamisk med inställningen *data-defined override*, vilket är användbart t.ex. vid generering av en atlas;
- Välj vilket *Map*-objekt som den aktuella förklaringen ska hänvisa till. Som standard väljs den karta över vilken legendobjektet ritas. Om det inte finns någon, faller den tillbaka till *referenskarta*.

> **Observera**
>
> *Variabler* för det länkade kartobjektet (@map_id, @map_scale, @map_extent…) är också tillgängliga från datadefinierade egenskaper för förklaringen.

- Ombryter texten i förklaringen på ett visst tecken: varje gång tecknet visas ersätts det med en radbrytning;
- Ange symbolernas och textens placering i teckenförklaringen: *Arrangement* kan vara *Symboler till vänster* eller *Symboler till höger*. Standardvärdet beror på vilken lokal som används (höger till vänster eller inte).
- Använd *Resize to fit contents* för att kontrollera om en teckenförklaring automatiskt ska anpassas till innehållet eller inte. Om kryssrutan är avmarkerad kommer teckenförklaringen aldrig att ändra storlek utan istället hålla sig till den storlek som användaren har ställt in. Allt innehåll som inte passar storleken beskärs.

Förklaringsobjekt

Gruppen *Legend items* i legendpanelen *Item Properties* ger följande funktioner (se Fig. 18.29):

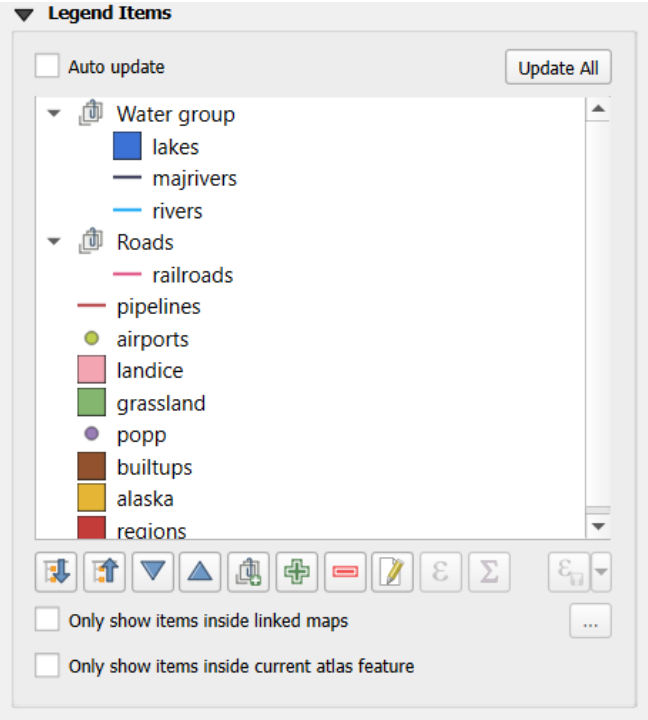


Fig. 18.29: Gruppen Legendobjekt

- Förklaringen uppdateras automatiskt om *Automatisk uppdatering* är markerad. När *Automatisk uppdatering* är avmarkerad får du mer kontroll över förklaringsobjekten. Alla ikoner under listan med förklaringsobjekt kommer att aktiveras.
- I fönstret Legend Items listas alla legendobjekt och du kan ändra objektordning, gruppera lager, ta bort och återställa objekt i listan, redigera lagernamn och symbologi samt lägga till ett filter.
  - Använd knapparna Expand all och Collapse all för att expandera respektive kollapsa alla grupper och undergrupper i förklaringsträdet. Se till att *Auto update* är avmarkerad för att använda dessa knappar.
  - Objektordningen kan ändras med hjälp av knapparna och eller med ”dra-och-släpp”-funktionalitet. Ordningen kan inte ändras för WMS-förklaringsgrafik.
  - Använd knappen för att lägga till en legendgrupp.

- Använd knappen för att lägga till lager och knappen för att ta bort grupper, lager eller symbolklasser.
- Knappen används för att redigera lagret, gruppnamnet eller titeln. Först måste du välja legendobjektet. Om du dubbelklickar på objektet öppnas också textrutan för att byta namn på det.
- Knappen använder uttryck för att anpassa varje symboletikett för det valda lagret (se *Data-definiera legendens etiketter*)
- Knappen lägger till en funktionsräkning för varje klass i vektorlagret.
- Med Filter förklaring by expression kan du filtrera vilka av legendobjekten i ett lager som ska visas, dvs. om du använder ett lager som har olika legendobjekt (t.ex. från en regelbaserad eller kategoriserad symbologi) kan du ange ett booleskt uttryck för att ta bort stilar från förklaringsträdet som inte har någon funktion som uppfyller ett villkor. Observera att funktionerna ändå behålls och visas i kartobjektet för layouten.

Standardbeteendet för legendobjektet är att efterlikna panelträdet *Layers* och visa samma grupper, lager och klasser av symbologi, men om du högerklickar på ett objekt kan du dölja lagrets namn eller visa det som en grupp eller undergrupp. Om du har gjort några ändringar i ett lager kan du återställa dem genom att välja *Reset to defaults* från den kontextuella menyn i legendposten.

När du har ändrat symbologin i QGIS huvudfönster kan du klicka på *Update All* för att anpassa ändringarna i teckenförklaringselementet i utskriftslayouten.

- Med kryssrutan *Visa endast objekt i länkade kartor* kommer endast de legendobjekt som är synliga i den länkade kartan att visas i förklaringen. Om du har mer än en karta kan du klicka på … och välja andra kartor från din layout. Detta verktyg är tillgängligt när *Auto-update* är aktiv.
- När du genererar en atlas med polygonfunktioner kan du filtrera bort legendobjekt som ligger utanför den aktuella atlasfunktionen. För att göra det, markera alternativet *Visa endast objekt inom aktuell atlasfunktion*.

Data-definiera legendens etiketter

kan du lägga till *expressions* till varje symbolmärkning i ett visst lager. Nya variabler (`@symbol_label`, `@symbol_id` och `@symbol_count`) hjälper dig att interagera med legendposten.

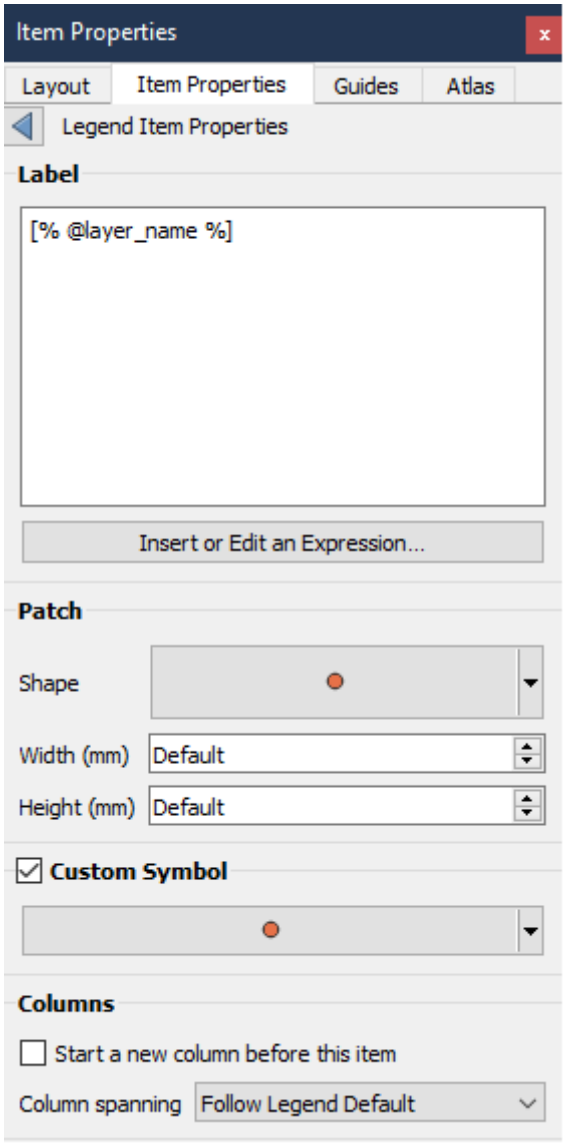
Om du t.ex. har ett lager med beteckningen ”regioner” som kategoriserats enligt fältet ”typ”, kan du till varje klass i teckenförklaringen lägga till antalet objekt och den totala ytan, t.ex. ”stadsdel (3) - 850 ha”:

1. Välj lagrets post i förklaringsträdet
2. Tryck på knappen för att öppna dialogrutan *Expression String Builder*
3. Ange följande uttryck (*förutsatt att symbolerna inte har ändrats*):

```
format( '%1 (%2) - %3ha',
        @symbol_label,
        @symbol_count,
        round( aggregate(@layer, 'sum', $area, filter:= "type"=@symbol_label)/
↪10000 )
      )
```

4. Tryck på *OK*

Anpassa legendobjekt



Legend-objekt kan också anpassas individuellt i *Legend Items Properties*. Men dessa anpassningar kan endast göras med *Auto update* avaktiverad.

Om du dubbelklickar på ett objekt eller trycker på Edit selected item properties kan du göra ytterligare anpassningar.

Label

För alla objekttyper är det möjligt att ändra etikettexten genom att skriva in eller infoga uttryck med hjälp av *Infoga eller redigera ett uttryck*. Uttryck kan också läggas till direkt var som helst i objektets etikett med hjälp av notationen [% expression %].

Kolumner

Med Legend Item Property kan du också styra kolumnindelningen genom att tvinga kolumnindelningen att ske efter ett visst objekt eller efter alla symboler i ett lager. Automatisk delning av ett lager och dess underordnade objekt kan också tillåtas eller blockeras på lagerbasis i den här widgeten.

Patch

För objekt med en symbol kan du i egenskapen Legend Item ange den maximala höjd och bredd som symbolen får ha.

För vektorsymboler kan du ange en anpassad form för symbolen. Formerna definieras vanligtvis av ett uttryck för att representera geometrin i ett enkelt plan, men dessa symboler kan också sparas i stilhanteraren och importeras senare. Standardsymbolen för varje geometrityp kan också styras via stilhanteraren.

Kundanpassad symbol

En anpassad symbol kan också anges för vektorsymboler. Detta kan vara användbart för att justera renderingen av en specifik symbol, för att förbättra den i förklaringen eller ha en symbol som är oberoende av förhandsgranskningen av den verkliga symbolen. Den här anpassade symbolen åsidosätter symbolen i teckenförklaringen, men tar hänsyn till symbolen *Patch* som angetts.

Typsnitt och textformatering

Gruppen *Typsnitt och textformatering* i legendpanelen *Postegenskaper* innehåller följande funktioner:

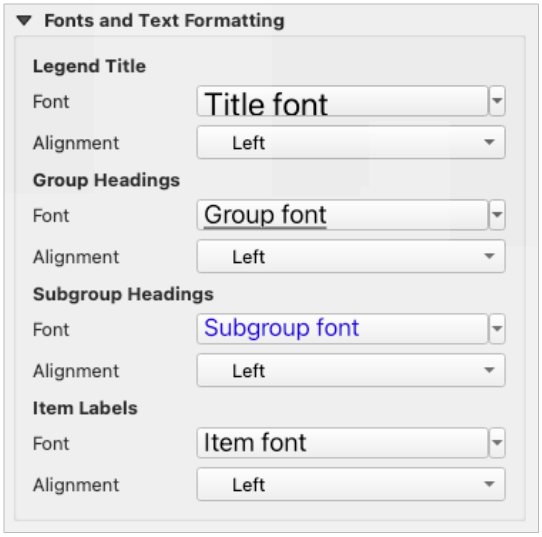


Fig. 18.30: Legend Fonts egenskaper

- Du kan ändra teckensnittet för legendtiteln, gruppen, undergruppen och objektet (funktionen) i legendobjektet med hjälp av widgeten *font selector*, som ger alla funktioner i *text formatting* (kontroll av teckensnittsavstånd, blandad HTML-formatering, färgläggning, blandning, bakgrund, textbuffert, skugga, …)
- För var och en av dessa nivåer kan du ange texten *Alignment*: den kan vara *Left* (standard för vänster-till-höger-baserade lokala), *Center* eller *Right* (standard för höger-till-vänster-baserade lokala).

Kolumner

Under gruppen *Columns* i panelen förklaring *Item Properties* kan legendobjekt ordnas över flera kolumner:

- Ställ in antalet kolumner i fältet *Count* 1,00. Detta värde kan göras dynamiskt, t.ex. genom att följa atlasfunktioner, legendinnehåll, ramstorlek…
- *Equal column widths* anger hur kolumnerna i förklaringen ska justeras.
- Med alternativet *Split layers* kan en kategoriserad eller graderad lagerförklaring delas upp mellan kolumnerna.

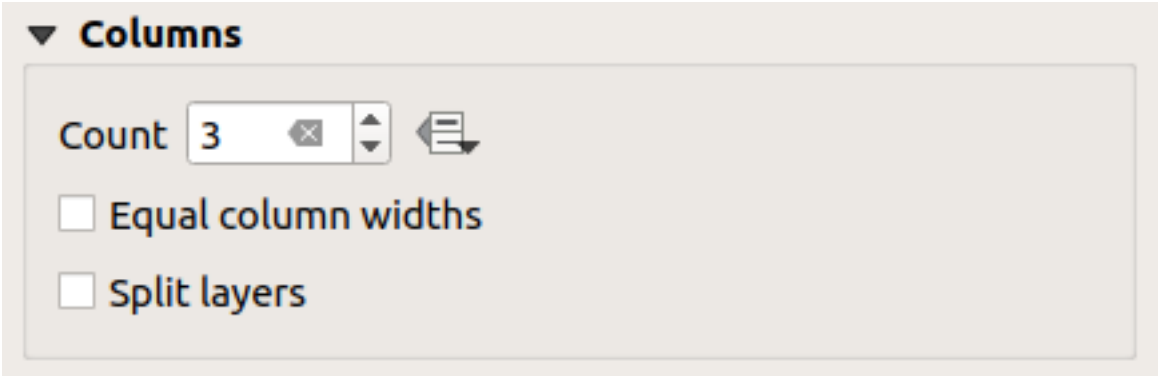


Fig. 18.31: Inställningar för Legend-kolumner

Symbol

Gruppen *Symbol* på panelen *Item Properties* i teckenförklaringen konfigurerar storleken på de symboler som visas bredvid teckenförklaringens etiketter. Du kan göra följande:

- Ställ in *Symbol width* och *Symbol height*
- Ställ in markörernas *Min symbolstorlek* och *Max symbolstorlek*: `0.00mm` betyder att inget värde har ställts in.
- *Draw stroke for raster symbols*: detta lägger till en kontur till symbolen som representerar bandfärgen i rasterlagret; du kan ställa in både *Stroke color* och *Tickness*.

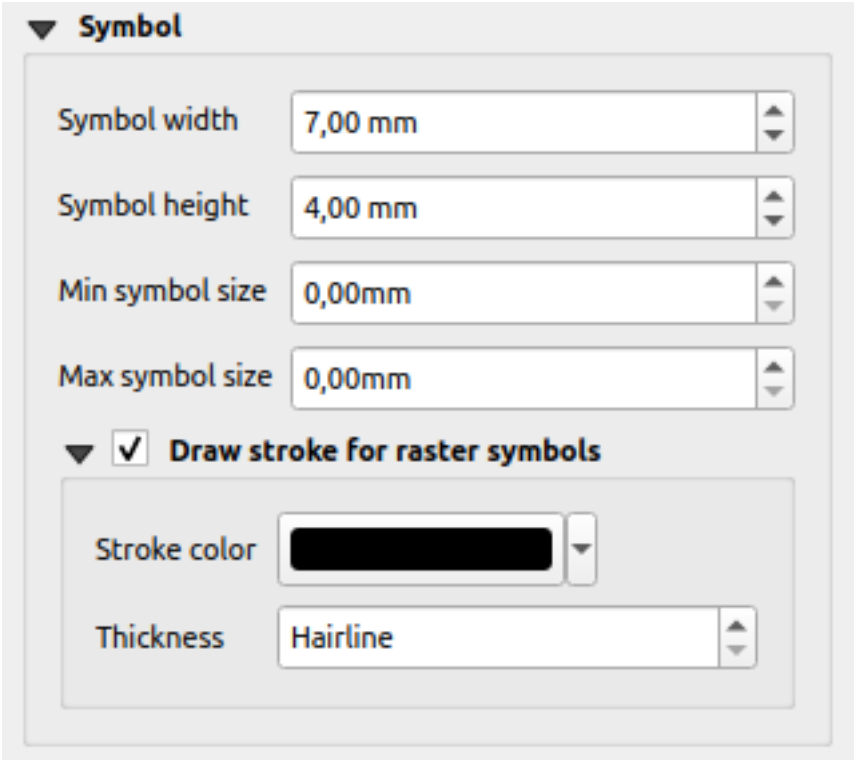


Fig. 18.32: Teckenförklaring Symbolkonfiguration

WMS LegendGrafik

Avsnittet *WMS LegendGraphic* i panelen förklaring *Item Properties* ger följande funktioner (se Fig. 18.33):

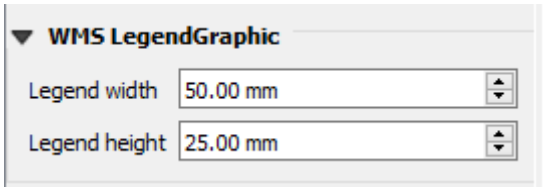
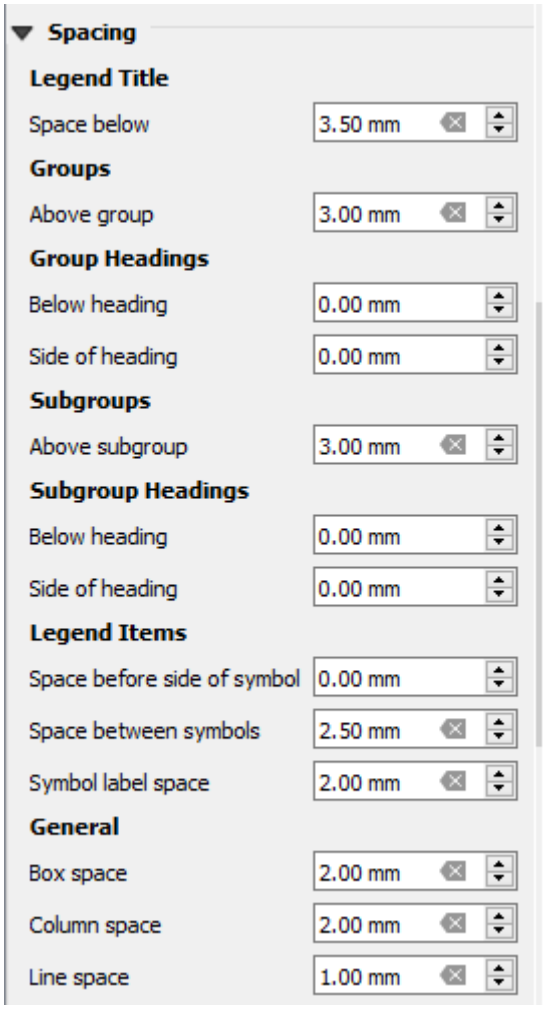


Fig. 18.33: WMS LegendGrafik

När du har lagt till ett WMS-lager och infogar ett legendobjekt skickas en begäran till WMS-servern om att tillhandahålla en WMS-förklaring. Denna förklaring kommer endast att visas om WMS-servern tillhandahåller GetLegendGraphic-funktionen. Innehållet i WMS-förklaringen kommer att tillhandahållas som en rasterbild.

WMS LegendGraphic används för att kunna justera *Legend width* och *Legend height* för WMS förklaring rasterbild.

Avstånd



I avsnittet *Spacing* kan du anpassa avståndet i teckenförklaringen. Avståndet kan vara till stor hjälp för att ange grupperingen av objekt i teckenförklaringen och deras relation.

Avståndet runt och före titel, grupper, undergrupper, symboler, etiketter, rutor, kolumner och rader kan anpassas via denna dialogruta.

18.2.6 Objektet för skalfältet

Skalstreck ger en visuell indikation på storleken på objekten och avståndet mellan objekten på kartobjektet. Ett objekt med skalstreck kräver ett kartobjekt. Använd verktyget *Add Scale Bar* efter *instruktioner för skapande av objekt* för att lägga till ett nytt skalstrecksobjekt som du senare kan manipulera på samma sätt som i *Interagera med layoutobjekt*.

Som standard visar ett nytt skalstapelobjekt skalan för det kartobjekt som det ritas över. Om det inte finns något kartobjekt nedanför används *referenskartan*. Du kan anpassa den i panelen *Item Properties*. Förutom *objektens gemensamma egenskaper* har den här funktionen följande funktioner (se Fig. 18.34):

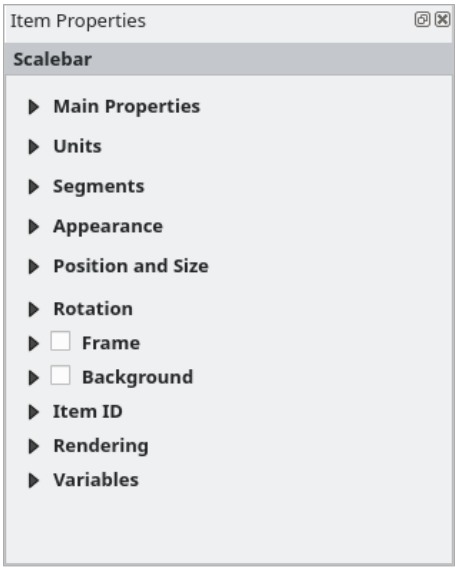


Fig. 18.34: Panel för egenskaper för objekt i skalfält

Huvudsakliga egenskaper

Gruppen *Huvudegenskaper* i skalstapelns panel *Egenskaper för objekt* ger följande funktioner (se Fig. 18.35):

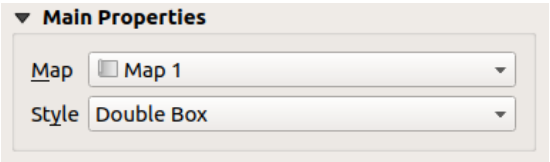


Fig. 18.35: Scale Bar Huvudsakliga egenskaper grupp

1. Välj först den karta som skalstocken ska fästas på
2. Välj sedan stil för skalfältet. Tillgängliga stilar är:
   - *stilarna *Single box** och **Double box**, som innehåller en eller två rader med rutor som växlar färg;
   - **Mitten**, **Upp** eller **Ned** linjetickar;
   - **Stepped line** stil som ritar en steppad linje som representation av en scalebar

- **Hollow**-stil som ritar en enda ruta med alternerande färg för segmenten, med horisontella linjer genom alternerande segment
- **Numerisk**, där skalförhållandet skrivs ut (t.ex. `1:50000`).

3. Ställ in egenskaper på lämpligt sätt

Enheter

Gruppen *Units* i skalstapelns panel *Item Properties* ger funktioner för att ställa in visningsenheter och viss textformatering (se Fig. 18.36):

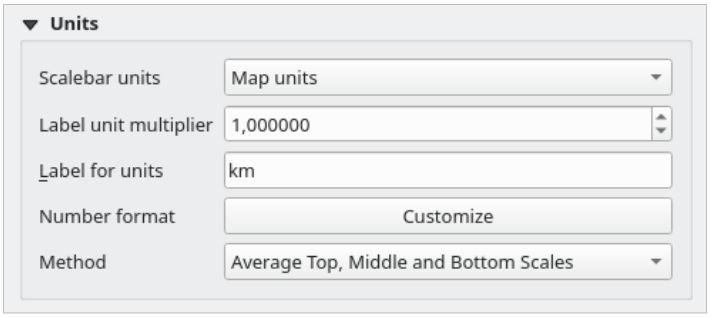


Fig. 18.36: Skalstreck Enheter grupp

- Välj de enheter som du vill använda med *Scalebar units*. Det finns många möjliga val: **Map Units** (standard), **Meters**, **Feet**, **Miles** eller **Nautical Miles**… och några derivat. Konvertering av enheter hanteras automatiskt.
- Multiplikatorn för *Märkningsenhet* anger hur många skalstrecksenheter per märkt enhet. Om t.ex. skalstrecksenheterna är inställda på ”meter”, kommer en multiplikator på 1000 att resultera i skalstrecksmärken i ”kilometer”.
- Fältet *Label for units* definierar den text som används för att beskriva enheterna i skalstrecket, t.ex. `m` eller `km`. Detta bör matchas för att återspegla multiplikatorn ovan.
- Tryck på *Customize* bredvid *Number format* för att få kontroll över alla formateringsegenskaper för siffrorna i skalfältet, inklusive tusentalsavgränsare, decimaler, vetenskaplig notation etc. (se *Nummerformatering* för mer information). Mycket användbart när du gör kartor för målgrupper utanför den aktuella QGIS-lokalen, eller när du vill variera stilen från de lokala standardinställningarna (t.ex. lägga till tusentalsavgränsare när den lokala standardinställningen är att dölja dem).
- Välj *Metod* för beräkning av skalstreck: Beroende på kartans CRS och omfattning är det inte säkert att det avstånd som rapporteras i skalstrecket är detsamma som mäts över kartan. Med det här alternativet kan du ange vilken metod du vill använda: *Average Top, Middle and Bottom Scales*, *Calculate along Top of Map*, *Calculate along Middle of Map* eller *Calculate along Bottom of Map*.

Segment

Gruppen *Segments* i skalstapelns panel *Item Properties* innehåller funktioner för att konfigurera antalet och storleken på segment och underavdelningar (se Fig. 18.37):

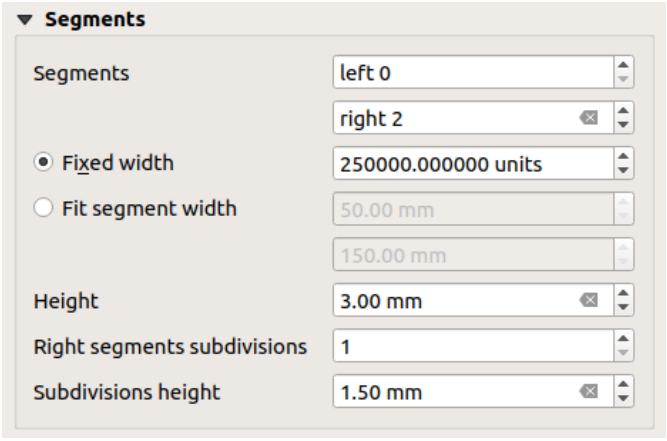


Fig. 18.37: Skala Bar Segment grupp

- Du kan definiera antalet *Segment* som ska ritas på vänster och höger sida av skalstapelns `0`:
  - antal underavdelningar av ett unikt segment på *Vänster* sida
  - antal segment på den *högra* sidan
- Du kan ställa in bredden på ett segment eller ett intervall för den totala längden på skalstrecket:
  - Ange hur långt ett segment ska vara i dina skalstapel-enheter (*Fixed width*)
  - Eller begränsa den totala storleken på skalstrecket i `mm` med alternativet *Fit segment width*. I det senare fallet ändras skalstrecket (och dess etikett uppdateras) varje gång kartskalan ändras så att det passar mellan det övre och det nedre intervallet.
- *Height* används för att definiera höjden på stapeln.
- *Underindelning av högersegment* används för att definiera antalet sektioner som skalstreckets högersidessegment kan ha (för skalstrecksstilarna *Line Ticks Down*, *Line Ticks Middle* och *Line Ticks Up*) .
- *Subdivision height* används för att definiera höjden på subdivisionssegmentet.

Utseende

Gruppen *Appearance* i panelen Scale Bar *Item Properties* har följande funktioner:

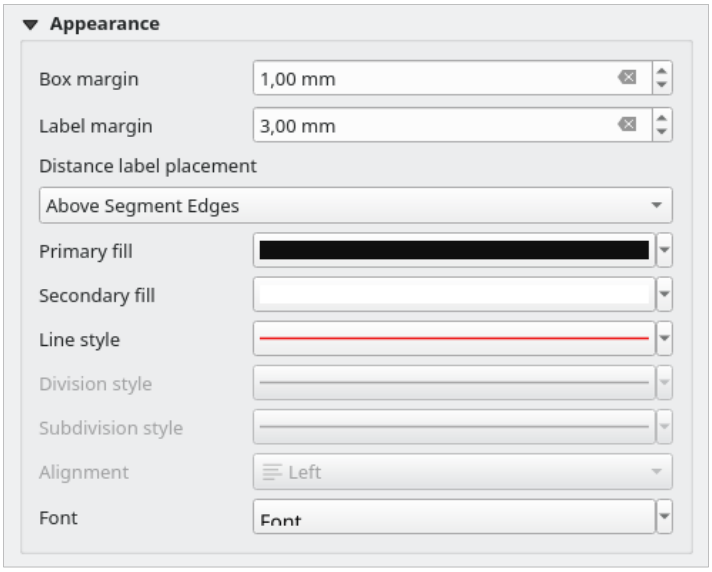


Fig. 18.38: Grupp Utseende för skalstreck

Du kan definiera hur skalstrecket ska visas i sin ram.

- *Boxmarginal* : utrymme mellan text och ramkanter
- *Label margin* : utrymme mellan text och skalstreck
- *Distance label placement* definierar textens position i förhållande till skalstapelsegmenten: *Above Segment Edges*, *Above Segment Centers*, *Below Segment Edges* eller *Below Segment Centers*
- *Primär fyllning* och *Sekundär fyllning* av skalstapeln med *egenskaper för fyllningssymboler* (färg, opacitet, mönster, effekter …) — för stilarna *Single Box*, *Double Box* och *Hollow*
- *Linjestil* för skalstrecket med hjälp av *linjesymbolernas egenskaper* (färg, streck, skarv, lockstil, mönster, effekter…) — för alla utom *Numerisk* stil
- *Division style* och *Subdivision style* för delnings- respektive underindelningssegment i *Line Ticks Up*, *Line Ticks Middle* och *Line Ticks Down* skalstapelstilar med *linjesymbolers egenskaper* (färg, stroke, join, cap style, mönster, effekter…)
- *Alignment* placerar texten på vänster, mitten eller höger sida av ramen (endast för *Numeric* skalfältsstil)
- *Font* för att ställa in *egenskaperna* (storlek, typsnitt, färg, bokstavsavstånd, skugga, bakgrund…) för skalstapelns etikett.

Eftersom de flesta visningsegenskaperna för skalstrecket är beroende av symboler vars egenskaper kan vara datadefinierade, är det möjligt att rendera datadefinierade skalstreck.

Exempel: Följande kod som tillämpas på fetstilsegenskapen för skaletiketterna visar siffror i fetstil när de är en multipel av 500:

```
-- returns True (or 1) if the value displayed on the bar
-- is a multiple of 500

@scale_value % 500 = 0
```

18.2.7 Tabellobjekten

Du kan använda tabellposter för att dekorera och förklara din karta:

- *Attribute table*: extraherar automatiskt en delmängd av attributen för ett lager, baserat på fördefinierade regler
- *Fixed table*: gör det möjligt att skapa tabeller med innehåll som matas in manuellt (dvs. i kalkylbladsstil), så att du kan skapa helt anpassade tabeller.

Attributtabellens objekt

Alla lager i projektet kan få sina attribut visade i utskriftslayouten. Använd verktyget *Add Attribute Table* efter *instruktioner för skapande av objekt* för att lägga till ett nytt tabellobjekt som du senare kan manipulera på samma sätt som i *Interagera med layoutobjekt*.

Som standard laddar en ny attributtabell de första raderna i det första (alfabetiskt sorterade) lagret, med alla fält. Du kan dock anpassa tabellen med hjälp av panelen *Item Properties*. Förutom *items common properties*, har denna funktion följande funktioner (se Fig. 18.39):

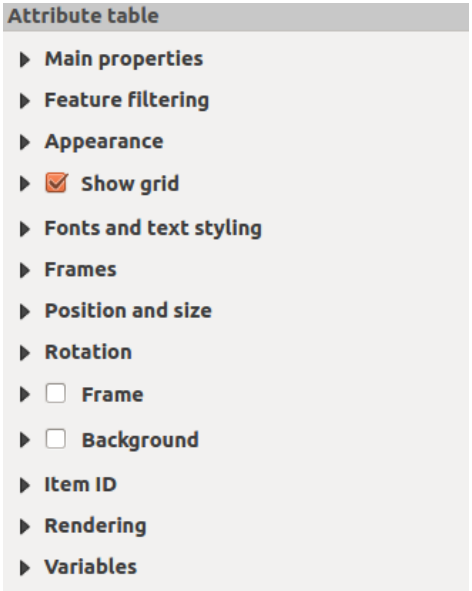


Fig. 18.39: Attributtabell Panel för artikelegenskaper

Huvudsakliga egenskaper

Gruppen *Main properties* i attributtabellen tillhandahåller följande funktioner (se Fig. 18.40):

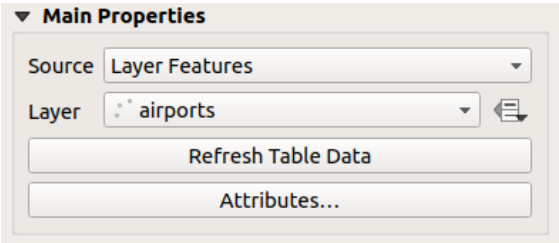


Fig. 18.40: Attributtabell Huvudsakliga egenskaper Grupp

- För *Source* kan du som standard bara välja **Layer features** så att du kan välja ett *Layer* från de vektorlager som laddats i projektet.

Med knappen Data-defined override nära lagerlistan kan du dynamiskt ändra det lager som används för att fylla i tabellen, t.ex. kan du fylla attributtabellen med olika lagerattribut per atlassida. Observera att den tabellstruktur som används (Fig. 18.43) är den för det lager som visas i rullgardinslistan *Layer* och den lämnas intakt, vilket innebär att om du ställer in en datadefinierad tabell på ett lager med olika fält kommer det att resultera i tomma kolumner i tabellen.

Om du aktiverar alternativet *Generate an atlas* i panelen *Atlas* (se *Skapa en Atlas*) finns det ytterligare två *Source*-alternativ:

- **Aktuell atlasfunktion** (se Fig. 18.41): du kommer inte att se något alternativ för att välja lager, och tabellobjektet kommer bara att visa en rad med attributen från den aktuella funktionen i atlasens täckningslager.
- och **Relationsbarn** (se Fig. 18.42): ett alternativ med relationsnamnen kommer att visas. Den här funktionen kan endast användas om du har definierat en *relation* med ditt atlastäckningsskikt som överordnat, och tabellen kommer att visa barnraderna i atlastäckningsskiktets aktuella funktion.

- Knappen *Refresh Table Data* kan användas för att uppdatera tabellen när det faktiska innehållet i tabellen har ändrats.

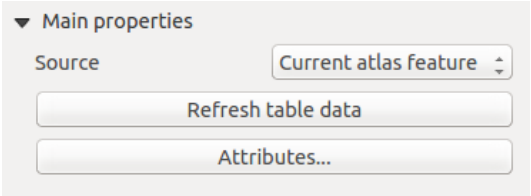


Fig. 18.41: Attributtabell Huvudegenskaper för ”Aktuell atlasfunktion

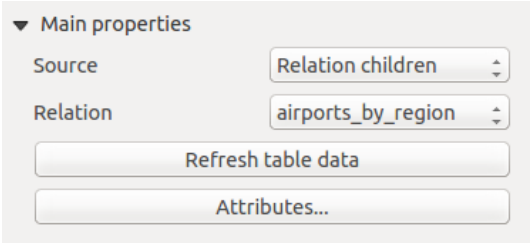


Fig. 18.42: Attributtabell Huvudegenskaper för ’Relation barn’

- Knappen *Attributes…* startar dialogen *Select Attributes*, (se Fig. 18.43) som kan användas för att ändra det synliga innehållet i tabellen. Den övre delen av fönstret visar listan över de attribut som ska visas och den nedre delen hjälper dig att sortera data.

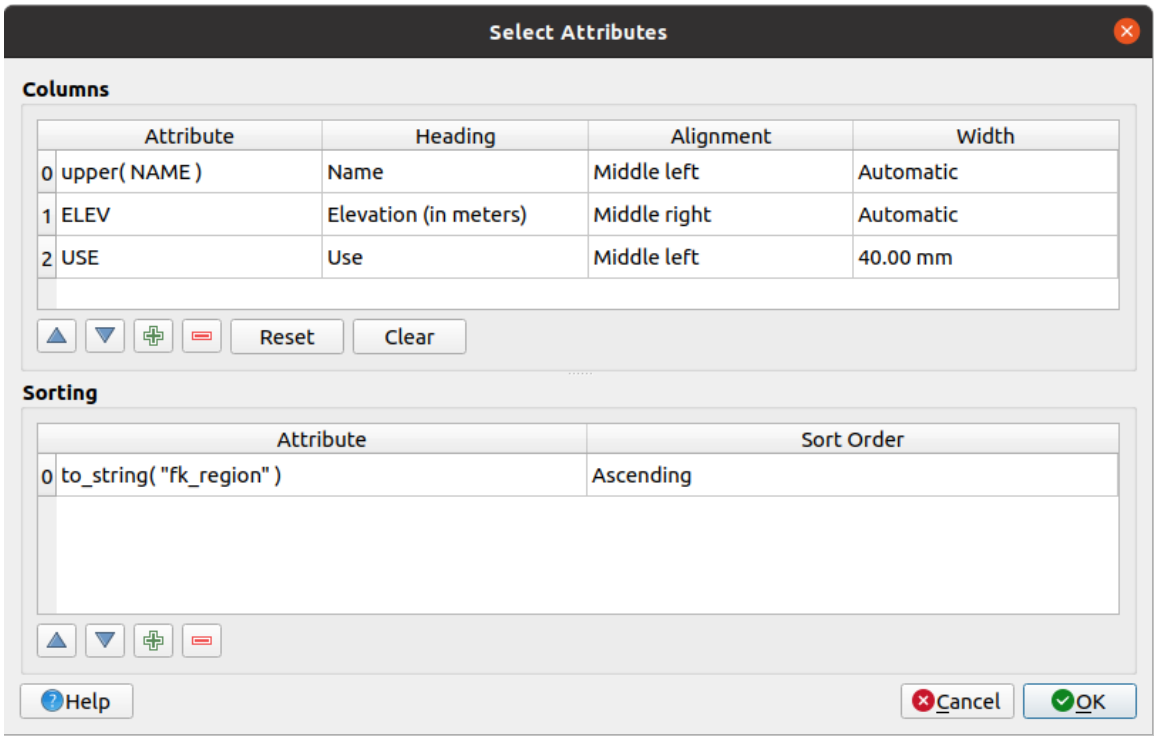


Fig. 18.43: Attributtabell Välj attribut Dialog

I avsnittet *Columns* kan du:

- Flytta attribut uppåt eller nedåt i listan genom att markera raderna och sedan använda knapparna och för att flytta raderna. Flera rader kan markeras och flyttas samtidigt.
- Lägg till ett attribut med knappen . Detta lägger till en tom rad längst ner i tabellen där du kan välja ett fält som ska vara attributvärdet eller skapa ett attribut via ett reguljärt uttryck.
- Ta bort ett attribut med knappen . Flera rader kan markeras och tas bort samtidigt.
- Återställ attributtabellen till dess standardläge med knappen *Reset*.
- Rensa tabellen med hjälp av knappen *Clear*. Detta är användbart när du har en stor tabell men bara vill visa ett litet antal attribut. Istället för att manuellt ta bort varje rad kan det gå snabbare att rensa tabellen och lägga till de rader som behövs.
- Cellrubrikerna kan ändras genom att lägga till den anpassade texten i kolumnen *Heading*.
- Celljustering kan hanteras med kolumnen *Alignment* som bestämmer textens position i tabellcellen.
- Cellbredden kan hanteras manuellt genom att lägga till egna värden i kolumnen *width*.

I avsnittet *Sortering* kan du:

- Lägg till ett attribut att sortera tabellen med: tryck på knappen så läggs en ny tom rad till. Infoga ett fält eller ett uttryck i kolumnen *Attribute* och ställ in *Sort order* till **Ascending** eller **Descending**.
- Markera en rad i listan och använd knapparna och för att ändra sorteringsprioriteten på attributnivå. Om du markerar en cell i kolumnen *Sorteringsordning* kan du ändra sorteringsordningen för attributfältet.
- Använd knappen för att ta bort ett attribut från sorteringslistan.

Filtrering av funktioner

Gruppen *Feature filtering* i attributtabellen tillhandahåller följande funktioner (se Fig. 18.44):

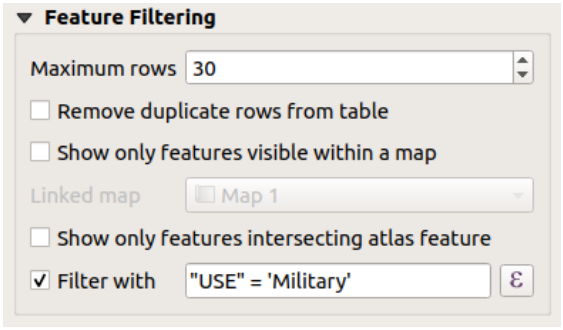


Fig. 18.44: Attributtabell Funktionsfiltrering Grupp

Det kan du:

- Definiera *Maximum rows* som ska visas.
- Aktivera *Remove duplicate rows from table* för att endast visa unika poster.
- Aktivera kryssrutan *Visa endast synliga egenskaper inom en karta* och välj motsvarande *Länkad karta* vars attribut för synliga egenskaper ska visas.
- Activate *Visa endast objekt som skär Atlasobjektet* är endast tillgänglig när *Generera en atlas* är aktiverad. När den är aktiverad visas en tabell med endast de objekt som korsar det aktuella atlasobjektet.
- Aktivera *Filter with* och ange ett filter genom att skriva in indataraden eller infoga ett reguljärt uttryck med hjälp av den givna -uttrycksknappen. Några exempel på filtreringssatser som du kan använda när du har laddat flygplatslagret från Sample-datasetet:
  - `ELEV > 500`
  - `NAME = 'ANIAK'`
  - `NAME NOT LIKE 'AN%'`
  - `regexp_match( attribut( $aktuell egenskap, 'USE' ) , '[i]')`

  Det sista reguljära uttrycket kommer endast att inkludera de flygplatser som har bokstaven ”i” i attributfältet ”USE”.

Utseende

Gruppen *Appearance* för attributet tabell ger följande funktioner (se Fig. 18.45):

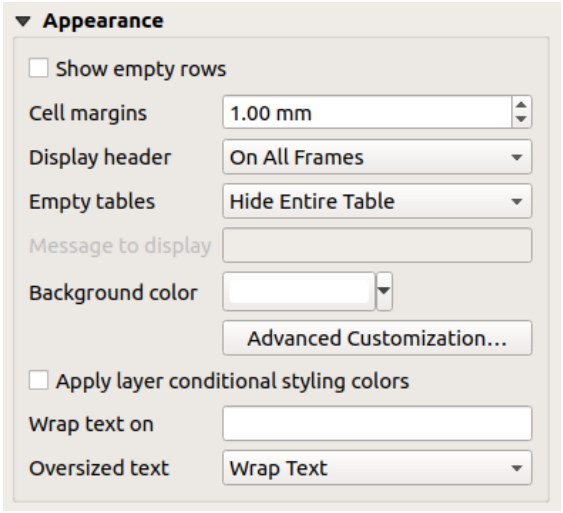


Fig. 18.45: Attributtabellens utseende Grupp

- Klicka på *Visa tomma rader* för att fylla attributtabellen med tomma celler. Detta alternativ kan också användas för att ge ytterligare tomma celler när du har ett resultat att visa!
- Med *Cellmarginaler* kan du definiera marginalen runt texten i varje cell i tabellen.
- Med *Display header* kan du från en lista välja ett av standardalternativen ”On first frame”, ”On all frames” eller ”No header”.
- Alternativet *Empty table* styr vad som ska visas när resultaturvalet är tomt.
  - **Draw headers only**, kommer bara att rita rubriken utom om du har valt ”No header” för *Display header*.
  - **Hide entire table**, kommer endast att rita bakgrunden till tabellen. Du kan aktivera *Don't draw background if frame is empty* i *Frames* för att helt dölja tabellen.
  - **Visa inställt meddelande**, ritar rubriken och lägger till en cell som sträcker sig över alla kolumner och visar ett meddelande som ”Inget resultat” som kan anges i alternativet *Meddelande att visa*
- Alternativet *Meddelande att visa* är endast aktiverat när du har valt **Visa inställt meddelande** för *Tom tabell*. Det angivna meddelandet visas i tabellen på den första raden när resultatet är en tom tabell.
- Med *Background color* kan du ställa in bakgrundsfärgen för tabellen med hjälp av widgeten *color selector*. Alternativet *Advanced customization* hjälper dig att definiera olika bakgrundsfärger för varje cell (se Fig. 18.46)

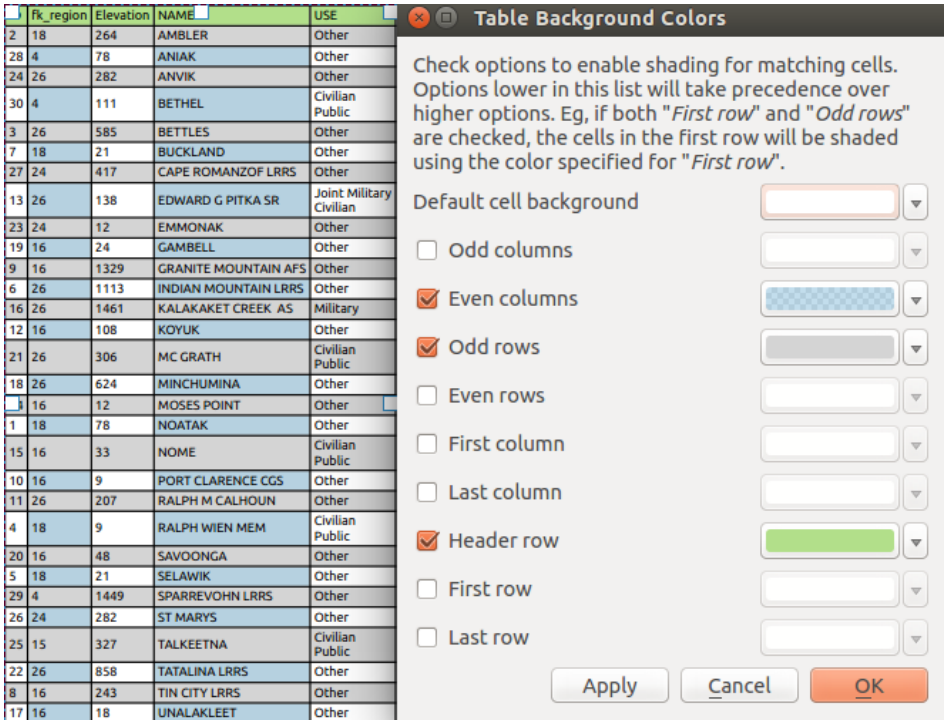


Fig. 18.46: Attributtabell Avancerad bakgrundsdialog

- *Apply layer conditional styling colors*: den *conditional table formatting* som finns i lagret tillämpas i layoutattributtabellen (bakgrundsfärg, teckensnittsfamilj och egenskaper som fet, kursiv, strykning, understrykning, färg, …). Regler för villkorlig formatering har företräde framför andra formateringsinställningar i layouttabellen, t.ex. åsidosätter de andra inställningar för cellbakgrundsfärg, t.ex. alternerande radfärger.
- Med alternativet *Wrap text on* kan du definiera ett tecken som cellinnehållet ska brytas om varje gång det uppfylls
- Med *Oversized text* definierar du beteendet när den bredd som ställts in för en kolumn är mindre än innehållets längd. Det kan vara **Viktig text** eller **Skära av text**.

Observera

Fler egenskaper för attributtabellobjektet beskrivs i avsnittet *Tabeller gemensamma funktioner*.

Den fasta tabellposten

Ytterligare information om kartan kan infogas manuellt i en tabell genom att välja *Add Fixed Table* och genom att följa *items creation instructions* för att lägga till ett nytt tabellobjekt som du senare kan manipulera på samma sätt som beskrivs i *Interagera med layoutobjekt*.

Som standard visas en tom tabell med två minimerade kolumner och rader i kartlayouten. Du måste anpassa tabellen i panelen *Item Properties*. Förutom *objektens gemensamma egenskaper* har den här funktionen följande funktioner:

Huvudsakliga egenskaper

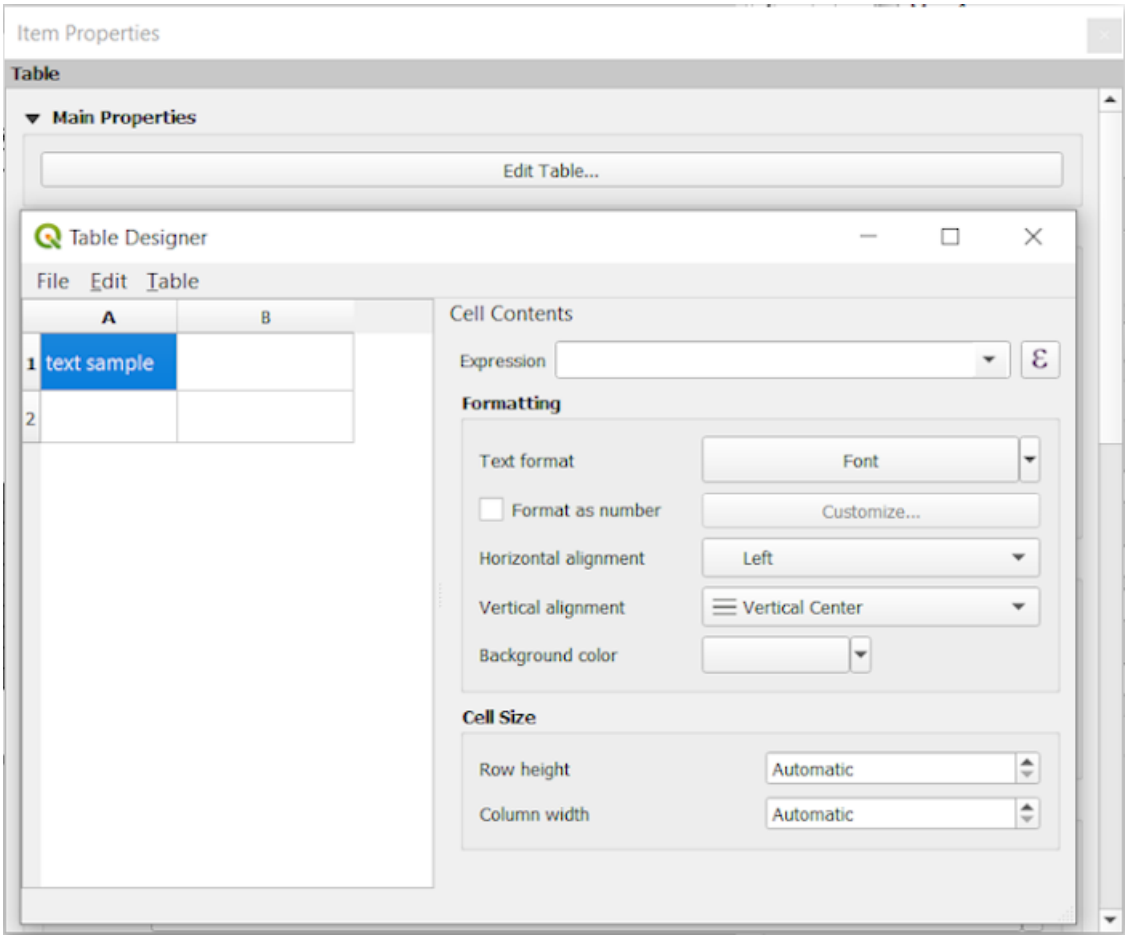


Fig. 18.47: Fast panel för tabellobjektsegenskaper med tabelldesigner

I *Main properties*, tryck på *Edit table…* för att öppna *Table designer* dialogen där du kan bygga din tabell:

- Till vänster i dialogrutan finns en tabellram som visar det aktiva tabellobjektet med dess innehåll och inställningar. Klicka eller dubbelklicka på en cell och skriv en text för att ändra dess innehåll. Tryck på `Ctrl+Enter` för att lägga till radbrytning (för en flerradig text). Logiken för att välja flera celler i tabellen är densamma som i vektorlagrets *attributtabell*. Ta tag i och flytta en rad- eller kolumnseparator för att ändra storlek på raden ovanför eller kolumnen före. Om du dubbelklickar på separatorn ändras storleken på raden ovanför eller kolumnen före så att den passar innehållet. Om du högerklickar på kolumn- eller radrubrikerna visas en kontextuell meny där du kan infoga kolumner eller rader runt markeringen eller ta bort dem.
- Via menyerna ovanför dialogrutan är det möjligt att:
  - *File* ► *Import Content From Clipboard*: det åsidosätter innehållet i tabellen.
  - *Arkiv* ► *Stäng redigerare*: stänger dialogrutan *Tabelldesigner*.
  - Från menyn *Edit* kan du arbeta med urvalsfunktioner för rader och kolumner:
    - *Välj alla* celler i tabellen
    - *Select columns* eller *Select Rows*: när celler är markerade i tabellen kan du utöka markeringen till deras kolumn(er) respektive rad(er)
    - *Clear Cells*: raderar innehållet i de markerade cellerna
  - Menyn *Table* är den plats där du utformar bordets struktur. Du kan göra följande:

        * *Infoga rader* ► ovanför eller under urvalet
        * *Infoga kolumner* ► före eller efter urvalet
        * *Delete Rows* eller *Delete Columns* för markerade celler
        * *Merge selected cells*: markera flera celler i en rektangulär form och du kan slå samman dem och sammanfoga deras innehåll till den övre vänstra cellen (vars styling också tillämpas på den sammanslagna cellen, såvida det inte finns en sista rad/kolumn-cell i markeringen).
        * *Split selected cells*: välj en sammanslagen cell så kan du dela upp den i sina enskilda celler. Den aktuella texten behålls i cellen längst upp till vänster och de andra cellerna fylls med sitt innehåll innan de slås samman. De återfår också sin ursprungliga styling.
        * *Include Header Row* vars styling endast kan kontrolleras från widgeten *Typsnitt och textstyling*

- Markera cellerna och till höger om tabellramen kan du ange *Cell Contents*:
  - Ange antingen innehållet i varje cell manuellt eller använd ett *Uttryck* för att automatiskt fylla i den.
  - Under gruppen *Formatting* anger du specifika alternativ som gäller för urvalet och som har företräde framför *global styling av tabellinnehåll*:
    * genom att välja *Textformat*
    * by *Formatera som nummer* (flera format är tillgängliga)
    * genom att definiera *Horizontal alignment* och *Vertical alignment*
    * genom att välja en *Bakgrundsfärg*
  - Definiera *Cell Size* med *Row height* och *Column width*.

Utseende

Gruppen *Appearance* i den fasta tabellen tillhandahåller följande funktioner:

- Klicka på *Visa tomma rader* för att fylla attributtabellen med tomma celler.
- Med *Cellmarginaler* kan du definiera marginalen runt texten i varje cell i tabellen.
- Med *Display header* kan du från en lista välja ett av standardalternativen ”On first frame”, ”On all frames” eller ”No header”.
- Med *Background color* kan du ställa in tabellens bakgrundsfärg med hjälp av widgeten *color selector*. Alternativet *Avancerad anpassning* hjälper dig att definiera olika bakgrundsfärger för varje cell.
- Med *Oversized text* definierar du beteendet när den bredd som ställts in för en kolumn är mindre än innehållets längd. Det kan vara **Viktig text** eller **Skära av text**.

Observera

Fler egenskaper för det fasta tabellobjektet beskrivs i avsnittet *Tabeller gemensamma funktioner*.

Tabeller gemensamma funktioner

Visa rutnät

Gruppen *Visa rutnät* i tabellobjekten ger följande funktioner (se Fig. 18.48):

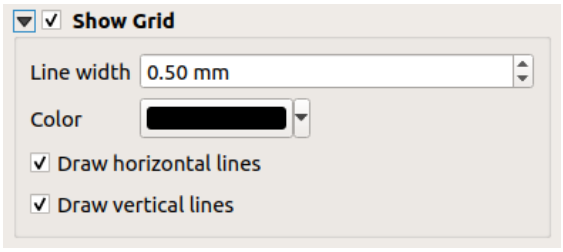


Fig. 18.48: Attributtabell Visa rutnät Grupp

- Aktivera kryssrutan *Show grid* när du vill visa rutnätet, dvs. konturerna av tabellcellerna. Du kan också välja att antingen *Draw horizontal lines* eller *Draw vertical lines* eller båda.
- Med *Line width* kan du ställa in tjockleken på de linjer som används i rutnätet.
- Rutnätets *Färg* kan ställas in med hjälp av widgeten för färgval.

Typsnitt och textstyling

Gruppen *Fonts and text styling* i tabellobjekten tillhandahåller följande funktioner (se Fig. 18.49):

Fig. 18.49: Attributtabell Typsnitt och textstyling Grupp

- Du kan definiera *Font*-egenskaper för *Table heading* och *Table contents* med hjälp av den avancerade *text settings*-widgeten (med buffert, skugga, färgeffekter, transparens, bakgrund, färg, …). Observera att dessa ändringar inte påverkar de celler som har tilldelats ett anpassat teckensnitt, antingen från avsnittet *Appearance* eller från dialogrutan *Table Designer*. Endast celler med standardrendering skrivs över.
- För *Table heading* kan du dessutom ställa in *Alignment* till `Follow column alignment` eller åsidosätta denna inställning genom att välja `Left`, `Center` eller `Right`. Kolumninriktningen ställs in med hjälp av dialogrutan *Select Attributes* (se Fig. 18.43).

Ramar

Gruppen *Frames* i egenskaperna för tabellobjekt ger följande funktioner (se Fig. 18.50):

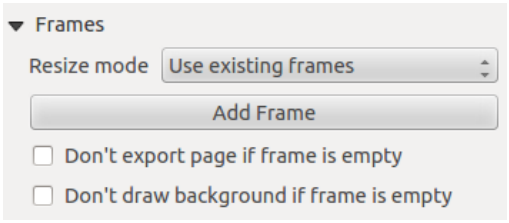


Fig. 18.50: Attributtabell Ramar Grupp

- Med *Resize mode* kan du välja hur innehållet i attributtabellen ska återges:
  - `Use existing frames` visar resultatet endast i den första ramen och i tillagda ramar.
  - `Extend to next page` skapar så många ramar (och motsvarande sidor) som behövs för att visa hela urvalet i attributtabellen. Varje ram kan flyttas runt på layouten. Om du ändrar storlek på en ram kommer den resulterande tabellen att delas upp mellan de andra ramarna. Den sista ramen trimmas så att den passar tabellen.
  - `Upprepa tills det är klart` skapar också lika många ramar som alternativet *Fortsätt till nästa sida*, förutom att alla ramar har samma storlek.
- Använd knappen *Add Frame* för att lägga till en ny ram med samma storlek som den valda ramen. Resultatet av tabellen som inte får plats i den första ramen kommer att fortsätta i nästa ram när du använder storleksändringsläget `Use existing frames`.
- Aktivera *Exportera inte sidan om ramen är tom* förhindrar att sidan exporteras när tabellramen inte har något innehåll. Detta innebär att alla andra layoutobjekt, kartor, skalfält, legender etc. inte kommer att synas i resultatet.
- Aktivera *Don't draw background if frame is empty* förhindrar att bakgrunden ritas när tabellramen inte har något innehåll.

18.2.8 Markören, bilden och föremålen med norrpil

- *Bildobjektet*
  - *Huvudsakliga egenskaper*
  - *Storlek och placering*
  - *Bildrotation*
- *North Arrow-objektet*
- *Markeringsobjektet*

Tillsammans med kart- eller legendobjekten i utskriftslayouten kanske du vill dekorera din realisering med bilder eller anteckningar. QGIS tillhandahåller olika verktyg för att uppnå detta:

- *bildobjekt*: dekorerar layouten med en bildraster- eller SVG-fil (t.ex. logotyper, bilder, norrpilar, …)
- *north arrow item*: ett bildelement som är fördefinierat med en bild av en norrpil

- *markeringsobjekt*: dekorerar layouten med QGIS-vektorn *symboler*. Det kan användas för att placera markörer över ett kartobjekt eller för att skapa avancerade anpassade legender.

Bildobjektet

Du kan lägga till en bild genom att dra den från din filhanterare till duken, klistra in den direkt i layouten med `Ctrl+V` eller *Redigera* ► *Klistra in* och genom att använda Lägg till bild, enligt *instruktioner för skapande av objekt*. Sedan kan du manipulera den, som förklaras i *Interagera med layoutobjekt*.

När du använder Lägg till bild, kommer bildobjektet att vara en tom ram som du kan anpassa med hjälp av dess *Objektegenskaper*-panel. Förutom *objektens gemensamma egenskaper* har den här funktionen följande funktioner:

Huvudsakliga egenskaper

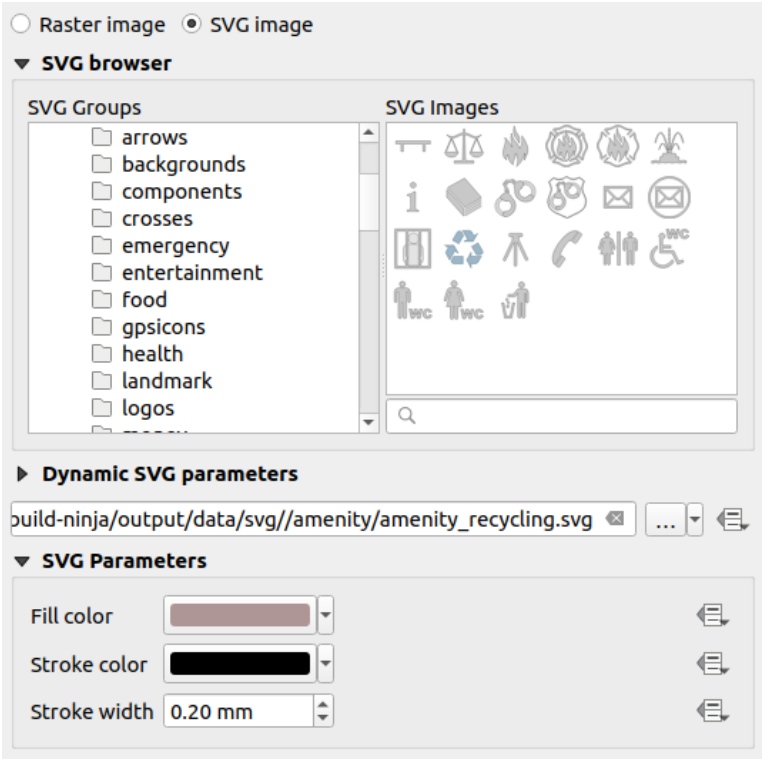


Fig. 18.51: Panelen Egenskaper för bildobjekt

Bildobjektet stöder två typer av bilder:

- *Raster Image*: En filväljarwidget kan användas för att hämta data. Använd … Browse-knappen för att välja en fil på din dator eller ange sökvägen direkt i textfältet. Du kan även ange en fjärr-URL som pekar på en bild. Den associerade bilden kan också *inbäddas* i layouten.

  Använd knappen data defined override för att ange bildkällan från ett funktionsattribut eller med hjälp av ett reguljärt uttryck.

- *SVG-bild*: använder som standard SVG-biblioteken som finns i *Inställningar* ► *Alternativ* ► *System* ► *SVG-sökvägar*. Du kan dock använda vilken annan fil som helst, och filvalet följer samma regler som för rasterbilden. SVG-parametrarna kan också ställas in dynamiskt.

  De QGIS-levererade (standard) `.SVG`-filerna är anpassningsbara, vilket innebär att du enkelt kan använda andra *Fyll color*, *Stroke color* (inklusive opacitet) och *Stroke width* än originalet, med hjälp av motsvarande funktion i gruppen *SVG Parameters*. Dessa egenskaper kan också vara *data-defined*.

Om du lägger till en `.SVG`-fil som inte aktiverar dessa egenskaper kan du behöva lägga till följande taggar i filen för att lägga till stöd för t.ex. transparens:

- *fill-opacity="param(fill-opacity)"*
- *stroke-opacity="param(outline-opacity)"*

Mer information finns på *Parametriserbar SVG*.

Observera

Om du drar och släpper en bildfil (raster eller SVG) på layoutsidan skapas ett layoutbildobjekt med motsvarande inställningar.

Storlek och placering

Fig. 18.52: Egenskaper för storlek och placering av layoutbilder

Med alternativet *Resize mode* kan du ställa in hur bilden ska visas när ramen ändras i storlek:

- `Zoom`: förstorar/förminskar bilden till ramen samtidigt som bildens bildförhållande bibehålls
- `Stretch`: sträcker ut bilden så att den får plats i ramen
- `Clip`: använd detta läge endast för rasterbilder, det ställer in bildens storlek till originalbildens storlek utan skalning, och ramen används för att klippa bilden. Så endast den del av bilden som är inuti ramen kommer att vara synlig.
- `Zoom and resize frame`: förstorar bilden så att den passar i ramen och ändrar sedan storleken på ramen så att den passar de resulterande bildmåtten
- `Resize frame to image size`: ställer in storleken på ramen så att den matchar bildens originalstorlek (ingen skalning)

Beroende på det valda *Resize mode* kan alternativen *Placement* och *Image rotation* vara inaktiverade. *Placement* låter dig välja bildens position inom ramen (topp/mitt/botten och vänster/centrum/höger).

Bildrotation

Bilder kan roteras med fältet *Image rotation*. Om kryssrutan *Sync with map* aktiveras synkroniseras bildens rotation med den rotation som tillämpas på det valda kartobjektet. Detta är en praktisk funktion för att få en bild att fungera som en norrpil. Den *North alignment* kan vara:

- **Grid north**: riktningen för en gridlinje som är parallell med mittmeridianen i det nationella/lokala gridnätet
- **True north**: riktning för en längdmeridian.

Du kan också lägga till en deklination *Offset* till bildrotationen.

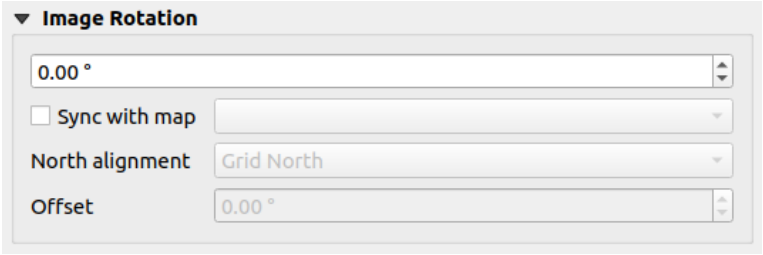


Fig. 18.53: Layoutbilder Egenskaper för bildrotation

North Arrow-objektet

Du kan lägga till en pil mot norr med knappen Add North Arrow, enligt *instruktioner för skapande av objekt* och manipulera den på samma sätt som i *Interagera med layoutobjekt*.

Eftersom norrpilar är bilder har objektet *North Arrow* samma egenskaper som objektet *picture item*. De viktigaste skillnaderna är:

- En standardpil mot norr används när objektet läggs till, i stället för en tom ram
- Nordpilens objekt synkroniseras med ett kartobjekt som standard: egenskapen *Sync with map* är den karta över vilken nordpilens objekt ritas. Om det inte finns någon faller den tillbaka till *referenskarta*.

Observera

Många av norrpilarna har inte ett ”N” tillagt i norrpilen. Detta är avsiktligt, eftersom det finns språk som inte använder ett ”N” för norr.

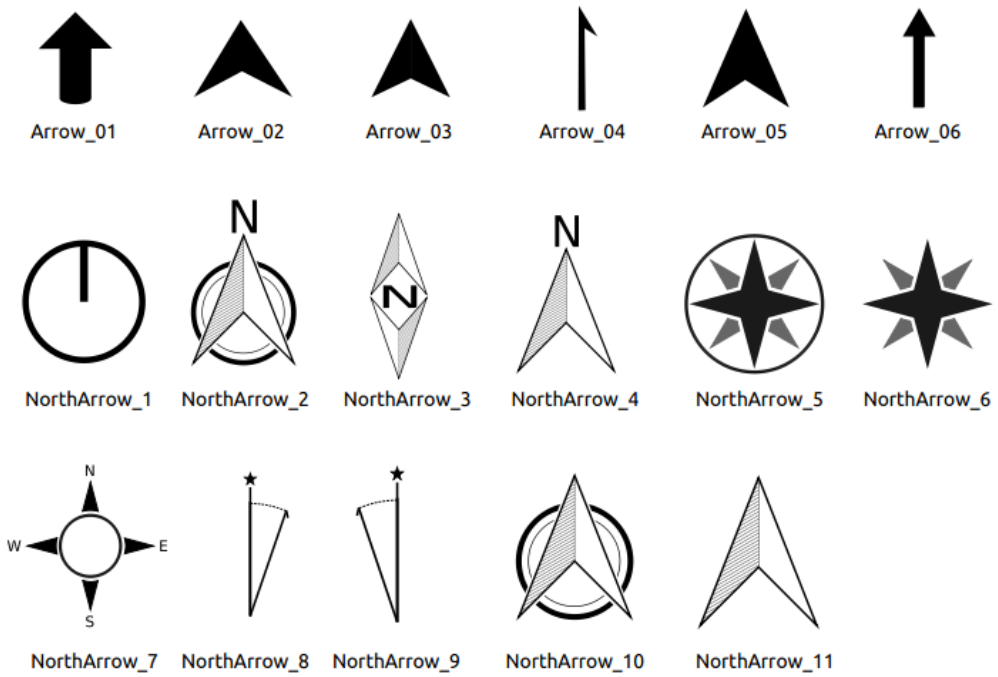


Fig. 18.54: Nordpilar tillgängliga för urval i det SVG-bibliotek som tillhandahålls

Markeringsobjektet

Om du vill lägga till en markör väljer du knappen Add Marker och klickar på sidan. En standardsymbol för en punktmarkör läggs till. Sedan kan du manipulera den, enligt förklaringen i *Interagera med layoutobjekt*. Men observera att till skillnad från de flesta andra objekt kan du ändra storlek på objektet eftersom dess storlek styrs av de inbäddade symbolernas egenskaper.

Markörobjektet kan anpassas från panelen *Item Properties*. Förutom *objektens gemensamma egenskaper* kan du också:

- ändra *Symbol*, med stöd av alla symbolens *widgetfunktioner*
- synkroniserar markörobjektets rotation med kartans (se *Bildrotation*), och fungerar som en norrpil. Kartans rotation läggs till eventuell befintlig rotation på markörsymbolnivå (så om du t.ex. måste rotera triangelmarkören 90° för att få den att peka rakt upp, kommer den fortfarande att fungera fint i norrpilsläge)

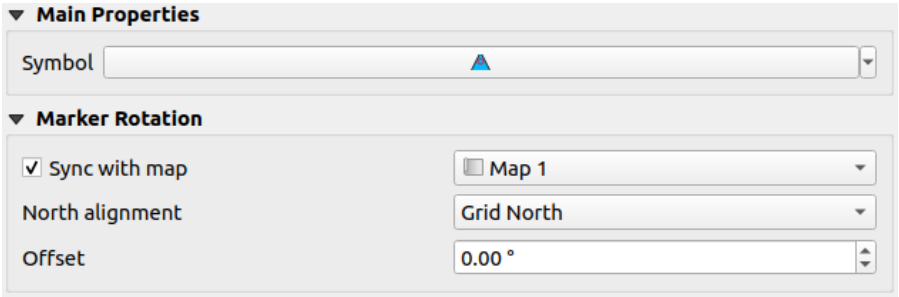


Fig. 18.55: Markörobjektets anpassade egenskaper

18.2.9 Höjdprofilens objekt

Höjdprofilobjektet används för att visa en *höjdprofilvy* i en layout. Använd knappen *Add Elevation Profile* och följ *instruktioner för skapande av objekt* för att lägga till ett nytt höjdprofilobjekt som du senare kan manipulera på samma sätt som demonstreras i *Interagera med layoutobjekt*.

Ett nytt höjdprofilobjekt har standardinställningar som gör att det renderar ett tomt diagram. Du kan anpassa dess egenskaper i panelen *Item Properties*. Förutom *common properties* har den här funktionen följande funktioner:

Panelen Elevation profile *Item Properties* innehåller ett övre verktygsfält med följande funktioner:

- Aktualisera höjdprofil för att uppdatera objektets rendering
- Kopiera från höjdprofil: en rullgardinsmeny för att välja en höjdprofilvy från. Inställningarna för vyn tillämpas på höjdprofilobjektet i layouten och kan ändras i efterhand.

Sikt

Under gruppen *Layers* markerar du i trädvyn de lager som du vill återge i profilobjektet. Kom ihåg att konfigurera egenskaperna *Elevation* för de valda lagren på rätt sätt.

Profilerad kurva

- *Kontrolleras av atlas*: *profilkurvan* hämtas från den aktuella atlasfunktionen och höjdprofilvyn uppdateras när du går igenom atlasfunktionerna. Detta stöds för närvarande för en aktiv layoutatlas eller rapport som använder ett täckningslager av linjegeometrityp.
- Avståndet *Tolerance*, som kan vara datadefinierat, hjälper dig att kontrollera hur långt en egenskap i de synliga lagren ska vara från profilkurvan för att visas i layoutens höjdvy. För närvarande returneras endast punktobjekt.

Diagrammets intervall

Ett höjdprofilobjekt i en layout visar inte nödvändigtvis hela omfattningen av den höjdprofilvy som det baseras på. Du kan begränsa området som ska återges genom att tillhandahålla:

- på X-axeln, *Minimum distance* och *Maximum distance* från profilkurvans startpunkt
- På Y-axeln visas *Minsta höjd* och *Maxsta höjd*

Avstånd och höjdaxlar

Grupperna *Distance axis* och *Elevation axis* ger alternativ för att justera rutnätet över höjdprofilobjektet, på X- respektive Y-axlarna:

- avståndsvisning *Unit*, som gör det möjligt att åsidosätta kartans enhet
- gradering på axeln med både *Major interval* och *Minor interval*
- linjesymbolerna som ska användas för motsvarande *Major grid lines* och *Minor grid lines*
- hur ofta graderingsobjekten ska märkas (*Label interval*) samt deras *Label format* och *Label font*
- *Distance labels*: anger om symbolen för avståndsenheten ska placeras bredvid *Every value*, *First value*, *Last value*, *First and last values*, eller hoppas över (*None*)

Diagramområde

Under *Chart area* kan du konfigurera återgivningen av det område där höjdprofilplotten faktiskt visas:

- en *Bakgrund* fyllnadssymbol
- en *Border* linjesymbol
- marginalerna från förhöjningsprofilens artikelkant

18.2.10 HTML-ramobjektet

Det är möjligt att lägga till en ram som visar innehållet på en webbplats eller till och med skapa och utforma din egen HTML-sida och visa den! Du kan lägga till en bild med *Add HTML* enligt *instruktioner för skapande av objekt* och manipulera den på samma sätt som i *Interagera med layoutobjekt*. Observera att HTML-skalan styrs av layoutexportens upplösning vid den tidpunkt då HTML-ramen skapas.

HTML-objektet kan anpassas med hjälp av dess *Objektegenskaper*-panel. Förutom *objektens gemensamma egenskaper*, har denna funktion följande funktioner (se Fig. 18.56):

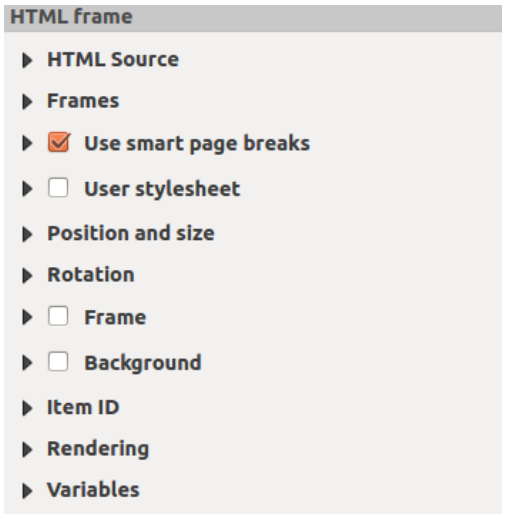


Fig. 18.56: HTML-ram, panelen för objektsegenskaper

HTML-källa

Gruppen *HTML Source* i HTML-ram *Objektegenskaper*-panelen ger följande funktioner (se Fig. 18.57):

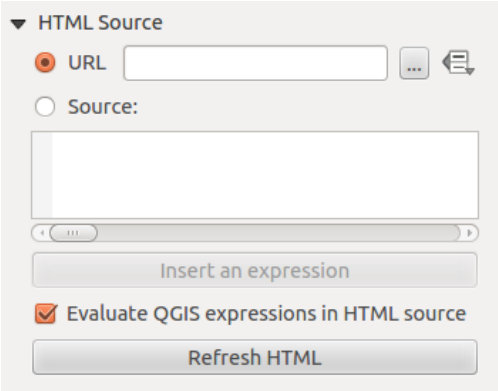


Fig. 18.57: HTML-ram, egenskaperna för HTML-källa

- I *URL* kan du ange URL:en till en webbsida som du kopierat från din webbläsare eller välja en HTML-fil med hjälp av … Bläddra-knappen. Det finns också möjlighet att använda knappen Data-defined override för att ange en URL från innehållet i ett attributfält i en tabell eller med hjälp av ett reguljärt uttryck.
- I *Source* kan du skriva in text i textrutan med några HTML-taggar eller tillhandahålla en fullständig HTML-sida.
- Knappen *Insert or Edit an Expression…* kan användas för att lägga till ett uttryck som `[%Year($now)%]` i textrutan Source för att visa aktuellt år. Den här knappen aktiveras endast när radioknappen *Source* är markerad. När du har infogat uttrycket måste du klicka någonstans i textrutan innan du uppdaterar HTML-rutan, annars förlorar du uttrycket.
- Aktivera *Evaluera QGIS-uttryck i HTML-kod* för att se resultatet av det uttryck du har inkluderat, annars ser du uttrycket istället.
- Använd knappen *Refresh HTML* för att uppdatera HTML-rutan/rutorna och se resultatet av ändringarna.

Ramar

Gruppen *Frames* i panelen HTML frame *Item Properties* ger följande funktioner (se Fig. 18.58):

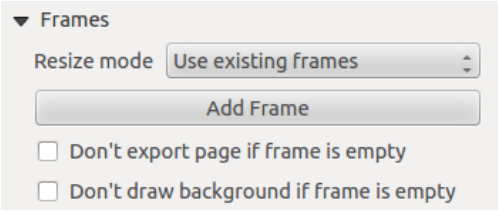


Fig. 18.58: HTML-ram, egenskaperna för ramar

- Med *Resize mode* kan du välja hur HTML-innehållet ska återges:
  - `Use existing frames` visar resultatet endast i den första ramen och i tillagda ramar.
  - `Extend to next page` skapar så många ramar (och motsvarande sidor) som behövs för att återge webbsidans höjd. Varje ram kan flyttas runt på layouten. Om du ändrar storlek på en ram kommer webbsidan att delas upp mellan de andra ramarna. Den sista ramen trimmas så att den passar webbsidan.
  - `Repeat on every page` kommer att upprepa den övre vänstra delen av webbsidan på varje sida i ramar av samma storlek.

  - `Upprepa tills det är klart` skapar också lika många ramar som alternativet `Förläng till nästa sida`, förutom att alla ramar har samma storlek.

- Använd knappen *Add Frame* för att lägga till en ny ram med samma storlek som den valda ramen. Om HTML-sidan inte ryms i den första ramen kommer den att fortsätta i nästa ram när du använder *Resize mode* eller *Use existing frames*.

- Aktivera *Exportera inte sidan om ramen är tom* förhindrar att sidan exporteras när ramen inte har något HTML-innehåll. Detta innebär att alla andra layoutobjekt, kartor, skalstreck, legender etc. inte kommer att synas i resultatet.

- Activate *Don't draw background if frame is empty* förhindrar att HTML-ramen ritas om ramen är tom.

Använd smarta sidbrytningar och User Style Sheet

Dialogrutorna *Use smart page breaks* och *User style sheet* i panelen HTML frame *Item Properties* har följande funktioner (se Fig. 18.59):

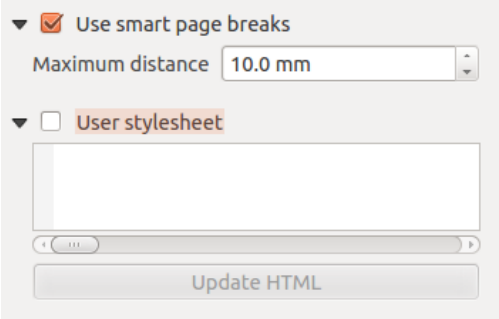


Fig. 18.59: HTML-ram, användning av smarta sidbrytningar och egenskaper för användarstilmallar

- Aktivera *Använd smarta sidbrytningar* för att förhindra att html-rutans innehåll bryts mitt i en textrad så att den fortsätter snyggt och smidigt i nästa ruta.

- Ställ in det *Maximala avstånd* som tillåts vid beräkning av var sidbrytningar ska placeras i html. Detta avstånd är den maximala mängd tomt utrymme som tillåts längst ned i en ram efter beräkning av den optimala brytplatsen. Ett större värde ger bättre val av plats för sidbrytning, men mer bortkastat utrymme längst ned i ramarna. Detta används endast när *Use smart page breaks* är aktiverat.

- Aktivera *User style sheet* för att tillämpa HTML-stilar som ofta tillhandahålls i kaskadstilmallar. Ett exempel på stilkod ges nedan för att ställa in färgen på `<h1>` header tag till grön och ställa in teckensnitt och teckenstorlek för text som ingår i stycketaggar `<p>`.

```
h1 {color: #00ff00;
}
p {font-family: "Times New Roman", Times, serif;
   font-size: 20px;
}
```

- Använd knappen *Update HTML* för att se resultatet av inställningarna för formatmallen.

18.2.11 De formade föremålen

QGIS tillhandahåller ett par verktyg för att rita regelbundna eller mer komplexa former över utskriftslayouten.

> **Observera**
>
> Till skillnad från andra utskriftslayoutobjekt kan du inte utforma ramen eller bakgrundsfärgen för formens begränsningsram (inställd på transparent som standard).

Det vanliga formobjektet

Objektet *Form* är ett verktyg som hjälper dig att dekorera din karta med vanliga former som triangel, rektangel, ellips … Du kan lägga till en vanlig form med hjälp av verktyget Add Shape som ger tillgång till särskilda verktyg som Add Rectangle, Lägg till ellips och Lägg till triangel. När du har valt rätt verktyg kan du rita objektet enligt *instruktioner för skapande av objekt*. Precis som andra layoutobjekt kan en vanlig form manipuleras på samma sätt som beskrivs i *Interagera med layoutobjekt*.

> **Observera**
>
> Genom att hålla ned `Shift`-tangenten medan du ritar grundformen med klick- och dragmetoden kan du skapa en perfekt kvadrat, cirkel eller triangel.

Standardformobjektet kan anpassas med hjälp av dess *Item Properties*-panel. Förutom *objektens gemensamma egenskaper*, har denna funktion följande funktioner (se Fig. 18.60):

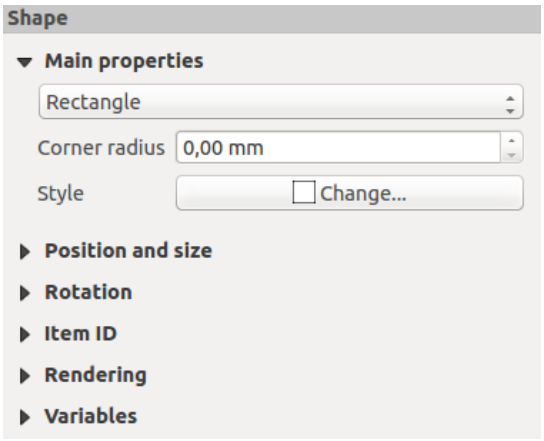


Fig. 18.60: Panel för egenskaper för formobjekt

Gruppen *Main properties* visar och låter dig ändra typen av formobjekt (**Ellips**, **Rectangle** eller **Triangle**) inom den givna ramen.

Du kan ställa in formens stil med hjälp av den avancerade *symbol* och *color* väljaren widget…

För rektangelformen kan du ange olika enheter för värdet på *Corner radius* för att runda hörnen.

De nodbaserade formobjekten

Medan verktyget *Add Shape* ger möjlighet att skapa enkla och fördefinierade geometriska objekt, hjälper verktyget *Add Node Item* dig att skapa ett anpassat och mer avancerat geometriskt objekt. För polylinjer eller polygoner kan du rita så många linjer eller sidor som du vill och objektens hörn kan manipuleras oberoende och direkt med hjälp av verktyget *Edit Nodes Item*. Själva objektet kan manipuleras på det sätt som beskrivs i *Interagera med layoutobjekt*.

För att lägga till en nodbaserad form:

1. Klicka på ikonen Add Node Item (lägg till nodobjekt)
2. Välj antingen verktyget Lägg till polygon eller Lägg till polylinje
3. Utför på varandra följande vänsterklick för att lägga till noder i ditt objekt. Om du håller ned `Shift`-tangenten medan du ritar ett segment, begränsas det till att följa en orienteringsmultipel av 45°.
4. När du är klar högerklickar du för att avsluta formen.

Du kan anpassa utseendet på formen i panelen *Item Properties*.

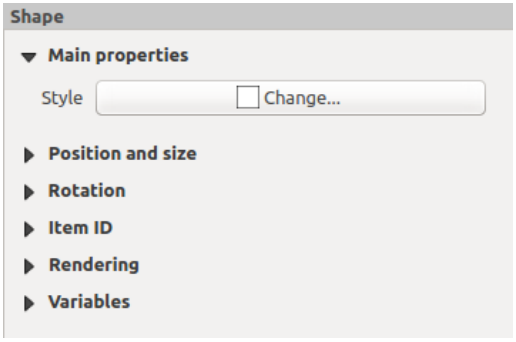


Fig. 18.61: Polygon Node Shape Item Egenskaper Panel

I *Main properties* kan du ställa in formens stil med hjälp av den avancerade *symbol* och *color* väljaren widget…

För polylinjenodens objekt kan du också parametrisera *Line markers*, dvs. lägga till:

- start- och/eller slutmarkörer med alternativ:
  - *None*: ritar en enkel polylinje.
  - *Arrow*: lägger till en vanlig triangulär pilspets som du kan anpassa.
  - *SVG*-markör: använder en `SVG`-fil som pilhuvud för objektet.
- anpassa pilspetsen:
  - *Pilens streckfärg*: anger streckfärgen för pilspetsen.
  - *Pilens fyllningsfärg*: anger fyllningsfärgen för pilhuvudet.
  - *Pilens slagbredd*: anger slagbredden för pilens huvud.
  - *Pilspetsens bredd*: anger pilspetsens storlek.

SVG-bilder roteras automatiskt med linjen. Färgerna för streck och fyllning i QGIS fördefinierade SVG-bilder kan ändras med hjälp av motsvarande alternativ. Anpassad SVG kan kräva några taggar som följer denna *instruktion*.

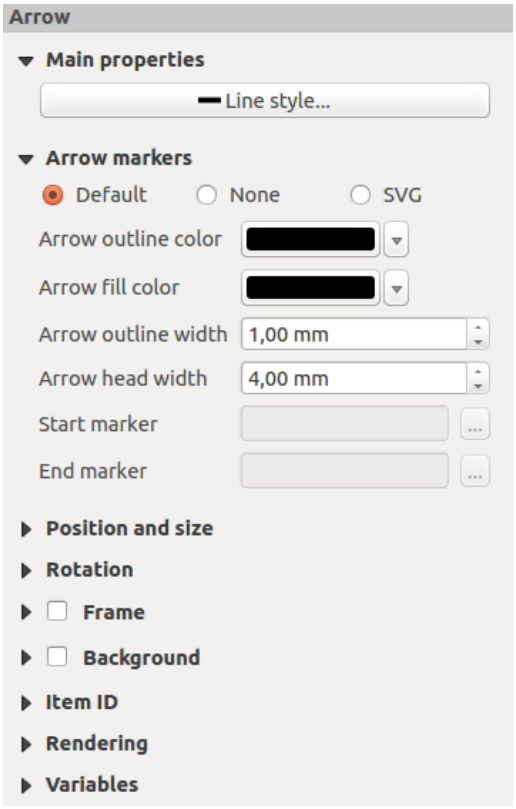


Fig. 18.62: Polyline Node Shape Item Egenskaper Panel

Föremålet Arrow

Verktyget Lägg till pil är en genväg för att skapa en pilaktiverad polylinje som standard och har därmed samma egenskaper och beteende som ett *polylinjenodobjekt*.

I själva verket kan pilobjektet användas för att lägga till en enkel pil, t.ex. för att visa relationen mellan två olika utskriftslayoutobjekt. För att skapa en norrpil bör dock *bildobjektet* övervägas först eftersom det ger tillgång till en uppsättning norrpilar i `.SVG`-format som du kan synkronisera med ett kartobjekt så att det roterar automatiskt med det.

Redigera geometrin för ett nodobjekt

Ett särskilt verktyg finns för att redigera nodbaserade former genom Edit Nodes Item. I det här läget kan du välja en nod genom att klicka på den (en markör visas på den valda noden). En vald nod kan flyttas antingen genom att dra den eller genom att använda piltangenterna. I det här läget kan du dessutom lägga till noder i en befintlig form: dubbelklicka på ett segment så läggs en nod till på den plats där du klickar. Slutligen kan du ta bort den markerade noden genom att trycka på `Del`-tangenten.

18.3 Skapa en utdata

Fig. 18.63 visar ett exempel på en utskriftslayout som innehåller alla typer av layoutobjekt som beskrivs i föregående avsnitt.

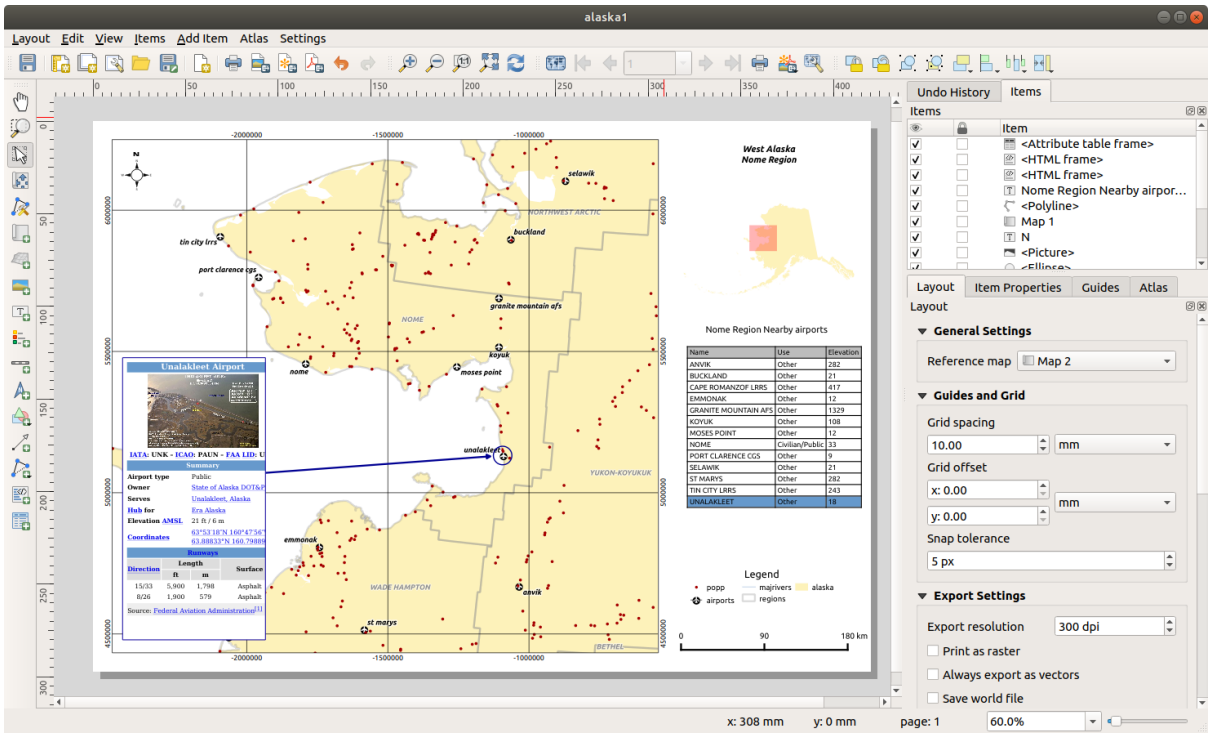


Fig. 18.63: Utskriftslayout med kartvy, teckenförklaring, bild, skalstreck, koordinater, text och HTML-ram tillagd

Från menyn eller verktygsfältet *Layout* kan du mata ut utskriftslayouten till olika filformat, och det är möjligt att ändra upplösning (utskriftskvalitet) och pappersstorlek:

- Med ikonen Print kan du skriva ut layouten till en ansluten skrivare eller en PostScript-fil, beroende på vilka skrivardrivrutiner som är installerade.
- Ikonen Export as image exporterar utskriftslayoutens bildformat som `PNG`, `BMP`, `TIF`, `JPG` och många andra…
- Ikonen Export as SVG sparar utskriftslayouten som en `SVG` (Scalable Vector Graphic).
- Ikonen Export as PDF sparar den definierade utskriftslayouten direkt som en `PDF`-fil (Portable Document Format).

18.3.1 Inställningar för export

När du exporterar en utskriftslayout finns det ett urval av exportinställningar som QGIS måste kontrollera för att kunna producera den mest lämpliga utskriften. Dessa konfigurationer är:

- *Exportinställningar* i panelen *Layout*, till exempel *Exportupplösning*, *Skriv ut som raster Exportera alltid som vektorer* eller *Spara världsfil*
- *Uteslut sidan från export* i *page item properties*-panelen
- *Uteslut objekt från export* i panelen *objektets egenskaper*

Dessutom tillämpas ett antal fördefinierade kontroller automatiskt på layouten. För närvarande omfattar dessa kontroller testning av att skalfält är korrekt länkade till kartobjekt och att kartöversiktsobjekt också är korrekt länkade till en karta. Om kontrollerna misslyckas visas en fin varning som informerar dig om problemet.

18.3.2 Exportera som bild

Så här exporterar du en layout som en bild:

1. Klicka på ikonen Exportera som bild
2. Välj det bildformat, den mapp och det filnamn (t.ex. `myill.png`) som ska användas. Om layouten innehåller mer än en sida exporteras varje sida till en fil med det angivna filnamnet med sidnumret som tillägg (t.ex. `myill_2.png`).
3. I nästa (*Bildexportalternativ*) dialogruta:
   - Du kan åsidosätta utskriftslayouten *Export resolution* och de exporterade sidornas mått (som anges i panelen *Layout*).
   - Bildrenderingen kan också förbättras med alternativet *Enable antialiasing*.
   - Om du vill exportera din layout som en **georefererad bild** (t.ex. för att dela med andra projekt), markera alternativet *Generate world file* och en *ESRI World File* med samma namn som den exporterade bilden, men med ett annat tillägg (`.tfw` för TIFF, `.pnw` för PNG, `jgw` för JPEG, …) kommer att skapas vid export. Det här alternativet kan också markeras som standard i *layoutpanel*.

     Observera

     För utdata på flera sidor kommer endast den sida som innehåller *referenskarta* att få en världsfil (förutsatt att alternativet *Generera världsfil* är markerat).

   - Genom att markera alternativet *Crop to content* kommer den bild som matas ut av layouten att innehålla det minimala område som omsluter alla objekt (karta, teckenförklaring, skalstreck, former, etikett, bild…) på varje sida i kompositionen:
     – Om kompositionen innehåller en enda sida ändras storleken på utdata så att ALLT på kompositionen inkluderas. Sidan kan sedan förminskas eller utökas till alla objekt beroende på deras position (på, ovanför, under, till vänster eller höger om sidan).
     – Om layouten består av flera sidor kommer varje sida att storleksändras så att den omfattar objekt inom sitt område (vänster och höger sida för alla sidor, plus toppen för den första sidan och botten för den sista sidan). Varje storleksändrad sida exporteras till en separat fil.

     I dialogrutan *Crop to content* kan du också lägga till marginaler runt de beskurna gränserna.
   - Genom att markera *Öppna fil efter export* kommer den exporterade filen automatiskt att öppnas i standardbildvisaren.

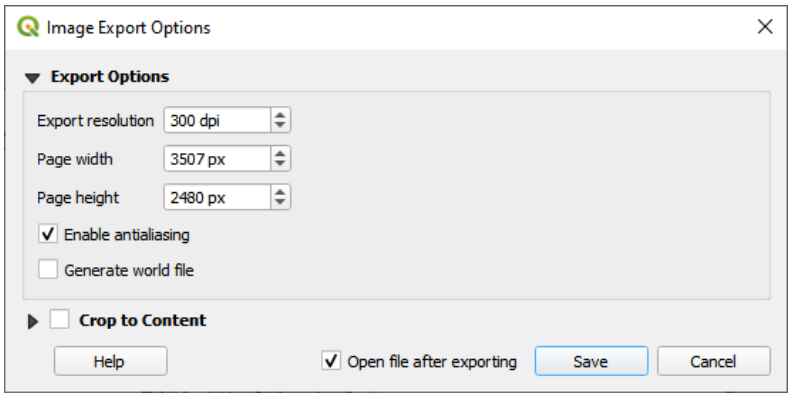


Fig. 18.64: Alternativ för bildexport

Tips

Använd bildformat som stöder transparens när objekten sträcker sig utanför pappersytan

Layoutobjekt kan placeras utanför pappersområdet. När du exporterar med alternativet *Crop to content* kan den resulterande bilden därför sträcka sig utanför pappersutbredningen. Eftersom bakgrunden utanför pappersområdet kommer att vara transparent, kommer den transparenta bakgrunden att återges som helsvart i bildformat som inte stöder transparens (t.ex. `BMP` och `JPG`), vilket ”förstör” bilden. Använd transparens-kompatibla format (t.ex. `TIFF` och `PNG`) i sådana fall.

Observera

När det stöds av formatet (t.ex. `PNG`) och det underliggande Qt-biblioteket kan den exporterade bilden innehålla *projektmetadata* (författare, titel, datum, beskrivning…)

18.3.3 Exportera som SVG

Så här exporterar du en layout som SVG:

1. Klicka på ikonen Exportera som SVG
2. Fyll i sökvägen och filnamnet (används som ett basnamn för alla filer vid flersidig komposition, som vid bildexport)
3. I nästa dialogruta *SVG Export Options* kan du åsidosätta layoutens standardinställningar *export settings* eller konfigurera nya:
   - ☐ *Exportera kartlager som SVG-grupper*: exporterade objekt grupperas inom lager vars namn matchar lagernamnen från QGIS, vilket gör det mycket lättare att förstå innehållet i dokumentet.
   - ☐ *Alltid exportera som vektorer*: vissa renderingsalternativ kräver att objekten rastreras för att få en bättre rendering. Markera det här alternativet om du vill behålla objekten som vektorer, med risk för att utseendet på utdatafilen kanske inte stämmer överens med förhandsgranskningen av utskriftslayouten (för mer information, se *Inställningar för export*).
   - ☑ *Exportera RDF-metadata* för dokumentet, t.ex. titel, författare, datum, beskrivning …
   - ☑ *Förenkla geometrier för att minska storleken på utdatafilen*: detta undviker att exportera ALLA geometriska hörn, vilket kan resultera i en löjligt komplex och stor exportfil som inte kan laddas i andra applikationer. Geometrier kommer att förenklas när layouten exporteras för att ta bort alla överflödiga hörn som inte är märkbart olika vid exportupplösningen (t.ex. om exportupplösningen är `300 dpi`, kommer hörn som är mindre än `1/600 tum` från varandra att tas bort).
   - Ange *Text export*: styr om textetiketter alltid eller helst ska exporteras som *text- eller konturobjekt*.
   - Tillämpa ☑ *Beskär till innehåll alternativ*
   - ☐ *Avaktivera export av kaklade rasterlager*: Vid export av filer använder QGIS en inbyggd rasterlagerstiliserad rendering som sparar minne. Ibland kan detta orsaka synliga ”sömmar” i rastrerna för genererade filer. Om du markerar det här alternativet åtgärdas detta, till priset av en högre minnesanvändning under exporten.
   - Genom att markera ☑ *Öppna fil efter export* kommer den exporterade filen automatiskt att öppnas i standard SVG-visningsprogrammet.

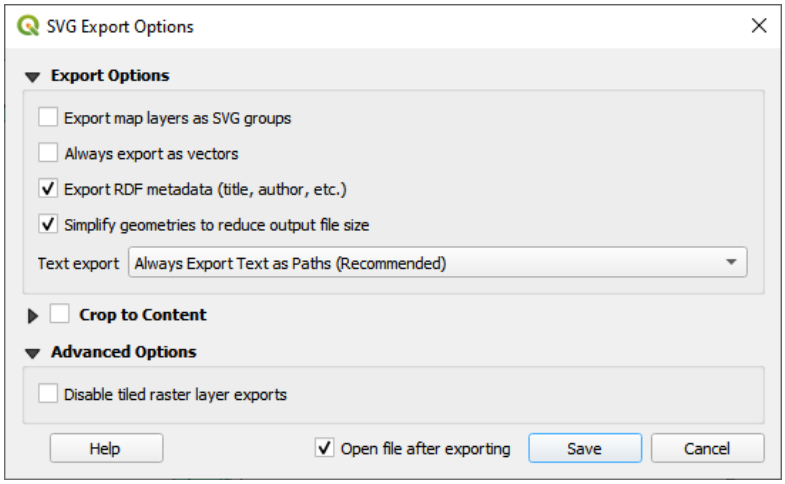


Fig. 18.65: Alternativ för SVG-export

Observera

För närvarande är SVG-utdata mycket grundläggande. Detta är inte ett QGIS-problem, utan ett problem med det underliggande Qt-biblioteket. Förhoppningsvis kommer detta att lösas i framtida versioner.

18.3.4 Exportera som PDF

Så här exporterar du en layout som PDF:

1. Klicka på ikonen :sup:`Exportera som PDF`
2. Fyll i sökväg och filnamn: till skillnad från bild- och SVG-export exporteras alla sidor i layouten till en enda PDF-fil.
3. I nästa dialogruta *PDF Export Options* kan du åsidosätta layoutens standardinställningar *export settings* eller konfigurera nya:
   - *Alltid exportera som vektorer*: vissa renderingsalternativ kräver att objekten rastreras för att få en bättre rendering. Markera det här alternativet om du vill behålla objekten som vektorer, med risk för att utseendet på utdatafilen kanske inte stämmer överens med förhandsgranskningen av utskriftslayouten (för mer information, se *Inställningar för export*).

     Observera

     Tänk på opacitetsinställningar för vektorlager

     När du ställer in en lageromfattande opacitet (eller opacitet för en vectorsymbolik) till mindre än 100%, tvingar det fram en rastrerad rendering av dessa objekt under PDF-exporten. Beroende på lagrets storlek kan detta öka PDF-filens storlek avsevärt.

   - *Lägg till georeferensinformation*: endast tillgänglig om *referenskarta*, från vilken informationen hämtas, finns på första sidan.
   - *Exportera RDF-metadata* för dokumentet, t.ex. titel, författare, datum, beskrivning …
   - Ange *Text export*: styr om textetiketter alltid eller helst ska exporteras som *text- eller konturobjekt*.
   - Styr PDF *Bildkomprimering* med hjälp av:

  - *Lossy (JPEG)*, vilket är standardkomprimeringsläget
  - eller *Lossless*, vilket skapar större filer i de flesta fall, men är mycket mer lämpligt för utskrift eller för efterbehandling i externa program (kräver Qt 5.13 eller senare).

- *Skapa geospatial PDF*: Genererar en georefererad PDF-fil.
- *Avaktivera export av rasterlager i mosaikformat*: När filer exporteras använder QGIS en tiledbaserad rendering som sparar minne. Ibland kan detta orsaka synliga ”sömmar” i rastrerna för genererade filer. Om du markerar det här alternativet åtgärdas detta, till priset av en högre minnesanvändning under exporten.
- *Förenkla geometrier för att minska storleken på utdatafilen*: Geometrier förenklas när layouten exporteras genom att toppar som inte skiljer sig märkbart åt vid exportupplösningen tas bort (t.ex. om exportupplösningen är `300 dpi` kommer toppar som är mindre än `1/600 tum` från varandra att tas bort). Detta kan minska exportfilens storlek och komplexitet (mycket stora filer kan inte laddas i andra program).
- Genom att markera *Öppna fil efter export* kommer den exporterade filen automatiskt att öppnas i standard PDF-visningsprogrammet.

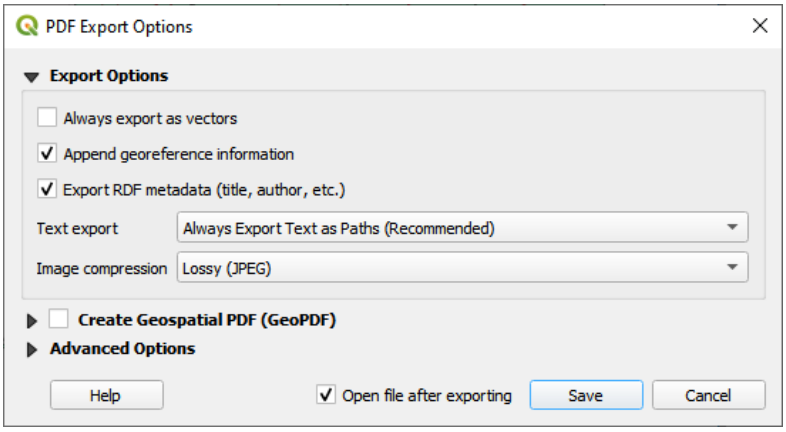


Fig. 18.66: Alternativ för PDF-export

Observera

Geospatial PDF-export stöds och ett antal specifika alternativ för Geospatial PDF finns tillgängliga:

- *Format* (Geospatialt PDF-format - det finns vissa variationer),
- *Inkludera flera kartteman* (ange vilka kartteman som ska inkluderas),
- *Include vector feature information* (välj lager och gruppera dem i logiska PDF-grupper).

Observera

Om du exporterar en utskriftslayout till format som stöder georeferenser (t.ex. `PDF` och `TIFF`) skapas en georefererad utskrift som standard.

18.3.5 Skapa en Atlas

Med Atlas funktioner kan du skapa kartböcker på ett automatiserat sätt. Atlas använder funktionerna i en tabell eller ett vektorlager (*Coverage layer*) för att skapa en utdata för varje funktion (**atlasfunktion**) i tabellen / lagret. Den vanligaste användningen är att zooma ett kartobjekt till den aktuella atlasfunktionen. Ytterligare användningsfall inkluderar:

- ett kartobjekt som för ett annat lager endast visar objekt som har samma attribut som atlasobjektet eller som ligger inom dess geometri.
- en etikett eller ett HTML-objekt vars text ersätts när funktioner itereras över
- ett tabellobjekt som visar attribut för associerade *föräldrar eller barn*-objekt för det aktuella atlasobjektet…

För varje funktion bearbetas utdata för alla sidor och artiklar i enlighet med deras exportinställningar.

Tips

Använd variabler för mer flexibilitet

QGIS tillhandahåller en stor panel av funktioner och *variabler*, inklusive atlasrelaterade sådana, som du kan använda för att manipulera layoutobjekten, men också lagrens symbologi, enligt atlasstatus. Kombinationen av dessa funktioner ger dig mycket flexibilitet och hjälper dig att enkelt producera avancerade kartor.

För att aktivera generering av en atlas och komma åt atlasparametrar, se panelen *Atlas*. Denna panel innehåller följande (se Fig. 18.67):

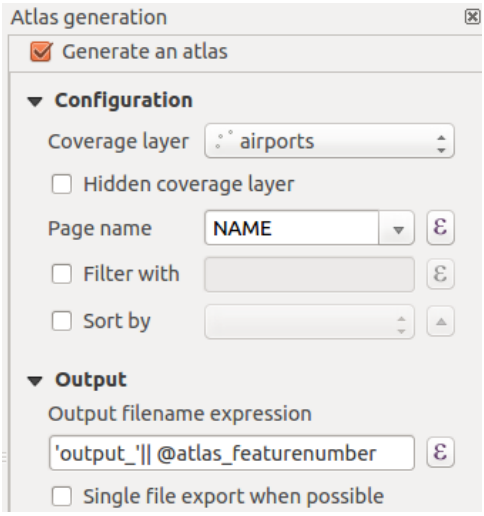


Fig. 18.67: Atlaspanelen

- *Generera en atlas* aktiverar eller inaktiverar generering av atlas.
- *Konfiguration*
  - En kombinationsruta *Coverage layer* där du kan välja det tabell- eller vektorlager som innehåller de funktioner som ska itereras över.
  - En valfri *Hidden coverage layer* som, om den är markerad, kommer att dölja täckningslagret (men inte de andra lagren) under genereringen.
  - En valfri *Sidnamn*-kombinationsruta för att ange namnet på funktionssidan/funktionssidorna. Du kan välja ett fält i täckningslagret eller ange en *expression*. Om detta alternativ är tomt kommer QGIS att använda ett internt ID, enligt det filter och/eller den sorteringsordning som tillämpas på lagret.

  - En valfri textruta (*Filter with*) där du kan ange ett uttryck för filtrering av funktioner från täckningslagret. Om uttrycket inte är tomt kommer endast funktioner som utvärderas till `True` att bearbetas.
  - En valfri kryssruta *Sort by* som gör att du kan sortera funktioner i täckningslagret (och utdata) med hjälp av ett fält i täckningslagret eller ett uttryck. Sorteringsordningen (antingen stigande eller fallande) anges av den tvådelade knappen *Sort direction* som visar en uppåt- eller nedåtpil.

- *Output* - här kan atlasens utdata konfigureras:
  - En textruta för *Output filename expression* som används för att generera ett filnamn för varje atlasfunktion. Den är baserad på uttryck. är endast meningsfull för rendering till flera filer.
  - En kryssruta *Export av en enda fil när det är möjligt* som gör att du kan tvinga fram generering av en enda fil om det är möjligt med det valda utdataformatet (t.ex. `PDF`). Om det här fältet är markerat är värdet i fältet *Output filename expression* meningslöst.
  - En rullgardinslista med *Bildexportformat* för att välja utdataformat när du använder knappen Exportera atlas som bilder….

Kontroll karta efter atlas

Den vanligaste användningen av atlas är med kartobjektet, som zoomar till den aktuella atlasfunktionen när iterationen går över täckningslagret. Detta beteende ställs in i *Controlled by atlas*-gruppegenskaperna för kartobjektet. Se *Kontrolleras av atlas* för olika inställningar som du kan tillämpa på kartobjektet.

Anpassa etiketter med uttryck

För att anpassa etiketter till den funktion som atlasen itererar över kan du inkludera uttryck. Se till att du placerar uttrycksdelen (inklusive funktioner, fält eller variabler) mellan `[%` och `%]` (se *Etikettobjektet* för mer information).

För ett stadslager med fälten `CITY_NAME` och `ZIPCODE` kan du till exempel infoga detta:

```
The area of [% concat( upper(CITY_NAME), ',', ZIPCODE, ' is ',
format_number($area/1000000, 2) ) %] km2
```

eller en annan kombination:

```
The area of [% upper(CITY_NAME)%],[%ZIPCODE%] is
[%format_number($area/1000000,2) %] km2
```

Informationen `[% concat( upper(CITY_NAME), ',', ZIPCODE, ' is ', format_number($area/1000000, 2) ) %]` är ett uttryck som används inuti etiketten. Båda uttrycken skulle resultera i följande typ av etikett i den genererade atlasen:

```
The area of PARIS,75001 is 1.94 km2
```

Utforska datadefinierade åsidosättningsknappar med atlas

Det finns flera ställen där du kan använda en Data defined override-knapp för att åsidosätta den valda inställningen. Detta är särskilt användbart vid atlasgenerering. Se *Inställning av datadefinierad åsidosättning* för mer information om den här widgeten.

I följande exempel används lagret `Regions` i QGIS exempeldataset och väljs som *Coverage layer* för atlasgenerering. Vi antar att det är en layout med en enda sida som innehåller ett kartobjekt och ett etikettobjekt.

När höjden (nord-sydlig) på en region är större än bredden (öst-västlig) bör du använda *Portrait* istället för *Landscape* för att optimera pappersanvändningen. Med knappen Data Defined Override kan du dynamiskt ställa in pappersorienteringen.

Högerklicka på sidan och välj *Page Properties* för att öppna panelen. Vi vill ställa in orienteringen dynamiskt med hjälp av ett uttryck som beror på regionens geometri, så tryck på knappen i fältet *Orientation*, välj *Edit…* för att öppna dialogrutan *Expression string builder* och ange följande uttryck:

```
CASE WHEN bounds_width(@atlas_geometry) > bounds_height(@atlas_geometry)
THEN 'Landscape' ELSE 'Portrait' END
```

Om du nu *förhandsvisar atlasen* orienterar sig papperet automatiskt, men objektens placering kanske inte är idealisk.

För varje region måste du också omplacera layoutobjekten. För kartobjektet kan du använda knappen för dess egenskap *Width* för att ställa in den dynamiskt med hjälp av följande uttryck:

```
@layout_pagewidth - 20
```

På samma sätt kan du använda knappen för egenskapen *Height* för att ange följande uttryck för att begränsa kartobjektets storlek:

```
@layout_pageheight - 20
```

För att säkerställa att kartobjektet är centrerat på sidan ställer du in dess *Reference point* på den övre vänstra alternativknappen och anger `10` för dess *X*- och *Y*-positioner.

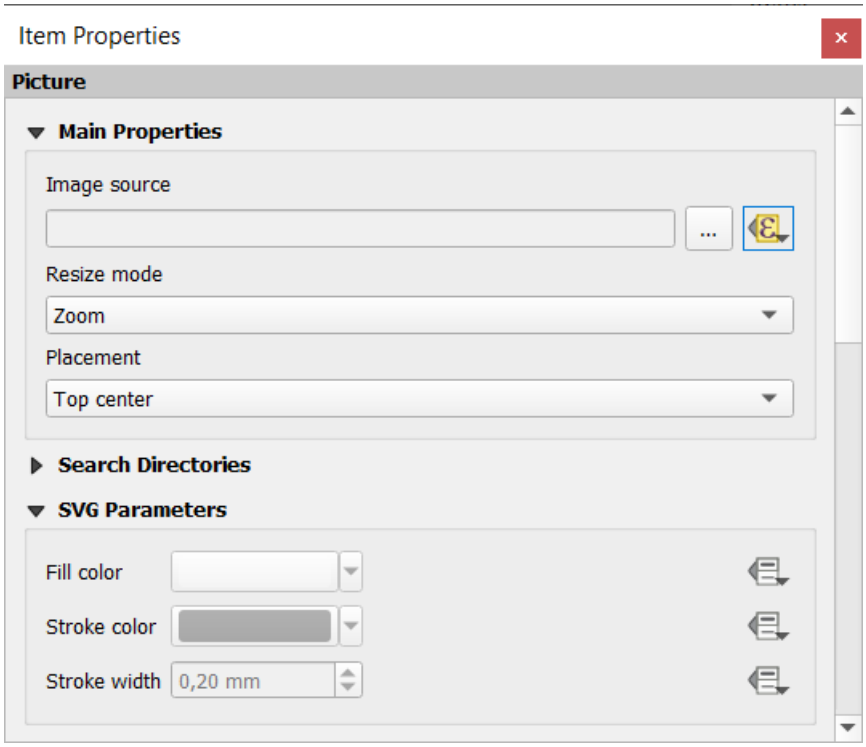
Låt oss lägga till en titel ovanför kartan i mitten av sidan. Välj etikettobjektet och ställ in den horisontella justeringen till *Center*. Flytta sedan etiketten till höger position, välj mittknappen för *Reference point* och ange följande uttryck för fältet *X*:

```
@layout_pagewidth / 2
```

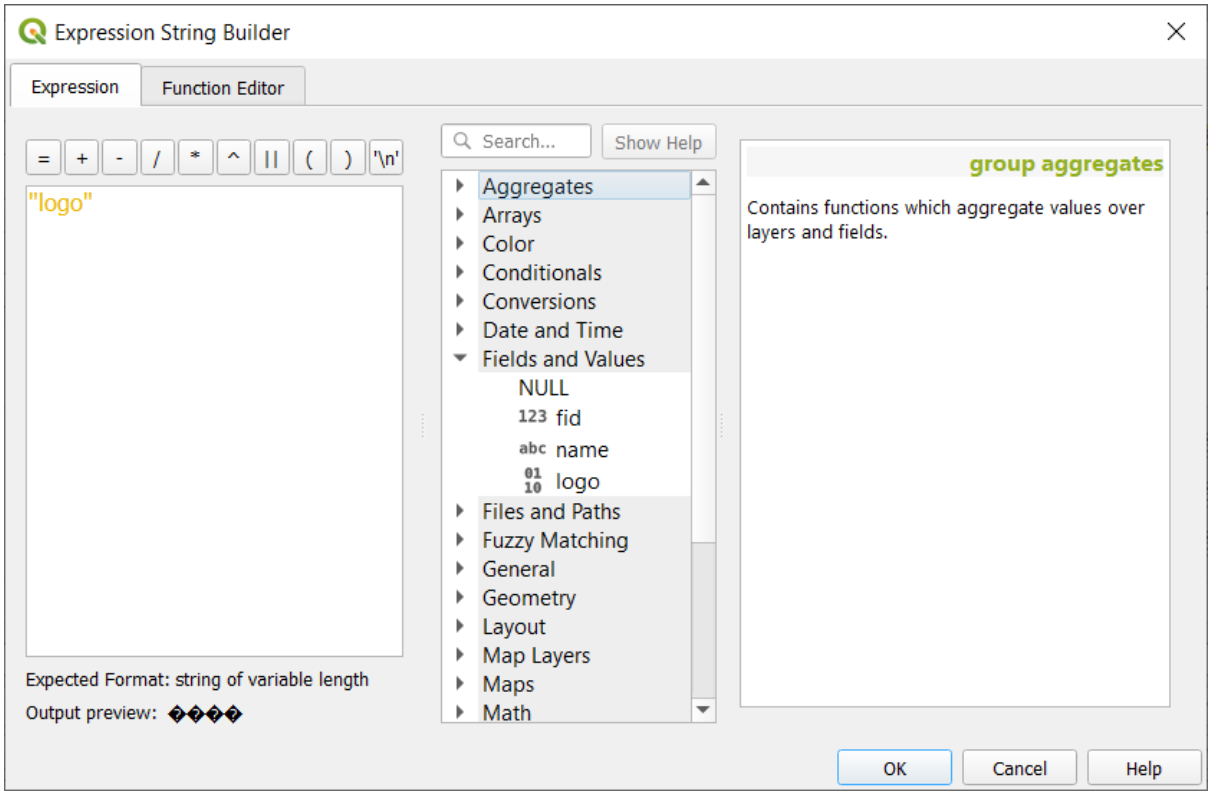
För alla andra layoutobjekt kan du ställa in positionen på ett liknande sätt så att de är korrekt placerade både för stående och liggande format. Du kan också göra fler justeringar, t.ex. anpassa titeln med funktionsattribut (se exemplet *Anpassa etiketter med uttryck*), ändra bilder, ändra antalet kolumner i teckenförklaringen beroende på sidans orientering, …

Den information som ges här är en uppdatering av den utmärkta bloggen (på engelska och portugisiska) om de datadefinierade åsidosättningsalternativen Multiple_format_map_series_using_QGIS_2.6 .

Ett annat exempel på användning av datadefinierade åsidosättningsknappar är användning av en dynamisk bild. I följande exempel använder vi ett geopackage-lager som innehåller ett BLOB-fält som heter `logo` med fälttypen binär (se *Skapa ett nytt GeoPackage-lager*). För varje funktion definieras en annan bild så att atlasen kan iterera över enligt beskrivningen i *Förhandsgranska och skapa en atlas*. Allt du behöver göra är att lägga till en bild i utskriftslayouten och gå till dess *Item properties* i atlaskontexten. Där kan du hitta en datadefinierad åsidosättningsknapp i avsnittet *Image source* i *Main Properties*.



I det följande fönstret väljer du *Edit* så att *Expression String Builder* öppnas. I avsnittet *Fields and values* hittar du det BLOB-fält som definierades i geopackage-lagret. Dubbelklicka på fältnamnet `logo` och klicka på *OK*.



Atlasen itererar över posterna i BLOB-fältet förutsatt att du väljer geopackage-lagret som *Coverage layer* (ytterligare instruktioner finns i *Förhandsgranska och skapa en atlas*).

Det här är bara två exempel på hur du kan använda avancerade inställningar med Atlas.

Förhandsgranska och skapa en atlas

Fig. 18.68: Atlas verktygsfält för förhandsgranskning

När atlasinställningarna har konfigurerats och layoutobjekt (karta, tabell, bild …) har länkats till den kan du skapa en förhandsgranskning av alla sidor genom att välja *Atlas* ► *Preview Atlas* eller klicka på ikonen Preview Atlas. Du kan sedan använda pilarna för att navigera genom alla funktioner:

- Första funktionen
- Föregående funktion
- Nästa funktion
- Sista funktion

Du kan också använda kombinationsrutan för att välja och förhandsgranska en specifik funktion. Kombinationsrutan visar namn på atlasfunktioner enligt det uttryck som anges i alternativet atlas *Sidnamn*.

När det gäller enkla kompositioner kan en atlas genereras på olika sätt (se *Skapa en utdata* för mer information - använd bara verktyg från *Atlas*-menyn eller verktygsfältet istället för *Layout*-menyn.

Detta innebär att du kan skriva ut dina kompositioner direkt med *Atlas* ► *Print Atlas*. Du kan också skapa en PDF-fil med hjälp av *Atlas* ► *Export Atlas as PDF…*: Du kommer att bli ombedd att ange en katalog för att spara alla genererade PDF-filer, utom om kryssrutan *Export av enstaka filer när det är möjligt* har valts. I så fall kommer du att uppmanas att ange ett filnamn.

Med verktyget *Atlas* ► *Exportera atlas som bilder…* eller *Atlas* ► *Exportera atlas som SVG…* blir du också ombedd att välja en mapp. Varje sida i varje atlasfunktionskomposition exporteras till det bildfilformat som anges i *Atlas*-panelen eller till SVG.

Observera

Med flersidig utmatning beter sig en atlas som en layout på så sätt att endast den sida som innehåller *Allmänna inställningar* får en world-fil (för varje funktion som matas ut).

Tips

Skriv ut en specifik atlasfunktion

Om du vill skriva ut eller exportera sammansättningen av endast en funktion i atlasen, startar du bara förhandsgranskningen, väljer önskad funktion i rullgardinsmenyn och klickar på *Layout* ► *Print* (eller *Export…* till valfritt filformat).

Använd projektdefinierade relationer för att skapa atlaser

För användare med HTML- och Javascript-kunskaper är det möjligt att arbeta med GeoJSON-objekt och använda projektdefinierade relationer från QGIS-projektet. Skillnaden mellan detta tillvägagångssätt och att använda uttryck som infogas direkt i HTML är att du får en fullständig, ostrukturerad GeoJSON-funktion att arbeta med. Detta innebär att du kan använda befintliga Javascript-bibliotek och funktioner som arbetar med GeoJSON-objektrepresentationer.

Följande kod inkluderar alla relaterade underordnade funktioner från den definierade relationen. Med hjälp av JavaScript-funktionen `setFeature` kan du skapa flexibel HTML som representerar relationer i vilket format du vill (listor, tabeller etc.). I kodexemplet skapar vi en dynamisk punktlista över de relaterade underordnade funktionerna.

```
// 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>
```

Under skapandet av atlasen sker en iteration över det täckningslager som innehåller de överordnade objekten. På varje sida visas en punktlista över de relaterade underordnade objekten efter den överordnades identifierare.

18.4 Skapa en rapport

I det här avsnittet får du hjälp med att skapa en rapport i QGIS.

18.4.1 Vad är det?

En GIS-rapport är per definition ett dokument som innehåller information som är organiserad på ett berättande sätt och som innehåller kartor, text, grafik, tabeller etc. En rapport kan upprättas ad hoc, periodiskt, återkommande, regelbundet eller efter behov. Rapporter kan avse specifika perioder, händelser, företeelser, ämnen eller platser.

I QGIS är en *Report* en förlängning av en *Layouts*.

Med hjälp av rapporter kan användarna lägga ut sina GIS-projekt på ett enkelt, snabbt och strukturerat sätt.

En rapport kan skapas med *Project* ► *New Report* eller i *Project* ► *Layout Manager*.

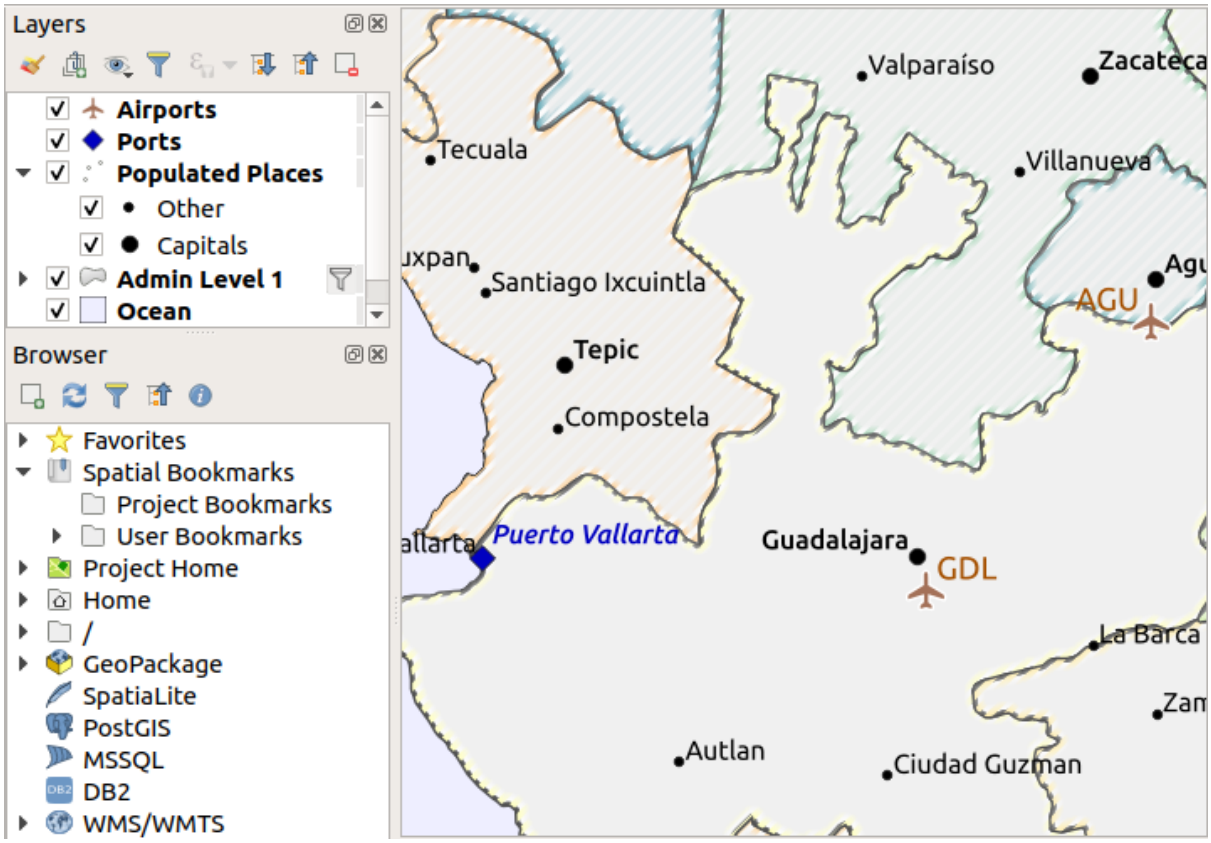
Observera

Kartorna i QGIS-rapporter beter sig på samma sätt som kartor i tryckta layouter och atlaser. Vi kommer att koncentrera oss på det som är specifikt för QGIS-rapporter. För detaljer om karthantering, se avsnitten om *print layouts* och *atlases*.

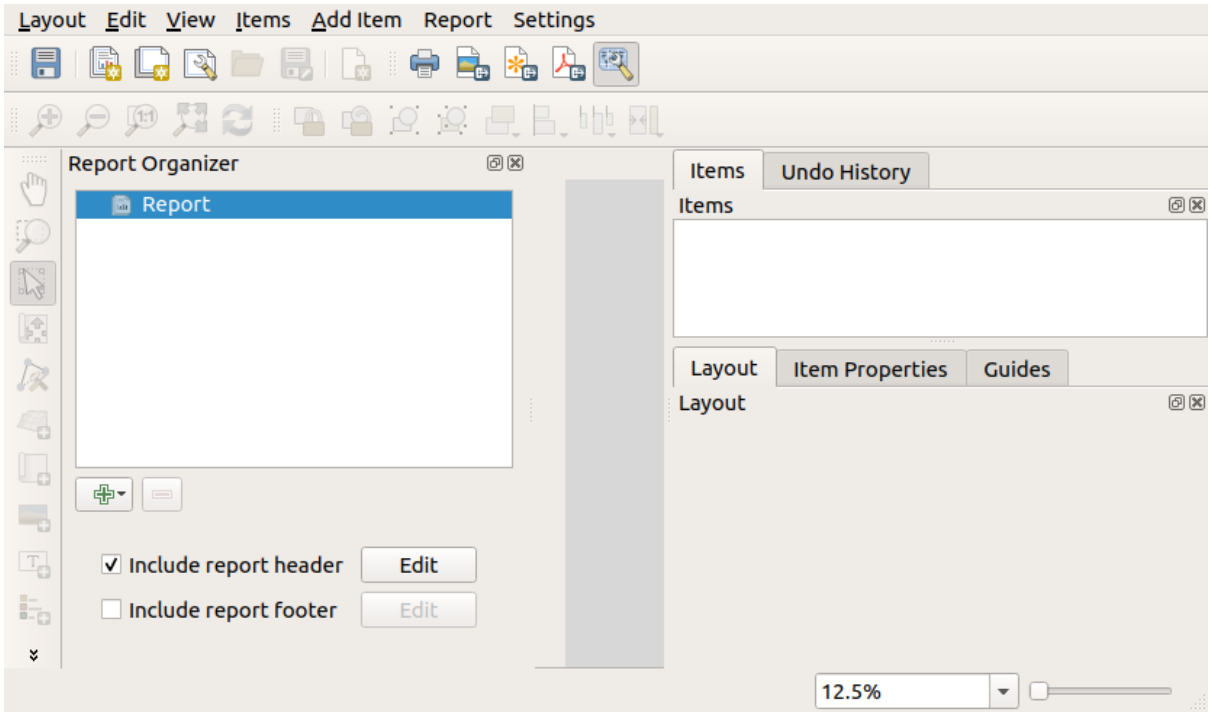
18.4.2 Kom igång

I dialogrutan *Layout Manager* kan en rapport skapas genom *New from template* genom att välja rullgardinsalternativet *Empty Report* och trycka på knappen *Create*….

I det här exemplet använder vi några administrativa gränser, befolkade platser, hamnar och flygplatser från datasetet Natural Earth (1:10M).

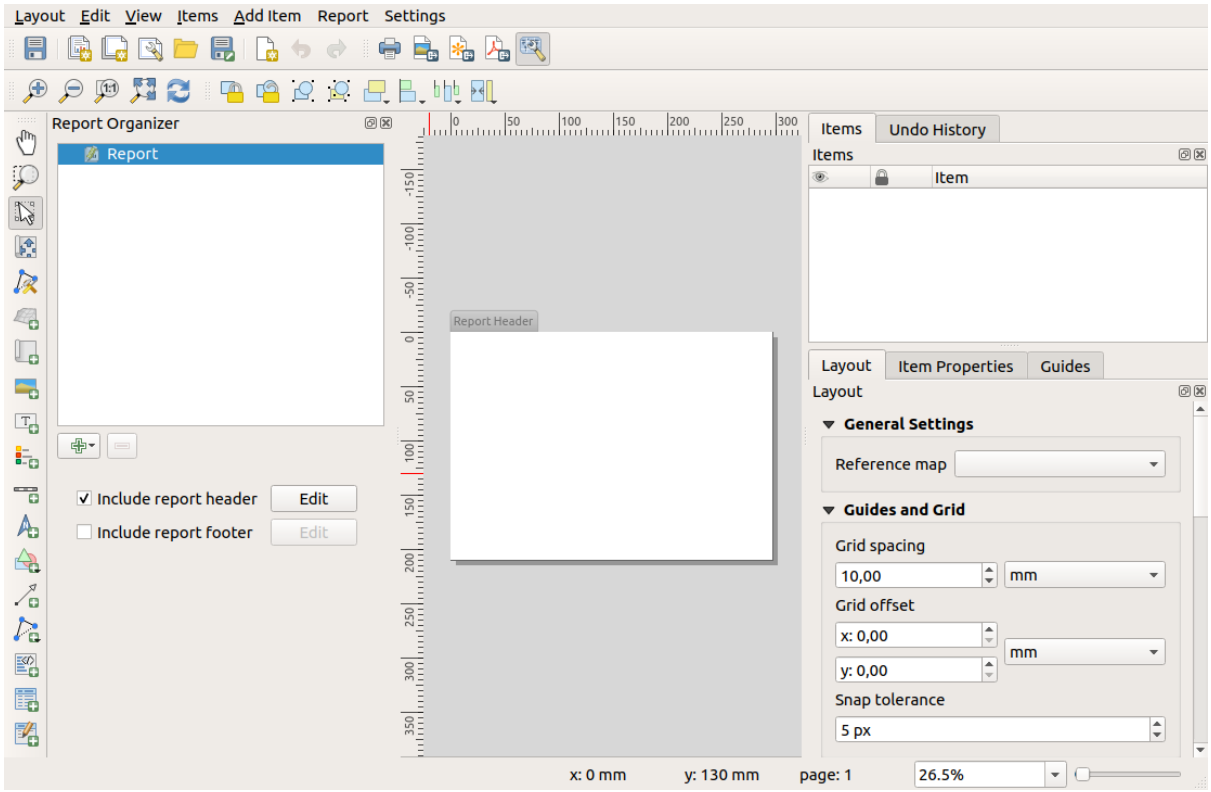


Med hjälp av kommandot *Project* ► *New Report* skapar vi en tom rapport. Till en början finns det inte mycket att titta på - dialogrutan som visas ser ungefär ut som utskriftslayoutdesignern, förutom *Report Organizer*-panelen till vänster:



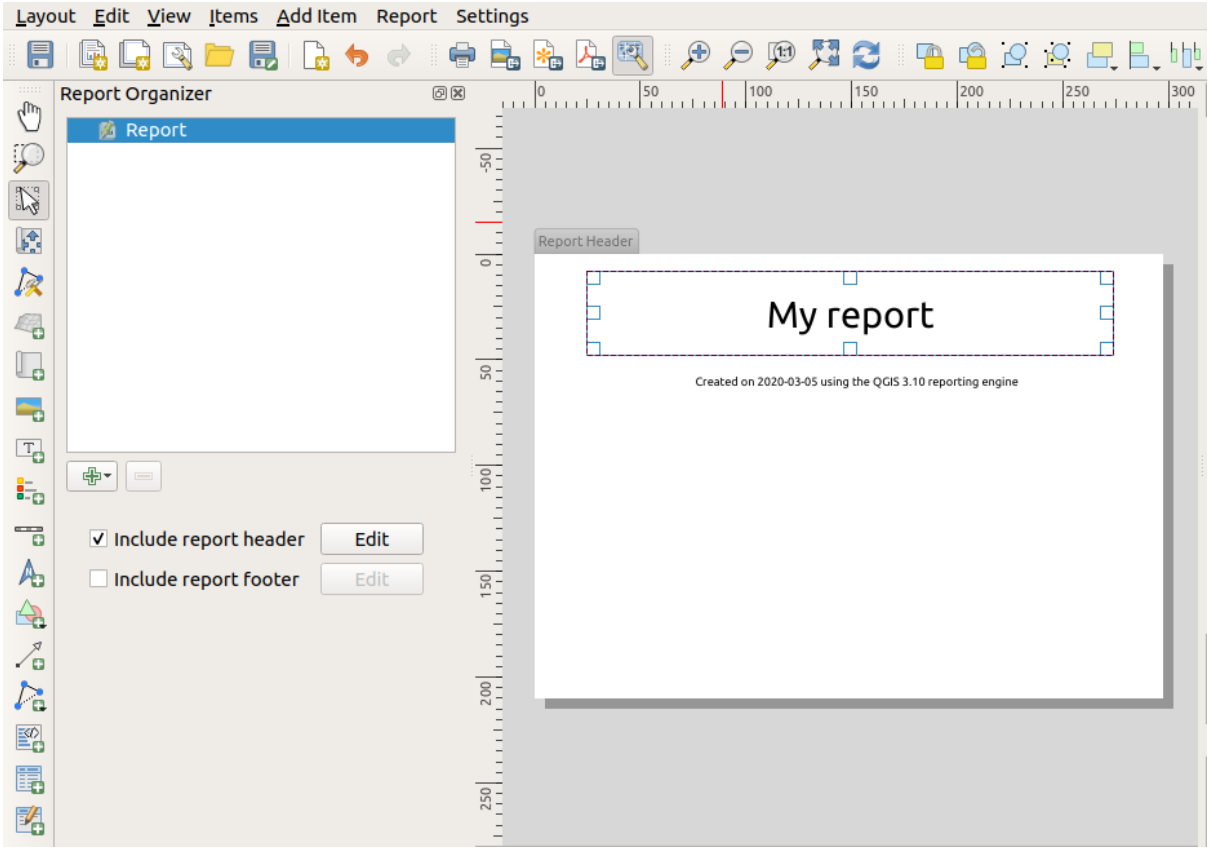
18.4.3 Layout Rapport Arbetsyta

QGIS-rapporter kan bestå av flera, nästlade avsnitt. I vår nya tomma rapport har vi inledningsvis bara huvudrapportavsnittet. De enda alternativen för detta rapportavsnitt är *Include report header* och *Include report footer*. Om vi aktiverar dessa alternativ kommer ett sidhuvud att inkluderas som första sida(n) (enskilda delar av rapporter kan vara flersidiga om så önskas) i rapporten, och en sidfot kommer att utgöra den sista sidan(n). Aktivera sidhuvudet (*Include report header*) och tryck på *Edit*-knappen bredvid det:

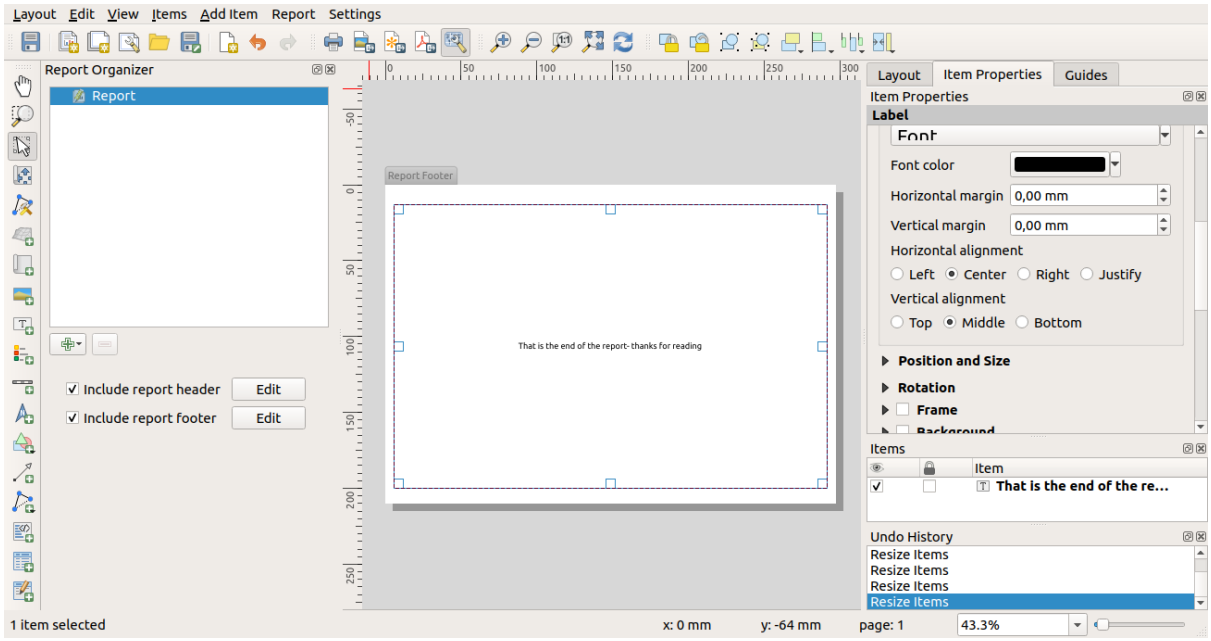


Några saker händer som ett resultat av detta. För det första visas en redigeringspenna bredvid *Report* i *Report Organizer*, vilket indikerar att rapportavsnittet för närvarande redigeras i designern. Vi ser också en ny sida med en liten *Report Header*-titel. Sidan har *landscape*-orientering som standard, men detta (och andra egenskaper hos sidan) kan ändras genom att högerklicka på sidan och välja *Page properties*. Då visas fliken *Item properties* för sidan, och sidans *Size*, *Width*, *Height* med mera kan anges.

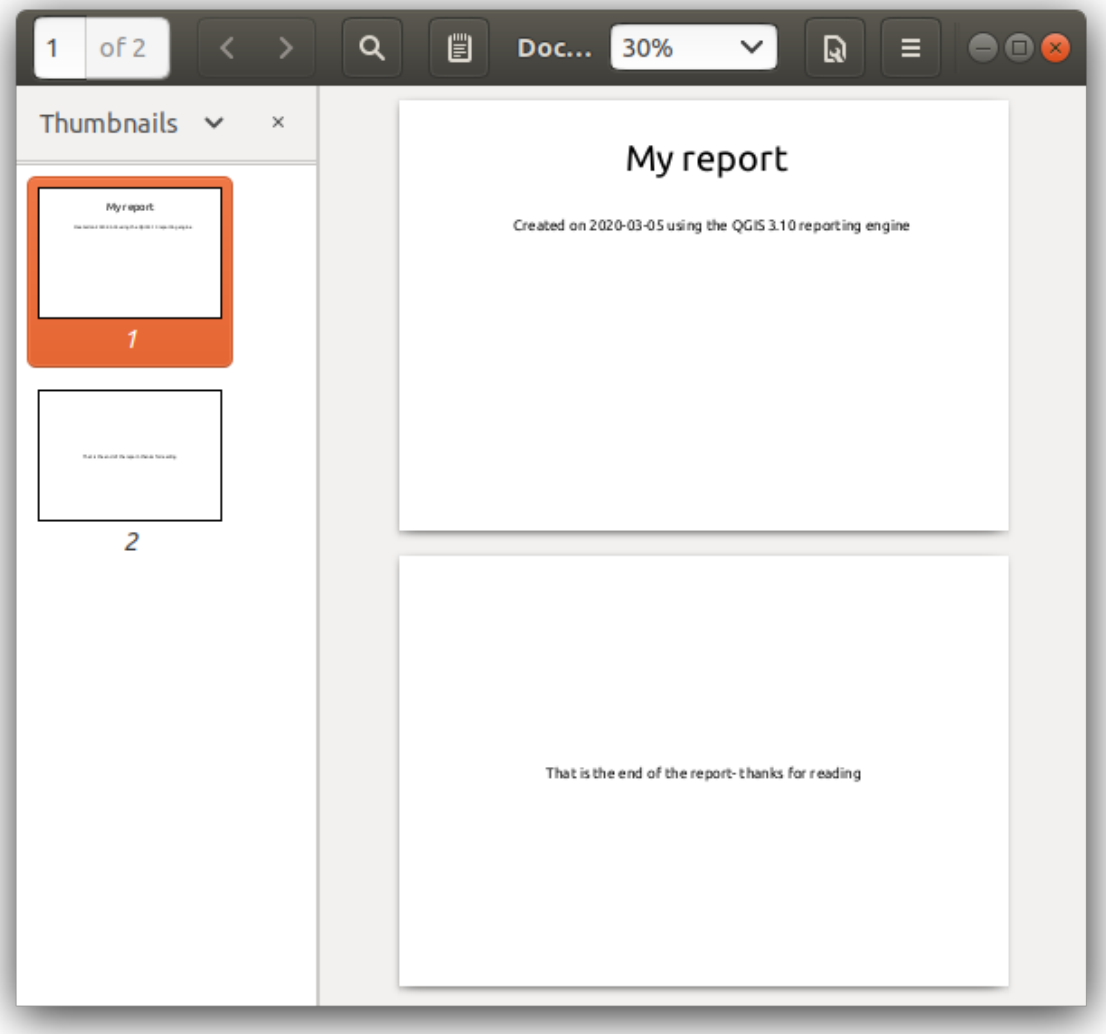
I QGIS-rapporter består varje komponent i rapporten av individuella layouter. De kan skapas och ändras med samma verktyg som för vanliga utskriftslayouter - så du kan använda vilken kombination av etiketter, bilder, kartor, tabeller etc. som helst. Låt oss lägga till några objekt i vårt rapporthuvud för att demonstrera:



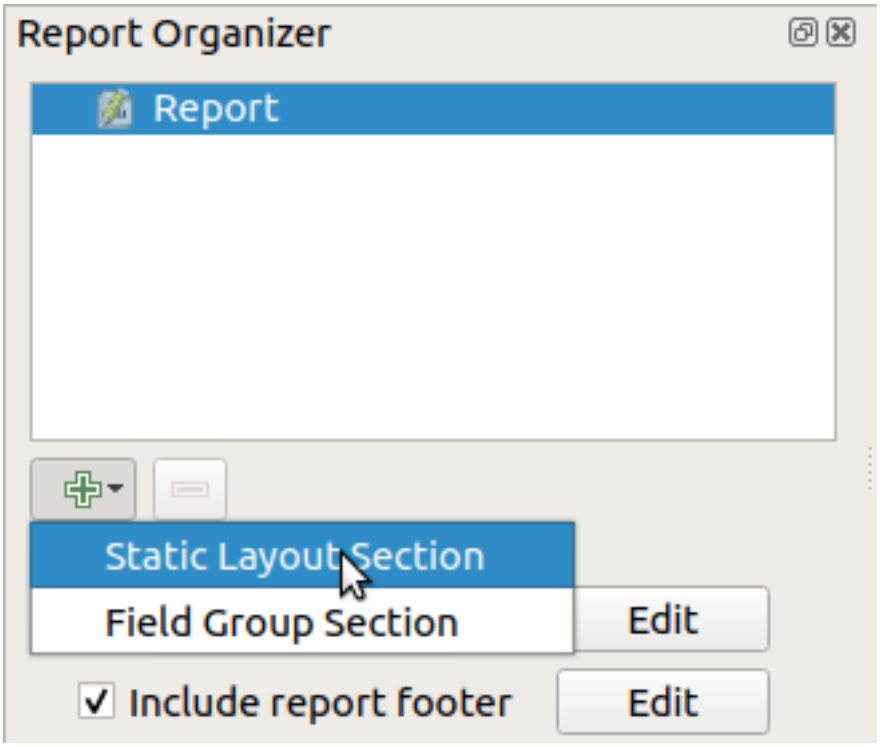
Vi kommer också att skapa en enkel sidfot för rapporten genom att markera alternativet *Include report footer* och trycka på *Edit*.



Innan vi går vidare, låt oss exportera denna rapport och se vad vi får. Exporten görs från menyn *Report* - i det här fallet väljer vi *Export Report as PDF…* för att göra om hela rapporten till en PDF-fil. Här är det inte särskilt imponerande resultatet - en PDF på två sidor som består av vårt sidhuvud och sidfot:



Låt oss göra saker och ting mer intressanta. Genom att trycka på knappen Add Section i *Report Organizer* får vi ett urval av nya avsnitt att lägga till i vår rapport.

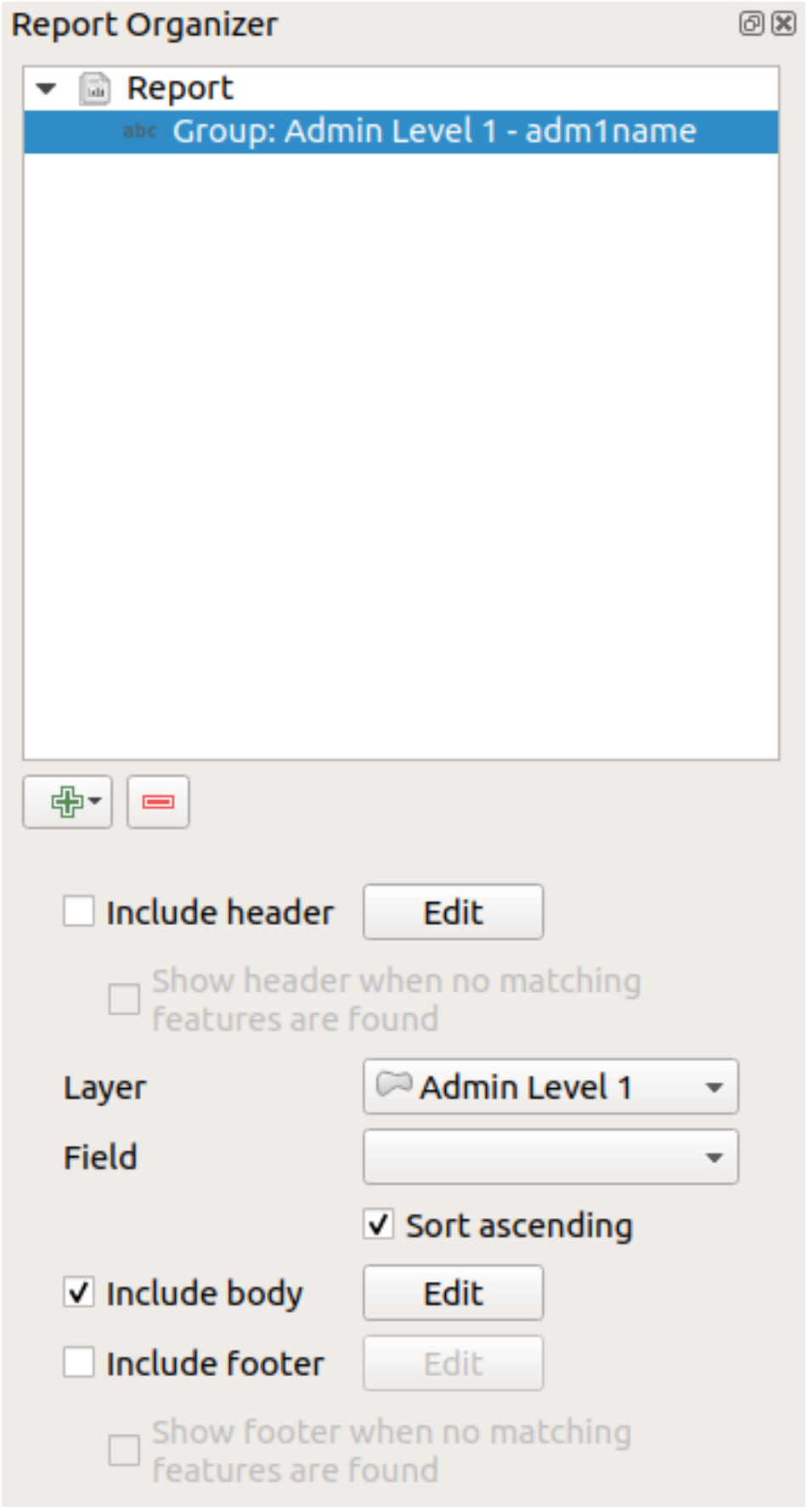


Det finns två alternativ: *Static Layout Section* och *Field Group Section*.

Avsnittet *Add Static Layout Section* är en enda, statisk layout för hela rapporten. Detta kan användas för att bädda in statiska layouter halvvägs genom en rapport.

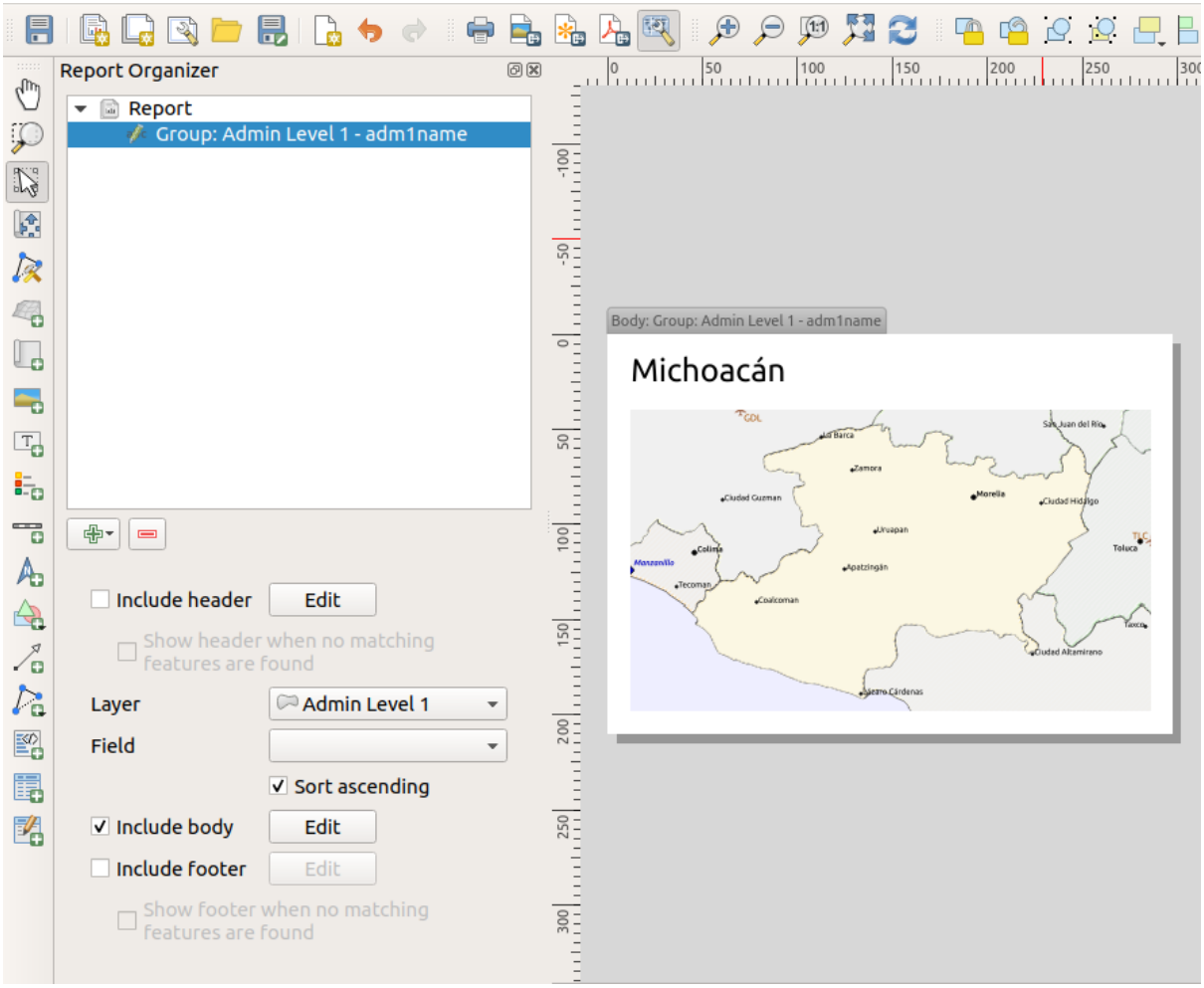
Field Group Section upprepar sin layout för varje funktion i ett lager. Objekten sorteras efter den valda grupperingsfunktionen (med möjlighet till stigande/fallande sortering). Om ett fältgruppsavsnitt har underordnade avsnitt (t.ex. ett annat fältgruppsavsnitt med ett annat fält) itereras endast funktioner med unika värden för gruppfunktionen. Detta möjliggör nästlade rapporter.

Nu ska vi lägga till en *Field Group Section* i vår rapport. På den mest grundläggande nivån kan du tänka på en *Field Group Section* som motsvarigheten till en *print atlas*: du väljer ett lager att iterera över, och rapporten infogar ett avsnitt för varje funktion som hittas. Om du väljer den nya *Field Group Section* visas ett antal nya relaterade inställningar:



I det här fallet har vi konfigurerat vår fältgrupp så att vi itererar över alla tillstånd från lagret *Adminnivå 1* med hjälp

av värdena från fältet *adm1name*. Samma alternativ för att inkludera sidhuvud och sidfot finns, tillsammans med ett nytt alternativ för att inkludera en *body* för det här avsnittet. Vi gör det och redigerar brödtexten:

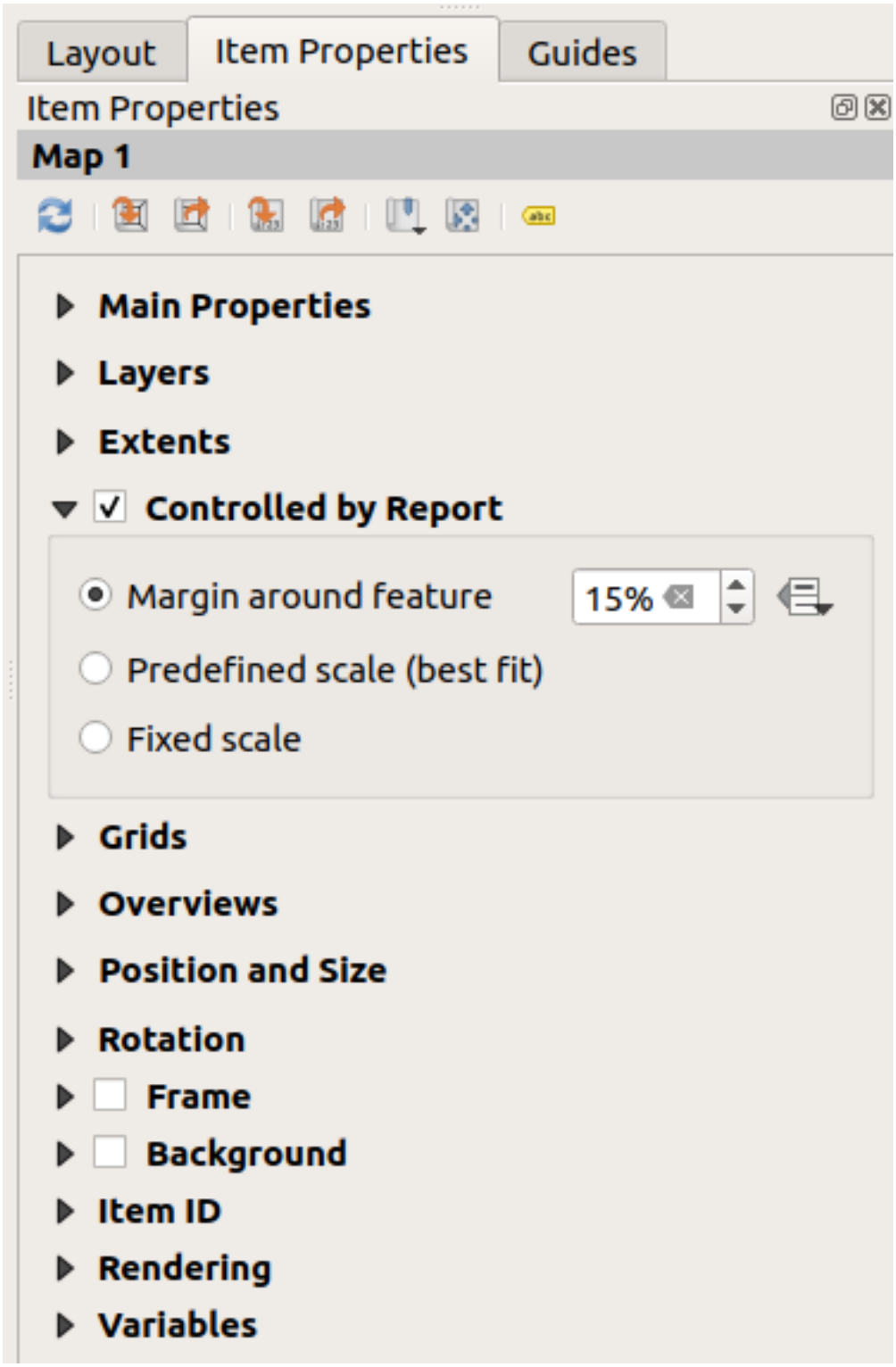


Vår body består nu av en karta och en etikett som visar namnet på delstaten. För att inkludera namnet på delstaten valde vi *Add Item* ► *Add Label* och datadefinierade texten under *Main Properties* med hjälp av *Insert or Edit an Expression*….

Resultatet blev följande uttryck (*name* är namnet på det attribut i *Admin Level 1*-lagret som innehåller namnet på delstaten):

```
[% "name" %]
```

Kartan är inställd på att följa den aktuella rapportfunktionen (aktiveras genom att markera *Controlled by Report* - precis som ett kartobjekt i en atlas kommer att följa den aktuella atlasfunktionen när *Controlled by Atlas* är markerad):



Om vi gick vidare och exporterade vår rapport nu, skulle vi få något liknande detta:

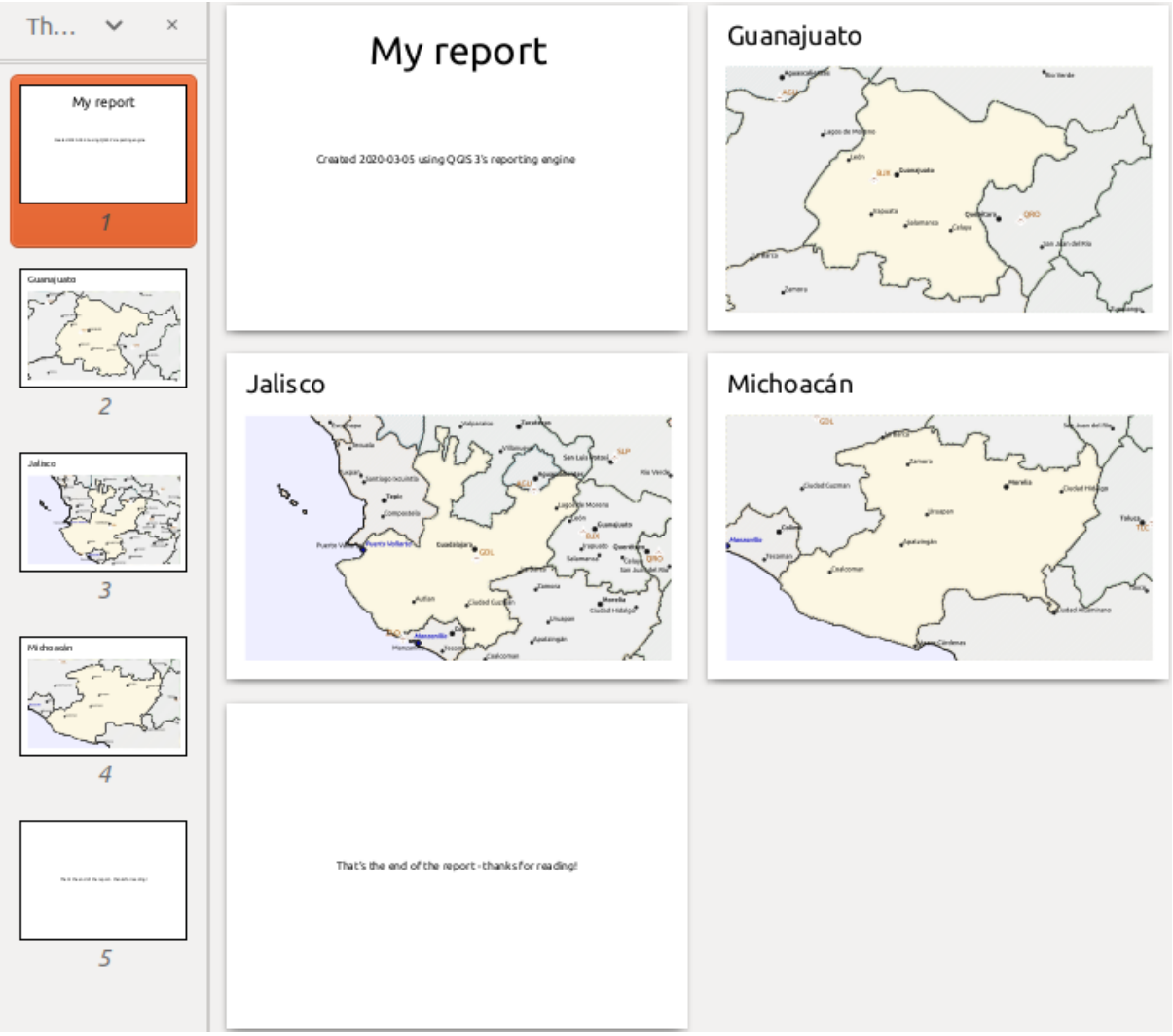
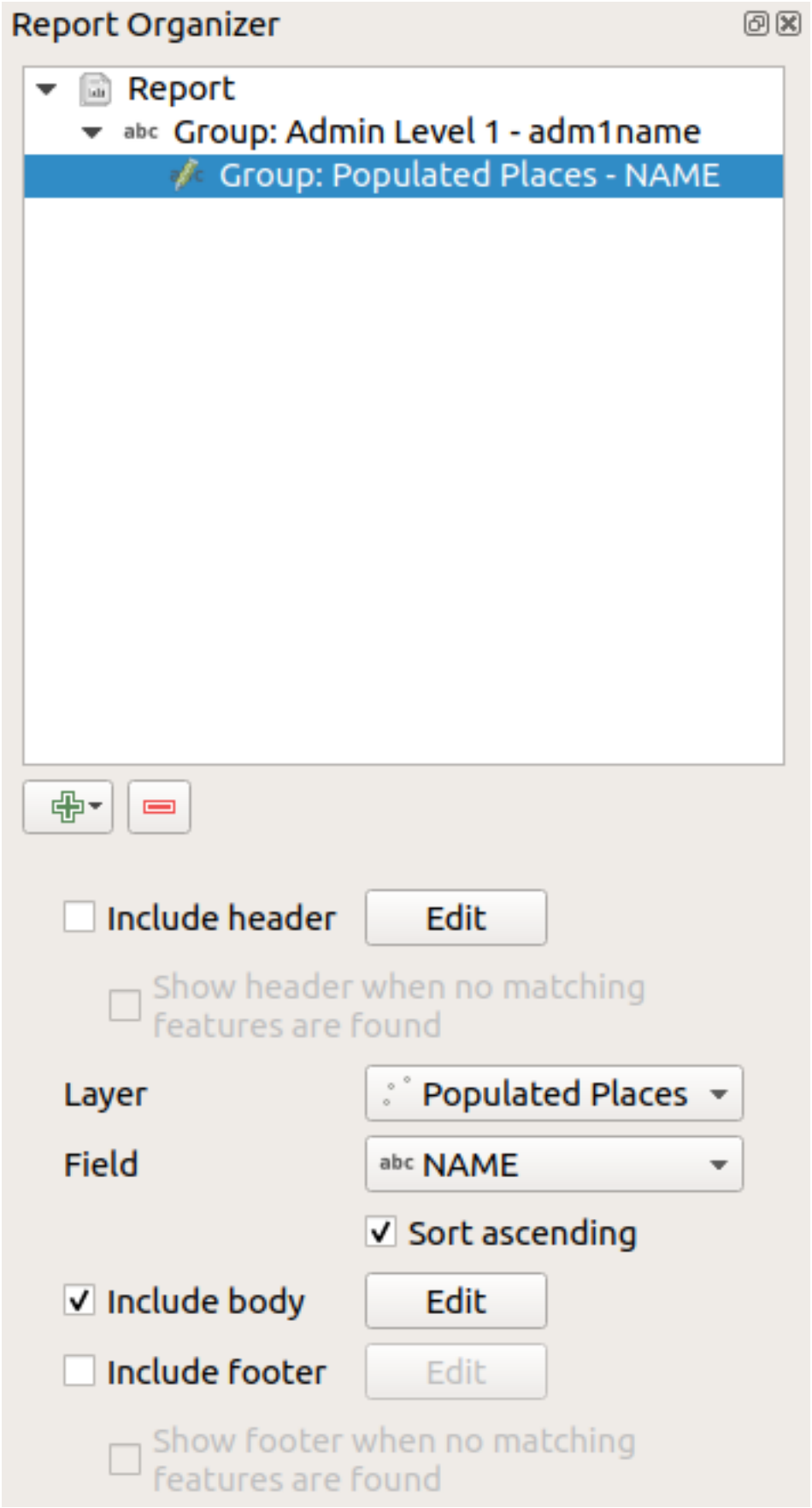


Fig. 18.69: Rapporthuvudet, en sida för varje delstat och rapportfoten.

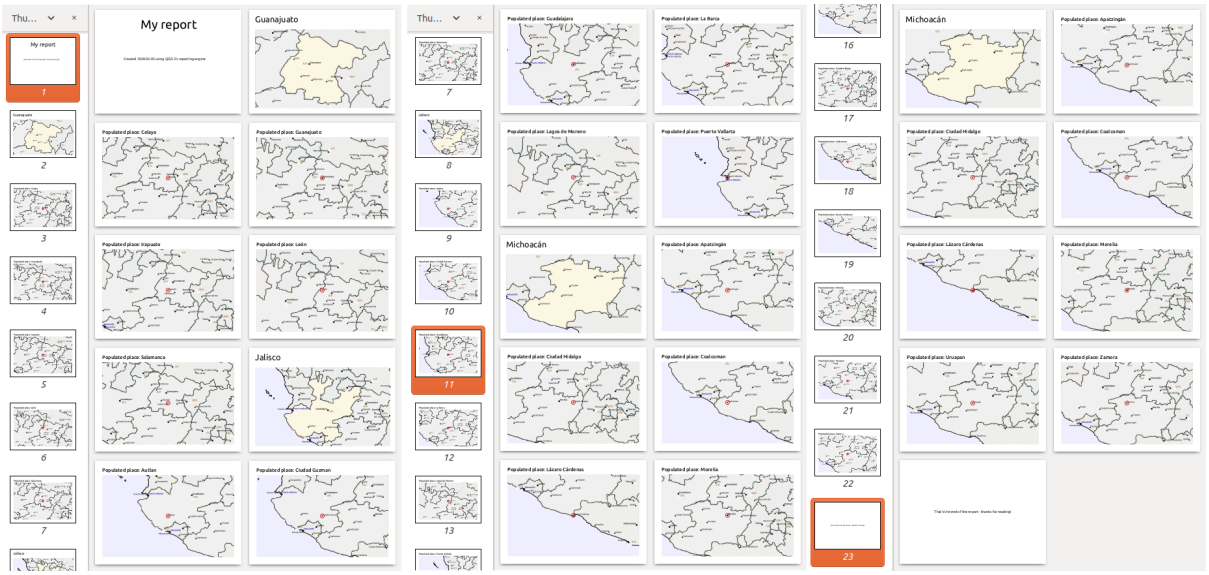
Alltså mer eller mindre en atlas, men med en sidhuvud- och sidfotssida.

Låt oss göra saker och ting mer intressanta genom att lägga till en underavdelning till vår statsgrupp. Det gör vi genom att först välja fältgruppen *Admin Level 1* i organisatören, sedan trycka på knappen Add Field och lägga till en ny *Field Group Section*:



När du itererar över funktionerna i ett *Field Group Section* filtreras funktionerna så att de matchar det definierande

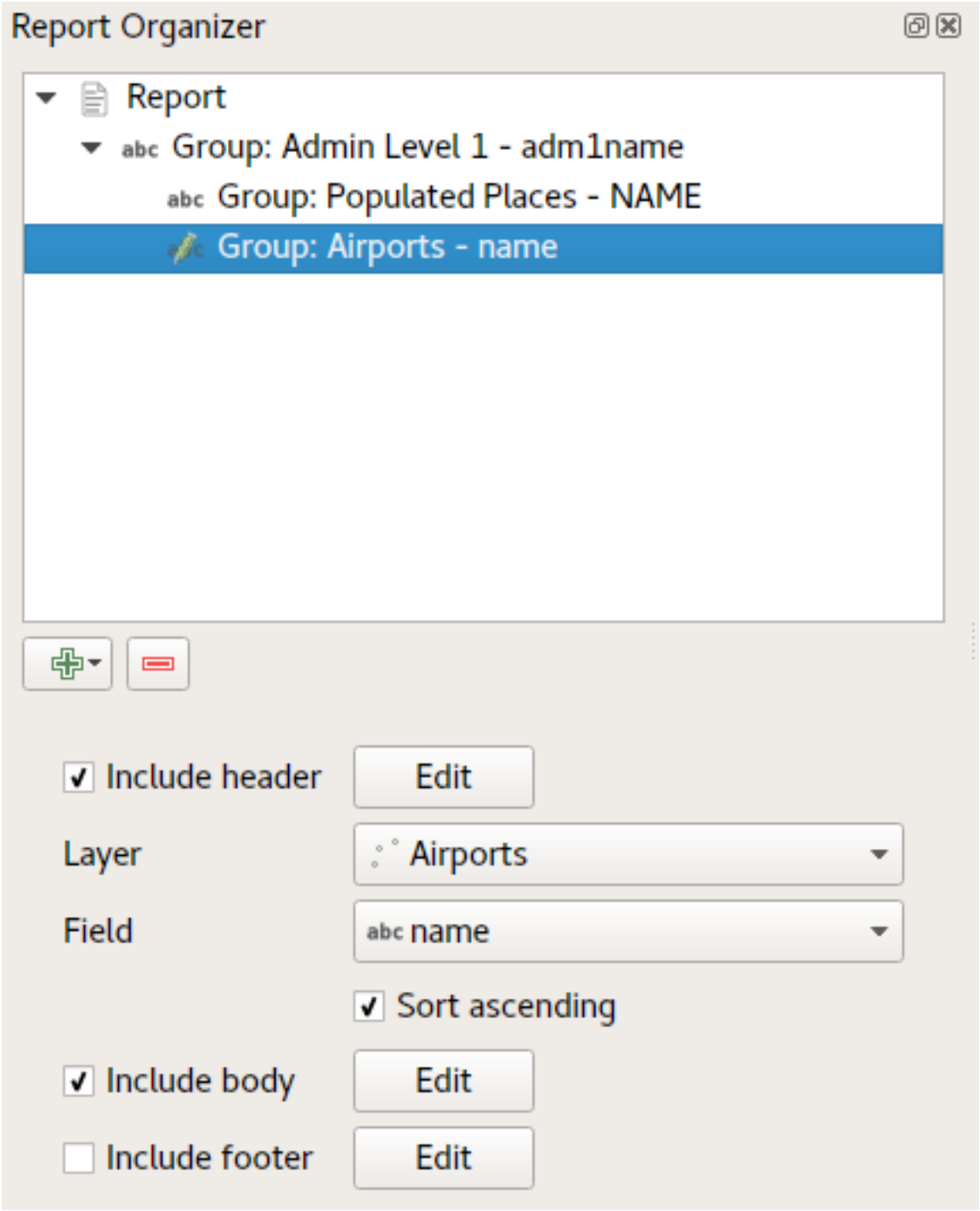
fältet i dess överordnade grupp (`adm1name` i det här fallet). Här kommer det underavsnitt vi lagt till att iterera över ett *Populated Places*-lager, inklusive ett kroppsavsnitt för varje plats som påträffas. Det magiska här är att lagret *Populated Places* har ett attribut med samma namn som det definierande fältet i det överordnade lagret, *adm1name*, som taggar varje plats med den stat som den ingår i (om du har tur är dina data redan strukturerade så här - om inte, kör *Join Attributes by Location* Processing-algoritmen och skapa ditt eget fält). När vi exporterar den här rapporten kommer QGIS att hämta den första staten från lagret *Admin Level 1* och sedan iterera över alla *Populated Places* med ett matchande *adm1name*-värde. Här är vad vi får:



Här skapade vi en grundläggande kropp för gruppen Populated Places, inklusive en karta över platsen och en tabell med några platsattribut. Så vår rapport består nu av ett rapporthuvud, en sida för den första delstaten, följt av en sida för varje befolkad plats i den delstaten, sedan resten av delstaterna med sina befolkade platser och slutligen rapportens sidfot. Om vi skulle lägga till en rubrik för gruppen Populated Places skulle den inkluderas precis innan vi listar de befolkade platserna för varje delstat, som visas i illustrationen nedan.

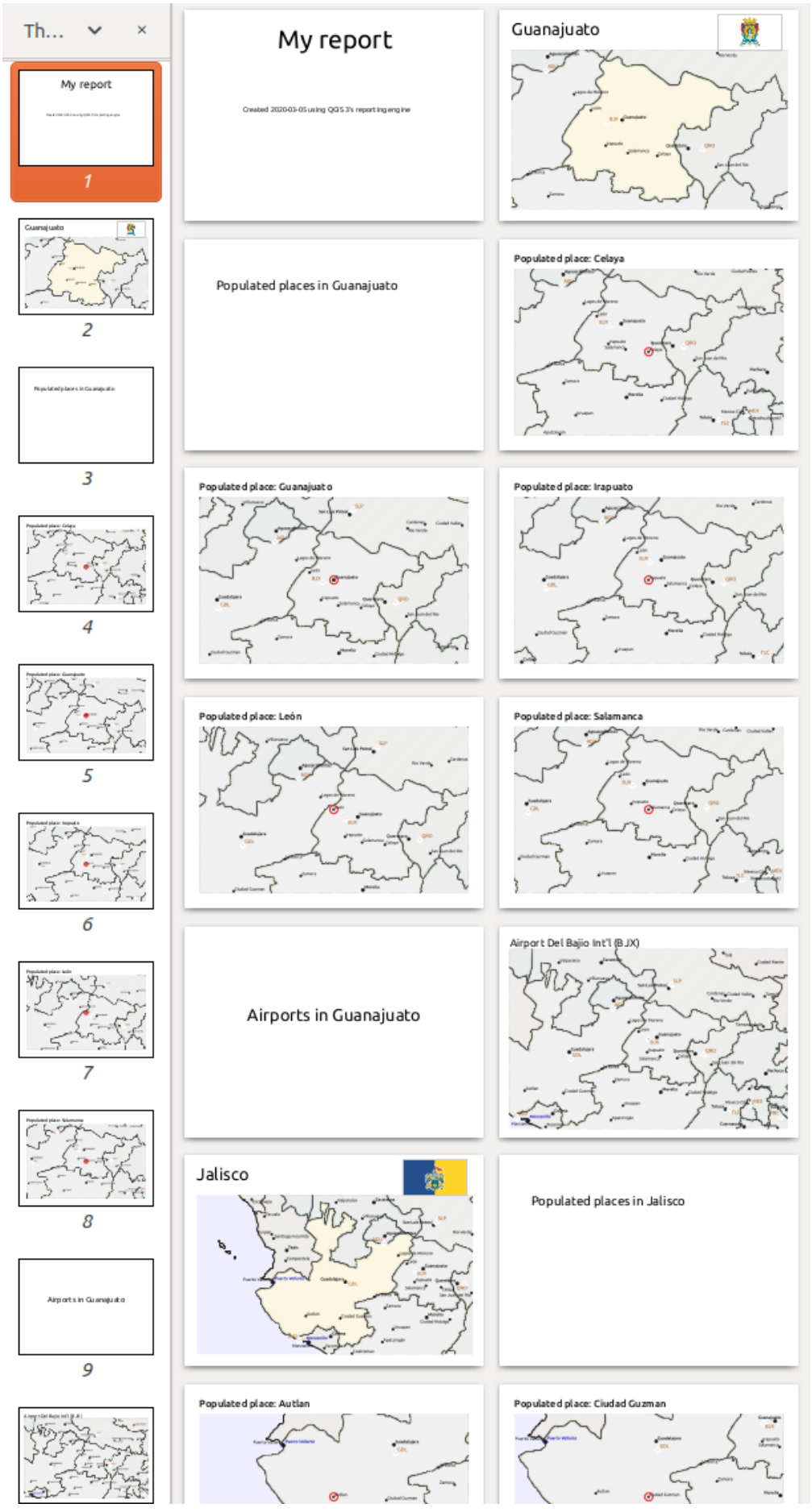
På samma sätt skulle en sidfot för gruppen Populated Places infogas efter att den sista platsen för varje delstat har inkluderats.

Förutom nästlade underavsnitt kan underavsnitt i en rapport också inkluderas i följd. Om vi lägger till ett andra underavsnitt till *Admin Level 1 group* för *Airports*, så (om *Airports*-lagret har ett attribut *adm1name* som kan länka det till den överordnade gruppen) kommer vår rapport först att lista ALLA befolkade platser för varje stat, följt av alla flygplatser i den staten, innan vi fortsätter till nästa stat.



Den viktigaste punkten här är att vår *Airports group* är en underavdelning av *Admin Level 1 group* - inte *Populated Places group*.

I det här fallet skulle vår rapport se ut så här (observera att delstatsflaggor också har inkluderats - hur man lägger till specifika bilder på det här sättet beskrivs nedan):


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

Inkludera bilder i en rapport

Bilder kan vara ganska användbara i rapporter, och QGIS tillåter bilder i både de statiska och dynamiska delarna av en rapport. Bilder läggs till på samma sätt som för vanliga utskriftslayouter, och för de statiska rapportdelarna (och statiska bilder i dynamiska delar) är det inte mer med det.

Men om du vill ha illustrationer som är anpassade till rapportens funktioner måste ditt lager ha ett attribut som kan användas för att definiera vilken bild som ska inkluderas.

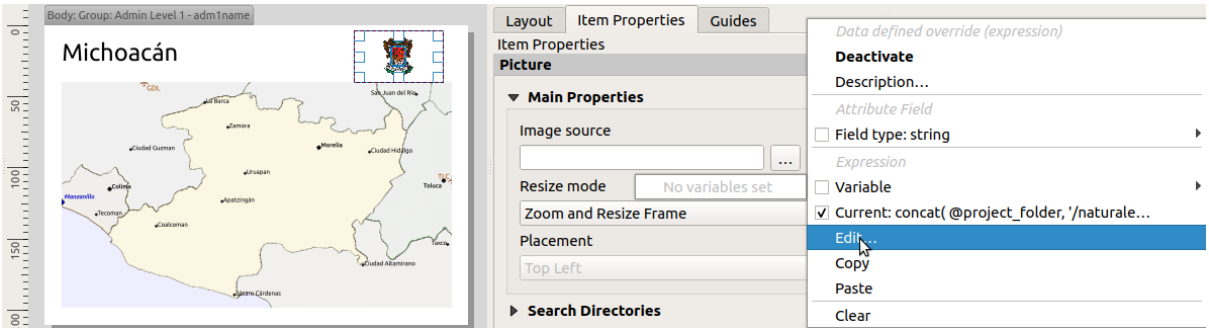
QGIS är beroende av absoluta filnamn för bilder i rapporter.

För dynamiska bilder lägger du först till en bild i gruppens kroppsdel, som vanligt. I *Item properties* för bilden anger du *Image Source* med hjälp av knappen Data defined override och väljer antingen ett attribut som innehåller bildernas absoluta sökväg eller *Edit…* (för att ange ett uttryck som genererar den absoluta bildsökvägen).

Nedan visas ett exempel på ett uttryck som använder strängkonkatenering för att ange den absoluta sökvägen till bilderna, med hjälp av katalogen där projektfilen finns (`@project_path`) och ett attribut (`adm1name`) från vilket filnamnet genereras (i det här fallet genom att omvandla strängen i attributet `adm1name` till versaler och lägga till '_flag.png'):

```
concat(@project_folder, '/naturalearth/pictures/' ,
       upper("adm1name"), '_flag.png')
```

Detta innebär att bilderna finns i underkatalogen `naturalearth/pictures` i projektets filkatalog.



Markera det aktuella rapportobjektet på en karta

I ovanstående rapport framhävs rapportfunktionerna i kartorna med hjälp av markeringar (stat) och cirklar (befolkade platser). För att framhäva rapportens funktioner på kartorna (förutom att placera dem i mitten av kartorna) måste du datadefiniera stilen med hjälp av en jämförelse mellan dess `@id` och `@atlas_featureid`, som för atlaser.

Om du t.ex. vill använda en tjockare linje/kant för rapportfunktionen än för de andra funktionerna kan du definiera linjebredden:

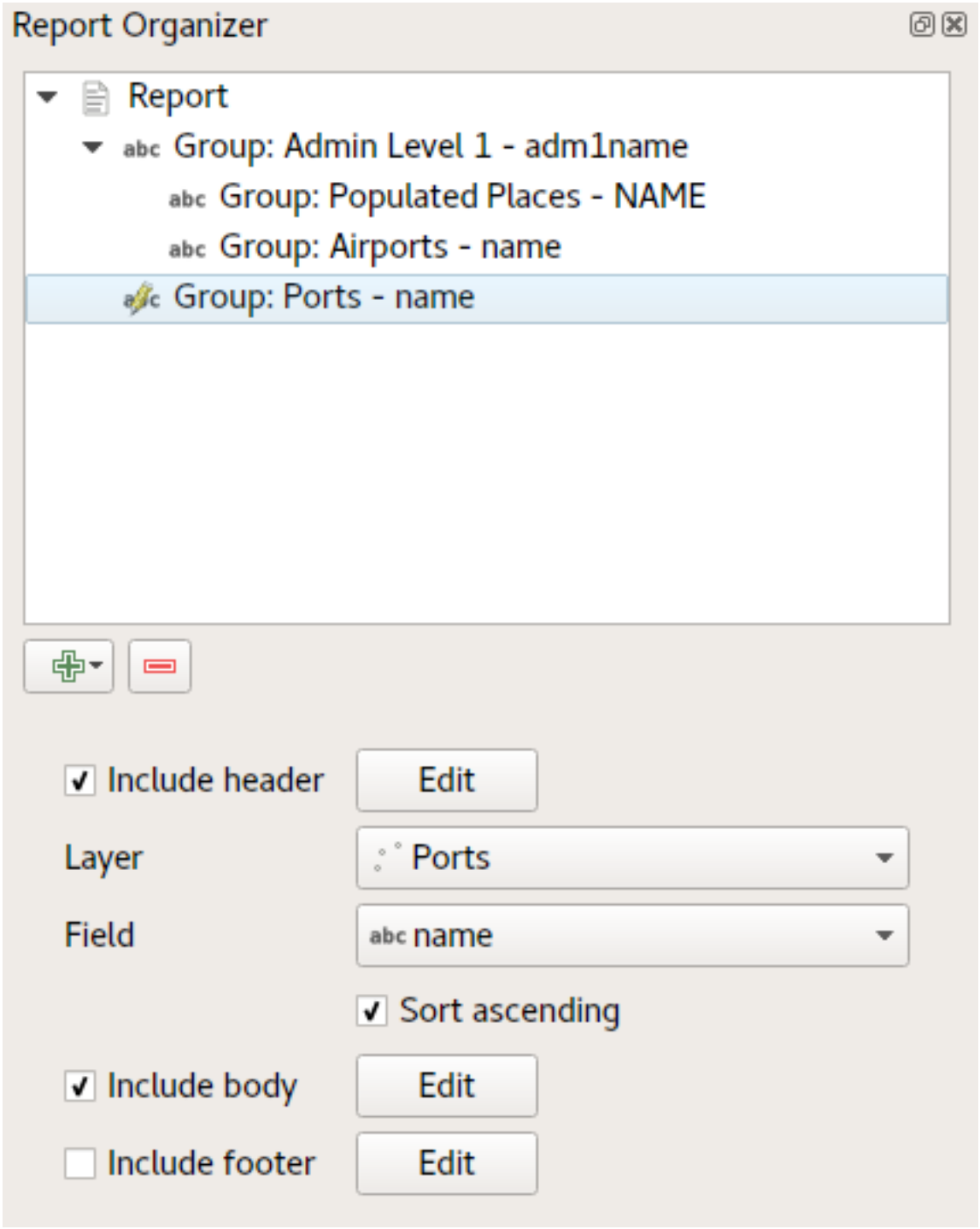
```
if($id=@atlas_featureid, 2.0, 0.1)
```

Rapportfunktionen får en 2 enheter bred polygonkontur, medan alla andra funktioner får en 0,1 enheter bred linje. Det är också möjligt att datadefiniera färgen (icke-transparent mörk magenta för rapportfunktionen och halvtransparent ljusgrå för de andra funktionerna):

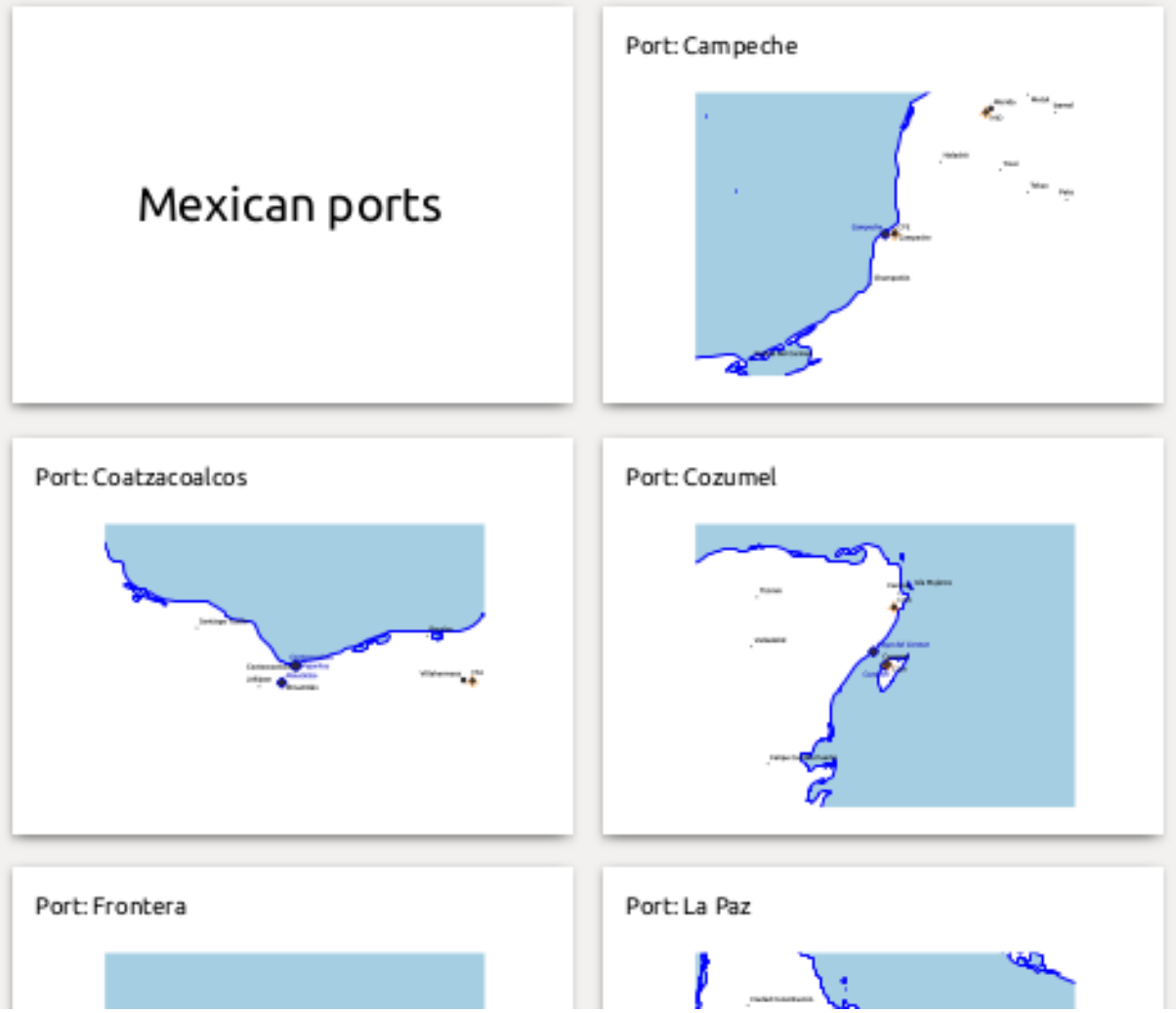
```
if($id=@atlas_featureid, '#FF880088', '#88CCCCCC')
```

Fler grupper på nivå 1

Kombinationen av nästlade och på varandra följande avsnitt, tillsammans med avsnittsrubriker och sidfötter, ger stor flexibilitet. I rapporten nedan lägger vi till exempel till en annan fältgrupp som underordnad huvudrapporten för lagret *Ports*. Nu, efter att ha listat staterna tillsammans med deras befolkade platser och flygplatser, får vi en sammanfattande lista över alla hamnar i regionen:



Detta resulterar i att den sista delen av vår rapport exporteras som:



18.4.4 Inställningar för export

När du exporterar en rapport (*Rapport* ► *Exportera rapport som bilder… / SVG… / PDF…*) blir du ombedd att ange ett filnamn, och sedan får du möjlighet att justera exportinställningarna för att få den mest lämpliga utskriften.

Som du ser är rapporter i QGIS extremt kraftfulla och flexibla!

Observera

Den aktuella informationen har hämtats från en North Road-blogg, Exploring Reports in QGIS 3.0 - the Ultimate Guide!

KAPITEL 19

Arbeta med OGC/ISO-protokoll

Open Geospatial Consortium (OGC) är en internationell organisation med mer än 300 medlemmar från kommersiella, statliga, ideella och forskningsorganisationer över hela världen. Dess medlemmar utvecklar och implementerar standarder för geospatialt innehåll och tjänster, GIS-databehandling och utbyte.

OGC beskriver en grundläggande datamodell för geografiska objekt och utvecklar ett ökande antal specifikationer för att tillgodose specifika behov av interoperabel plats- och geospatial teknik, inklusive GIS. Ytterligare information finns på https://www.ogc.org/.

Viktiga OGC-specifikationer som stöds av QGIS är:

- **WMS** — Web Map Service (*WMS/WMTS-klient*)
- **WMTS** — Web Map Tile Service (*WMS/WMTS-klient*)
- **WFS** — Web Feature Service (*WFS och WFS-T Klient*)
- **WFS-T** — Web Feature Service - Transactional (*WFS och WFS-T Klient*)
- **WCS** — Web Coverage Service (*WCS-klient*)
- **WPS** — Web Processing Service
- **CSW** — Katalogtjänst för webben
- **SFS** — Enkla funktioner för SQL (*PostGIS-lager*)
- **GML** — Geografiskt Markup Language
- **SensorThings API** — SensorThings API (*SensorThings*)

OGC-tjänster används alltmer för att utbyta geospatiala data mellan olika GIS-implementeringar och datalager. QGIS kan hantera ovanstående specifikationer som en klient och är **SFS** (genom stöd för PostgreSQL / PostGIS-dataleverantören, se avsnitt *PostGIS-lager*).

Du kan också dela dina kartor och data via protokollen WMS, WMTS, WFS, WFS-T och WCS med hjälp av en webbserver med QGIS Server, UMN MapServer eller GeoServer installerad.

19.1 WMS/WMTS-klient

19.1.1 Översikt över WMS-support

QGIS kan för närvarande fungera som en WMS-klient som förstår WMS 1.1-, 1.1.1- och 1.3-servrar. I synnerhet har det testats mot allmänt tillgängliga servrar som DEMIS.

En WMS-server agerar på förfrågningar från klienten (t.ex. QGIS) om en rasterkarta med en given utsträckning, uppsättning lager, symboliseringsstil och transparens. WMS-servern konsulterar sedan sina lokala datakällor, rastrerar kartan och skickar tillbaka den till klienten i ett rasterformat. För QGIS är detta format vanligtvis JPEG eller PNG.

WMS är generellt sett en REST-tjänst (Representational State Transfer) snarare än en fullfjädrad webbtjänst. Som sådan kan du faktiskt ta de webbadresser som genereras av QGIS och använda dem i en webbläsare för att hämta samma bilder som QGIS använder internt. Detta kan vara användbart vid felsökning, eftersom det finns flera olika WMS-servermärken på marknaden och de har alla sin egen tolkning av WMS-standarden.

WMS-lager kan läggas till ganska enkelt, så länge du känner till URL:en för att komma åt WMS-servern, du har en fungerande anslutning till den servern och servern förstår HTTP som datatransportmekanism.

Dessutom kommer QGIS att cacha dina WMS-svar (dvs. bilder) i 24 timmar så länge som GetCapabilities-begäran inte utlöses. GetCapabilities-begäran utlöses varje gång knappen *Connect* i dialogrutan *WMS/WMTS* används för att hämta WMS-serverns kapacitet. Detta är en automatisk funktion som är avsedd att optimera projektets laddningstid. Om ett projekt sparas med ett WMS-lager kommer motsvarande WMS-plattor att laddas från cacheminnet nästa gång projektet öppnas, så länge de inte är äldre än 24 timmar.

19.1.2 Översikt över WMTS-stöd

QGIS kan också fungera som en WMTS-klient. WMTS är en OGC-standard för distribution av kakeluppsättningar med geospatiala data. Det här är ett snabbare och effektivare sätt att distribuera data än WMS, eftersom WMTS innebär att kakeluppsättningarna är förgenererade och att klienten bara begär överföring av kakeluppsättningarna, inte produktion av dem. En WMS-begäran omfattar vanligtvis både generering och överföring av data. Ett välkänt exempel på en icke-OGC-standard för visning av kaklade geospatiala data är Google Maps.

För att kunna visa data i en mängd olika skalor som ligger nära vad användaren kan tänkas vilja ha, produceras WMTS-kakelsatserna i flera olika skalnivåer och görs tillgängliga för GIS-klienten som begär dem.

Detta diagram illustrerar konceptet med kakeluppsättningar:

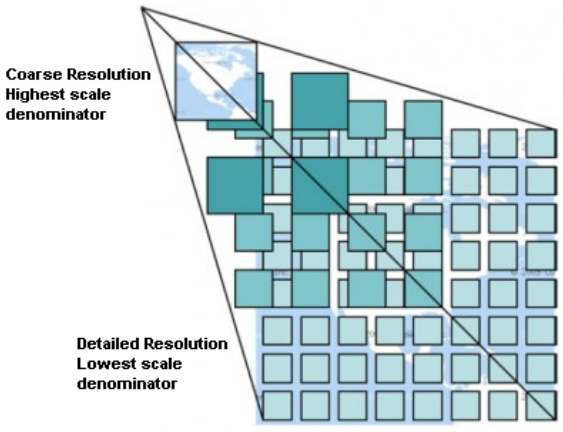


Fig. 19.1: Koncept för WMTS-kakeluppsättningar

De två typerna av WMTS-gränssnitt som QGIS stöder är via KVP (Key-Value-Pairs) och RESTful. Dessa två gränssnitt är olika och du måste ange dem på olika sätt för QGIS.

1. För att komma åt en **WMTS KVP**-tjänst måste en QGIS-användare öppna WMS/WMTS-gränssnittet och lägga till följande sträng i URL:en för WMTS-kakeltjänsten:

```
"?SERVICE=WMTS&REQUEST=GetCapabilities"
```

Ett exempel på denna typ av adress är:

```
https://opencache.statkart.no/gatekeeper/gk/gk.open_wmts?service=WMTS&
↪request=GetCapabilities
```

För testning fungerar topo2-lagret i denna WMTS bra. Om du lägger till denna sträng anger du att en WMTS-webbtjänst ska användas i stället för en WMS-tjänst.

2. Tjänsten **RESTful WMTS** har en annan form, en enkel URL. Det format som rekommenderas av OGC är:

```
{WMTSBaseURL}/1.0.0/WMTSCapabilities.xml
```

Detta format hjälper dig att känna igen att det är en RESTful-adress. En RESTful WMTS nås i QGIS genom att helt enkelt lägga till dess adress i WMS-konfigurationen i URL-fältet i formuläret. Ett exempel på den här typen av adress för en österrikisk baskarta är:

```
https://maps.wien.gv.at/basemap/1.0.0/WMTSCapabilities.xml
```

Observera

Du kan fortfarande hitta några gamla tjänster som kallas WMS-C. Dessa tjänster är ganska lika WMTS (dvs. samma syfte men fungerar lite annorlunda). Du kan hantera dem på samma sätt som du hanterar WMTS-tjänster. Lägg bara till `?tiled=true` i slutet av webbadressen. Se https://wiki.osgeo.org/wiki/Tile_Map_Service_Specification för mer information om den här specifikationen.

När man läser WMTS kan man ofta tänka WMS-C också.

19.1.3 Välja WMS/WMTS-servrar

Första gången du använder WMS/WMTS-funktionen i QGIS finns det inga servrar definierade.

Du måste sedan skapa anslutningar till den server du riktar in dig på:

1. Gå till fliken *WMS/WMTS* i dialogrutan *Data Source Manager*, antingen genom att:
   - klicka på knappen Open Data Source Manager (eller tryck på `Ctrl+L`) och aktivera fliken
   - klicka på knappen Lägg till WMS-lager i verktygsfältet *Manage Layers* (hantera lager)
   - eller genom att välja *Lager* ► *Lägg till lager* ► *Lägg till WMS/WMTS-lager…*-menyn
2. Tryck på *New* från fliken *Layers*. Dialogrutan *Create a New WMS/WMTS Connection…* visas.

   Tips

   Högerklicka på posten *WMS/WMTS* i *Nätläsarpanelen* och välj *New Connection…* öppnar också dialogrutan *Create a New WMS/WMTS Connection…*.

3. Ange sedan parametrarna för att ansluta till önskad WMS-server enligt listan nedan:

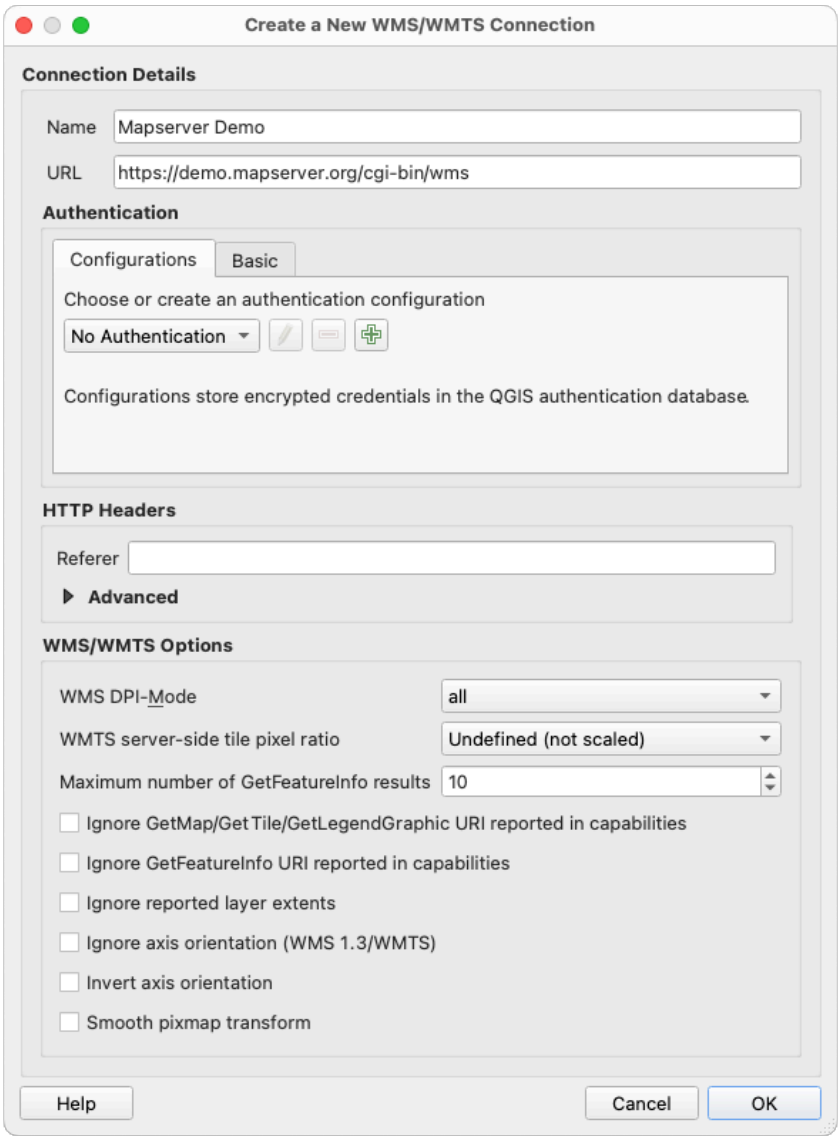


Fig. 19.2: Skapa en anslutning till en WMS-server

- *Namn*: Ett namn för anslutningen. Detta namn kommer att användas i rullgardinsmenyn Serveranslutningar så att du kan skilja den från andra WMS-servrar.
- *URL*: URL till den server som tillhandahåller data. Detta måste vara ett upplösningsbart värdnamn - samma format som du skulle använda för att öppna en telnet-anslutning eller pinga en värd, dvs. endast bas-URL:en. Du bör t.ex. inte ha fragment som `request=GetCapabilities` eller `version=1.0.0` i din URL.
- *Autentisering* (valfritt): med hjälp av en *lagrad konfiguration* eller en grundläggande autentisering med *Användarnamn* och *Lösenord*.

> **Varning**
>
> Om du anger **användarnamn** och **lösenord** på fliken *Autentication* kommer oskyddade uppgifter att sparas i anslutningskonfigurationen. Dessa **uppgifter kommer att vara synliga** om du t.ex. delar projektfilen med någon. Därför är det lämpligt att spara dina autentiseringsuppgifter i en *Autentication-konfiguration* istället (fliken *Configurations*). Se *System för autentisering* för mer information.

- HTTP *Referer*

- *WMS DPI-Mode*: Tillgängliga alternativ är **all**, **off**, **QGIS**, **UMN** och **GeoServer**
- *WMTS pixelförhållande på serversidan*: Vid rendering av WMTS-lager kan du skala upp eller ner brickorna baserat på enhetens skärm-DPI. Tillgängliga alternativ är *Undefined (inte skalad)*, *Standard (96 DPI)* och *High (192 DPI)*.
- *Maximalt antal GetFeatureInfo-resultat*: anger ett standardvärde för det maximala antalet resultat som returneras per lager av en GetFeatureInfo-begäran som använder den här anslutningen (se FEATURE_COUNT parameter). Standardvärdet är `10`. Ställ in till `0` för att använda serverns standardvärde (vanligtvis `1`): ingen FEATURE_COUNT-parameter läggs till i begäran.
- ☐ *Ignorera GetMap/GetTile/GetLegendGraphic URI reported in capabilities*: om markerat, använd angiven URI från fältet *URL* ovan.
- ☐ *Ignorera GetFeatureInfo URI reported in capabilities*: om markerat, använd angiven URI från fältet *URL* ovan.
- ☐ *Ignore reported layer extents*: eftersom den utsträckning som rapporteras av rasterlager kan vara mindre än det faktiska område som kan återges (särskilt för WMS-servrar med symbologi som tar mer plats än datautsträckningen), markera det här alternativet för att undvika beskärning av rasterlager till deras rapporterade utsträckning, vilket resulterar i avkortade symboler på gränserna för dessa lager.
- ☐ *Ignorera axelorientering (WMS 1.3/WMTS)*
- ☐ *Invertera axelns orientering*
- ☐ *Smooth pixmap transformation*

4. Tryck på *OK*

När den nya WMS/WMTS-serveranslutningen har skapats kommer den att bevaras för framtida QGIS-sessioner. Observera att det också är möjligt att *Load* anslutningsparametrarna från en `.XML`-fil eller *Save* dem till en `.XML`-fil.

Om du behöver ställa in en proxyserver för att kunna ta emot WMS-tjänster från Internet kan du lägga till din proxyserver i alternativen. Välj *Inställningar* ► *Alternativ* och klicka på fliken *Nätverk*. Där kan du lägga till dina proxyinställningar och aktivera dem genom att markera kryssrutan *Use proxy for web access*. Se till att du väljer rätt proxytyp från rullgardinsmenyn *Proxytyp* .

19.1.4 Laddning av WMS/WMTS-lager

När du har fyllt i dina parametrar kan du använda knappen *Connect* för att hämta den valda serverns funktioner. Detta inkluderar bildkodning, lager, lagerstilar och projektioner. Eftersom detta är en nätverksoperation beror svarshastigheten på kvaliteten på din nätverksanslutning till WMS-servern. Under nedladdningen av data från WMS-servern visualiseras nedladdningsförloppet i det nedre vänstra hörnet av QGIS huvuddialogruta.

Din skärm bör nu se ut ungefär som Fig. 19.3, som visar svaret från en WMS-server.

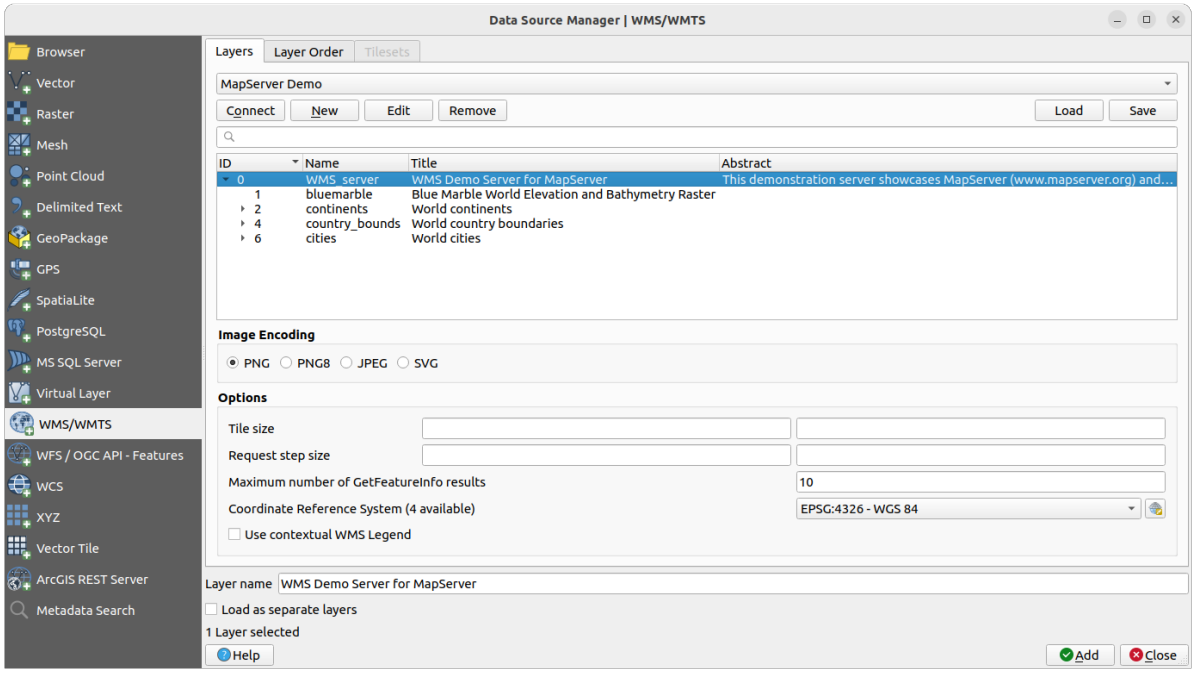


Fig. 19.3: Dialog för att lägga till en WMS-server, med filter på tillgängliga lager

Den övre delen av fliken *Layers* i dialogrutan visar en trädstruktur som kan innehålla lagergrupper som bäddar in lager med tillhörande bildstil(ar) som serveras av servern. Varje objekt kan identifieras genom:

- en *ID*
- a *Namn*
- a *Title*
- och en *Abstract*.

Listan kan filtreras med hjälp av widgeten i det övre högra hörnet.

Bildkodning

Avsnittet *Bildkodning* innehåller en lista över de format som stöds av både klienten och servern. Välj ett beroende på dina krav på bildnoggrannhet.

Tips

Bildkodning

Du kommer vanligtvis att upptäcka att en WMS-server erbjuder dig valet av JPEG- eller PNG-bildkodning. JPEG är ett komprimeringsformat med förlust, medan PNG troget återger de råa rasterdata.

Använd JPEG om du förväntar dig att WMS-data ska vara av fotografisk karaktär och/eller om du inte har något emot en viss förlust i bildkvalitet. Denna kompromiss minskar vanligtvis dataöverföringsbehovet med fem gånger jämfört med PNG.

Använd PNG om du vill ha exakta representationer av originaldata och du inte har något emot de ökade kraven på dataöverföring.

Menyalternativ: *Lager* ► *Lägg till ett rasterlager* eller *Inställningar* ► *Verktygsrader* ► *Digitalisering*

I dialogrutans område Options kan du konfigurera WMS-begärandena. Du kan definiera:

- *Tile size* om du vill ställa in kakelstorlekar (t.ex. 256x256) för att dela upp WMS-begäran i flera begäranden.

- *Request step size*: om du vill minska effekten av klippta etiketter vid kakelplattornas kanter, skapar en ökning av stegstorleken större förfrågningar, färre plattor och färre kanter. Standardvärdet är 2000.
- *Maximalt antal GetFeatureInfo-resultat*: anger det maximala antalet resultat som returneras av en GetFeatureInfo-begäran för det eller de lager som laddas. Gör det möjligt att åsidosätta *anslutningens standardvärde* för specifika lager.
- Varje WMS-lager kan presenteras i flera CRS:er, beroende på WMS-serverns kapacitet. Om du väljer ett WMS i listan visas ett fält med den standardprojektion som tillhandahålls av webbservern. Tryck på widgeten :sup:`Select CRS` för att ersätta standardprojektionen för WMS med en annan CRS som stöds av WMS-servern.

  En dialog som liknar den som visas i Fig. 6.4 kommer att visas. Den största skillnaden med WMS-versionen av dialogrutan är att endast de CRS som stöds av WMS-servern kommer att visas.
- Slutligen kan du aktivera *Use contextual WMS Legend* om WMS-servern stöder den här funktionen. Då visas endast den relevanta förklaringen för den aktuella kartvyns omfattning och inkluderar således inte legendobjekt för objekt som du inte kan se på den aktuella kartan.

Längst ned i dialogrutan finns ett textfält *Layer name* som visar det valda objektets *Title*. Du kan ändra namnet när du vill. Detta namn kommer att visas i panelen *Layers* efter att du har tryckt på knappen *Add* och laddat lagret/lagren i QGIS.

Du kan välja flera lager samtidigt, men bara en bildstil per lager. När flera lager väljs kombineras de på WMS-servern och överförs till QGIS på en gång, som ett enda lager. Standardnamnet är en snedstrecksseparerad (/) lista med deras ursprungliga titel. Du kan dock välja att *Ladda som separata lager*.

Lagerordning

Fliken *Layer Order* listar de valda lager som är tillgängliga från den aktuella anslutna WMS-servern.

WMS-lager som återges av en server överlagras i den ordning som anges på fliken *Layers*, från toppen till botten av listan. Om du vill ändra överläggningsordningen kan du använda knapparna *Up* och *Down* på fliken *Layer Order*.

Öppenhet

Inställningen *Global transparency* från *Lageregenskaper* är hårdkodad till att alltid vara på, om den finns tillgänglig.

19.1.5 Mosaiker

När du använder WMTS (Cached WMS) tjänster kan du bläddra igenom fliken *Tilesets* som ges av servern. Ytterligare information som kakelstorlek, format och stödda CRS listas i denna tabell.

```
# example of WMTS service

https://opencache.statkart.no/gatekeeper/gk/gk.open_wmts?service=WMTS&
↪request=GetCapabilities
```

När du väljer ett lager att ladda är det också möjligt att tillämpa en *Tolkningsmetod*, som omvandlar det till ett rasterlager av typen single band float, redo för styling med QGIS vanliga *rasterrenderare*.

I kombination med den här funktionen kan du använda skjutreglaget för kakelskalan genom att välja *Visa* ► *Paneler* (eller *Inställningar* ► *Paneler*) och sedan välja *Tile Scale Panel*. Detta ger dig de tillgängliga skalorna från kakelservern med ett snyggt reglage dockat i.

19.1.6 Använda identifieringsverktyget

När du har lagt till en WMS-server, och om något lager från en WMS-server är sökbart, kan du använda verktyget Identify för att välja en pixel på kartbilden. En fråga ställs till WMS-servern för varje val som görs. Resultatet av förfrågan returneras i klartext. Formateringen av denna text är beroende av vilken WMS-server som används.

Val av format

Om flera utdataformat stöds av servern läggs en kombinationsruta med format som stöds automatiskt till i dialogrutan för identifieringsresultat och det valda formatet kan lagras i projektet för lagret.

Stöd för GML-format

Verktyget Identifiera stöder WMS-serverns svar (GetFeatureInfo) i GML-format (det kallas Funktion i QGIS grafiska gränssnitt i det här sammanhanget). Om ”Funktion”-formatet stöds av servern och väljs, är resultaten från Identifiera-verktyget vektorfunktioner, som från ett vanligt vektorlager. När en enskild funktion väljs i trädet markeras den på kartan och den kan kopieras till urklippet och klistras in i ett annat vektorlager. Se exemplet på konfiguration av UMN Mapserver nedan för att stödja GetFeatureInfo i 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)
# 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 Visning av egenskaper

När du har lagt till en WMS-server kan du visa dess egenskaper genom att högerklicka på den i teckenförklaringen och välja *Properties*. WMS/WMTS-lagrets egenskaper liknar rasterlagrets egenskaper, så du hittar en detaljerad beskrivning i *Dialog för egenskaper för raster*. Det finns dock vissa skillnader, som förklaras nedan.

Informationsegenskaper

Fliken Metadata

Fliken *Metadata* visar en mängd information om WMS-servern, i allmänhet hämtad från den capabilities statement som returneras från den servern. Många definitioner kan hämtas genom att läsa WMS-standarderna (se OPEN-GEOSPATIAL-CONSORTIUM i *Litteratur- och webbreferenser*), men här är några praktiska definitioner:

- **Egenskaper för server**
  - **WMS Version** — Den WMS-version som stöds av servern.
  - **Image Formats** — Listan över MIME-typer som servern kan svara med när kartan ritas. QGIS stöder alla format som de underliggande Qt-biblioteken byggdes med, vilket vanligtvis är åtminstone `image/png` och `image/jpeg`.
  - **Identity Formats** — Listan över MIME-typer som servern kan svara med när du använder Identify-verktyget. För närvarande stöder QGIS typen `text-plain`.
- **Lagrets egenskaper**
  - **Vald** — Huruvida detta lager var valt eller inte när dess server lades till i detta projekt.
  - **Visible** — Huruvida detta lager är markerat som synligt i förklaringen (används ännu inte i den här versionen av QGIS).
  - **Can Identify** — Huruvida detta lager kommer att ge några resultat när Identify-verktyget används på det eller inte.
  - **Can be Transparent** — Huruvida detta lager kan återges med transparens eller inte. Denna version av QGIS kommer alltid att använda transparens om detta är `Yes` och bildkodningen stöder transparens.
  - **Can Zoom In** — Om detta lager kan zoomas in av servern eller inte. Denna version av QGIS förutsätter att alla WMS-lager har detta inställt på `Yes`. Lager som inte har det kan återges på ett konstigt sätt.
  - **Cascade Count** — WMS-servrar kan fungera som en proxy till andra WMS-servrar för att hämta rasterdata för ett lager. Denna post visar hur många gånger begäran för detta lager vidarebefordras till andra WMS-servrar för att få ett resultat.
  - **Fixed Width, Fixed Height** — Om detta lager har fasta pixeldimensioner för källan eller inte. Denna version av QGIS antar att alla WMS-lager har detta inställt till ingenting. Lager som inte har det kan återges på ett konstigt sätt.
  - **WGS 84 Bounding Box** — Lagrets avgränsande box, i WGS 84-koordinater. Vissa WMS-servrar ställer inte in detta korrekt (t.ex. används UTM-koordinater i stället). Om detta är fallet kan den första vyn av detta lager återges med ett mycket ”utzoomat” utseende av QGIS. WMS webmaster bör informeras om detta fel, som de kanske känner till som WMS XML-elementen `LatLonBoundingBox`, `EX_GeographicBoundingBox` eller CRS:84 `BoundingBox`.
  - **Tillgänglig i CRS** — De projektioner som detta lager kan återges i av WMS-servern. Dessa listas i det WMS-nativa formatet.
  - **Available in style** — De bildstilar som detta lager kan återges i av WMS-servern.

Temporala egenskaper

Raster *temporala egenskaper* (nämligen *Dynamic Temporal Control*) kan ställas in för WMS- och WMTS-lager. När ett WMS- eller WMTS-lager med tidsdimension läggs till i projektet visas det som standard i panelen *Layers* med ikonen Temporal Layer bredvid. Dess *Temporal*-egenskaper är som standard inställda på *Automatic* temporalt läge, vilket innebär att lagret som standard följer den temporala styrenhetens aktuella tidsintervall.

Du kan sedan välja att visa ett specifikt statiskt tidsvärde för lagret genom att avmarkera *Dynamic Temporal Control* och välja ett alternativ under *Static WMS-T Temporal Range*:

- *Serverstandard*
- *Predefined date* med en server som exponerar data för icke sammanhängande tidsintervall eller *Predefined range* med en server som exponerar ett intervall av tillgängliga datum. En *Start date* och *End date* är nödvändiga i det senare fallet. Deras förväntade formatering kan härledas från alternativet referenstid (se nedan). beroende på om leverantören har data för en sammanhängande period eller inte
- *Följ projektets tidsintervall* enligt definitionen i projektets egenskapsdialog

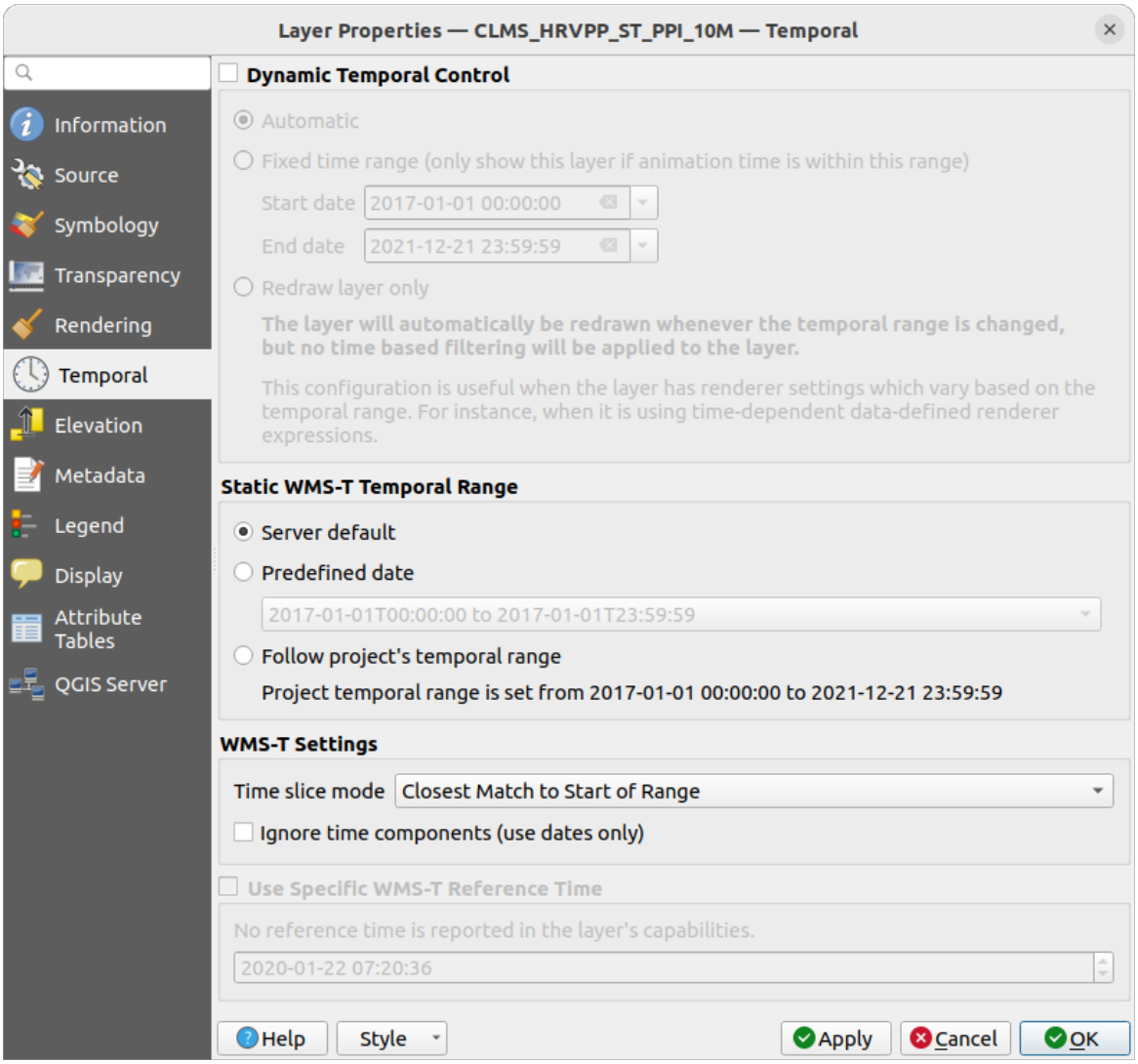


Fig. 19.4: Temporala egenskaper hos ett WMTS-lager

Oavsett vilken kontroll av tidsdata som används finns det några *WMS-T Inställningar* som hjälper till att visa rätt tidsdata:

- *Time slice mode* vilket kan vara:
  - *Utnyttja hela tidsintervallet*
  - *Matcha med början av intervallet*
  - *Matchning till slutet av intervallet*
  - *Närmaste matchning till början av intervallet*
  - *Närmaste matchning till slutet av intervallet*
- *Ignorera tidskomponenter (använd endast datum)*: Om detta markeras kommer tidskomponenten i temporala frågor att ignoreras och endast datumkomponenten kommer att användas i serverförfrågningar.

Du kan också *Use Specific WMS-T Reference Time* plocka från tider som rapporteras i lagrets funktioner. Praktiskt för servrar som exponerar en icke sammanhängande uppsättning av datumtidsinstanser (istället för ett intervall av datum).

Egenskaper för QGIS Server

Förutom rasterlagrets *properties* visas följande alternativ när du publicerar ett WMS/WMTS-lager med QGIS Server:

- *WMS-utskriftslager*: Gör det möjligt att ange ett alternativt WMS-lager som ska användas för utskrift (i svaret GetProjectSettings). Lämpligt för WMTS-lager som i allmänhet inte är lämpliga för utskrift.
- *Publicera WMS/WMTS datakälla uri*: Tillåter en webbklient att hämta WMS/WMTS-data direkt
- *Annonser som bakgrundslager*

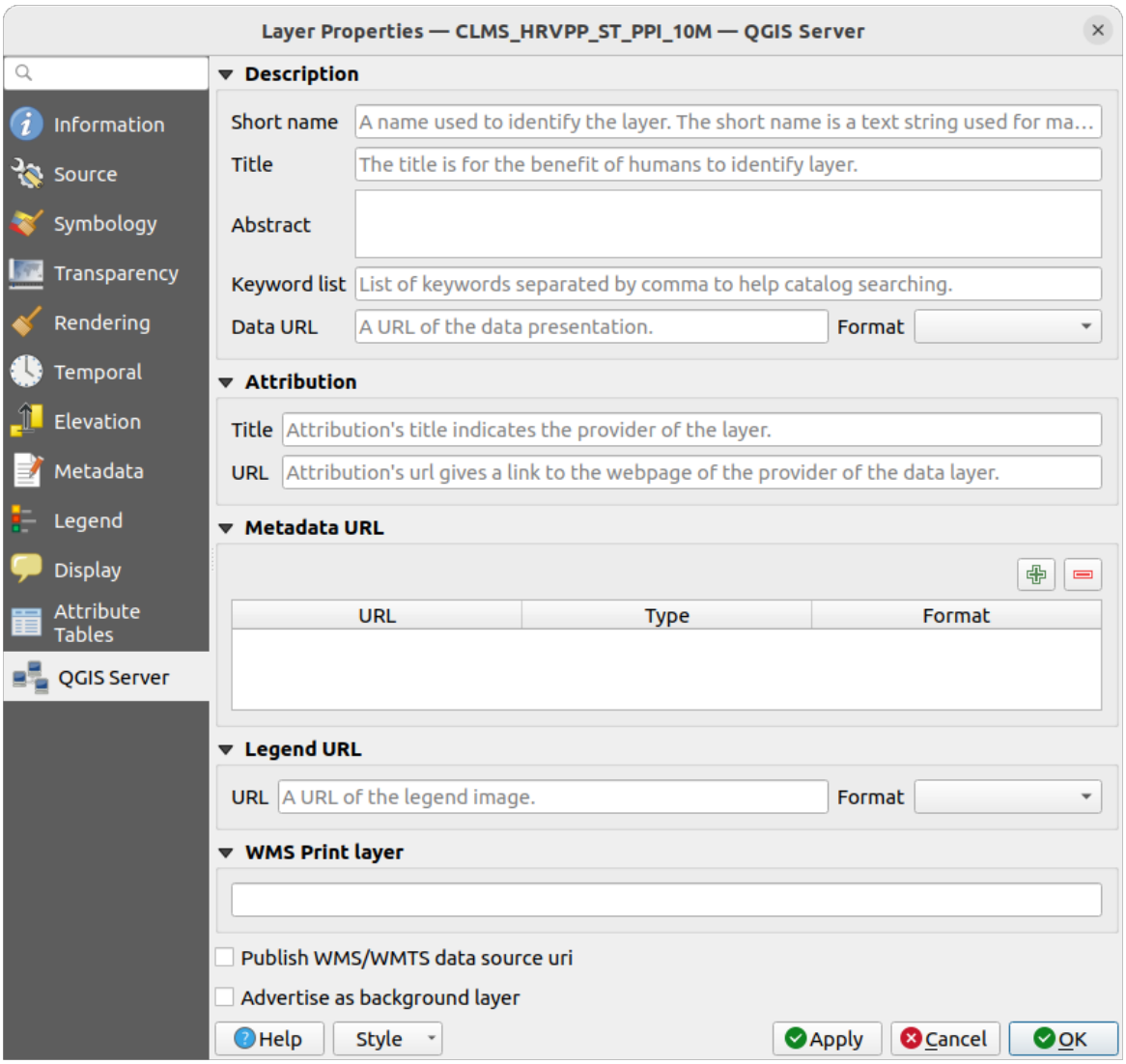


Fig. 19.5: QGIS Server-egenskaper för ett WMS/WMTS-lager

19.1.8 Visa WMS legendgrafik i innehållsförteckningen och i layouten

QGIS WMS-dataleverantören kan visa en grafisk förklaring i innehållsförteckningens lagerlista och i utskriftslayouten. WMS-förklaringen visas endast om WMS-servern har GetLegendGraphic-kapacitet och lagret har getCapability url specificerad, så du måste dessutom välja en styling för lagret.

Om en legendGraphic är tillgänglig visas den under lagret. Den är liten och du måste klicka på den för att öppna den i verklig storlek (på grund av QgsLegendInterface arkitektoniska begränsning). Om du klickar på lagrets förklaring öppnas en ram med förklaringen i full upplösning.

I utskriftslayouten kommer teckenförklaringen att integreras i sin ursprungliga (nedladdade) dimension. Upplösningen på legendgrafiken kan ställas in i artikelegenskaperna under *Legend* ► *WMS LegendGraphic* för att matcha dina utskriftskrav.

Legenden kommer att visa kontextuell information baserat på din aktuella skala. WMS-förklaringen visas endast om WMS-servern har GetLegendGraphic-kapacitet och lagret har getCapability url specificerad, så du måste välja en styling.

19.2 WCS-klient

En Web Coverage Service (WCS) ger tillgång till rasterdata i former som är användbara för rendering på klientsidan, som indata i vetenskapliga modeller och för andra klienter. WCS kan jämföras med WFS och WMS. I likhet med WMS- och WFS-tjänstinstanser gör en WCS det möjligt för klienter att välja delar av en servers informationsinnehav baserat på spatiala begränsningar och andra frågekriterier.

QGIS har en inbyggd WCS-leverantör och stöder både version 1.0 och 1.1 (som skiljer sig avsevärt), men för närvarande föredrar den 1.0, eftersom 1.1 har många problem (dvs. varje server implementerar den på olika sätt med olika särdrag).

Den inbyggda WCS-providern hanterar alla nätverksförfrågningar och använder alla standardinställningar för QGIS-nätverk (särskilt proxy). Det är också möjligt att välja cache-läge (”always cache”, ”prefer cache”, ”prefer network”, ”always network”), och leverantören stöder också val av tidsposition, om tidsdomän erbjuds av servern.

Varning

Om du anger **användarnamn** och **lösenord** på fliken *Autentisering* kommer oskyddade uppgifter att sparas i anslutningskonfigurationen. Dessa **uppgifter kommer att vara synliga** om du t.ex. delar projektfilen med någon. Därför är det lämpligt att spara dina autentiseringsuppgifter i en *Autentiseringskonfiguration* istället (fliken *configurations*). Se *System för autentisering* för mer information.

19.3 WFS och WFS-T Klient

I QGIS beter sig ett WFS-lager ungefär som vilket annat vektorlager som helst. Du kan identifiera och välja funktioner och visa attributtabellen. QGIS stöder WFS 1.0.0, 1.1.0, 2.0 och OGC API - Features (OAPIF), inklusive redigering (genom WFS-T). QGIS stöder också nedladdning i bakgrunden och progressiv rendering, cachelagring på disk av nedladdade funktioner och automatisk versionsdetektering.

Lager av servrar som implementerar OGC API - Funktioner - Del 4: Skapa, ersätta, uppdatera och ta bort kan omvandlas till redigeringsläge för att göra det möjligt att skapa, ändra och ta bort funktioner. Observera att varje skapad/ändrad/raderad funktion kräver en dedikerad nätverksbegäran, så prestandan kan bli lidande vid samtidig ändring av hundratals eller fler funktioner åt gången.

Att lägga till ett WFS-lager är i allmänhet mycket likt det förfarande som används med WMS. Det finns inga standardservrar definierade, så du måste lägga till dina egna. Du kan hitta WFS-servrar genom att använda *MetaSearch plugin* eller din favoritsökmotor på webben. Det finns ett antal listor med offentliga webbadresser, vissa av dem underhålls och andra inte.

Laddning av en WFS Layer

Som ett exempel använder vi Gateway Geomatics WFS-server och visar ett lager.

```
https://demo.gatewaygeomatics.com/cgi-bin/wfs_gateway?REQUEST=GetCapabilities&
↪VERSION=1.0.0&SERVICE=WFS
```

För att kunna ladda ett WFS-lager måste du först skapa en anslutning till WFS-servern:

1. Öppna dialogrutan *Data Source Manager* genom att trycka på knappen Open Data Source Manager
2. Aktivera fliken *WFS / OGC API - Funktioner*
3. Klicka på *New…* för att öppna dialogrutan *Create a New WFS Connection*
4. Ange `Gateway Geomatics` som namn
5. Ange webbadressen (se ovan)

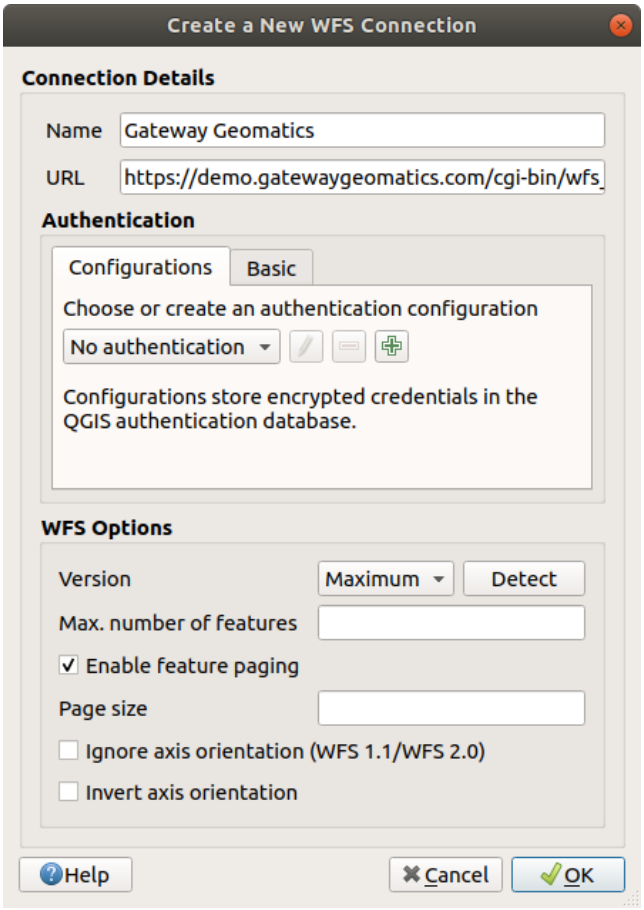


Fig. 19.6: Skapa en anslutning till en WFS-server

Observera

När det gäller ett OGC API - Features (OAPIF) ska den URL som ska anges vara landing page, dvs. den huvudsida från vilken det är möjligt att navigera till alla tillgängliga serviceendpoints.

6. I dialogrutan för WFS-inställningar kan du:
   - Anger serverns WFS-version. Om den är okänd, tryck på knappen *Detect* för att automatiskt hämta den.
   - Definiera det *maximala antalet funktioner* som hämtas i en enda GetFetFeature-begäran. Om den är tom sätts ingen gräns.
   - Och beroende på WFS-versionen, ange om du vill:
     - *Enable feature paging* och ange det maximala antalet funktioner som ska hämtas med *Page size*. Om ingen gräns definieras tillämpas serverns standard.
     - Tvinga till *Ignore axis orientation (WFS 1.1/WFS 2.0)*
     - *Inverterad axelorientering*.
     - *Använd GML2-kodning för transaktioner*.

Varning

Om du anger **användarnamn** och **lösenord** på fliken *Autentication* kommer oskyddade uppgifter att sparas i anslutningskonfigurationen. Dessa **uppgifter kommer att vara synliga** om du t.ex. delar projektfilen med

någon. Därför är det lämpligt att spara dina autentiseringsuppgifter i en *Autentication-konfiguration* istället (fliken *Configurations*). Se *System för autentisering* för mer information.

7. Tryck på *OK* för att skapa anslutningen.

Observera att eventuella proxyinställningar som du kan ha angett i dina inställningar också beaktas. Observera också att det är möjligt att *Load* anslutningsparametrarna från en `.XML`-fil eller *Save* dem till en `.XML`-fil.

Nu är vi redo att ladda WFS-lager från ovanstående anslutning.

1. Välj ”Gateway Geomatics” i rullgardinsmenyn *Server Connections*.
2. Klicka på *Connect*
3. Välj lagret *Parks* i listan
4. Du kan också välja om du vill:
   - *Använd titel för lagernamn*, visar lagrets titel enligt definitionen på servern i panelen *Lager* i stället för dess *Namn*
   - *Begära endast funktioner som överlappar vyns omfattning*
   - *Ändra…* lagrets CRS till något annat som stöds av tjänsten
   - eller skapa en fråga för att ange särskilda funktioner som ska hämtas från tjänsten: dubbelklicka på lagrets rad för att öppna dialogrutan *SQL Query Composer*. I den dialogen finns widgetar för att skriva en avancerad SQL-fråga som bygger på tillgängliga tabeller och kolumner i tjänsten, med sortering och filtrering och en mängd SQL-funktioner, spatiala predikat och operatorer.

     Den fråga du bygger kommer att visas efter validering i kolumnen *SQL* i tabellen *WFS / OGC API - Features*, och det filtrerade lagret kommer att visa ikonen bredvid det i panelen *Layers*. Det är således möjligt att justera frågan när som helst.
5. Klicka på *Add* för att lägga till lagret på kartan.

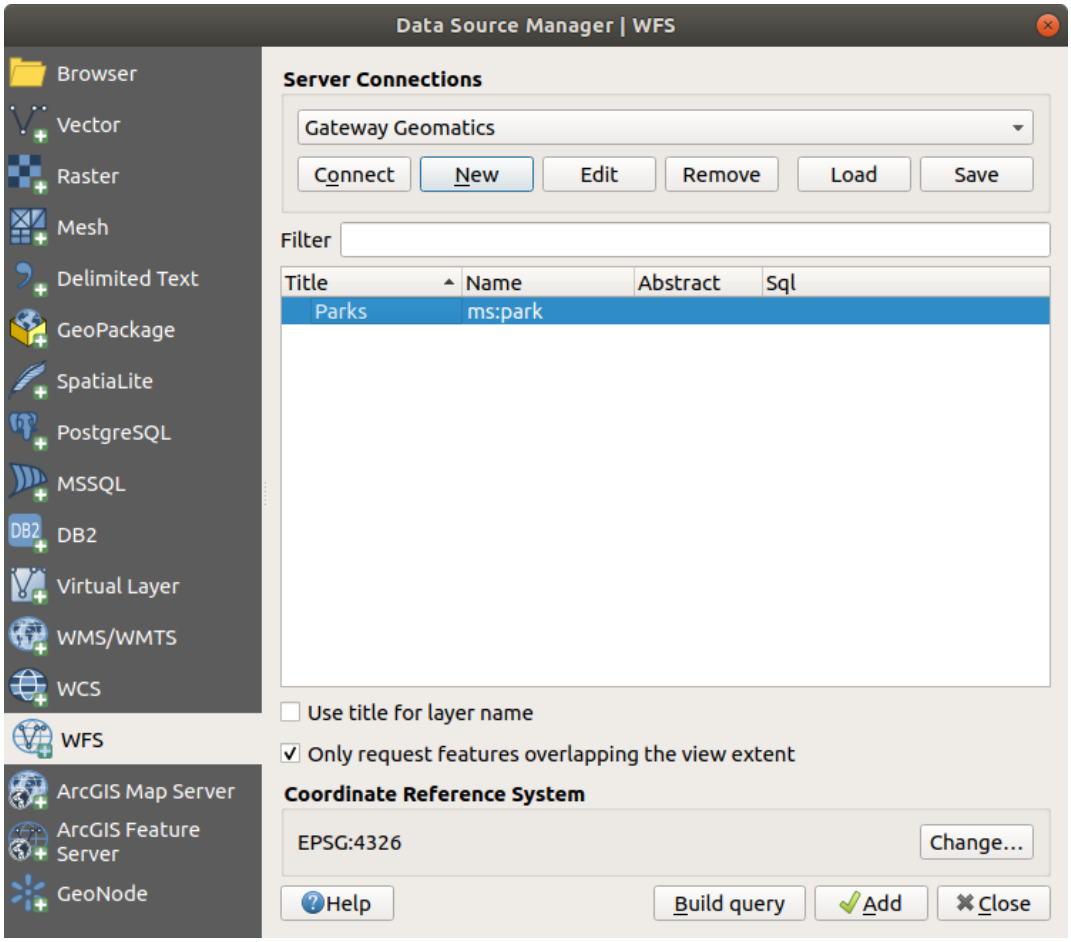


Fig. 19.7: Lägga till ett WFS-lager

Du kommer att märka att nedladdningsförloppet visualiseras längst ner till vänster i QGIS huvudfönster. När lagret har laddats kan du identifiera och välja ett par funktioner och visa attributtabellen.

19.4 SensorThings

QGIS stöder anslutningar till OGC SensorThings API, en standard som tillhandahåller ett öppet och enhetligt ramverk för att sammankoppla IoT-avkänningsenheter, data och applikationer över webben. Det är en öppen standard som behandlar syntaktisk och semantisk interoperabilitet för Internet of Things. Den baseras på datamodellen ”Observations and Measurements <https://www.ogc.org/publications/standard/om/>`_”, en standardiserad modell för observationer och för funktioner som ingår i provtagning vid observationer.

19.4.1 Inställning av anslutning

Lägg till SensorThings data i QGIS genom att använda fliken *SensorThings* i dialogrutan *Data Source Manager*.

För att upprätta en ny anslutning trycker du på *Ny* (eller *Ny SensorThings-anslutning* från bläddrarpanelen) och anger *Namn* och *URL*. Avancerade alternativ, t.ex. *authentication* och en *Referer*, kan också konfigureras.

Tryck på *OK* för att upprätta anslutningen. Sedan kommer du att kunna:

- *Ändra* inställningarna för SensorThings-anslutningen
- *Avlägsna* SensorThings-anslutningen

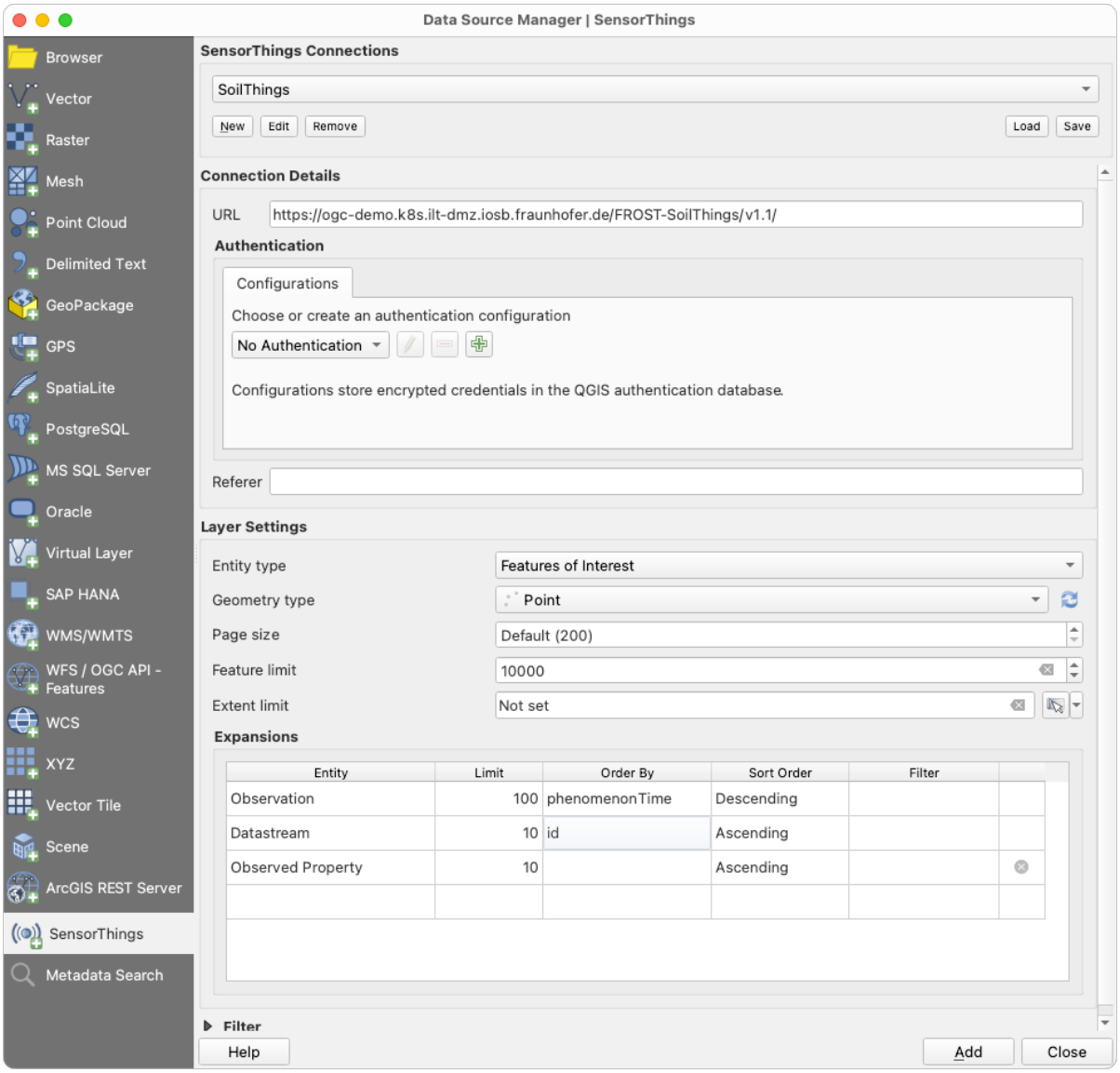


Fig. 19.8: SensorThings Anslutningsdialog

Konfigurationer kan sparas till en `.XML`-fil (*Save*) via *SensorThings*-posten i *Data Source Manager*-dialogrutan eller dess kontextuella meny i *Browser*-panelen (*Save Connections*). På samma sätt kan konfigurationer läggas till från en fil (*Load*).

19.4.2 Ladda SensorThings data

Relationer mellan lager (s.k. entiteter) som lagras i en SensorThings-datauppsättning uttrycks i diagrammet nedan.

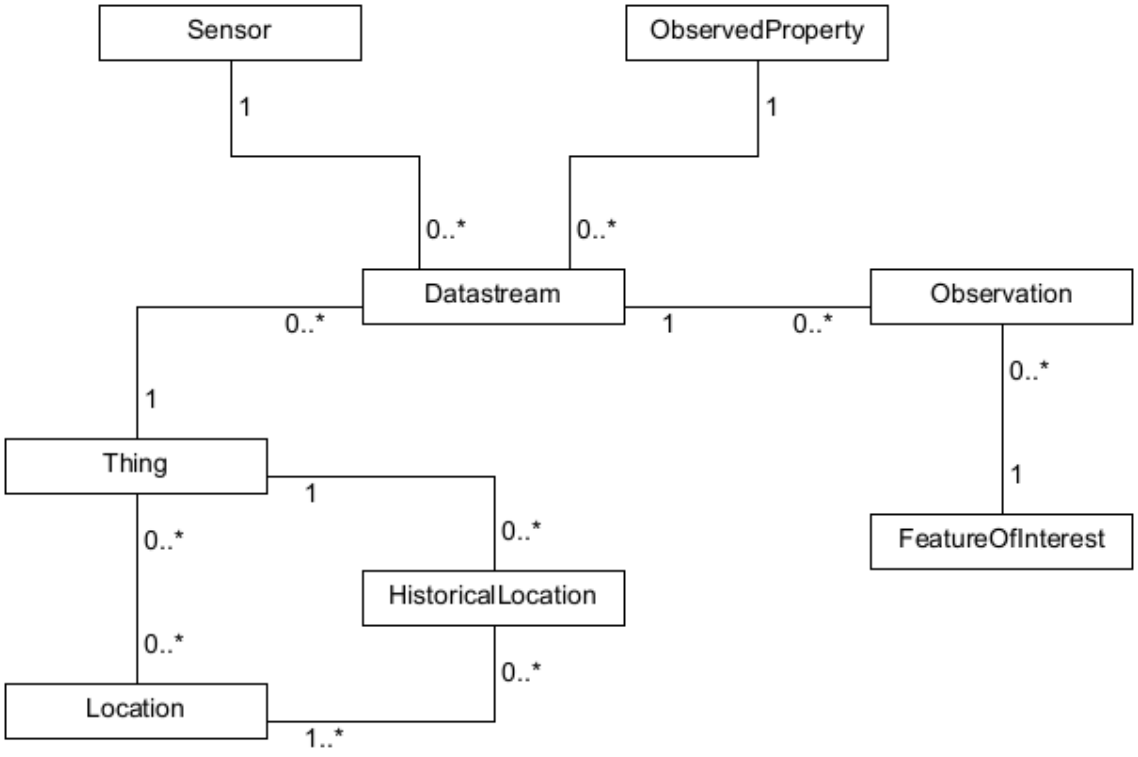


Fig. 19.9: Datamodell Observationer och mätningar (Källa: Sensor Web Tutorial by SIST network)

Alla typer av entiteter kan läsas in i QGIS, men alla är inte spatiala data. För att ladda en entitet finns det *Lagerinställningar* som kan konfigureras:

- *Entity Type*: den entitet som ska laddas från datamodellen som lager i QGIS
- *Geometry Type*: geometritypen för den valda entiteten som ska laddas. Tryck på Check available types för att begränsa listan till de geometrityper som faktiskt stöds.
- *Sidstorlek*
- *Feature Limit* anger ett maximalt antal funktioner som ska begäras från tjänsten
- *Extent Limit* anger en maximal omfattning för lagret, så att endast funktioner inom omfattningen begärs
- *Expansioner*: Datamodellen för SensorThings tillhandahåller en mekanism för utvidgning av resultaten till relaterade enheter, på samma sätt som tabeller sammanfogas i en relationsdatabas. Med hjälp av detta tillvägagångssätt kan du expandera det valda lagret till att omfatta data från andra objekt. Detta gör relationen plattare och skapar lika många överordnade funktioner som underordnade, och ytterligare egenskaper läggs till som kolumner i attributtabellen.

Använd *Filter* för att skapa en fråga för att filtrera data med hjälp av SensorThings filtersyntax.

Observera

Ovanstående inställningar och filtreringsalternativ är också tillgängliga för uppdatering i dialogrutan för lagrets egenskaper, fliken *Source*, när lagret har laddats i QGIS.

Tryck på *Add* för att ladda den valda enhetstypen som lager i QGIS.

19.4.3 Arbeta med ett vektorlager från SensorThings

Ett SensorThings-lager laddas i QGIS som ett vektorlager. Som sådant visar det samma flikar i *lagrets egenskaper* och tillåter samma funktionsinteraktioner med hjälp av urvals- eller identifieringsverktygen. Det finns dock några specifika egenskaper som du bör tänka på när du arbetar med SensorThings-data.

På grund av datamodellen är resultategenskapen för en SensorThings Observation ett strängfält. Om du vill använda dess numeriska representation, t.ex. i en graderad stil, använder du ett uttryck för att konvertera värdet till verkligt och try() om detta misslyckas (t.ex. `try( to_real("Observation_result"), Null)`).

Om du vill skapa ett diagram över observationerna på en eller flera platser kan du installera QGIS-tillägget *Data Plotly*.

1. Välj nu observationerna på en punkt i kartvyn.
2. Öppna plotly-panelen och aktivera kryssrutan *Use only selected features*.
3. Välj i x-kolumnen en egenskap för datum och tid och i y-kolumnen *Observation_result*. Detta kommer att rita upp observationerna på den platsen över tiden.
4. Verifiera för att filtrera efter en enda observerad egenskap.
5. Lägg märke till att diagrammet ändras så snart du väljer andra platser på kartan.

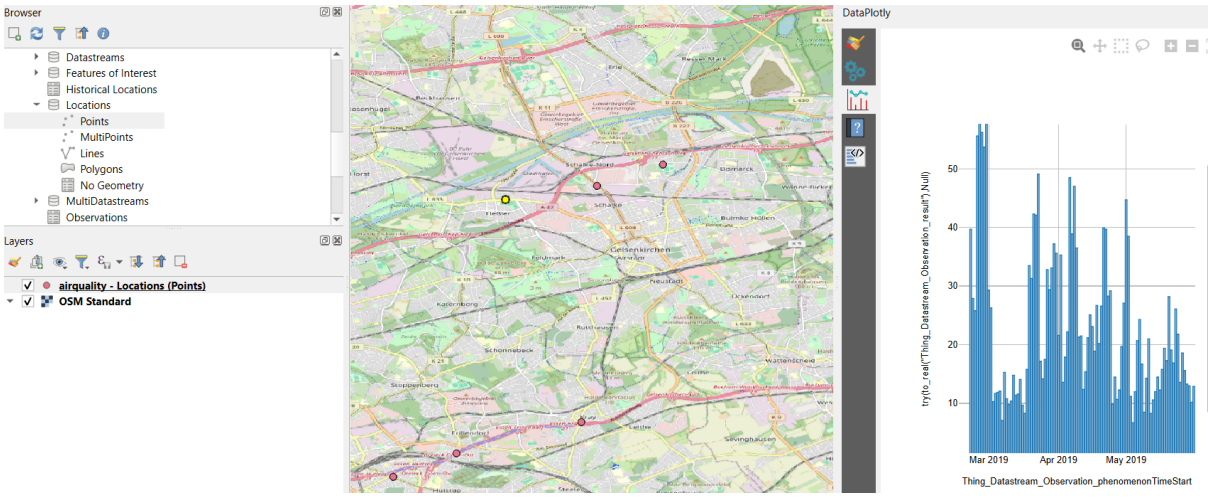


Fig. 19.10: Använd Data plotly för att plotta luftkvalitetsobservationerna på en plats

KAPITEL 20

Arbeta med GPS-data

20.1 Introduktion av GNSS/GPS-data

20.1.1 Vad är GPS?

GPS, **Global Positioning System**, är ett satellitbaserat system som gör det möjligt för alla med en GPS-mottagare att hitta sin exakta position var som helst i världen. GPS används som ett hjälpmedel vid navigering, t.ex. i flygplan, båtar och av vandrare. GPS-mottagaren använder signalerna från satelliterna för att beräkna latitud, longitud och (ibland) elevation. De flesta mottagare har också möjlighet att lagra:

- platser (så kallade **vägpunkter**)
- sekvenser av platser som utgör en planerad **rutt**
- och en **track**-logg över mottagarens förflyttning över tiden.

Waypoints, rutter och spår är de tre grundläggande funktionstyperna i GPS-data. QGIS visar waypoints i punktlager, medan rutter och spår visas i linjestringlager.

Observera

QGIS stöder även GNSS-mottagare. Men vi fortsätter att använda termen GPS i den här dokumentationen.

20.1.2 Överföra eller ladda GPS-data

Läsa in en GPX-fil

Det finns dussintals olika filformat för lagring av GPS-data. Det format som QGIS använder kallas GPX (GPS eXchange format), vilket är ett standardiserat utbytesformat som kan innehålla valfritt antal waypoints, rutter och spår i samma fil.

För att ladda en `GPX`-fil:

1. Öppna fliken *GPS* i dialogrutan *Data Source Manager*, d.v.s:
   - Klicka på knappen Open Data Source Manager i verktygsfältet (eller tryck på `Ctrl+L`) och aktivera målfliken
   - Eller välj *Lager* ► *Lägg till lager* ► *Lägg till GPX-lager…*
2. Använd knappen … Browse bredvid alternativet *GPX dataset* för att välja GPX-filen
3. Använd kryssrutorna för att välja de *Feature types* som du vill ladda från filen. Varje funktionstyp (*Waypoints*, *Tracks* eller *Routes*) laddas i ett separat lager.

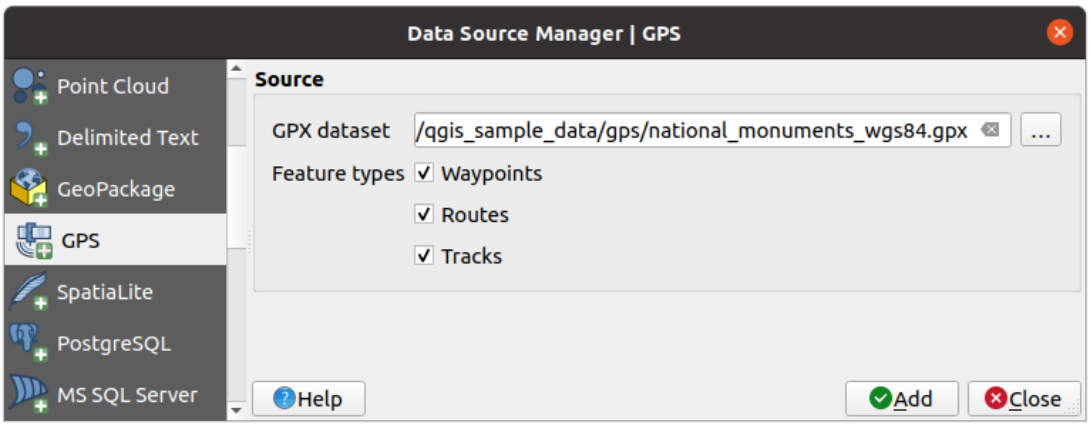


Fig. 20.1: Dialogrutan Ladda GPS-data

Laddning till eller från en enhet

Det finns många olika typer av GPS-enheter och format. Eftersom QGIS använder GPX-filer behöver du ett sätt att konvertera andra GPS-filformat till GPX. QGIS kan göra det med hjälp av gratisprogrammet GPSBabel. GPSBabel kan hjälpa dig att konvertera waypoints, spår och rutter mellan populära GPS-mottagare som Garmin eller Magellan och kartprogram som Google Earth eller Basecamp. Bokstavligen talat stöds hundratals GPS-mottagare och program. Du kan också överföra GPS-data mellan datorn och en GPS-enhet.

Under *Inställningar* ► *Alternativ* ► *GPS* ► *GPSBabel* kan du i QGIS definiera din egen enhetstyp och ange parametrar för konvertering som senare kan användas av *Processing GPS algorithms*.

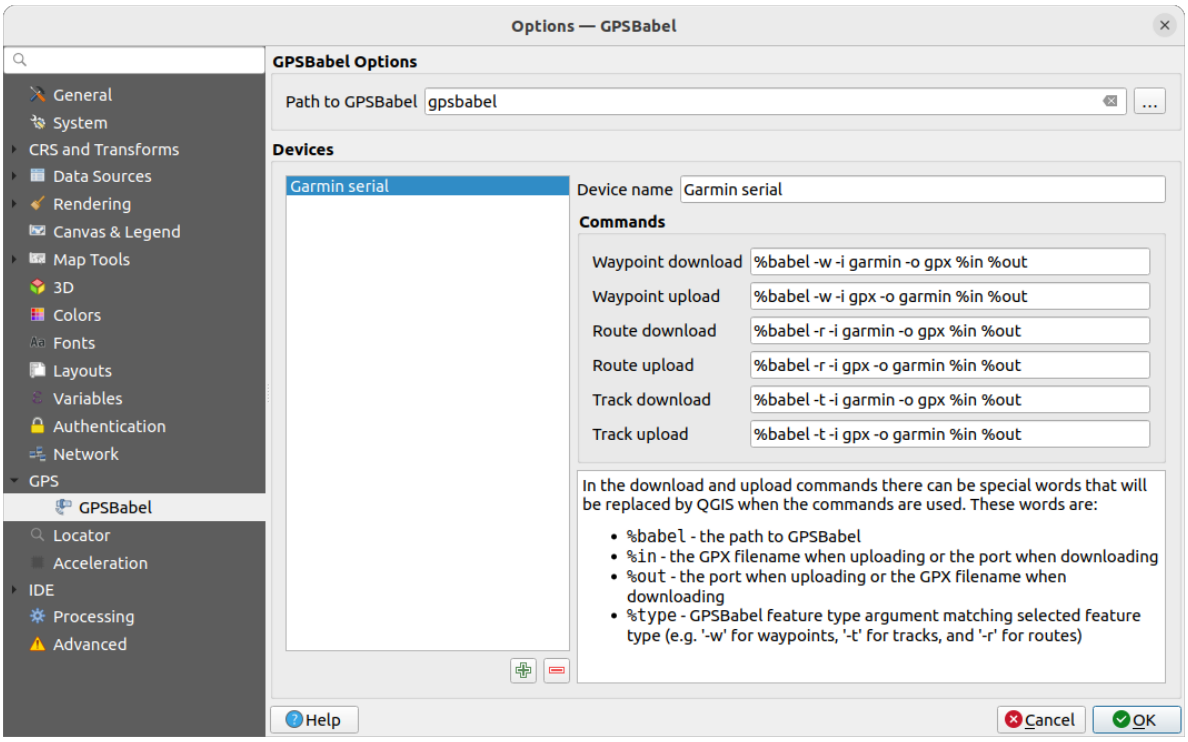


Fig. 20.2: GPS Babel-inställningar

1. Först måste du definiera *Sökväg till GPSBabel*-binärfiler.
2. Då kanske du vill lägga till din enhet. Du kan uppdatera enhetslistan med hjälp av knappen Add new device eller Remove device.
3. För varje enhet:
   - du anger ett *Enhetens namn*
   - konfigurerar du olika *Commands* som QGIS ska använda när det interagerar med det, t.ex:
     - *Waypoint nedladdning* från enheten
     - *Waypoint-uppladdning* till enheten
     - *Ruta nedladdning* från enheten
     - *Routera uppladdning* till enheten
     - *Track download* från enheten
     - *Spåruppladdning* till enheten

   Kommandona är vanligtvis GPSBabel-kommandon, men du kan också använda vilket annat kommandoradsprogram som helst som kan skapa en GPX-fil. QGIS kommer att ersätta nyckelorden `%type`, `%in` och `%out` när kommandot körs.

   Om du t.ex. skapar en enhetstyp med nedladdningskommandot `gpsbabel %type -i garmin -o gpx %in %out` och sedan använder den för att ladda ner waypoints från port `/dev/ttyS0` till filen `output.gpx`, kommer QGIS att ersätta nyckelorden och köra kommandot `gpsbabel -w -i garmin -o gpx /dev/ttyS0 output.gpx`.

   Läs GPSBabel-manualen för de kommandoradsalternativ som kan vara specifika för ditt användningsfall.

När du har skapat en ny enhetstyp visas den i enhetslistorna för GPS-algoritmerna för nedladdning och uppladdning.

Observera

Med GPS-enheter kan du lagra data i olika koordinatsystem. När du hämtar en GPX-fil (från din GPS-enhet eller en webbplats) och sedan laddar den i QGIS, se till att de data som lagras i GPX-filen använder WGS 84 (latitud/longitud). QGIS förväntar sig detta, och det är den officiella GPX-specifikationen. Se GPX 1.1 Schema Documentation.

20.2 GPS-spårning i realtid

QGIS kan hjälpa dig att göra fältkartläggning med en GPS-mottagare. En sådan live-spårning görs med hjälp av *GPS Toolbar*. Viss *enhetskonfiguration* kan behövas innan QGIS och GPS-mottagaren ansluts.

20.2.1 GPS-verktygsfält

Verktygsfältet *GPS* innehåller de viktigaste verktygen för att styra en live tracking-session. Det kan aktiveras från *View* ► *Toolbars* ► *GPS*. Det följer projektets, GPS:ens och det aktuella GPS-spårets tillstånd och aktiverar åtgärder endast när de är meningsfulla. När en enhet har upptäckts av QGIS kommer du att kunna interagera med den:

- Anslut till GPS
- Koppla från GPS
- Recentrera kartan på GPS-positionen: kartan kommer omedelbart att recentreras på den aktuella GPS-positionen oavsett *parametern för automatisk recentrering*.
- Ställ in destinationslager för GPS-digitaliserade funktioner: Som standard kommer QGIS att *Follow active layer* för digitalisering av funktioner, vilket innebär att GPS-digitaliseringsverktyg kommer att anpassa sig till det lager som valts i *Layers*-panelen och skapade funktioner kommer att lagras i det i enlighet med detta. Detta kan vara användbart i vissa sammanhang, men det innebär också att man måste vara noga med att interagera med andra lager för att undvika att funktioner oavsiktligt lagras i oönskade lager. Med det här alternativet kan du uttryckligen ange ett lager för datalagring under live tracking-sessionen, och du kan växla beroende på dina behov. Exponerade verktyg kommer också att anpassas till den valda lagertypen.

  GPS-destinationslagret görs automatiskt redigerbart när en funktion skapas, och användaren meddelas om detta.
- åtgärder för skapande av funktioner:
  - Lägg till vertex i GPS-spår med hjälp av aktuell GPS-position
  - Skapa en ny punktfunktion på aktuell GPS-position
  - Skapa en ny linjefunktion med hjälp av det aktuella GPS-spåret
  - Skapa en ny polygonfunktion med hjälp av det aktuella GPS-spåret
  - Återställ spår
- Visa GPS-information: öppnar panelen *GPS-information*
- en displaylåda för snabb åtkomst till vissa GPS-informationskomponenter som du kan behöva:
  - Visa plats
  - Visa höjd (Geoid)
  - Visa höjd (WGS 84 ellipsoid)

  - ☐ Visa markhastighet
  - ☐ Visa bäring
  - ☐ Visa total spårlängd
  - ☐ Visa avstånd från spårets början

- Knappen Settings öppnar en rullgardinsmeny med vanliga inställningar som förväntas ändras mitt under sessionen:
  - ☐ *Visa platsmarkör*
  - ☐ *Visa bäringslinje*
  - ☐ *Rotera kartan för att matcha GPS-riktningen*
  - Alternativ för att styra kartans återställning:
    - ○ *Alltid centrera om kartan* när GPS-positionen är förskjuten från kartans mitt med ett visst avstånd (som en kvot av kartans utbredning)
    - ◉ *Återställ kartan när du lämnar omfattningen*
    - ○ *Aldrig återgå till centrum*
  - ☐ *Automatiskt lägga till spårkorsningar* när nya platser tas emot från GPS-enheten.
  - ☐ *Automatiskt spara tillagda funktioner*: funktioner som skapas från GPS-positioner kopplas omedelbart till sina mållager (hoppar över den vanliga lagerredigeringsbufferten)
  - *Tidsstämpel destination* för att justera fältet för tidsbestämd lagring
  - ☐ *Logga in på GeoPackage/SpatiaLite…*: När funktionen är aktiverad uppmanas användaren att välja en befintlig GeoPackage/SpatiaLite-fil eller ange ett nytt filnamn. I filen skapas tabellerna `gps_points` och `gps_tracks` med en fördefinierad struktur.

    Alla inkommande GPS-meddelanden loggas i lagret `gps_points` tillsammans med information om hastighet, bäring, höjd och noggrannhet från GPS:en.

    När GPS:en kopplas bort (eller QGIS stängs) läggs hela det inspelade GPS-spåret till i tabellen `gps_tracks` (tillsammans med viss beräknad information som spårlängd, start- och sluttider).
  - ☐ *Logga NMEA-meningar…*: aktiverar loggning av alla råa NMEA-strängar från enheten till en textfil
  - *GPS-inställningar…* för att komma till GPS *globala alternativ*-dialogrutan

Tips

Live status bar information om position

När en GPS-enhet är ansluten och användaren flyttar markören över kartbilden, visas ett statusfält som visar avstånd och bäring från markören till GPS-positionen. Projektets inställningar för avstånd och bäring respekteras i den här visningen. På en pekskärmsenhet kan du använda en tryck-och-håll-händelse för att utlösa statusfältmeddelandet i realtid.

20.2.2 GPS-informationspanel

För fullständig övervakning av en live GPS-spårning i QGIS kan du behöva aktivera *GPS Information Panel* (*Visa* ► *Paneler* ► eller tryck på `Ctrl+0`).

I det övre högra hörnet av *GPS Information Panel* trycker du på *Connect* för att initiera anslutningen mellan QGIS och en inkopplad GPS-mottagare, eller på *Disconnect*.

Längst upp till vänster på panelen finns följande knappar tillgängliga:

- Position: live-detaljer om GPS-position och sensorer
- Signal: signalstyrka för satellitanslutningar
- Inställningar: rullgardinsmeny till *alternativ för live-spårning* som du kan behöva ändra under en session

Position och ytterligare attribut

På fliken Position, om GPS:en tar emot signaler från satelliter, visas din position i latitud, longitud och höjd tillsammans med ytterligare attribut.

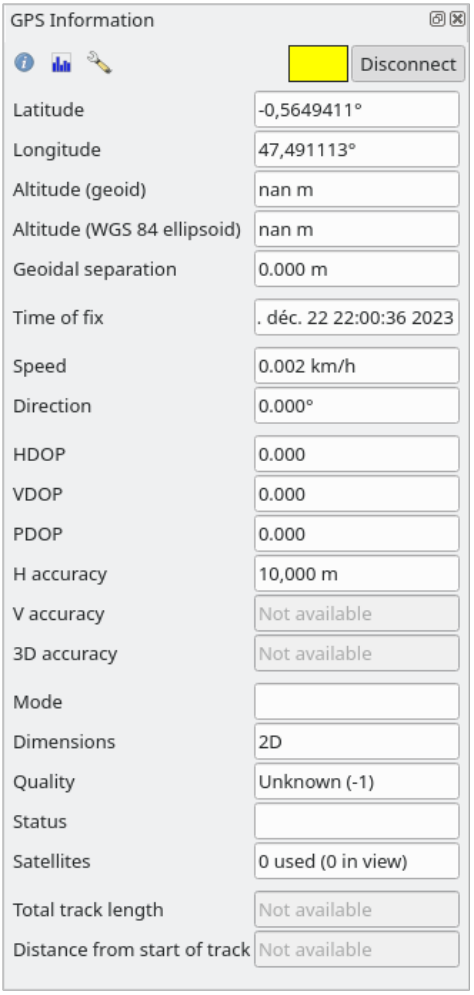


Fig. 20.3: GPS-spårning av position och ytterligare attribut

- *Latitud*
- *Longitud*

- *Altitud (Geoid)*: Altitud/höjd över eller under havets medelvattennivå
- *Altitud (WGS 84 ellipsoid)*: Altitud/höjd över eller under WGS-84 jordellipsoiden
- *Geoidal separation*: Skillnaden mellan WGS-84 jordellipsoiden och havets medelnivå (geoiden), − betyder att havets medelnivå ligger under ellipsoiden
- *Tidpunkt för fixering*
- *Hastighet*: Hastighet på marken
- *Riktning*: Bäring mätt i grader medurs från nordlig riktning i förhållande till färdriktningen
- *HDOP*: Horisontell utspädning av precision
- *VDOP*: Vertikal utspädning av precision
- *PDOP*: Utspädning av precision
- *H noggrannhet*: Horisontell noggrannhet i meter
- *V noggrannhet*: Vertikal noggrannhet i meter
- *3D-noggrannhet*: 3D Root Mean Square (RMS) i meter
- *Mode*: GPS-mottagarens konfiguration 2D/3D-läge, kan vara `automatisk` eller `manuell`
- *Dimensioner*: Dimension för positionsfixering, kan vara `2D`, `3D` eller `No fix`
- *Kvalitet*: Kvalitetsindikator för positionering
- *Status*: Status för positionsfixering, kan vara `Valid` eller `Invalid`
- *Satelliter*: Antal satelliter som använts för att erhålla fixen
- *Total spårlängd*: Totalt avstånd för aktuellt GPS-spår
- *Avstånd från början av spåret*: Direkt avstånd från första toppunkten i aktuellt GPS-spår till senaste toppunkten

Signal

Med fliken Signal kan du se signalstyrkan för de satelliter som du tar emot signaler från.

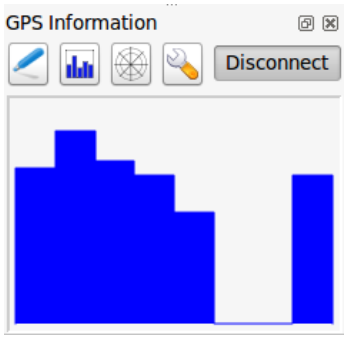


Fig. 20.4: Signalstyrka för GPS-spårning

20.2.3 Anslut till en Bluetooth GPS för spårning i realtid

Med QGIS kan du ansluta en Bluetooth GPS för insamling av fältdata. För att utföra den här uppgiften behöver du en Bluetooth GPS-enhet och en Bluetooth-mottagare på din dator.

Först måste du låta din GPS-enhet identifieras och kopplas ihop med datorn. Slå på GPS:en, gå till Bluetooth-ikonen i meddelandefältet och sök efter en ny enhet.

Se till att alla enheter är valda på höger sida av Device selection mask, så att din GPS-enhet förmodligen visas bland de tillgängliga enheterna. I nästa steg bör en seriell anslutningstjänst finnas tillgänglig, välj den och klicka på *Configure*-knappen.

Kom ihåg numret på den COM-port som tilldelats GPS-anslutningen enligt Bluetooth-egenskaperna.

Efter att GPS:en har identifierats, gör du parningen för anslutningen. Vanligtvis är auktoriseringskoden `0000`.

Öppna nu panelen *GPS information* och växla till skärmen GPS options. Välj den COM-port som tilldelats GPS-anslutningen och klicka på *Connect*. Efter en stund bör en markör som anger din position visas.

Om QGIS inte kan ta emot GPS-data bör du starta om GPS-enheten, vänta 5-10 sekunder och sedan försöka ansluta igen. Vanligtvis fungerar den här lösningen. Om du får ett anslutningsfel igen ska du kontrollera att du inte har en annan Bluetooth-mottagare i närheten som är parkopplad med samma GPS-enhet.

20.2.4 Exempel på anslutning av GPS-enheter i QGIS

Använda GPSMAP 60cs

MS Windows

Det enklaste sättet att få det att fungera är att använda ett mellanprogram (freeware, inte öppet) som heter GPSGate.

Starta programmet, låt det söka efter GPS-enheter (fungerar för både USB- och BT-enheter) och sedan i QGIS:

1. I *Inställningar* ► *Alternativ* ► *GPS*, upptäck den inkopplade enheten. Du kan använda läget *Autodetect*.
2. Tryck på *Connect* i panelen *GPS Information*

Ubuntu/Mint GNU/Linux

När det gäller Windows är det enklaste sättet att använda en server i mitten, i det här fallet GPSD, så

1. Installera programmet

```
sudo apt install gpsd
```

2. Ladda sedan kärnmodulen `garmin_gps`

```
sudo modprobe garmin_gps
```

3. Anslut sedan enheten och kontrollera med `dmesg` den faktiska enhet som används av enheten, till exempel `/dev/ttyUSB0`.
4. Nu kan du starta gpsd

```
gpsd /dev/ttyUSB0
```

5. Och slutligen ansluta till QGIS live tracking-verktyg.

Använda BTGP-38KM datalogger (endast Bluetooth)

Det är enkelt att använda GPSD (under Linux) eller GPSGate (under Windows).

Använder BlueMax GPS-4044 datalogger (både BT och USB)

MS Windows

Live-spårningen fungerar för både USB- och BT-lägen, genom att använda GPSGate eller till och med utan det, använd bara läget *Autodetect* eller rikta verktyget mot rätt port.

Ubuntu/Mint GNU/Linux

För USB

Live tracking fungerar både med GPSD

```
gpsd /dev/ttyACM3
```

eller utan, genom att ansluta QGIS live tracking tool direkt till enheten (t.ex. `/dev/ttyACM3`).

För Bluetooth

Live tracking fungerar både med GPSD

```
gpsd /dev/rfcomm0
```

eller utan, genom att ansluta QGIS live tracking tool direkt till enheten (t.ex. `/dev/rfcomm0`).

KAPITEL 21

System för autentisering

21.1 Översikt över autentiseringssystem

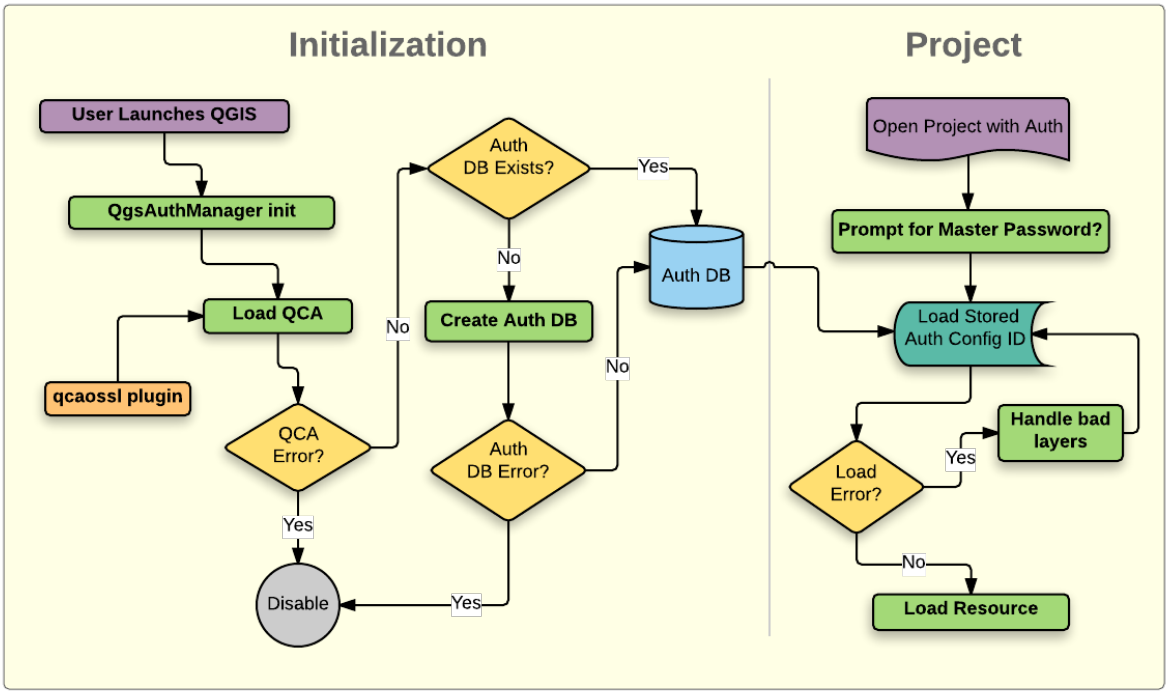


Fig. 21.1: Autentiseringssystemets anatomi

21.1.1 Autentiseringsdatabas

Autentiseringssystemet lagrar autentiseringskonfigurationer i en SQLite-databasfil som finns, som standard, på `<profile directory>/qgis-auth.db`.

Denna autentiseringsdatabas kan flyttas mellan QGIS-installationer utan att påverka andra aktuella QGIS-användarinställningar, eftersom den är helt skild från normala QGIS-inställningar. Ett konfigurations-ID (en slumpmässig alfanumerisk sträng med 7 tecken) genereras när en konfiguration initialt lagras i databasen. Detta representerar konfigurationen, vilket gör att ID:t kan lagras i programkomponenter i klartext (t.ex. projekt-, plugin- eller inställningsfiler) utan att de tillhörande autentiseringsuppgifterna avslöjas.

Observera

Den överordnade katalogen för *qgis-auth.db* kan ställas in med hjälp av följande miljövariabel, `QGIS_AUTH_DB_DIR_PATH`, eller ställas in på kommandoraden under start med alternativet `--authdbdirectory`.

21.1.2 Anpassade autentiseringsdatabaser

QGIS kan konfigureras att använda en anpassad autentiseringsdatabas istället för den ovan nämnda standard SQLite-databasen: alla databaser som stöds av Qt SQL-modulen kan användas (t.ex. PostgreSQL, MySQL, etc.), förutsatt att motsvarande Qt SQL-drivrutin finns tillgänglig i systemet.

Detta kan vara användbart i scenarier där en användare vill dela samma autentiseringsdatabas mellan flera QGIS-installationer, eller när en användare vill använda en annan autentiseringsdatabas än standard SQLite eller när en centraliserad autentiseringsdatabas används av QGIS-servern.

Det enda sättet att konfigurera en anpassad autentiseringsdatabas är genom att ställa in miljövariabeln `QGIS_AUTH_DB_URI` till URI för anslutningen, URI är i form av `driver://username:password@hostname:port/database?options`.

Var:

- `driver` är namnet på Qt SQL-drivrutinen som ska användas, t.ex. `QPSQL` för PostgreSQL, `QMYSQL` för MySQL, etc.
- `username` är det användarnamn som ska användas för att ansluta till databasen
- `password` är det lösenord som ska användas för att ansluta till databasen
- `hostname` är värdnamnet på databasservern
- `port` är porten till databasservern
- `database` är namnet på den databas som ska användas
- `options` är alternativen som ska skickas till drivrutinen, t.ex. `sslmode = require` för PostgreSQL

Observera

Schemat kan anges i URI-alternativen, t.ex. `QPSQL://användarnamn:lösenord@hostname:port/database?schema=schema_name`

Databasen måste finnas innan QGIS startas och användaren måste ha de nödvändiga behörigheterna för att ansluta till databasen och för att skapa de nödvändiga tabellerna om de inte redan finns.

Varning

Lösenordet i URI:n lagras i klartext i miljövariabeln, så vi rekommenderar att du använder en användare utan lösenord eller en användare med begränsade behörigheter för att ansluta till databasen.

Varning

Alla databaser som inte är baserade på SQLite anses vara skrivskyddade (detta kan ändras av Python-tillägg vid behov).

Detta är en avancerad funktion som är utformad för att QGIS ska kunna använda en eller flera anpassade autentiseringsdatabaser eller till och med anpassade Python-implementeringar av lagring av autentiseringsuppgifter.

Systemet är också utformat för att möjliggöra användning av flera autentiseringsdatabaser, men det finns för närvarande inget användargränssnitt för att hantera lagring av flera autentiseringsuppgifter, så användningen kräver manuell konfiguration och hantering, vanligtvis från ett Python-plugin.

21.1.3 Huvudlösenord

För att lagra eller få tillgång till känslig information i databasen måste användaren definiera ett ”huvudlösenord”. Ett nytt huvudlösenord begärs och verifieras när krypterade data första gången lagras i databasen. Vid åtkomst till känslig information uppmanas användaren att ange huvudlösenordet. Lösenordet cachelagras sedan under resten av sessionen (tills programmet avslutas), såvida inte användaren manuellt väljer en åtgärd för att rensa det cachelagrade värdet. Vid vissa tillfällen då autentiseringssystemet används krävs inte att huvudlösenordet anges, t.ex. när en befintlig autentiseringskonfiguration väljs eller när en konfiguration tillämpas på en serverkonfiguration (t.ex. när ett WMS-lager läggs till).

Du kan välja att spara lösenordet i datorns `Wallet/Keyring`.

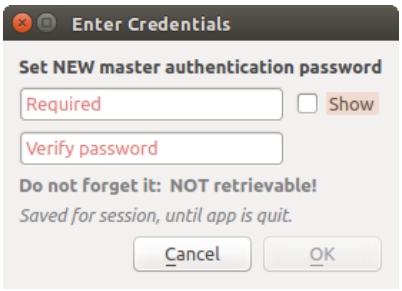


Fig. 21.2: Ange nytt huvudlösenord

Observera

En sökväg till en fil som innehåller huvudlösenordet kan anges med hjälp av följande miljövariabel, `QGIS_AUTH_PASSWORD_FILE`.

Hantering av huvudlösenordet

När huvudlösenordet har ställts in kan det återställas; det aktuella huvudlösenordet måste anges innan återställningen kan göras. Under denna process finns det möjlighet att generera en fullständig säkerhetskopia av den aktuella databasen.

Fig. 21.3: Återställning av huvudlösenord

Om användaren glömmer huvudlösenordet finns det inget sätt att hämta eller åsidosätta det. Det finns inte heller något sätt att hämta krypterad information utan att känna till huvudlösenordet.

Om en användare skriver in sitt befintliga lösenord felaktigt tre gånger kommer dialogrutan att erbjuda sig att radera databasen.

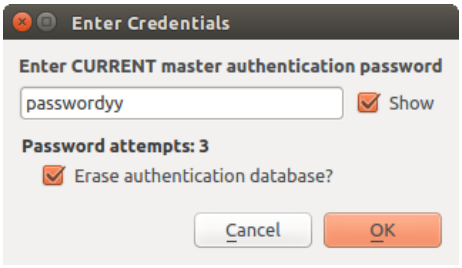


Fig. 21.4: Lösenordsuppmaning efter tre ogiltiga försök

21.1.4 Konfigurationer för autentisering

Du kan hantera autentiseringskonfigurationer från *Configurations* i *Autentisering*-fliken i QGIS Options-dialogrutan (*Inställningar* ► *Alternativ*).

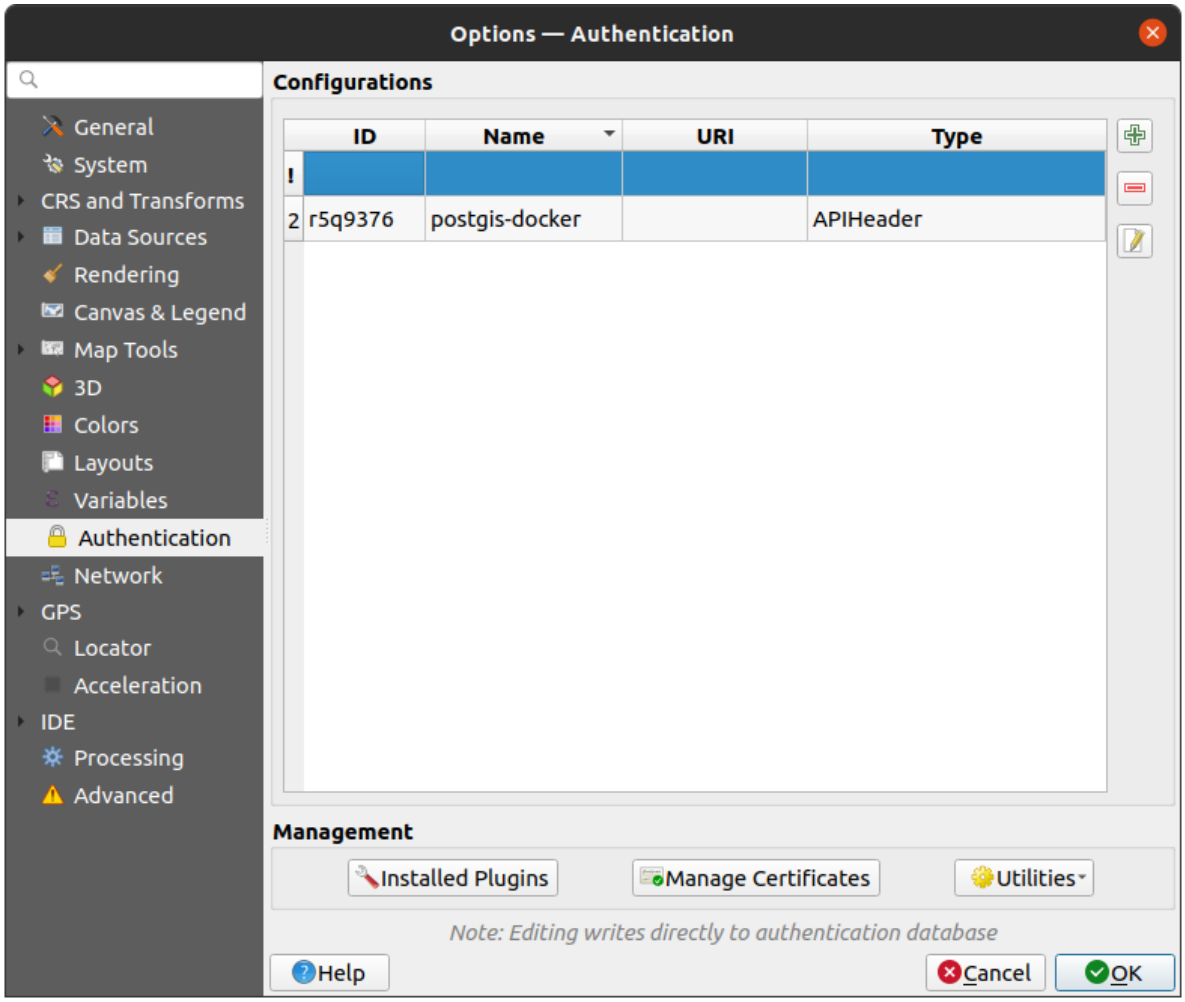


Fig. 21.5: Redaktör för konfigurationer

Använd knappen för att lägga till en ny konfiguration, knappen för att ta bort konfigurationer och knappen för att ändra befintliga konfigurationer.

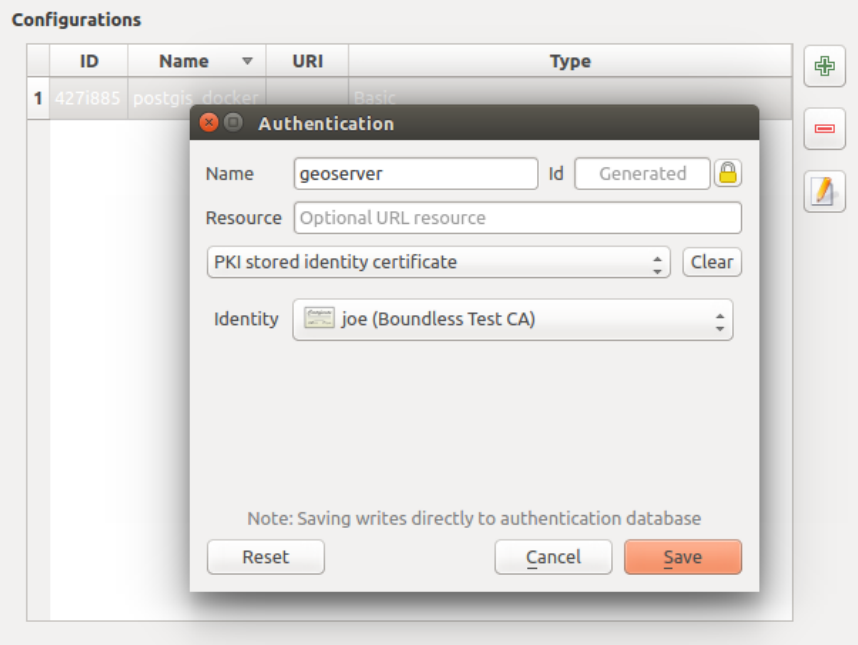


Fig. 21.6: Lägga till konfiguration från konfigurationsredigeraren

Samma typ av åtgärder för hantering av autentiseringskonfiguration (Lägg till, Redigera och Ta bort) kan göras när du konfigurerar en viss tjänsteanslutning, t.ex. konfigurering av en OWS-tjänsteanslutning. För detta ändamål finns det åtgärdsknappar i konfigurationsväljaren för fullständig hantering av konfigurationer som finns i autentiseringsdatabasen. I det här fallet behöver du inte gå till fliken *configurations* i *Authentication* i QGIS-alternativen om du inte behöver göra en mer omfattande konfigurationshantering.

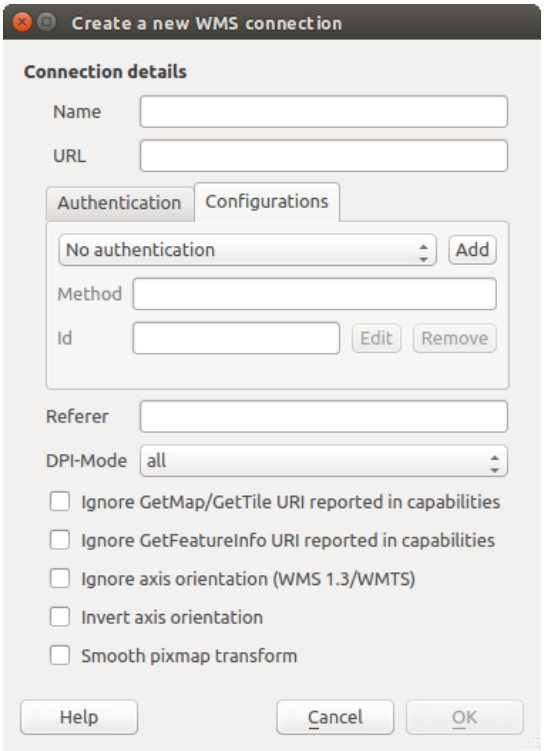


Fig. 21.7: WMS-anslutningsdialog med knapparna *Add*, *Edit* och *Remove* för autentiseringskonfiguration

När du skapar eller redigerar en autentiseringskonfiguration krävs ett namn, en autentiseringsmetod och

eventuell annan information som autentiseringsmetoden kräver (se mer om tillgängliga autentiseringstyper i *Autentiseringsmetoder*).

21.1.5 Autentiseringsmetoder

Tillgängliga autentiseringar tillhandahålls av C++-tillägg på ungefär samma sätt som tillägg för dataleverantörer stöds av QGIS. Vilken autentiseringsmetod som kan väljas beror på vilken åtkomst som behövs för resursen/providern, t.ex. HTTP(S) eller databas, och om det finns stöd i både QGIS-koden och ett tillägg. Därför kan det hända att vissa tillägg för autentiseringsmetoder inte är tillämpliga överallt där en väljare för autentiseringskonfiguration visas. En lista över tillgängliga tillägg för autentiseringsmetoder och deras kompatibla resurser/providers kan nås genom att gå till *Inställningar* ► *Alternativ* och, på fliken *Autentisering*, klicka på knappen *Installed Tillägg*.

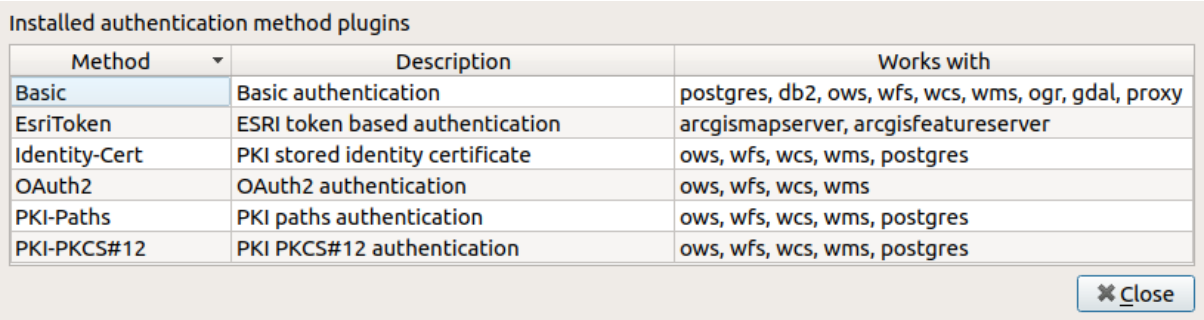


Fig. 21.8: Lista över tillgängliga metodplugins

Tillägg kan skapas för nya autentiseringsmetoder som inte kräver att QGIS kompileras om. Eftersom stödet för tillägg för närvarande endast är C++ måste QGIS startas om för att det nya tillägget ska bli tillgängligt för användaren. Se till att ditt tillägg är kompilerat mot samma målversion av QGIS om du tänker lägga till det i en befintlig målinstallation.

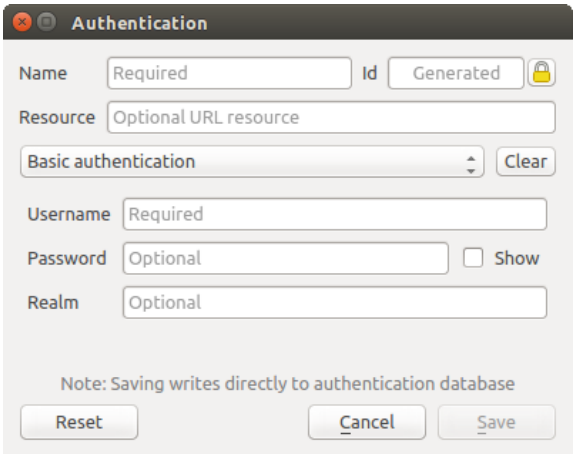


Fig. 21.9: Konfigurationer för grundläggande HTTP-autentisering

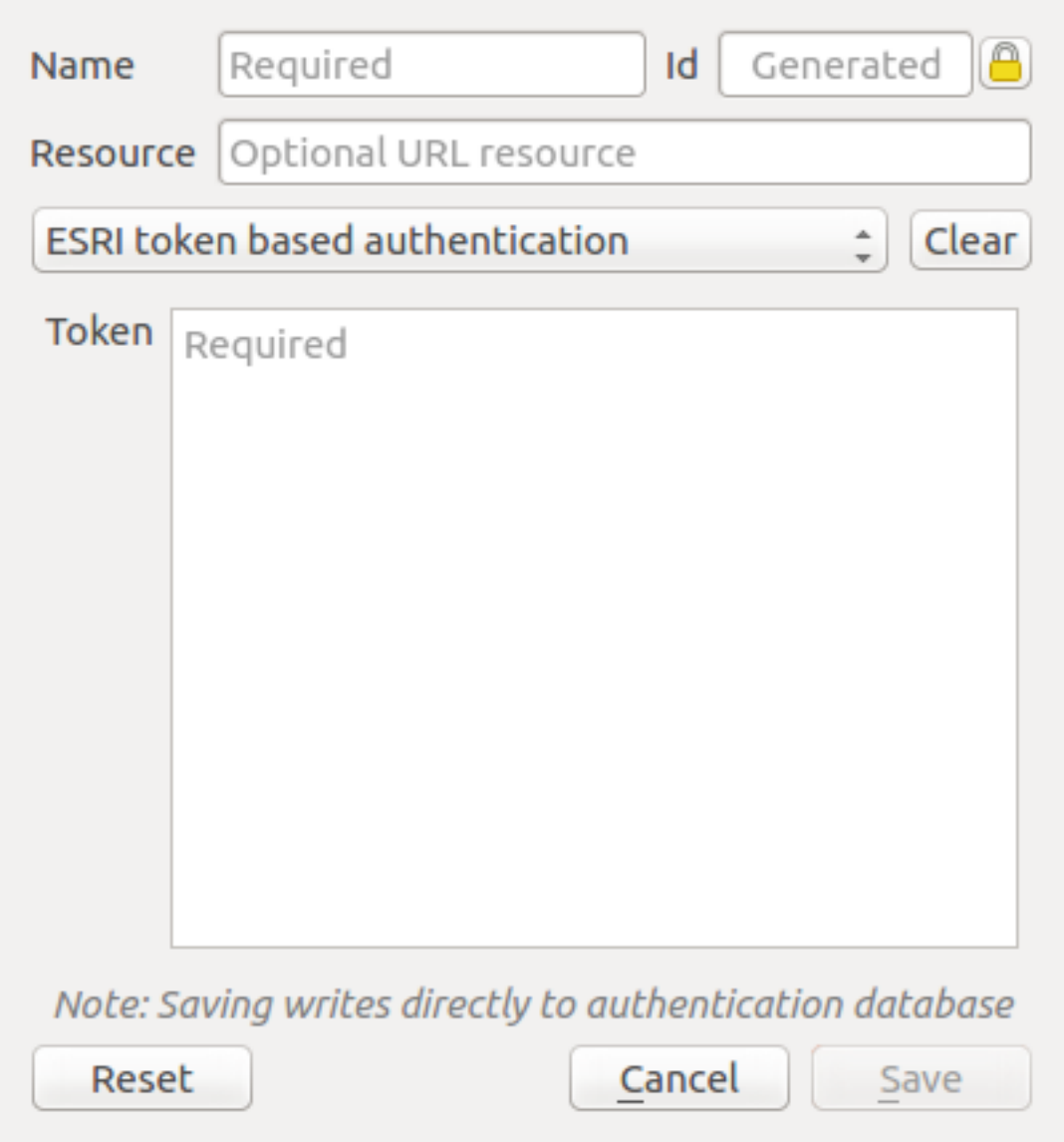


Fig. 21.10: Konfigurationer för ESRI Token-autentisering

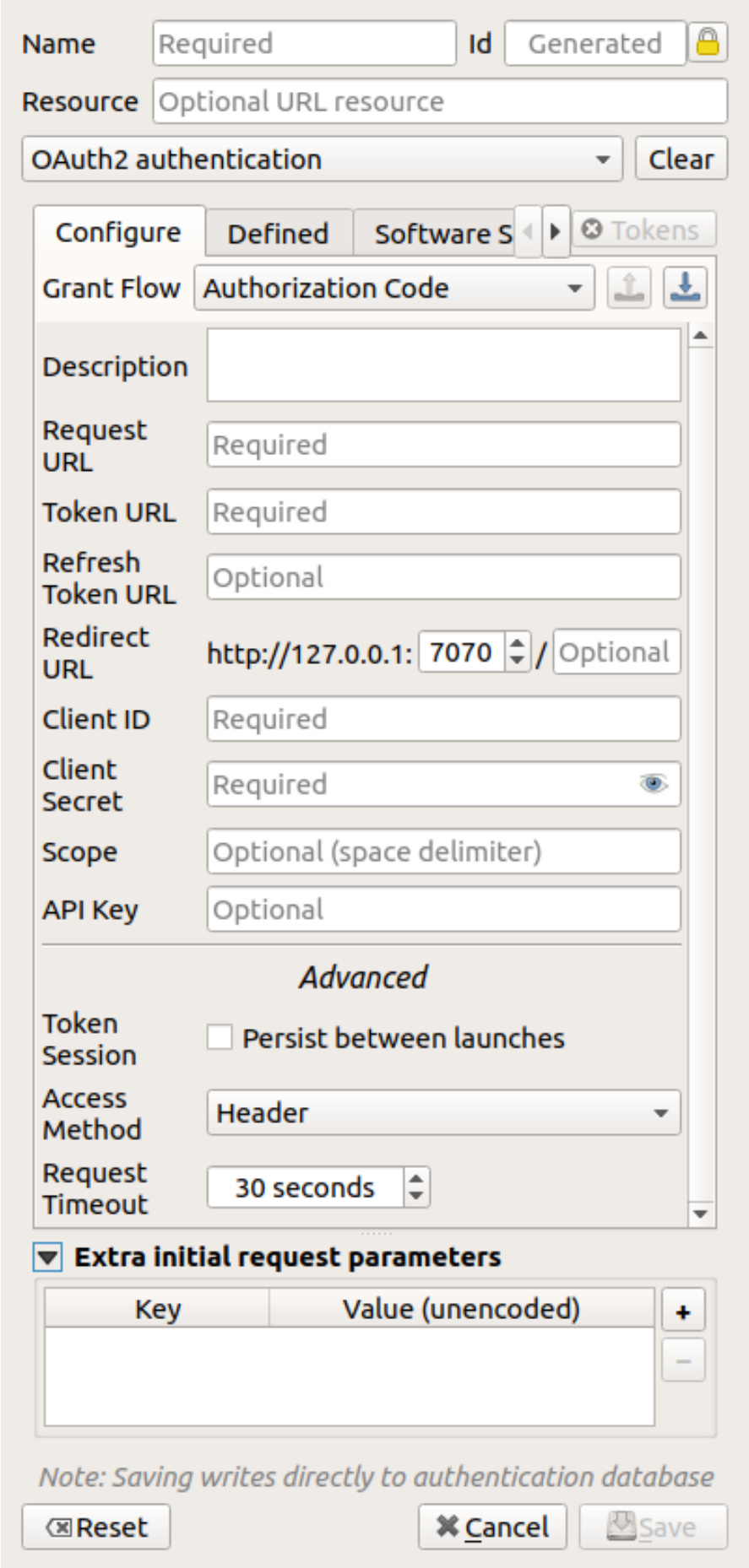


Fig. 21.11: Konfigurationer för OAuth2-autentisering

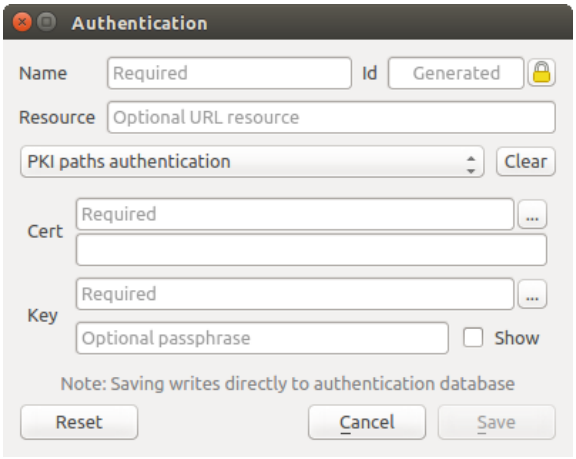


Fig. 21.12: PKI-sökvägar autentisering konfigurationer

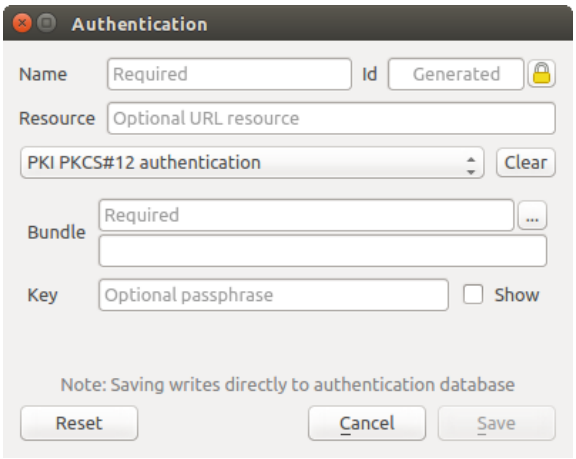


Fig. 21.13: PKI PKCS#12 filsökvägar autentisering konfigurationer

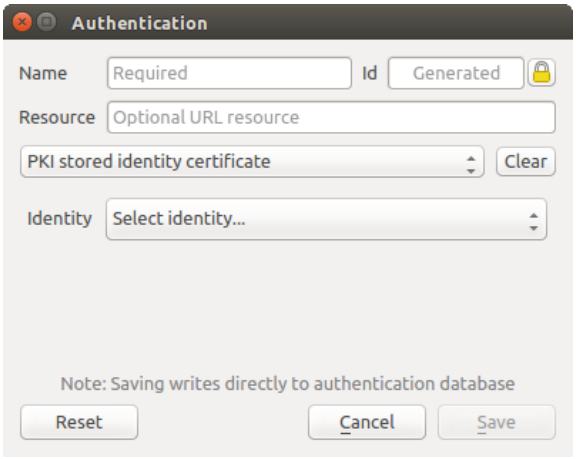


Fig. 21.14: Lagrade konfigurationer för identitetsautentisering

Observera

Resource URL är för närvarande en *ej implementerad* funktion som så småningom kommer att göra det möjligt att automatiskt välja en viss konfiguration vid anslutning till resurser på en viss URL.

21.1.6 Verktyg för Master Password och Auth Config

Under menyn Options (*Inställningar* ► *Alternativ*) på fliken *Authentication* finns det flera verktyg för att hantera autentiseringsdatabasen och konfigurationerna:

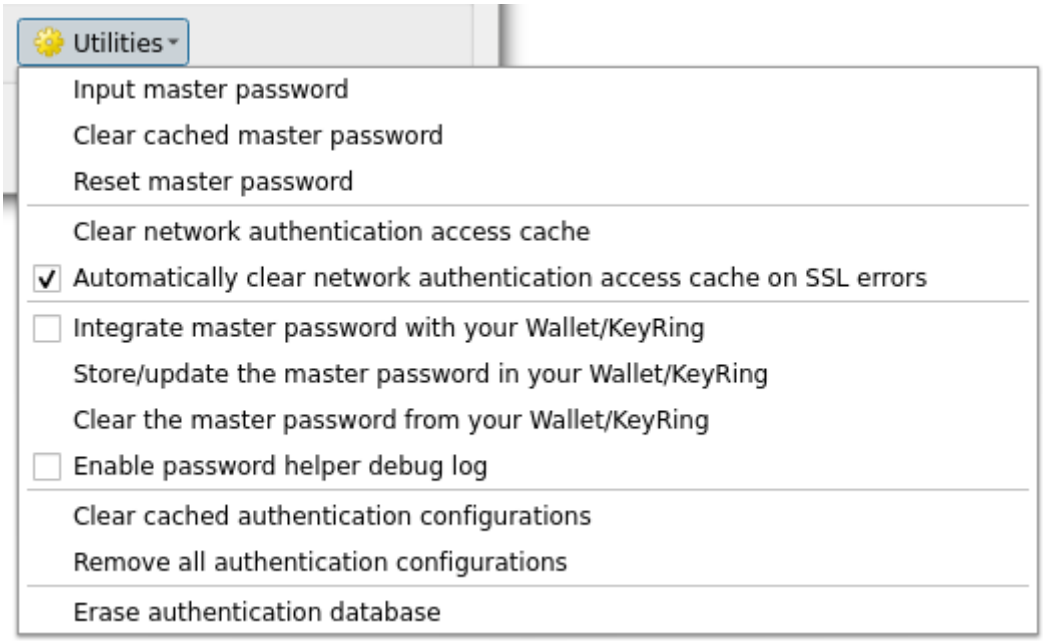


Fig. 21.15: Verktygsmeny

- **Input master password**: öppnar dialogrutan för inmatning av huvudlösenordet, oberoende av om något autentiseringsdatabaskommando utförs
- **Clear cached master password**: återställer masterlösenordet om det har ställts in
- **Reset master password**: öppnar en dialogruta för att ändra huvudlösenordet (det aktuella lösenordet måste vara känt) och eventuellt säkerhetskopiera den aktuella databasen
- **Clear network authentication access cache**: rensar autentiseringscachen för alla anslutningar
- **Automatiskt rensa nätverksautentiseringens åtkomstcache vid SSL-fel**: Anslutningscachen lagrar alla autentiseringsdata för anslutningar, även när anslutningen misslyckas. Om du ändrar autentiseringskonfigurationer eller certifikatutfärdare bör du rensa autentiseringscachen eller starta om QGIS. När det här alternativet är markerat rensas autentiseringscachen automatiskt varje gång ett SSL-fel inträffar och du väljer att avbryta anslutningen
- **Integrera huvudlösenordet med din plånbok/nyckelring**: lägger till huvudlösenordet i din personliga plånbok/nyckelring
- **Lagra/uppdatera huvudlösenordet i din plånbok/nyckelring**: uppdaterar det ändrade huvudlösenordet i din plånbok/nyckelring
- **Radera huvudlösenordet från din plånbok/nyckelring**: raderar huvudlösenordet från din plånbok/nyckelring
- **Enable password helper debug log**: aktiverar ett felsökningsverktyg som kommer att innehålla all logginformation för autentiseringsmetoderna

- **Rensa cachade autentiseringskonfigurationer**: rensar den interna cacheminnet för konfigurationer, som används för att snabba upp nätverksanslutningar. Detta rensar inte QGIS centrala nätverksåtkomsthanterares cache, vilket kräver en omstart av QGIS.
- **Remove all authentication configurations**: rensar databasen från alla konfigurationsposter, utan att ta bort andra lagrade poster.
- **Erase authentication database**: schemalägger en säkerhetskopia av den aktuella databasen och en fullständig återuppbyggnad av databasens tabellstruktur. Åtgärderna schemaläggs till en senare tidpunkt för att säkerställa att andra åtgärder, t.ex. projektladdning, inte avbryter åtgärden eller orsakar fel på grund av en tillfälligt saknad databas.

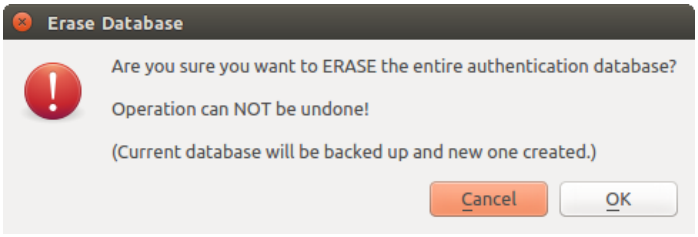


Fig. 21.16: Meny för verifiering av DB-radering

21.1.7 Använda autentiseringskonfigurationer

Vanligtvis väljs en autentiseringskonfiguration i en konfigurationsdialog för en nätverkstjänst (t.ex. WMS). Väljarwidgeten kan dock bäddas in var som helst där autentisering behövs eller i icke-kärnfunktioner, som i PyQGIS- eller C++-tillägg från tredje part.

När väljaren används visas *No authentication* i popup-menyn när ingenting har valts, när det inte finns några konfigurationer att välja mellan eller när en tidigare tilldelad konfiguration inte längre finns i databasen. Fälten *Type* och *Id* är skrivskyddade och innehåller en beskrivning av autentiseringsmetoden respektive konfigurationens ID.

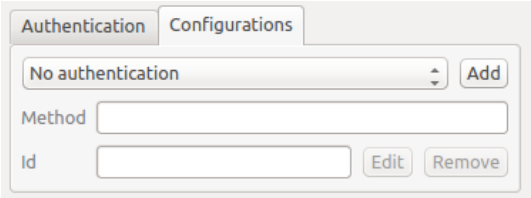


Fig. 21.17: Konfigurationsväljare för autentisering utan autentisering

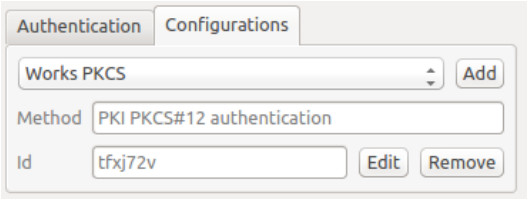


Fig. 21.18: Konfigurationsväljare för autentisering med vald konfiguration

21.1.8 Python-bindningar

Alla klasser och offentliga funktioner har sip-bindningar, utom `QgsAuthCrypto`, eftersom hanteringen av huvudlösenordets hashing och auth-databasens kryptering bör hanteras av huvudappen och inte via Python.

När huvudlösenordet har angetts är API:et öppet för åtkomst till autentiseringskonfigurationer i autentiseringsdatabasen, på samma sätt som Firefox fungerar. Ingen mur mot PyQGIS-åtkomst har dock definierats. Detta kan leda till problem där en användare laddar ner/installerar ett skadligt PyQGIS-plugin eller en fristående app som får tillgång till autentiseringsuppgifter.

21.2 Arbetsflöden för användarautentisering

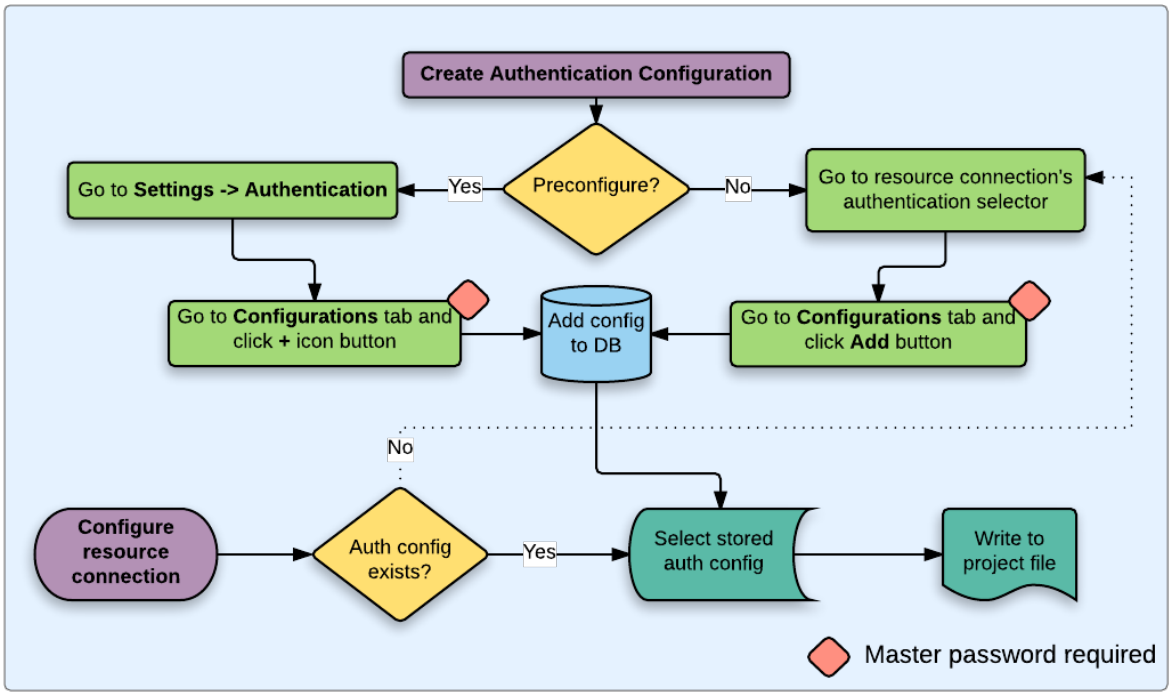


Fig. 21.19: Generiskt arbetsflöde för användare

21.2.1 HTTP(S)-autentisering

En av de vanligaste resursanslutningarna är via HTTP(S), t.ex. servrar för webbkartläggning, och tillägg för autentiseringsmetoder fungerar ofta för dessa typer av anslutningar. Metodtillägg har tillgång till HTTP-begärandeobjektet och kan manipulera både begäran och dess rubriker. Detta möjliggör många former av internetbaserad autentisering. Vid anslutning via HTTP(S) med standardmetoden för autentisering med användarnamn/lösenord kommer HTTP BASIC-autentisering att försöka vid anslutningen.

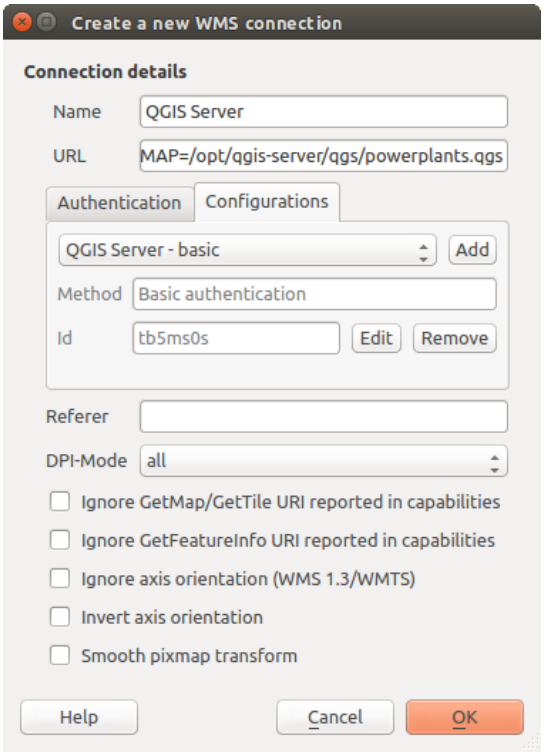


Fig. 21.20: Konfigurera en WMS-anslutning för HTTP BASIC

21.2.2 Autentisering av databas

Anslutningar till databasresurser lagras i allmänhet som par av typen `key=value`, vilket innebär att användarnamn och (eventuellt) lösenord visas om *inte* en autentiseringskonfiguration används. Vid konfiguration med autentiseringssystemet kommer `key=value` att vara en abstraherad representation av inloggningsuppgifterna, t.ex. `authfg=81t21b9`.

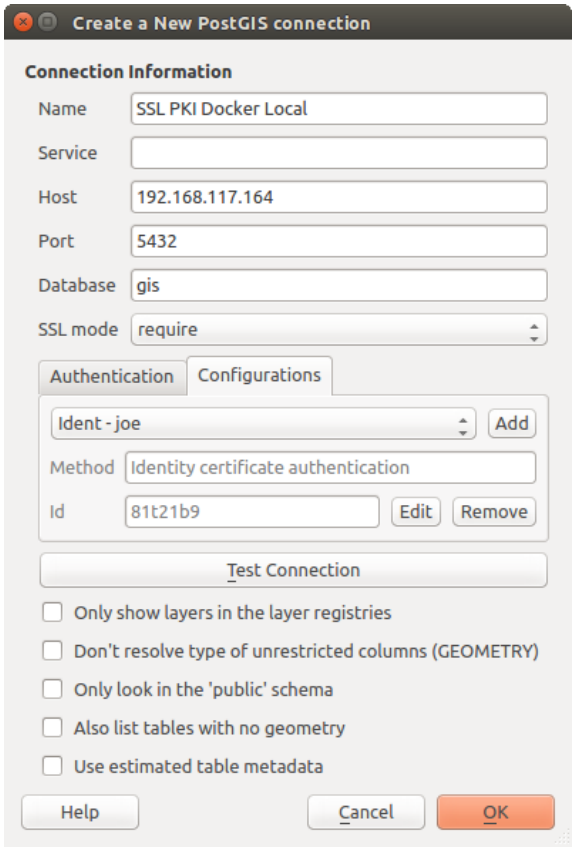


Fig. 21.21: Konfigurera en Postgres SSL-anslutning med PKI

21.2.3 PKI-autentisering

När du konfigurerar PKI-komponenter i autentiseringssystemet har du möjlighet att importera komponenter till databasen eller referera till komponentfiler som lagras på ditt filsystem. Det senare kan vara användbart om sådana komponenter ändras ofta eller om komponenterna kommer att ersättas av en systemadministratör. I båda fallen måste du lagra alla lösenfraser som behövs för att komma åt privata nycklar i databasen.

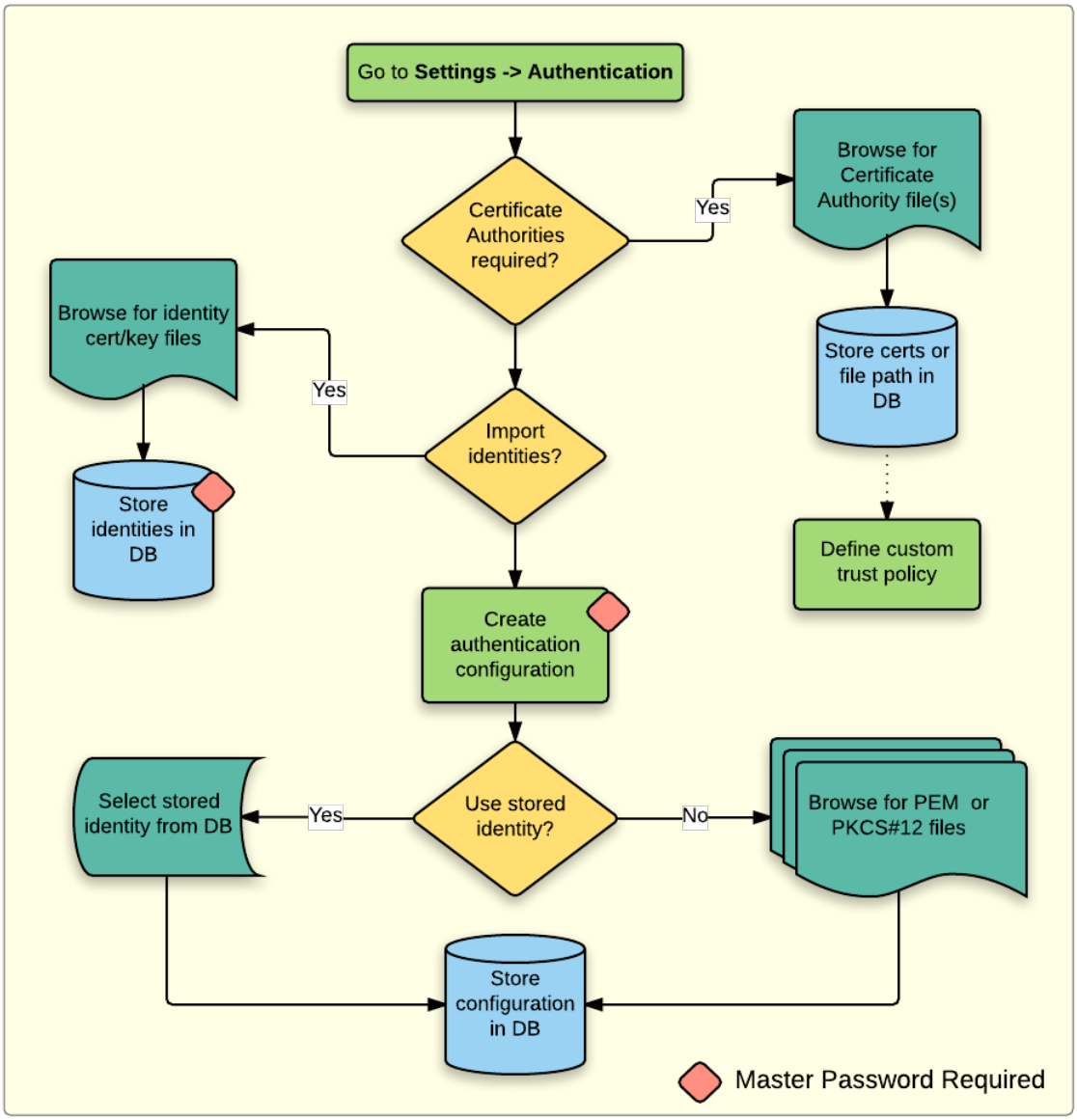


Fig. 21.22: Arbetsflöde för PKI-konfiguration

Alla PKI-komponenter kan hanteras i separata redigerare inom **Certifikathanterare**, som kan nås på fliken *Autentisering* i QGIS dialogruta *Alternativ* (*Inställningar* ► *Alternativ*) genom att klicka på knappen *Hantera certifikat*.

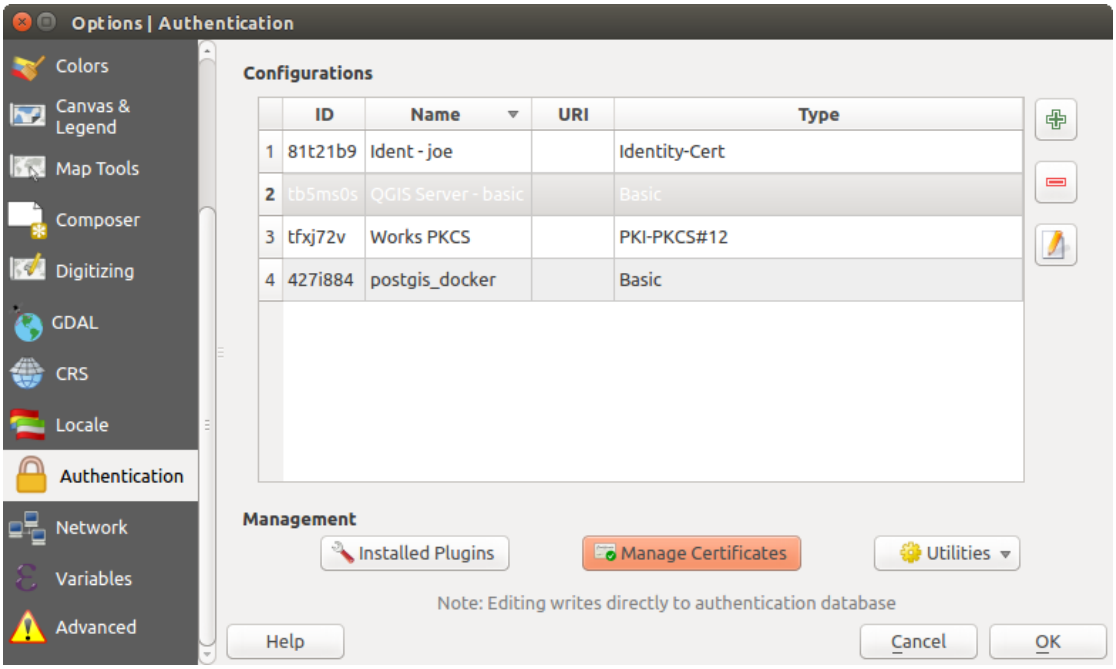


Fig. 21.23: Öppna certifikathanteraren

I *Certificate Manager* finns det redaktörer för **Identities**, **Servers** och **Authorities**. Var och en av dessa finns i sina egna flikar och beskrivs nedan i den ordning de förekommer i arbetsflödesdiagrammet ovan. Flikordningen är relativ till de redigerare som används ofta när du har vant dig vid arbetsflödet.

Observera

Eftersom alla ändringar i autentiseringssystemet omedelbart skrivs till autentiseringsdatabasen behöver du inte klicka på knappen *Options* i dialogrutan *OK* för att ändringarna ska sparas. Detta till skillnad från andra inställningar i dialogrutan Alternativ.

Myndigheter

Du kan hantera tillgängliga certifikatutfärdare (CA) på fliken **Authorities** i **Certificate manager** på fliken **Authentication** i dialogrutan QGIS **Options**.

Som framgår av arbetsflödesschemat ovan är det första steget att importera eller referera till en fil med certifikatutfärdare. Det här steget är valfritt och kan vara onödigt om PKI:s förtroendekedja härrör från rotcertifikatutfärdare som redan är installerade i operativsystemet, t.ex. ett certifikat från en kommersiell certifikatleverantör. Om din autentiserande rotcertifikatutfärdare inte finns bland operativsystemets betrodda rotcertifikatutfärdare måste den importeras eller få sin filsystemssökväg refererad. (Kontakta din systemadministratör om du är osäker.)

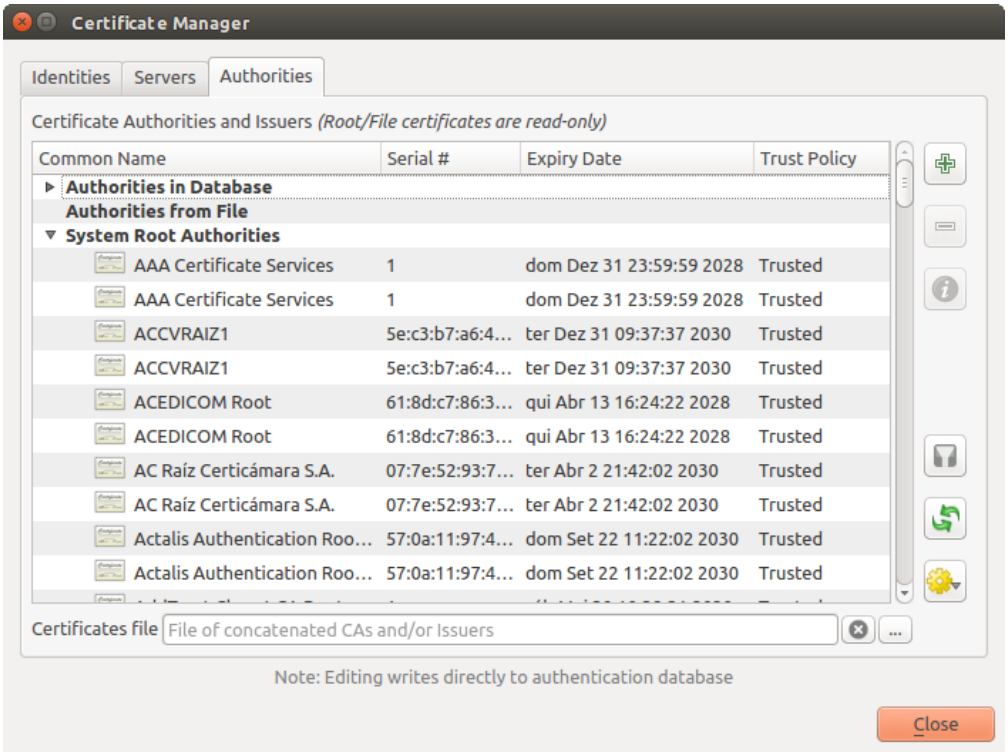


Fig. 21.24: Myndighetsredaktör

Som standard är rotcertifikatutfärdarna från ditt operativsystem tillgängliga, men deras förtroendeinställningar ärvs inte. Du bör granska inställningarna för certifikatförtroendeprinciperna, särskilt om OS-rotcertifikatutfärdarna har fått sina principer justerade. Alla certifikat som har gått ut kommer att ställas in som icke betrodda och kommer inte att användas i säkra serveranslutningar, såvida du inte specifikt åsidosätter dess förtroendepolicy. För att se den QGIS-avläsbara förtroendekedjan för ett certifikat, markera det och klicka på Visa information för certifikat.

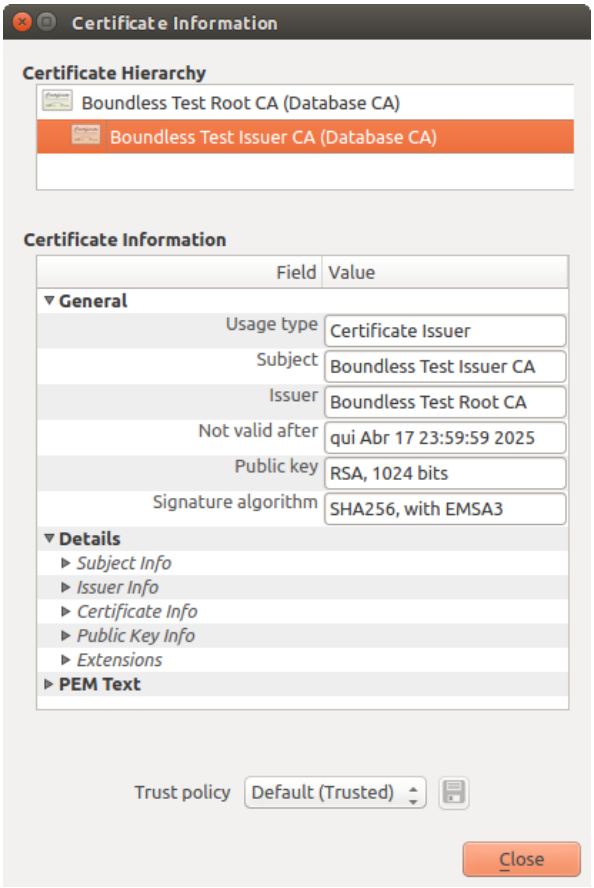


Fig. 21.25: Dialog för certifikatinformation

Du kan redigera *Trust policy* för alla valda certifikat i kedjan. En ändring av förtroendepolicyn för ett valt certifikat sparas inte i databasen om inte knappen Save certificate trust policy change to database klickas *för* varje valt certifikat. Om du stänger dialogrutan kommer policyändringarna **inte** att tillämpas.

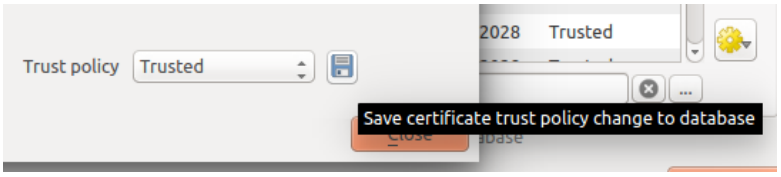


Fig. 21.26: Spara ändringarna av förtroendepolicyn

Du kan granska de filtrerade certifikatutfärdarna, både mellanliggande certifikat och rotcertifikat, som kommer att betros för säkra anslutningar eller ändra standardförtroendepolicyn genom att klicka på knappen **Alternativ**.

Varning

Om du ändrar standardförtroendeprincipen kan det leda till problem med säkra anslutningar.

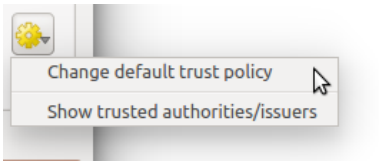


Fig. 21.27: Meny för myndighetsalternativ

Du kan importera certifikatutfärdare eller spara en filsystemssökväg från en fil som innehåller flera certifikatutfärdare, eller importera enskilda certifikatutfärdare. Standardformatet PEM för filer som innehåller flera certifikatutfärdarkedjor har rotcertifikatet längst ned i filen och alla efterföljande signerade underordnade certifikat ovanför, mot början av filen.

Dialogrutan för import av CA-certifikat hittar alla CA-certifikat i filen, oavsett ordning, och erbjuder även möjligheten att importera certifikat som anses vara ogiltiga (om du vill åsidosätta deras förtroendepolicy). Du kan åsidosätta förtroendepolicyn vid import eller göra det senare i redigeraren **Authorities**.

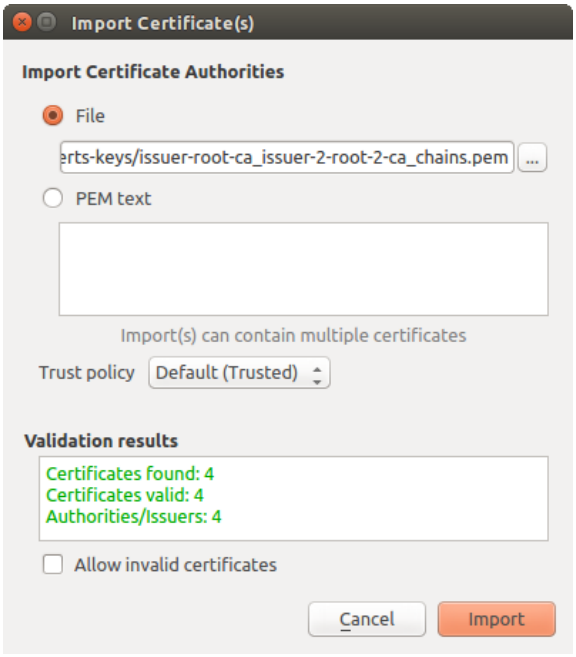


Fig. 21.28: Dialog för import av certifikat

Observera

Om du klistrar in certifikatinformation i fältet *PEM text* bör du observera att krypterade certifikat inte stöds.

Identiteter

Du kan hantera tillgängliga klientidentitetspaket från fliken *Identities* i *Certificate manager* från fliken **Authentication** i dialogrutan QGIS **Options**. En identitet är det som autentiserar dig mot en PKI-aktiverad tjänst och består vanligtvis av ett klientcertifikat och en privat nyckel, antingen som separata filer eller kombinerade till en enda ”buntad” fil. Paketet eller den privata nyckeln är ofta skyddad med en lösenfras.

När du har importerat alla certifikatutfärdare (CA) kan du eventuellt importera alla identitetsbuntar till autentiseringsdatabasen. Om du inte vill lagra identiteterna kan du hänvisa till komponenternas filsystemssökvägar i en enskild autentiseringskonfiguration.

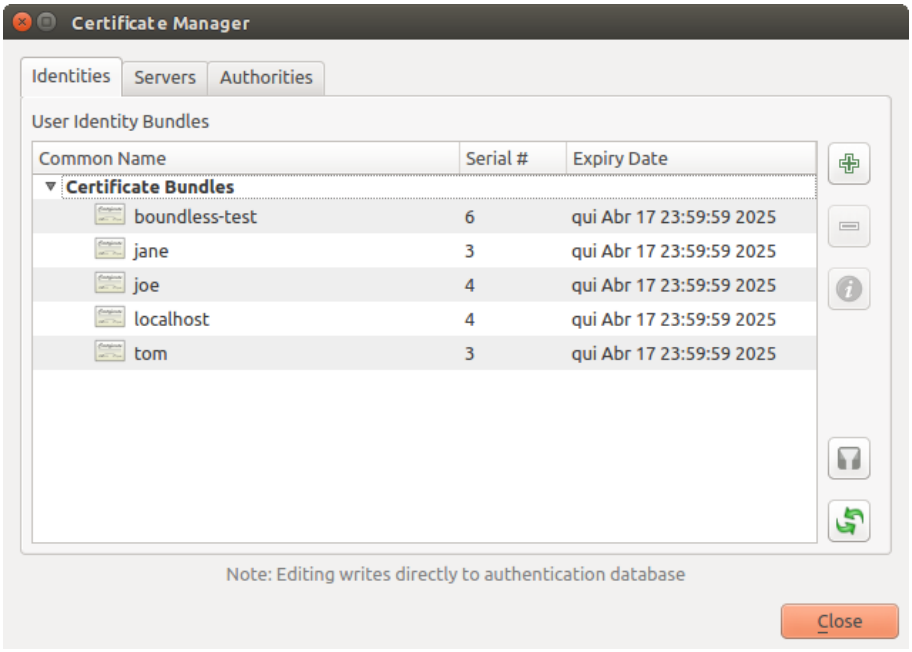


Fig. 21.29: Identiteter redaktör

När du importerar ett identitetspaket kan det vara lösenfrasskyddat eller oskyddat och kan innehålla CA-certifikat som bildar en förtroendekedja. Förtroendekedjans certifieringar importeras inte här, utan kan läggas till separat under fliken *Authorities*.

Vid import kommer paketets certifikat och privata nyckel att lagras i databasen, med nyckelns lagring krypterad med QGIS huvudlösenord. Efterföljande användning av det lagrade paketet från databasen kräver endast inmatning av huvudlösenordet.

Personliga identitetsbuntar som består av PEM/DER (.pem/.der) och PKCS#12 (.p12/.pfx) komponenter stöds. Om en nyckel eller ett paket är skyddat med lösenord krävs lösenordet för att validera komponenten före import. Om klientcertifikatet i paketet är ogiltigt (t.ex. om dess giltighetsdatum ännu inte har börjat gälla eller har löpt ut) kan paketet inte heller importeras.

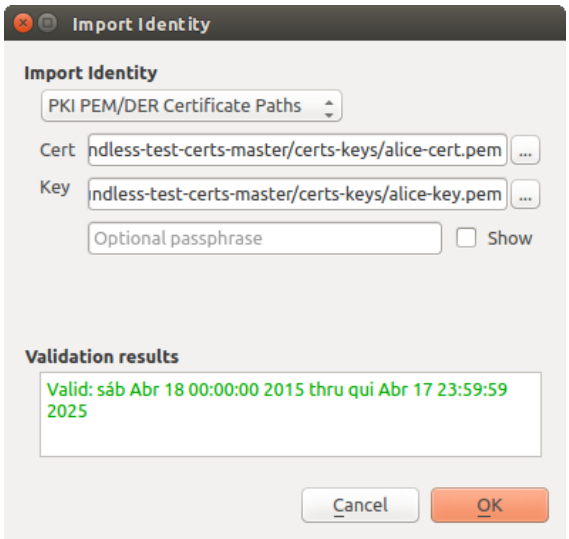


Fig. 21.30: Import av PEM/DER-identitet

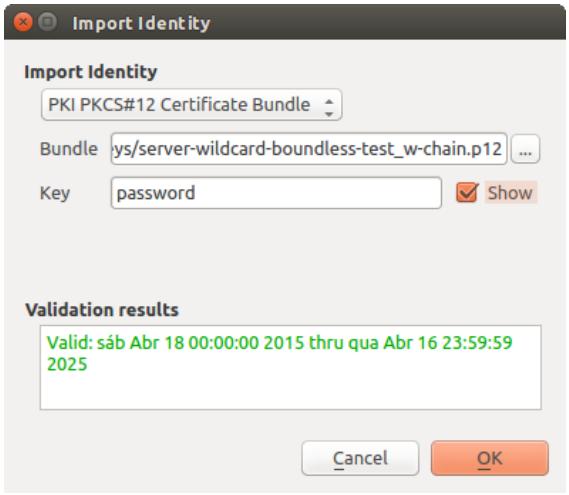


Fig. 21.31: Import av PKCS#12-identitet

21.2.4 Hantering av dåliga lager

Ibland är konfigurations-ID:t för autentisering som sparas med en projektfil inte längre giltigt, vilket kan bero på att den aktuella autentiseringsdatabasen är en annan än när projektet senast sparades, eller på att autentiseringsuppgifterna inte stämmer överens. I sådana fall kommer dialogrutan *Handle bad layers* att visas när QGIS startas.

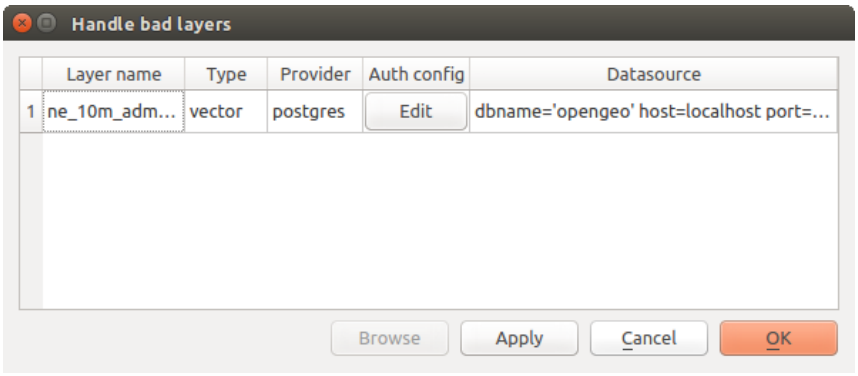


Fig. 21.32: Hantera dåliga lager med autentisering

Om en datakälla har ett konfigurations-ID för autentisering associerat med sig kan du redigera det. Då redigeras datakällans sträng automatiskt, ungefär på samma sätt som när du öppnar projektfilen i en textredigerare och redigerar strängen.

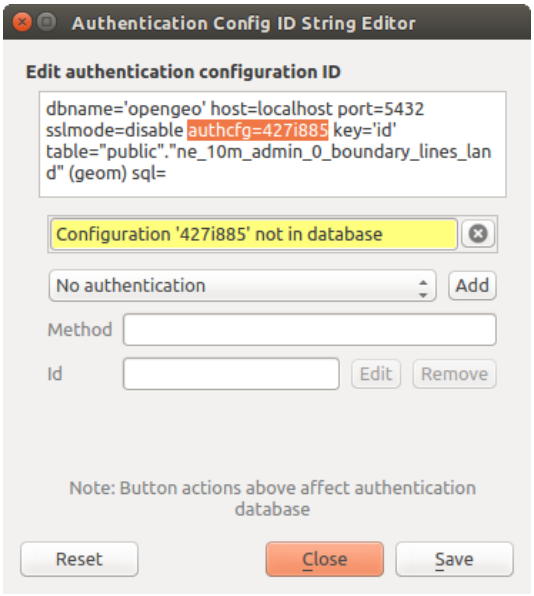


Fig. 21.33: Redigera konfigurations-ID för autentisering av dåligt lager

21.2.5 Ändra konfigurations-ID för autentisering

Ibland behöver du ändra det konfigurations-ID för autentisering som är kopplat till åtkomst till en resurs. Det finns fall där detta är användbart:

- **Resursens autentiserings-konfigurations-ID är inte längre giltigt**: Detta kan inträffa när du har bytt auth-databaser och behöver *anpassa* en ny konfiguration till det ID som redan är associerat med en resurs.
- **Delade projektfiler**: Om du har för avsikt att dela projekt mellan användare, t.ex. via en delad filserver, kan du *fördefiniera* ett 7-tecken (innehållande **a-z** och/eller **0-9**) som är associerat med resursen. Sedan ändrar enskilda användare ID:t i en autentiseringskonfiguration som är specifik för deras referenser för resursen. När projektet öppnas finns ID:t i autentiseringsdatabasen, men inloggningsuppgifterna är olika för varje användare.

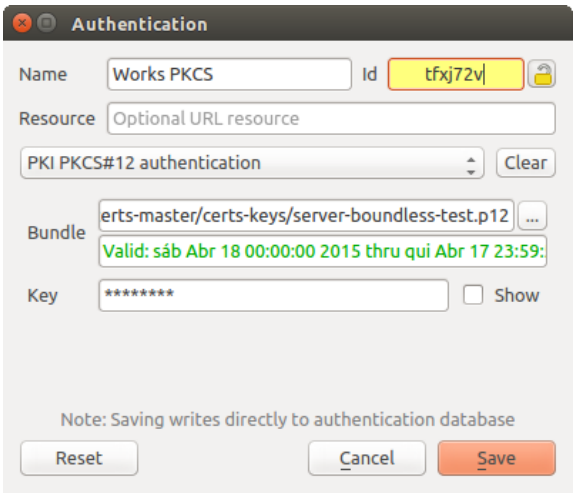


Fig. 21.34: Ändra ett lagers konfigurations-ID för autentisering (olåst gult textfält)

Varning

Att ändra auth-konfigurations-ID:t anses vara en avancerad åtgärd och bör endast göras med full kunskap om

varför det är nödvändigt. Därför finns det en låsknapp som du måste klicka på för att låsa upp ID:ts textfält innan du kan redigera ID:t.

21.2.6 Stöd för QGIS Server

När du använder en projektfil, med lager som har autentiseringskonfigurationer, som grund för en karta i QGIS Server, krävs det ett par ytterligare installationssteg för att QGIS ska kunna ladda resurserna:

- Autentiseringsdatabasen måste vara tillgänglig
- Autentiseringsdatabasens huvudlösenord måste vara tillgängligt

Vid instansiering av autentiseringssystemet skapar eller använder Servern `qgis-auth.db`-filen i den aktiva *användarprofilen*, eller den katalog som definieras av miljövariabeln `QGIS_AUTH_DB_DIR_PATH`. Det kan hända att serverns användare inte har någon HOME-katalog, i så fall ska du använda miljövariabeln för att definiera en katalog som serverns användare har läs- och skrivrättigheter till och som inte finns i de kataloger som är tillgängliga för webben.

För att skicka huvudlösenordet till Server, skriv det till den första raden i filen på en sökväg i filsystemet som kan läsas av Serverns processanvändare och som definieras med hjälp av miljövariabeln `QGIS_AUTH_PASSWORD_FILE`. Se till att begränsa filen så att den endast kan läsas av serverns processanvändare och att filen inte lagras i kataloger som är tillgängliga för webben.

Observera

variabeln `QGIS_AUTH_PASSWORD_FILE` kommer att tas bort från servermiljön omedelbart efter åtkomst.

21.2.7 Undantag för SSL-server

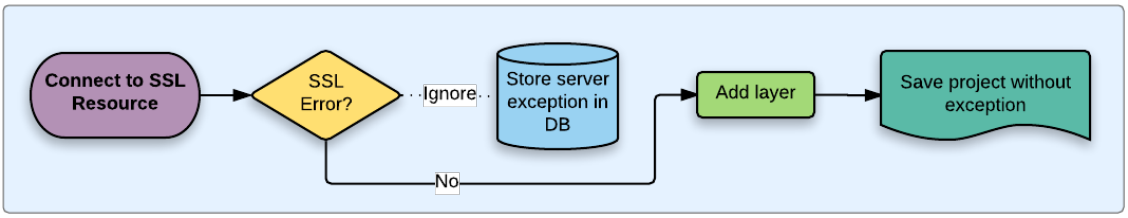


Fig. 21.35: Undantag för SSL-server

Du kan hantera SSL-serverkonfigurationer och undantag från fliken **Servers** i avsnittet **Authentication** i dialogrutan QGIS **Options**.

När du ansluter till en SSL-server uppstår ibland fel i SSL-”handskakningen” eller serverns certifikat. Du kan ignorera dessa fel eller skapa en SSL-serverkonfiguration som ett undantag. Detta liknar hur webbläsare tillåter dig att åsidosätta SSL-fel, men med mer detaljerad kontroll.

Varning

Du bör inte skapa en SSL-serverkonfiguration om du inte har fullständig kunskap om hela SSL-konfigurationen mellan servern och klienten. Rapportera i stället problemet till serveradministratören.

Observera

Vissa PKI-konfigurationer använder en helt annan CA-förtroendekedja för att validera klientidentiteter än den kedja som används för att validera SSL-servercertifikatet. Under sådana omständigheter kommer en konfiguration som skapats för den anslutande servern inte nödvändigtvis att åtgärda ett problem med valideringen av din klientidentitet, och det är bara utfärdaren av din klientidentitet eller serveradministratören som kan åtgärda problemet.

Du kan förkonfigurera en SSL-serverkonfiguration genom att klicka på knappen . Alternativt kan du lägga till en konfiguration när ett SSL-fel inträffar under en anslutning och du får upp en dialogruta **SSL Error** (där felet kan ignoreras tillfälligt eller sparas i databasen och ignoreras):

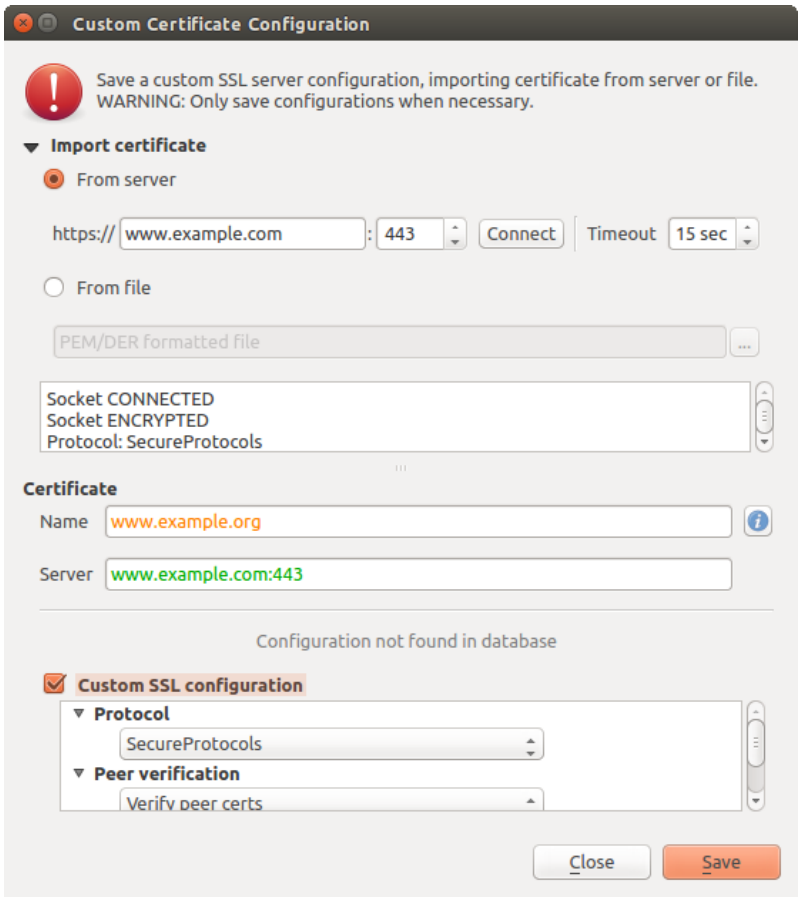


Fig. 21.36: Lägga till konfiguration manuellt

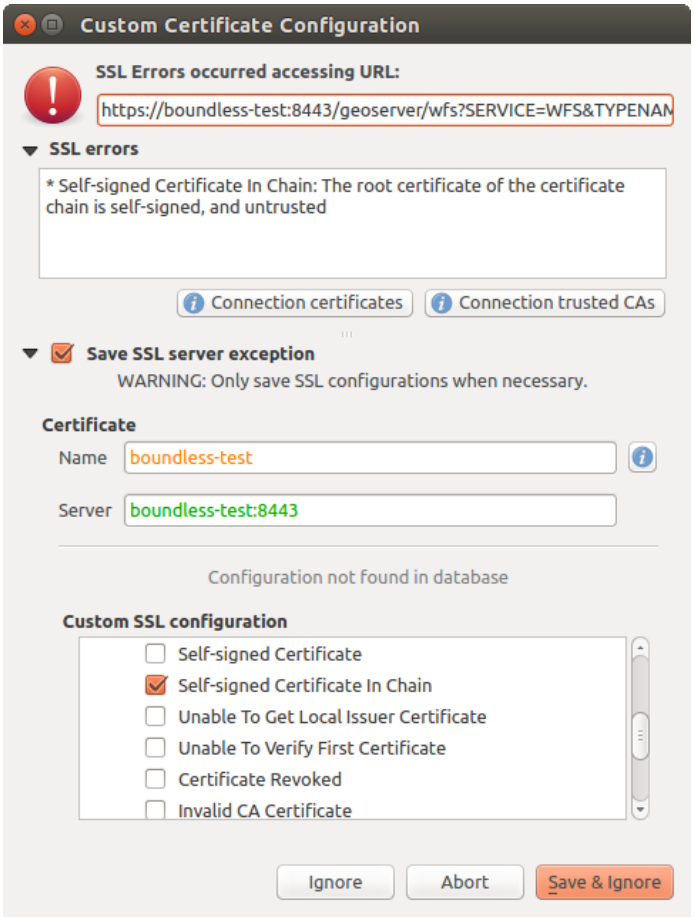


Fig. 21.37: Lägga till konfiguration under SSL-fel

När en SSL-konfiguration har sparats i databasen kan den redigeras eller tas bort.

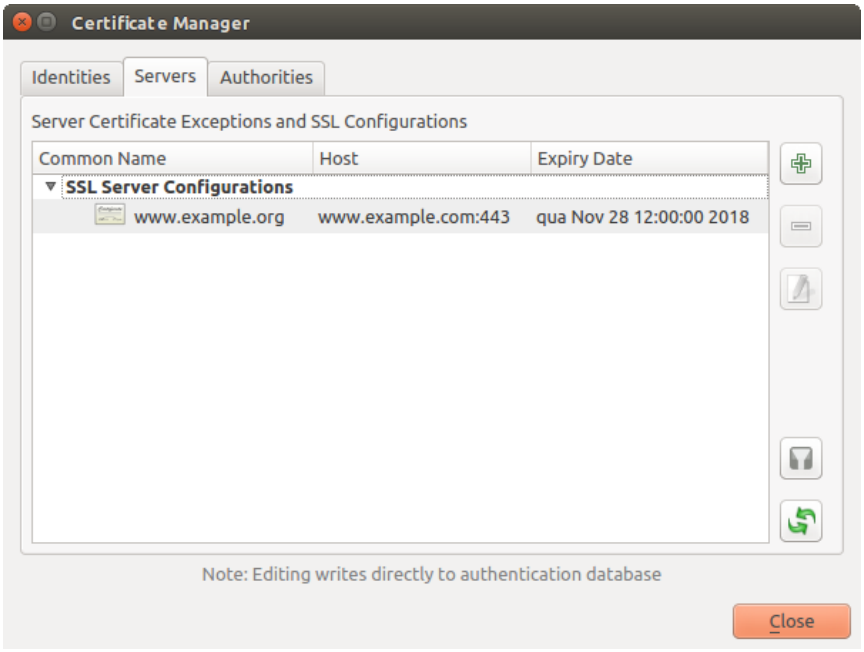


Fig. 21.38: Befintlig SSL-konfiguration

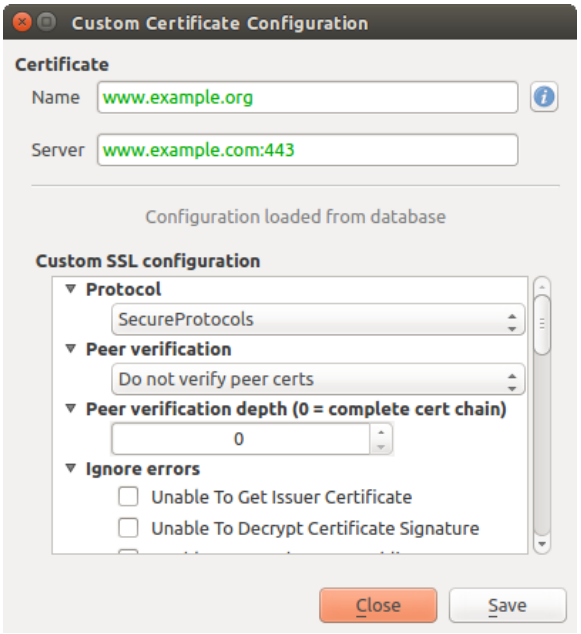


Fig. 21.39: Redigera en befintlig SSL-konfiguration

Om du vill förkonfigurera en SSL-konfiguration och importdialogen inte fungerar för serverns anslutning, kan du manuellt utlösa en anslutning via **Python Console** genom att köra följande kod (ersätt `https://bugreports.qt-project.org` med URL:en till din 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)
```

Om något fel uppstår öppnas en dialogruta för SSL-fel, där du kan välja att spara konfigurationen i databasen.

KAPITEL 22

GRASS GIS-integration

GRASS-integration ger tillgång till GRASS GIS-databaser och funktioner (se GRASS-PROJEKT i *Litteratur- och webbreferenser*). Integrationen består av två delar: datakälla och tillägg. Datakällan gör det möjligt att bläddra bland, hantera och visualisera GRASS raster- och vektorlager. Tillägget kan användas för att skapa nya GRASS-platser och kartuppsättningar, ändra GRASS-region, skapa och redigera vektorlager och analysera GRASS 2-D- och 3-D-data med mer än 400 GRASS-moduler. I det här avsnittet presenterar vi datakäll- och tilläggsfunktionerna och ger några exempel på hur man hanterar och arbetar med GRASS-data.

Datakällan stöder GRASS version 6 och 7, tillägget stöder GRASS 6 och 7 (från och med QGIS 2.12). QGIS-distributionen kan innehålla datakälla/tillägg för antingen GRASS 6 eller GRASS 7 eller för båda versionerna samtidigt (binärerna har olika filnamn). Endast en version av datakällan/tillägget kan dock laddas vid körning.

22.1 Demo dataset

Som exempel använder vi QGIS Alaska-datauppsättningen (se avsnitt *Nedladdning av exempeldata*). Den innehåller ett litet GRASS-prov `LOCATION` med tre vektorlager och en rasterhöjdkarta. Skapa en ny mapp som heter `grassdata`, ladda ner QGIS 'Alaska'-datasetet `qgis_sample_data.zip` från https://download.qgis.org/downloads/data/ och packa upp filen i `grassdata`.

Fler exempel på GRASS `LOCATIONs` finns tillgängliga på GRASS webbplats.

22.2 Ladda GRASS raster- och vektorlager

Om datakällan laddas i QGIS läggs platsobjektet med GRASS -ikonen till i bläddrarträdet under varje mappobjekt som innehåller GRASS-plats. Gå till mappen `grassdata` och expandera platsen `alaska` och mapset `demo`.

Du kan ladda GRASS raster- och vektorlager som vilket annat lager som helst från bläddraren, antingen genom att dubbelklicka på lagerobjektet eller genom att dra och släppa det på kartbilden eller i teckenförklaringen.

Tips

GRASS-dataladdning

Om du inte ser GRASS platsobjekt, kontrollera i *Hjälp* ► *Om* ► *Datakällor* om GRASS vektorprovider är laddad.

22.3 Importera data till en GRASS LOCATION via dra och släpp

I detta avsnitt ges ett exempel på hur man importerar raster- och vektordata till en GRASS-kartuppsättning.

1. I QGIS browser navigerar du till den mapset som du vill importera data till.
2. I QGIS browser hittar du ett lager som du vill importera till GRASS, observera att du kan öppna en annan instans av browsern (*Browser Panel (2)*) om källdata är för långt från kartuppsättningen i trädet.
3. Dra ett lager och släpp det på målkartan. Importen kan ta lite tid för större lager, du kommer att se den animerade ikonen framför det nya lagerobjektet tills importen är klar.

När rasterdata finns i olika CRS kan de återprojiceras med hjälp av en *Approximate* (snabb) eller *Exact* (exakt) transformation. Om en länk till källrastret skapas (med `r.external`), källdata finns i samma CRS och formatet är känt för GDAL, kommer källdatans CRS att användas. Du kan ställa in dessa alternativ på fliken *Browser* i *GRASS Alternativ*.

Om ett källraster har fler band skapas en ny GRASS-karta för varje lager med **.<band number>** suffix och en grupp av alla kartor med ikonen skapas. Externa raster har en annan ikon.

22.4 Hantera GRASS-data i QGIS Browser

- Kopiering av kartor: GRASS-kartor kan kopieras mellan kartuppsättningar inom samma plats med hjälp av dra och släpp.
- Radering av kartor: Högerklicka på en GRASS-karta och välj *Delete* från snabbmenyn.
- Byta namn på kartor: Högerklicka på en GRASS-karta och välj *Rename* från kontextmenyn.

22.5 GRASS Alternativ

GRASS-alternativ kan ställas in i dialogen *GRASS-alternativ*, som kan öppnas genom att högerklicka på platsen eller kartsetet i bläddraren och sedan välja *GRASS-alternativ*.

22.6 Starta tillägget GRASS

För att använda GRASS-funktioner i QGIS måste du välja och ladda GRASS-plugin med hjälp av Tilläggshanterare. Detta gör du genom att gå till menyn *Tillägg* ► *Manage and Install Tillägg…*, markera kryssrutan| *GRASS* och klicka på *OK*.

Följande huvudfunktioner finns i GRASS-menyn (*Tillägg* ► *GRASS*) när du startar GRASS-tillägget:

- Open Mapset
- New Mapset
- Close Mapset
- Öppna GRASS-verktyg
- Visa aktuell GRASS-region
- GRASS-alternativ

22.7 Öppna GRASS mapset

Ett GRASS-kapset måste öppnas för att få tillgång till GRASS-verktygen i tillägget (verktygen är inaktiverade om inget mapset är öppet). Du kan öppna ett mapset från bläddraren: högerklicka på mapset-objektet och välj sedan *Open mapset* från snabbmenyn.

22.8 GRASS LOCATION och MAPSET

GRASS-data lagras i en katalog som kallas GISDBASE. Den här katalogen, som ofta kallas `grassdata`, måste skapas innan du börjar arbeta med GRASS-tillägget i QGIS. I den här katalogen är GRASS GIS-data organiserade efter projekt som lagras i underkataloger som kallas `LOCATIONs`. Varje `LOCATION` definieras av sitt koordinatsystem, sin kartprojektion och sina geografiska gränser. Varje `LOCATION` kan ha flera `MAPSETs` (underkataloger till `LOCATION`) som används för att dela upp projektet i olika ämnen eller delregioner, eller som arbetsytor för enskilda teammedlemmar (se Neteler & Mitasova 2008 i *Litteratur- och webbreferenser*). För att kunna analysera vektor- och rasterlager med GRASS-moduler måste du i allmänhet importera dem till en GRASS `LOCATION`. (Detta är inte helt sant - med GRASS-modulerna `r.external` och `v.external` kan du skapa skrivskyddade länkar till externa GDAL-stödda dataset utan att importera dem. Detta är inte det vanliga sättet för nybörjare att arbeta med GRASS, därför kommer denna funktionalitet inte att beskrivas här)

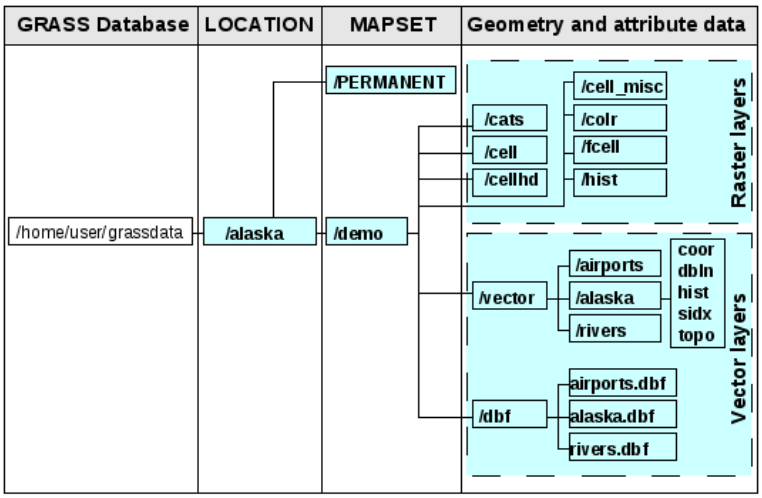


Fig. 22.1: GRASS-data i alaska LOCATION

22.9 Importera data till en GRASS-lokalisering

Se avsnitt *Importera data till en GRASS LOCATION via dra och släpp* för att få reda på hur data enkelt kan importeras genom att dra och släppa i bläddraren.

Detta avsnitt ger ett exempel på hur man importerar raster- och vektordata till GRASS `LOCATION` 'alaska' som tillhandahålls av QGIS 'Alaska'-datasetet på traditionellt sätt, med hjälp av standard GRASS-moduler. Därför använder vi landcover-rasterkartan `landcover.img` och vektor-GML-filen `lakes.gml` från QGIS 'Alaska'-datasetet (se *Nedladdning av exempeldata*).

1. Starta QGIS och se till att GRASS-tillägget är laddat.
2. I GRASS verktygsfält klickar du på ikonen Open MAPSET för att få upp guiden *MAPSET*.
3. Välj som GRASS-databas mappen `grassdata` i QGIS Alaska dataset, som `LOCATION` 'alaska', som `MAPSET` 'demo' och klicka på *OK*.
4. Klicka nu på ikonen Öppna GRASS-verktyg. Dialogrutan GRASS Toolbox (se avsnitt *GRASS verktygslåda*) visas.

5. För att importera rasterkartan `landcover.img`, klicka på modulen `r.in.gdal` i fliken *Modules Tree*. Med denna GRASS-modul kan du importera GDAL-stödda rasterfiler till en GRASS `LOCATION`. Dialogrutan för modulen för `r.in.gdal` visas.
6. Bläddra till mappen `raster` i QGIS-datasetet ”Alaska” och välj filen `landcover.img`.
7. Som namn på rasterutmatningen definierar du `landcover_grass` och klickar på *Run*. På fliken *Output* ser du det pågående GRASS-kommandot `r.in.gdal -o input=/path/to/landcover.img output=landcover_grass`.
8. När det står **Succesfully finished** klickar du på *View Output*. Rasterlagret `landcover_grass` är nu importerat till GRASS och kommer att visualiseras i QGIS canvas.
9. För att importera vektor GML filen `lakes.gml`, klicka på modulen `v.in.ogr` i *Modules Tree* fliken. Med denna GRASS-modul kan du importera OGR-stödda vektorfiler till en GRASS `LOCATION`. Dialogrutan för modulen för `v.in.ogr` visas.
10. Bläddra till mappen `gml` i QGIS-datasetet ”Alaska” och välj filen `lakes.gml` som OGR-fil.
11. Som namn på vektorutdata anger du `lakes_grass` och klickar på *Run*. Du behöver inte bry dig om de andra alternativen i det här exemplet. På fliken *Output* ser du det pågående GRASS-kommandot `v.in.ogr -o dsn=/path/to/lakes.gml output=lakes\_grass`.
12. När det står **Succesfully finished** klickar du på *View Output*. Vektorlagret `lakes_grass` är nu importerat till GRASS och kommer att visualiseras i QGIS canvas.

22.9.1 Skapa en ny GRASS LOCATION

Som ett exempel visas här exemplet GRASS `LOCATION alaska`, som är projicerat i Albers Equal Area-projektionen med fot som enhet. Detta exempel på GRASS `LOCATION alaska` kommer att användas för alla exempel och övningar i följande GRASS-relaterade avsnitt. Det är bra att ladda ner och installera datasetet på din dator (se *Nedladdning av exempeldata*).

1. Starta QGIS och se till att GRASS-tillägget är laddat.
2. Visualisera shapefilen `alaska.shp` (se avsnitt *Läsa in ett lager från en fil*) från QGIS Alaska-datasetet (se *Nedladdning av exempeldata*).
3. I GRASS verktygsfält klickar du på ikonen New mapset för att få upp guiden *MAPSET*.
4. Välj en befintlig GRASS databas (GISDBASE) mapp `grassdata`, eller skapa en för den nya `LOCATION` med hjälp av en filhanterare på din dator. Klicka sedan på *Next*.
5. Vi kan använda den här guiden för att skapa en ny `MAPSET` inom en befintlig `LOCATION` (se avsnitt *Lägga till en ny MAPSET*) eller för att skapa en helt ny `LOCATION`. Välj *Create new location* (se Fig. 22.2).
6. Ange ett namn för `LOCATION` – vi använde ’alaska’ – och klicka på *Next*.
7. Definiera projektionen genom att klicka på alternativknappen *Projection* för att aktivera projektionslistan.
8. Vi använder Albers Equal Area Alaska (feet) projektion. Eftersom vi råkar veta att den representeras av EPSG ID 2964, skriver vi in det i sökrutan. (Obs: Om du vill upprepa denna process för en annan `LOCATION` och projektion och inte har memorerat EPSG-ID:t, klicka på ikonen CRS Status i statusfältets nedre högra hörn (se avsnitt *Arbeta med projektioner*)).
9. I *Filter*, sätt in 2964 för att välja projektion.
10. Klicka på *Next*.
11. För att definiera standardregionen måste vi ange gränserna för `LOCATION` i nordlig, sydlig, östlig och västlig riktning. Här klickar vi helt enkelt på knappen *Set Current QGIS Extent*, för att tillämpa utbredningen av det laddade lagret `alaska.shp` som GRASS standardregionutbredning.
12. Klicka på *Next*.

13. Vi måste också definiera en MAPSET inom vår nya LOCATION (detta är nödvändigt när du skapar en ny LOCATION). Du kan namnge den som du vill - vi använde ”demo”. GRASS skapar automatiskt en speciell MAPSET som kallas PERMANENT, som är utformad för att lagra kärndata för projektet, dess standard spatiala utsträckning och koordinatsystemdefinitioner (se Neteler & Mitasova 2008 i *Litteratur- och webbreferenser*).
14. Kontrollera att sammanfattningen är korrekt och klicka sedan på *Finish*.
15. Den nya LOCATION, ’alaska’, och två MAPSETs, ’demo’ och ’PERMANENT’, skapas. Den arbetsuppsättning som för närvarande är öppen är ’demo’, som du definierade.
16. Lägg märke till att några av de verktyg i GRASS verktygsfält som var inaktiverade nu är aktiverade.

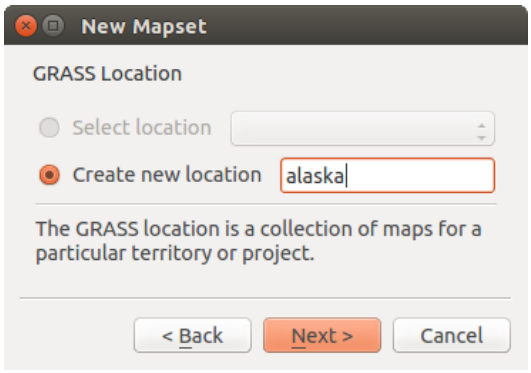


Fig. 22.2: Skapa en ny GRASS LOCATION eller en ny MAPSET i QGIS

Om du tyckte att det verkade vara många steg, så är det inte så farligt och ett mycket snabbt sätt att skapa en LOCATION. LOCATION ’alaska’ är nu redo för dataimport (se avsnitt *Importera data till en GRASS-lokalisering*). Du kan också använda de redan existerande vektor- och rasterdata i exemplet GRASS LOCATION ’alaska’, som ingår i QGIS-datasetet ’Alaska’ *Nedladdning av exempeldata*, och gå vidare till avsnitt *GRASS-vektordatamodellen*.

22.9.2 Lägga till en ny MAPSET

En användare har endast skrivrättigheter till en GRASS MAPSET som han eller hon har skapat. Detta innebär att förutom tillgång till din egen MAPSET, kan du läsa kartor i andra användares MAPSETs (och de kan läsa dina), men du kan bara ändra eller ta bort kartorna i din egen MAPSET.

Alla MAPSETs innehåller en WIND-fil som lagrar de aktuella gränskoordinatvärdena och den aktuella valda rasterupplösningen (se Neteler & Mitasova 2008 i *Litteratur- och webbreferenser*, och avsnitt *GRASS-regionverktyget*).

1. Starta QGIS och se till att GRASS-tillägget är laddat.
2. I GRASS verktygsfält klickar du på ikonen New mapset för att få upp guiden *MAPSET*.
3. Välj GRASS-databasens (GISDBASE) mapp grassdata med LOCATION ’alaska’, där vi vill lägga till ytterligare en MAPSET med namnet ’test’.
4. Klicka på *Next*.
5. Vi kan använda den här guiden för att skapa en ny MAPSET inom en befintlig LOCATION eller för att skapa en helt ny LOCATION. Klicka på alternativknappen *Select location* (se Fig. 22.2) och klicka på *Next*.
6. Ange namnet test för den nya MAPSET. Nedan i guiden ser du en lista över befintliga MAPSETs och deras ägare.
7. Klicka på *Next*, kolla sammanfattningen för att se att allt är korrekt och klicka på *Finish*.

22.10 GRASS-vektordatamodellen

Det är viktigt att förstå GRASS vektordatamodell före digitaliseringen. I allmänhet använder GRASS en topologisk vektormodell. Detta innebär att områden inte representeras som slutna polygoner, utan av en eller flera gränser. En gräns mellan två angränsande områden digitaliseras endast en gång, och den delas av båda områdena. Gränserna måste vara sammanhängande och slutna utan luckor. Ett område identifieras (och märks) med hjälp av områdets **centroid**.

Förutom gränser och centroider kan en vektorkarta också innehålla punkter och linjer. Alla dessa geometriska element kan blandas i en vektor och kommer att representeras i olika så kallade ”lager” i en GRASS-vektorkarta. I GRASS är ett lager alltså inte en vektor- eller rasterkarta utan en nivå i ett vektorlager. Detta är viktigt att skilja på noggrant. (Det är visserligen möjligt att blanda geometriska element, men det är ovanligt och används även i GRASS endast i speciella fall, t.ex. vid analys av vektornätverk. Normalt sett bör du föredra att lagra olika geometrielement i olika lager)

Det är möjligt att lagra flera ”lager” i ett vektordataset. Till exempel kan fält, skogar och sjöar lagras i en vektor. En intilliggande skog och sjö kan dela samma gräns, men de har separata attributtabeller. Det är också möjligt att koppla attribut till gränser. Ett exempel kan vara när gränsen mellan en sjö och en skog utgörs av en väg, och den kan därför ha en annan attributtabell.

Objektets ”lager” definieras av ”lagret” i GRASS. ”Layer” är det nummer som definierar om det finns mer än ett lager i datasetet (t.ex. om geometrin är skog eller sjö). För närvarande kan det bara vara ett nummer. I framtiden kommer GRASS även att stödja namn som fält i användargränssnittet.

Attribut kan lagras i GRASS `LOCATION` som dBase, SQLite3 eller i externa databastabeller, t.ex. PostgreSQL, MySQL, Oracle, etc.

Attribut i databastabeller är länkade till geometrielement med hjälp av ett ”category”-värde.

”Category” (nyckel, ID) är ett heltal som är kopplat till geometriprimitiver och används som länk till en nyckelkolumn i databastabellen.

Tips

Lär dig GRASS vektormodell

Det bästa sättet att lära sig GRASS vektormodell och dess möjligheter är att ladda ner en av de många GRASS handledningarna där vektormodellen beskrivs mer djupgående. Se https://grass.osgeo.org/learn/manuals/ för mer information, böcker och handledningar på flera språk.

22.11 Skapa ett nytt GRASS-vektorlager

Om du vill skapa ett nytt GRASS-vektorlager väljer du något av följande alternativ från mapset-kontextmenyn i bläddraren:

- Nytt punktlager
- Ny linje lager
- Nytt polygonlager

och ange ett namn i dialogrutan. En ny vektorkarta skapas och lagret läggs till i arbetsytan och redigeringen påbörjas. Valet av lagertyp begränsar inte vilka geometrityper som kan digitaliseras i vektorkartan. I GRASS är det möjligt att organisera alla typer av geometri (punkt, linje och polygon) i en vektorkarta. Typen används endast för att lägga till lagret på duken, eftersom QGIS kräver att ett lager har en viss typ.

Det är också möjligt att lägga till lager i befintliga vektorkartor genom att välja ett av de objekt som beskrivs ovan i snabbmenyn för en befintlig vektorkarta.

I GRASS är det möjligt att organisera alla möjliga geometrityper (punkt, linje och yta) i ett lager, eftersom GRASS använder en topologisk vektormodell, så du behöver inte välja geometrityp när du skapar en ny GRASS-vektor. Detta

skiljer sig från skapandet av shapefiler med QGIS, eftersom shapefiler använder vektormodellen Simple Feature (se avsnitt *Skapa nya vektorlager*).

22.12 Digitalisering och redigering av ett GRASS-vektorlager

GRASS-vektorlager kan digitaliseras med hjälp av QGIS standardverktyg för digitalisering. Det finns dock vissa särdrag som du bör känna till, på grund av

- GRASS topologisk modell jämfört med QGIS enkel funktion
- gRASS-modellens komplexitet
  - flera lager i en enda karta
  - flera geometrityper i en enda karta
  - geometrisk delning av flera funktioner från flera lager

De särskilda omständigheterna diskuteras i följande avsnitt.

Spara, kassera ändringar, ångra, göra om

> **Varning**
>
> Alla ändringar som görs under redigeringen skrivs omedelbart in i vektorkartan och relaterade attributtabeller.

Ändringar skrivs efter varje operation, men det är dock möjligt att ångra/återställa eller kassera alla ändringar när redigeringen avslutas. Om ångra eller kassera ändringar används, skrivs originaltillståndet om i vektorkartan och attributtabellerna.

Det finns två huvudsakliga skäl till detta beteende:

- Det ligger i GRASS-vektorns natur att komma från övertygelsen att användaren vill göra det han gör och att det är bättre att ha data sparade när arbetet plötsligt avbryts (t.ex. strömavbrott)
- En förutsättning för effektiv redigering av topologiska data är visualiserad information om topologisk korrekthet, sådan information kan endast erhållas från GRASS vektorkarta om ändringar skrivs till kartan.

Verktygsrad

Verktygsfältet ”Digitizing Toolbar” har några specifika verktyg när ett GRASS-lager redigeras:

Ikon	Verktyg	Syfte
	Ny punkt	Digitalisera ny punkt
	Ny linje	Digitalisera ny linje
	Ny gräns	Digitalisera ny gräns
	Ny centroid	Digitalisera ny centroid (märk ut befintligt område)
	Ny stängd gräns	Digitalisera ny stängd gräns

Tabell GRASS Digitalisering: Verktyg för GRASS-digitalisering

> **Tips**
>
> **Digitalisering av polygoner i GRASS**

Om du vill skapa en polygon i GRASS digitaliserar du först polygonens gräns. Sedan lägger du till en centroid (etikettpunkt) i den slutna gränsen. Anledningen till detta är att en topologisk vektormodell alltid länkar attributinformationen för en polygon till centroiden och inte till gränsen.

Kategori

Category, ofta kallat cat, är ett slags ID. Namnet kommer från tiden då GRASS-vektorer bara hade ett enda attribut ”category”. Kategorin används som en länk mellan geometri och attribut. En enda geometri kan ha flera kategorier och därmed representera flera funktioner i olika lager. För närvarande är det möjligt att tilldela endast en kategori per lager med hjälp av QGIS redigeringsverktyg. Nya funktioner har automatiskt tilldelats en ny unik kategori, utom gränser. Gränser bildar vanligtvis bara områden och representerar inte linjära funktioner, men det är dock möjligt att definiera attribut för en gräns senare, till exempel i ett annat lager.

Nya kategorier skapas alltid endast i det lager som för närvarande redigeras.

Det är inte möjligt att tilldela fler kategorier till geometri med hjälp av QGIS-redigering, sådana data representeras korrekt som flera funktioner och enskilda funktioner, även från olika lager, kan raderas.

Attribut

Attribut för det aktuella redigerade lagret kan endast ändras. Om vektorkartan innehåller fler lager kommer alla attribut för funktioner i andra lager att anges till ”<inte redigerbart (lager #)>” för att varna dig för att ett sådant attribut inte är redigerbart. Anledningen är att andra lager kan ha och vanligtvis har olika uppsättningar fält medan QGIS endast stöder en fast uppsättning fält per lager.

Om en geometriprimitiv inte har någon kategori tilldelad, tilldelas en ny unik kategori automatiskt och en ny post i attributtabellen skapas när ett attribut för geometrin ändras.

Tips

Om du vill göra en massuppdatering av attribut i tabellen, till exempel med hjälp av ”Field Calculator” (*Använda fältkalkylatorn*), och det finns funktioner utan kategori som du inte vill uppdatera (vanligtvis gränser), kan du filtrera bort dem genom att ställa in ”Advanced Filter” på ”cat is not null”.

Redigeringsstil

Den topologiska symbologin är viktig för effektiv redigering av topologiska data. När redigeringen startar ställs en specialiserad ”GRASS Edit”-renderare in på lagret automatiskt och den ursprungliga renderaren återställs när redigeringen avslutas. Stilen kan anpassas på fliken Style i lagrets egenskaper. Stilen kan också lagras i projektfilen eller i en separat fil som alla andra stilar. Om du anpassar stilen ska du inte ändra dess namn, eftersom det används för att återställa stilen när redigeringen startas igen.

Tips

Spara inte projektfilen när lagret redigeras, lagret skulle lagras med ”Edit Style” som inte har någon betydelse om lagret inte redigeras.

Stilen baseras på topologisk information som temporärt läggs till i attributtabellen som fältet ’topo_symbol’. Fältet tas bort automatiskt när redigeringen stängs.

Tips

Ta inte bort fältet ”topo_symbol” från attributtabellen, det skulle göra funktioner osynliga eftersom renderingsprogrammet är baserat på den kolumnen.

Snapping

För att bilda en yta måste hörnpunkterna i de anslutna gränserna ha **exakt** samma koordinater. Detta kan uppnås med hjälp av snäppverktyget endast om duken och vektorkartan har samma CRS. Annars kan koordinaterna bli något annorlunda på grund av konverteringen från kartkoordinater till canvas och tillbaka, på grund av representationsfel och CRS-transformationer.

Tips

Använd lagrets CRS även för canvas när du redigerar.

Begränsningar

Samtidig redigering av flera lager i samma vektor samtidigt stöds inte. Detta beror främst på att det är omöjligt att hantera flera ångra-stackar för en enda datakälla.

På Linux och macOS kan endast ett GRASS-lager redigeras åt gången. Detta beror på en bugg i GRASS som inte tillåter att databasdrivrutiner stängs i slumpmässig ordning. Detta problem håller på att lösas tillsammans med GRASS-utvecklare.

Tips

GRASS Redigera behörigheter

Du måste vara ägare till den GRASS `MAPSET` som du vill redigera. Det är omöjligt att redigera datalager i en `MAPSET` som inte är din, även om du har skrivbehörighet.

22.13 GRASS-regionverktyget

Regiondefinitionen (inställning av ett spatialt arbetsfönster) i GRASS är viktig för att arbeta med rasterlager. Vektoranalys är som standard inte begränsad till några definierade regiondefinitioner. Men alla nyskapade raster kommer att ha samma spatiala utsträckning och upplösning som den aktuella definierade GRASS-regionen, oavsett deras ursprungliga utsträckning och upplösning. Den aktuella GRASS-regionen lagras i filen `$LOCATION/$MAPSET/WIND` och den definierar gränser i norr, söder, öster och väster, antal kolumner och rader, horisontell och vertikal spatial upplösning.

Det är möjligt att slå på och av visualiseringen av GRASS-regionen i QGIS-rutan med hjälp av knappen
Display current GRASS region.

Regionen kan ändras på fliken ”Region” i dockningswidgeten ”GRASS Tolls”. Skriv in de nya regiongränserna och upplösningen och klicka på *Apply*. Om du klickar på *Select the extent by dragging on canvas* kan du välja en ny region interaktivt med musen på QGIS canvas genom att dra en rektangel.

GRASS-modulen `g.region` tillhandahåller många fler parametrar för att definiera en lämplig regionutbredning och upplösning för din rasteranalys. Du kan använda dessa parametrar med GRASS Toolbox, som beskrivs i avsnitt *GRASS verktygslåda*.

22.14 GRASS verktygslåda

Open GRASS Tools boxen tillhandahåller GRASS modulens funktioner för att arbeta med data i en vald GRASS `LOCATION` och `MAPSET`. För att använda GRASS Toolbox måste du öppna en `LOCATION` och en `MAPSET` som du har skrivrättigheter till (vanligtvis beviljade om du skapade `MAPSET`). Detta är nödvändigt eftersom nya raster- eller vektorlager som skapas under analysen måste skrivas till den aktuella `LOCATION` och `MAPSET`.

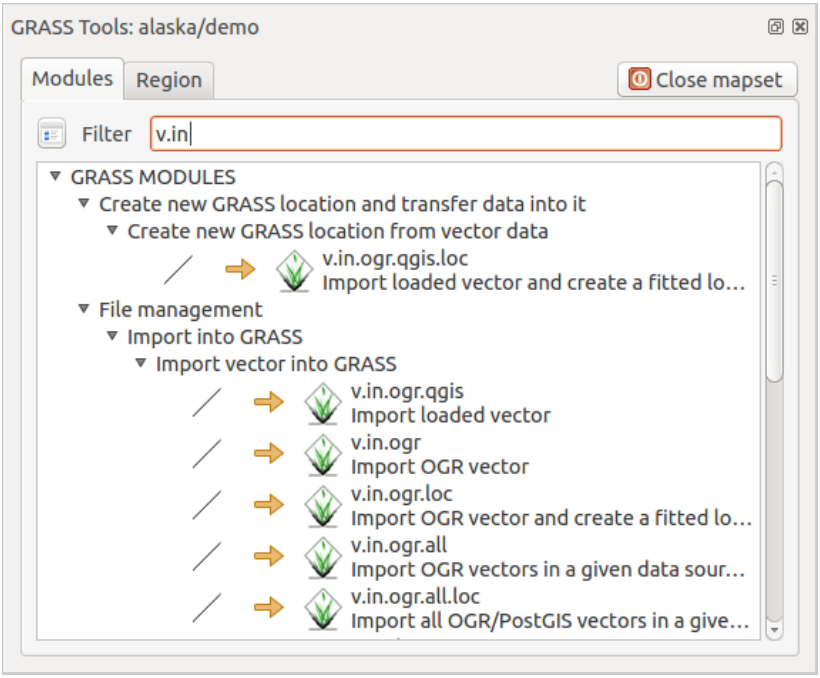


Fig. 22.3: GRASS verktygslåda och modulträd

22.14.1 Arbeta med GRASS-moduler

GRASS-skalet i GRASS Toolbox ger tillgång till nästan alla (mer än 300) GRASS-moduler via ett kommandoradsgränssnitt. För att erbjuda en mer användarvänlig arbetsmiljö tillhandahålls ca 200 av de tillgängliga GRASS-modulerna och funktionerna även via grafiska dialogrutor i GRASS plugin Toolbox.

En fullständig lista över GRASS-moduler som finns tillgängliga i den grafiska verktygslådan i QGIS version 3.40 finns i GRASS-wikin på https://grasswiki.osgeo.org/wiki/GRASS-QGIS_relevant_module_list.

Det är också möjligt att anpassa innehållet i GRASS Toolbox. Denna procedur beskrivs i avsnitt *Anpassa GRASS verktygslåda*.

Som visas i Fig. 22.3 kan du leta efter lämplig GRASS-modul med hjälp av den tematiskt grupperade *Modules Tree* eller den sökbara *Modules List*-fliken.

Genom att klicka på ikonen för en grafisk modul läggs en ny flik till i dialogrutan Toolbox, med tre nya underflikar: *Options*, *Output* och *Manual*.

Alternativ

Fliken *Options* ger en förenklad moduldialog där du vanligtvis kan välja ett raster- eller vektorlager som visualiseras i QGIS-kanvasen och ange ytterligare modulspecifika parametrar för att köra modulen.

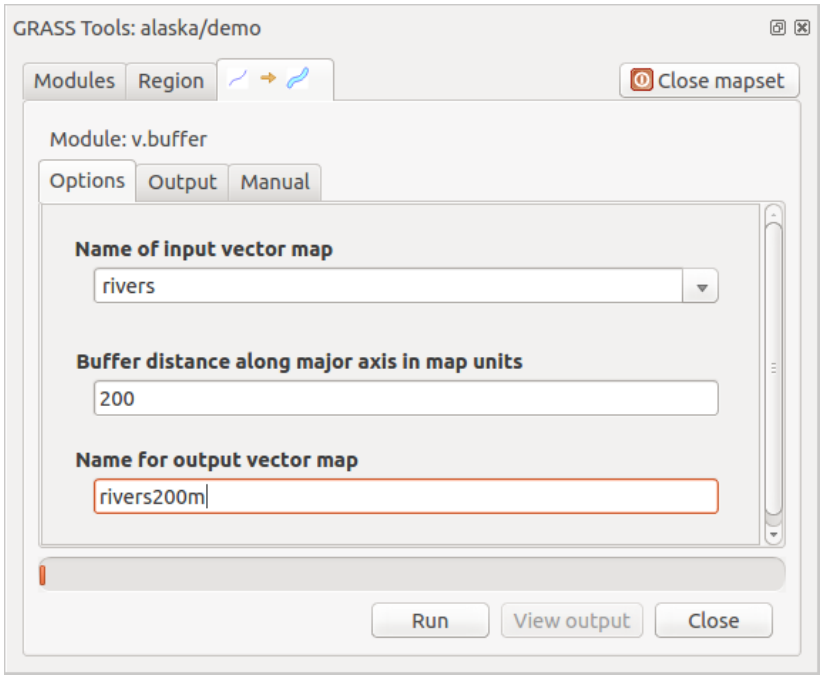


Fig. 22.4: Alternativ för GRASS Toolbox-modul

De medföljande modulparametrarna är ofta inte kompletta för att hålla dialogen enkel. Om du vill använda ytterligare modulparametrar och flaggor måste du starta GRASS-skalet och köra modulen på kommandoraden.

En ny funktion sedan QGIS 1.8 är stödet för en *Show Advanced Options*-knapp under den förenklade moduldialogen på *Options*-fliken. För tillfället är den bara tillagd i modulen `v.in.ascii` som ett exempel på användning, men den kommer förmodligen att vara en del av fler eller alla moduler i GRASS Toolbox i framtida versioner av QGIS. Detta gör att du kan använda de kompletta GRASS-modulalternativen utan att behöva byta till GRASS-skalet.

Utdata

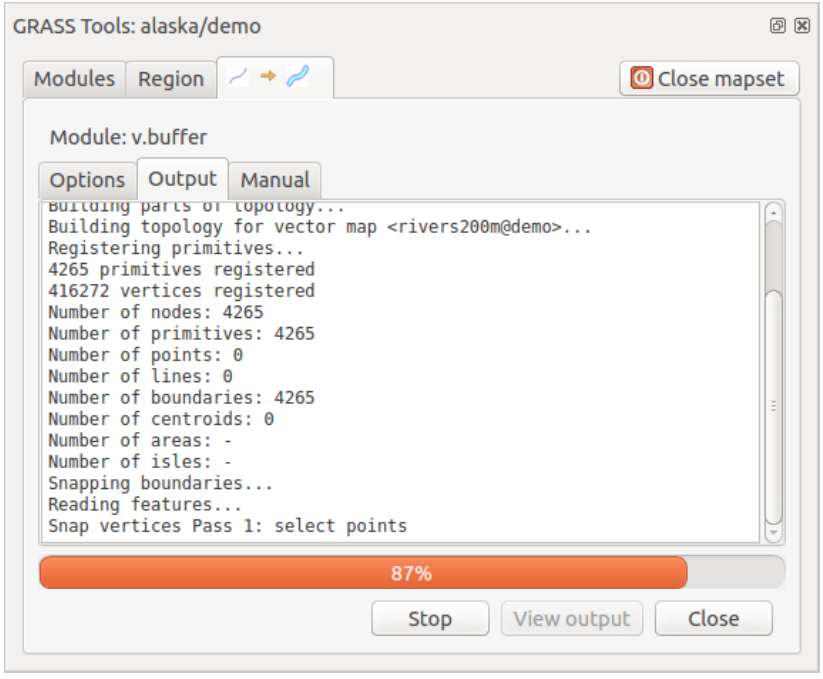


Fig. 22.5: GRASS Toolbox-modulens utdata

Fliken *Output* ger information om modulens utmatningsstatus. När du klickar på knappen *Run* växlar modulen till fliken *Output* och du ser information om analysprocessen. Om allt fungerar som det ska visas slutligen meddelandet `Successfully finished`.

Manual

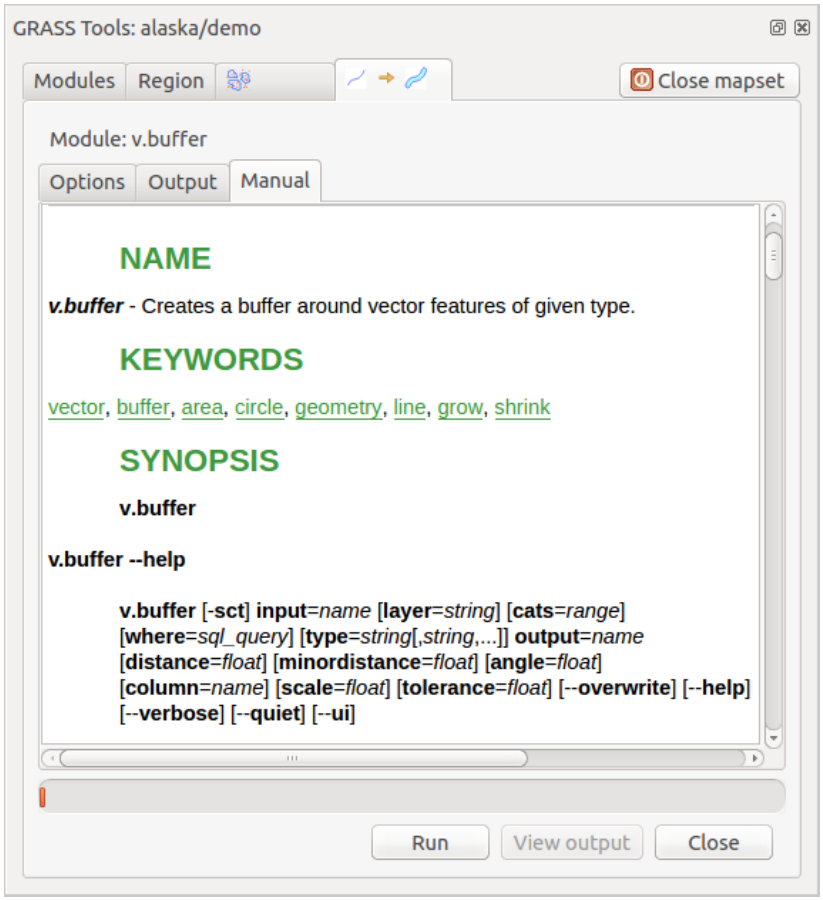


Fig. 22.6: GRASS Toolbox Modul Manual

Fliken *Manual* visar HTML-hjälpsidan för GRASS-modulen. Du kan använda den för att kontrollera ytterligare modulparametrar och flaggor eller för att få en djupare kunskap om modulens syfte. I slutet av varje sida i modulens manual finns länkar till `Main Help index`, `Thematic index` och `Full index`. Dessa länkar ger samma information som modulen `g.manual`.

Tips

Visa resultat omedelbart

Om du vill visa dina beräkningsresultat direkt i kartbilden kan du använda knappen ”View Output” längst ned på modulfliken.

22.14.2 Exempel på GRASS-moduler

Följande exempel visar hur kraftfulla några av GRASS-modulerna är.

Skapa konturlinjer

I det första exemplet skapas en vektorkonturkarta från ett höjdraster (DEM). Här förutsätts att du har Alaska `LOCATION` konfigurerad enligt beskrivningen i avsnitt *Importera data till en GRASS-lokalisering*.

- Först öppnar du platsen genom att klicka på knappen Open mapset och välja platsen Alaska.
- Öppna nu verktygslådan med knappen Öppna GRASS-verktyg.
- Dubbelklicka på *Raster* ► *Surface Management* ► *Generate vector contour lines* i listan över verktygskategorier.
- Nu kan du klicka på verktyget **r.contour** för att öppna verktygsdialogen enligt beskrivningen ovan (se *Arbeta med GRASS-moduler*).
- I *Namn på inmatad rasterkarta* skriver du `gtopo30`.
- Skriv in värdet 100 i fältet *Increment between Contour levels* . (Detta skapar konturlinjer med 100 meters intervall.)
- Skriv in namnet `ctour_100` i *Name for output vector map*.
- Klicka på *Run* för att starta processen. Vänta några ögonblick tills meddelandet `Successfully finished` visas i utdatafönstret. Klicka sedan på *View Output* och *Close*.

Eftersom detta är en stor region kommer det att ta ett tag att visa. När renderingen är klar kan du öppna fönstret för lageregenskaper och ändra linjefärgen så att konturerna syns tydligt över höjdrastret, som i *Dialogrutan Egenskaper för vektor*.

Zooma sedan in på ett litet bergsområde i mitten av Alaska. Om du zoomar in på nära håll kommer du att märka att konturerna har skarpa hörn. GRASS erbjuder verktyget **v.generalize** för att ändra vektorkartor något samtidigt som deras övergripande form bibehålls. Verktyget använder flera olika algoritmer med olika syften. Vissa av algoritmerna (t.ex. Douglas Peuker och Vertex Reduction) förenklar linjen genom att ta bort några av hörnen. Den resulterande vektorn laddas snabbare. Denna process är användbar när du har en mycket detaljerad vektor, men du skapar en mycket småskalig karta, så detaljerna är onödiga.

Tips

Det förenklade verktyget

Observera att QGIS har ett *Vector* ► *Geometry Tools* ► *Simplify geometries*-verktyg som fungerar precis som GRASS **v.generalize** Douglas-Peuker-algoritm.

Syftet med det här exemplet är dock annorlunda. De konturlinjer som skapats av `r.contour` har skarpa vinklar som bör jämnas ut. Bland algoritmerna **v.generalize** finns Chaikens, som gör just detta (även Hermite splines). Var medveten om att dessa algoritmer kan **lägga till** ytterligare hörn till vektorn, vilket gör att den laddas ännu långsammare.

- Öppna GRASS Toolbox och dubbelklicka på kategorierna *Vector* ► *Develop map* ► *Generalization*, klicka sedan på modulen **v.generalize** för att öppna dess alternativfönster.
- Kontrollera att vektorn 'ctour_100' visas som *Namn på indatavektor*.
- Välj Chaikens algoritm från listan över algoritmer. Låt alla andra alternativ vara standardinställningar och bläddra ner till sista raden för att i fältet *Namn på utdataväxelkartan* ange 'ctour_100_smooth' och klicka på *Kör*.
- Processen tar flera ögonblick. När `Successfully finished` visas i utdatafönstret klickar du på *View Output* och sedan på *Close*.

- Du kan ändra vektorfärgen för att visa den tydligt mot rasterbakgrunden och för att skapa kontrast mot de ursprungliga konturlinjerna. Du kommer att märka att de nya konturlinjerna har mjukare hörn än originalet, samtidigt som den ursprungliga övergripande formen bibehålls.

Fig. 22.7: GRASS-modul v.generalize för att jämna ut en vektorkarta

Tips

Andra användningsområden för r.contour

Det förfarande som beskrivs ovan kan användas i andra motsvarande situationer. Om du t.ex. har en rasterkarta med nederbördsdata kan samma metod användas för att skapa en vektorkarta med isohyetallinjer (konstant nederbörd).

Skapa en terrängskuggning med 3-D-effekt

Flera metoder används för att visa höjdlager och ge kartor en 3D-effekt. Konturlinjer, som visas ovan, är en populär metod som ofta används för att framställa topografiska kartor. Ett annat sätt att visa en 3-D-effekt är genom terrängskuggning. Terrängskuggningseffekten skapas från ett DEM-raster (höjd) genom att först beräkna lutningen och aspekten för varje cell, sedan simulera solens position på himlen och ge varje cell ett reflektansvärde. På så sätt får du solvända sluttningar upplysta; sluttningar som vetter bort från solen (i skugga) blir mörkare.

- Börja detta exempel med att ladda höjdrastret `gtopo30`. Starta GRASS Toolbox, och under kategorin Raster, dubbelklicka för att öppna *Spatial analysis* ► *Terrain analysis*.
- Klicka sedan på **r.shaded.relief** för att öppna modulen.
- Ändra *azimuth angle* 1,00 270 till 315.
- Ange `gtopo30_shade` för det nya terrängskuggningsrastret och klicka på *Kör*.
- När processen är klar lägger du till terrängskuggningsrastret på kartan. Du bör se att det visas i gråskala.
- För att se både terrängskuggning och färgerna i `gtopo30` tillsammans, flytta terrängskuggningskartan under `gtopo30` kartan i innehållsförteckningen, öppna sedan *Egenskaper*` fönstret för `gtopo30`, växla till *Transparens*`-fliken och ställ in dess transparensnivå till ca 25%.

Du bör nu ha `gtopo30`-höjden med dess färgkarta och transparensinställning visad **över** den gråskaliga terrängskuggningskartan. För att se de visuella effekterna av terrängskuggning, stäng av kartan `gtopo30_shade` och slå sedan på den igen.

Användning av GRASS-skalet

GRASS-tillägget i QGIS är utformat för användare som är nya i GRASS och inte känner till alla moduler och alternativ. Därför visar vissa moduler i verktygslådan inte alla tillgängliga alternativ, och vissa moduler visas inte alls. GRASS-skalet (eller konsolen) ger användaren tillgång till de ytterligare GRASS-moduler som inte visas i Toolbox-trädet, och även till några ytterligare alternativ för de moduler som finns i Toolbox med de enklaste standardparametrarna. Detta exempel demonstrerar användningen av ett ytterligare alternativ i **r.shaded.relief**-modulen som visades ovan.

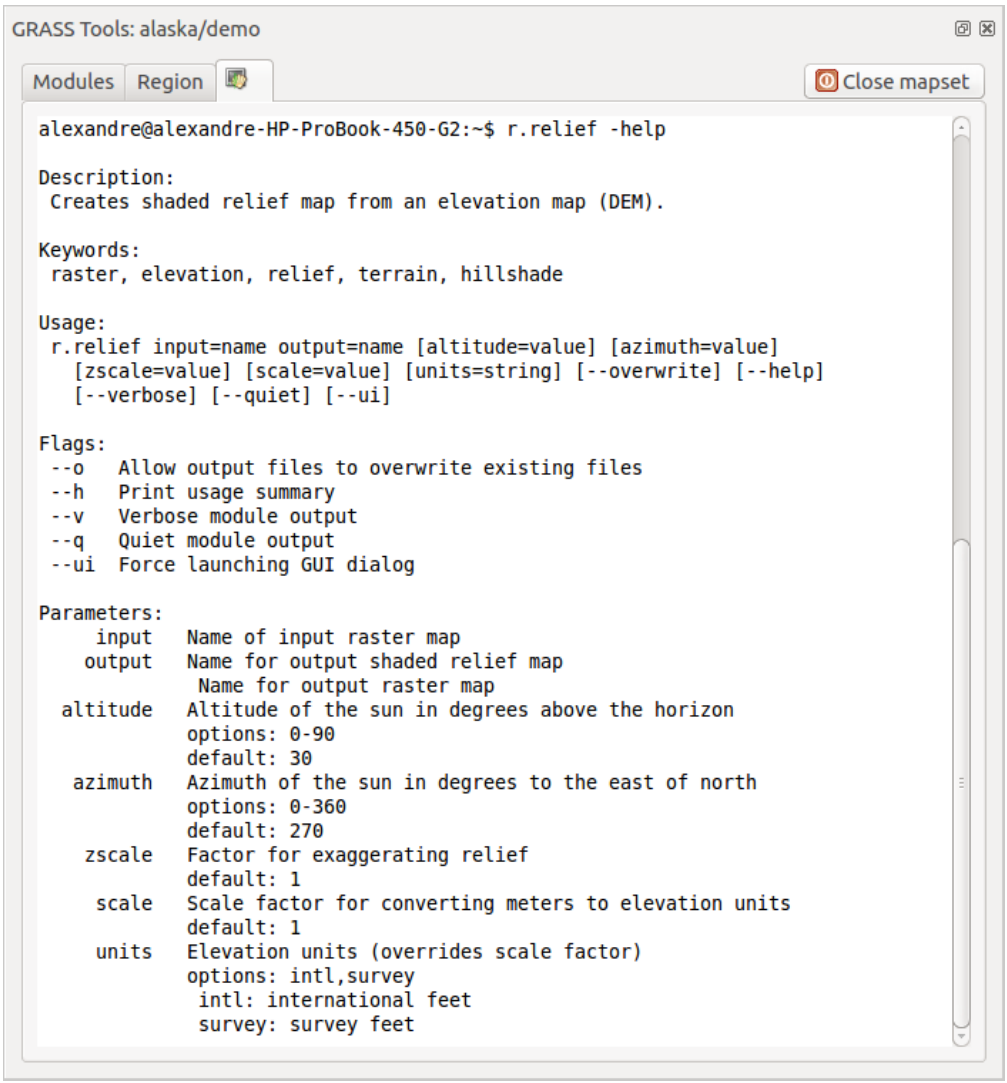


Fig. 22.8: GRASS-skalet, modulen r.shaded.relief

Modulen **r.shaded.relief** kan ta en parameter `zmult`, som multiplicerar höjdvärdena i förhållande till X-Y-koordinatenheterna så att terrängskuggningseffekten blir ännu mer uttalad.

- Ladda höjdrastret `gtopo30` enligt ovan, starta sedan GRASS Toolbox och klicka på GRASS-skalet. I skalfönstret skriver du kommandot `r.shaded.relief map=gtopo30 shade=gtopo30_shade2 azimuth=315 zmult=3` och trycker på `Enter`.
- När processen är klar växlar du till fliken *Browse* och dubbelklickar på det nya `gtopo30_shade2`-rastret för att visa det i QGIS.
- Som förklarats ovan flyttar du det skuggade reliefrastret under `gtopo30`-rastret i innehållsförteckningen och kontrollerar sedan transparensen i det färgade `gtopo30`-lagret. Du bör se att 3D-effekten framträder tydligare jämfört med den första skuggade reliefkartan.

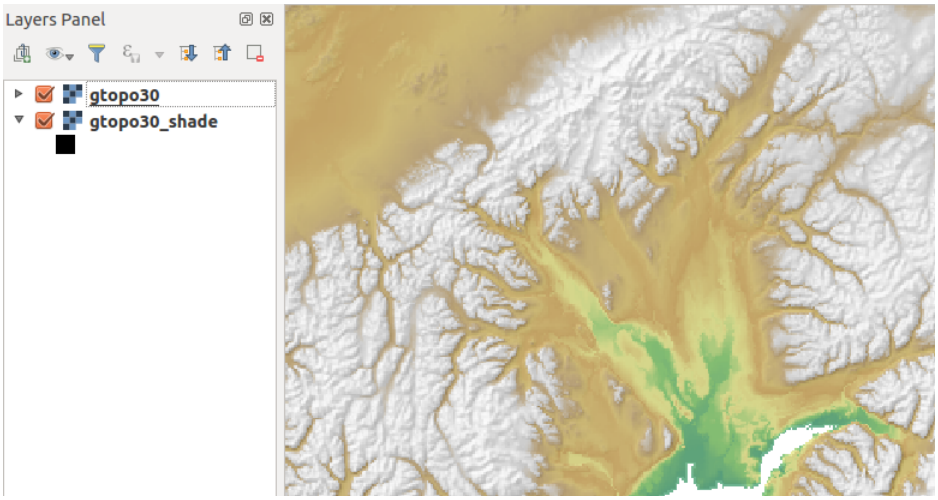


Fig. 22.9: Visning av skuggad relief skapad med GRASS-modulen r.shaded.relief

Rasterstatistik i en vektorkarta

Nästa exempel visar hur en GRASS-modul kan sammanställa rasterdata och lägga till kolumner med statistik för varje polygon i en vektorkarta.

- Använd Alaska-data igen, se *Importera data till en GRASS-lokalisering* för att importera `shapefiles/trees.shp`-filen till GRASS.
- Nu krävs ett mellanliggande steg: centroider måste läggas till den importerade trädkartan för att göra den till en komplett GRASS-områdesvektor (inklusive både gränser och centroider).
- I verktygslådan väljer du *Vector* ► *Manage features* och öppnar modulen **v.centroids**.
- Ange som *output vector map* 'forest_areas' och kör modulen.
- Ladda nu vektorn `forest_areas` och visa skogstyperna - lövskog, vintergrön skog, blandskog - i olika färger: I lagrets fönster *Egenskaper*, fliken *Symbologi*, välj från *Legend type* 'Unikt värde' och ställ in fältet *Klassificering* till 'VEGDESC'. (Se förklaringen av symbologifliken i *Egenskaper för symbologi* i vektoravsnittet)
- Därefter öppnar du GRASS Toolbox igen och öppnar *Vector* ► *Vector update* med andra kartor.
- Klicka på modulen **v.rast.stats**. Ange `gtopo30` och `forest_areas`.
- Endast en ytterligare parameter behövs: Ange *kolumnprefix* `elev`, och klicka på *Run*. Det här är en beräkningstung operation som kommer att pågå under lång tid (troligen upp till två timmar).
- Öppna slutligen attributtabellen `forest_areas` och kontrollera att flera nya kolumner har lagts till, inklusive `elev_min`, `elev_max`, `elev_mean`, etc., för varje skogspolygon.

22.14.3 Anpassa GRASS verktygslåda

Nästan alla GRASS-moduler kan läggas till i GRASS Toolbox. Ett XML-gränssnitt tillhandahålls för att tolka de ganska enkla XML-filer som konfigurerar modulernas utseende och parametrar i verktygslådan.

Ett exempel på en XML-fil för att generera modulen `v.buffer` (v.buffer.qgm) ser ut så här:

```
<?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"/>
```

(continues on next page)

(fortsättning från föregående sida)

```
        <option key="output" />
</qgisgrassmodule>
```

Parsern läser denna definition och skapar en ny flik i verktygslådan när du väljer modulen.

KAPITEL 23

Ramverk för bearbetning av QGIS

23.1 Introduktion

I det här kapitlet introduceras QGIS Processing Framework, en geoprocessmiljö som kan användas för att anropa egna algoritmer och algoritmer från tredje part från QGIS, vilket gör dina spatiala analysuppgifter mer produktiva och enkla att utföra.

Som ett *Core-plugin* är Processing installerat som standard men du måste aktivera det:

1. Gå till *Tillägg* ► *Hantera och installera tillägg…*
2. Klicka på fliken *Installed* till vänster
3. Markera rutan bredvid posten *Processing*
4. Stäng dialogrutan.

   En *Processing*-meny finns nu tillgänglig i den övre menyraden. Därifrån kan du nå de viktigaste komponenterna i detta ramverk.

I de följande avsnitten går vi igenom hur du använder de grafiska elementen i detta ramverk och får ut mesta möjliga av vart och ett av dem.

Det finns fyra grundläggande element i ramverkets grafiska gränssnitt, som används för att köra algoritmer för olika ändamål. Valet av det ena eller andra verktyget beror på vilken typ av analys som ska utföras och de särskilda egenskaperna hos varje användare och projekt. Alla (utom gränssnittet för batchbearbetning, som anropas från verktygslådan eller dialogrutan för algoritmkörning, som vi kommer att se) kan nås från menyalternativet *Processing* (du kommer att se fler poster; de återstående används inte för att köra algoritmer och förklaras senare i detta kapitel).

- Den *verktygslådan*: Huvudelementet i GUI, används för att köra en enskild algoritm eller köra en batchprocess baserad på den algoritmen.

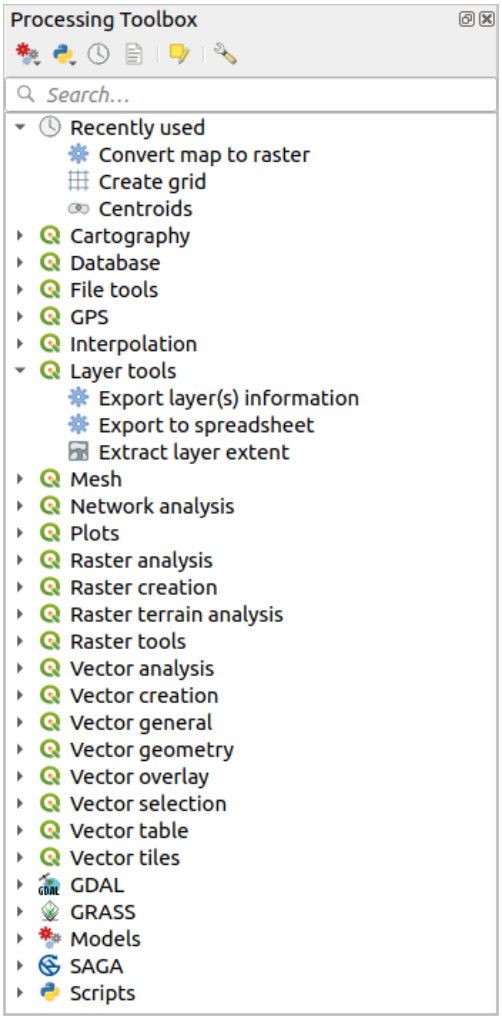


Fig. 23.1: Geobearbetning verktygslåda

- Den *Model Designer*: Flera algoritmer kan kombineras grafiskt med hjälp av modelleraren för att definiera ett arbetsflöde, vilket skapar en enda process som involverar flera delprocesser.

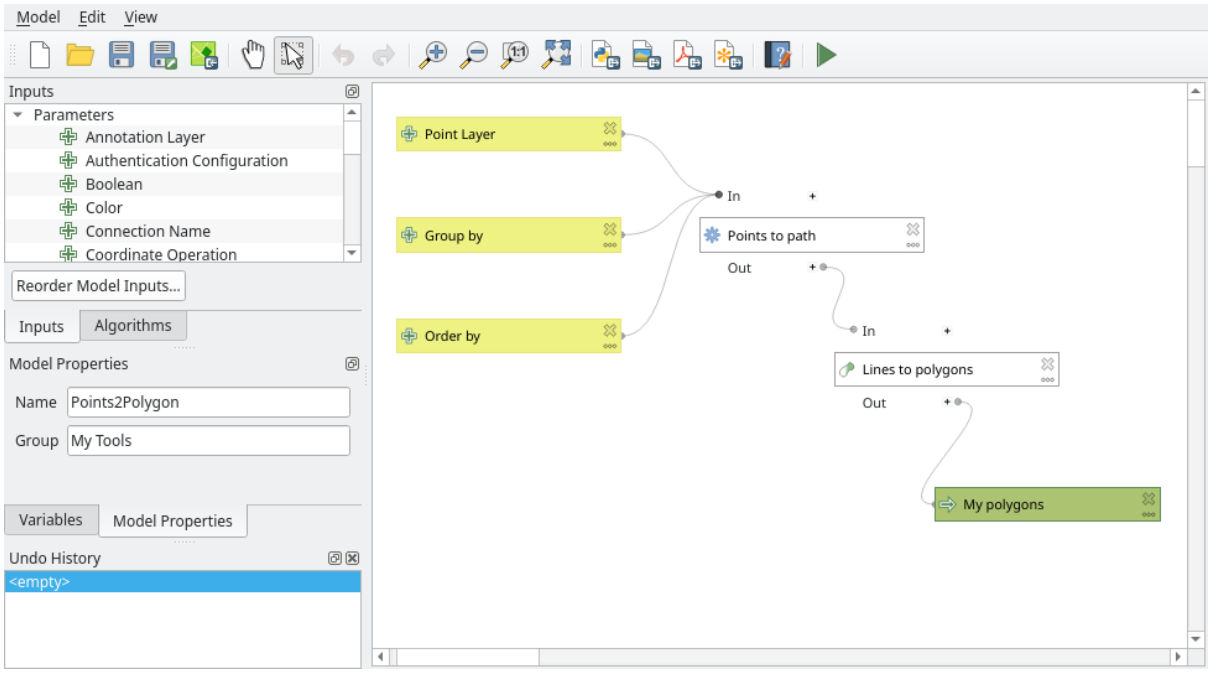


Fig. 23.2: Geobearbetning Modellbyggare

- Hanteraren *History*: Alla åtgärder som utförs med hjälp av något av de ovan nämnda elementen lagras i en historikfil och kan senare enkelt återskapas med hjälp av historikhanteraren.

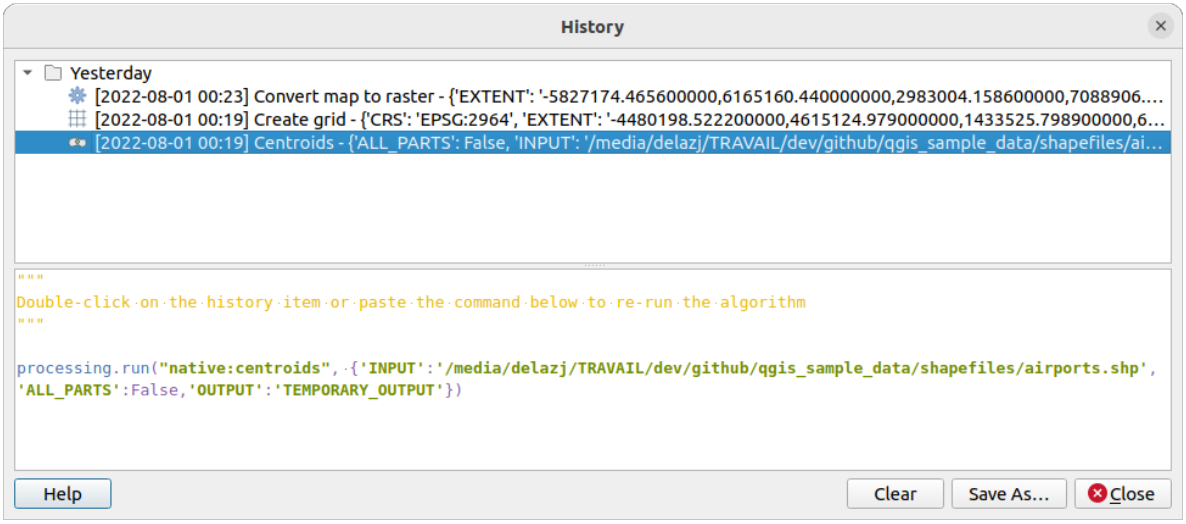


Fig. 23.3: Bearbetningshistorik

- Gränssnittet *Batch Processing*: Med det här gränssnittet kan du köra batchprocesser och automatisera körningen av en enda algoritm på flera dataset.

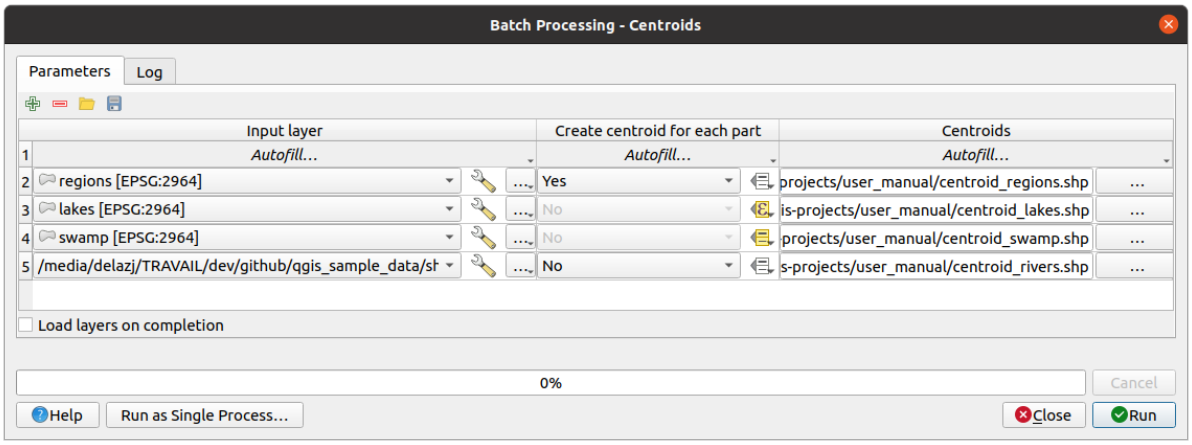


Fig. 23.4: Gränssnitt för batchbearbetning

I de följande avsnitten kommer vi att gå igenom vart och ett av dessa element i detalj.

23.2 Konfigurera ramverket för bearbetning

I menyn Processing Options (fliken *Inställningar*► *Alternativ* ► *Processing*) kan du konfigurera hur algoritmerna fungerar. Konfigurationsparametrarna är strukturerade i separata block som du kan välja på vänster sida av dialogrutan.

23.2.1 Allmänt

Blocket *General* innehåller standardinställningarna för att styra hur algoritmdialogen samt inmatnings- och utdataparametrar ska bete sig. Vissa av inställningarna kan dock åsidosättas på algoritmnivå, per *algoritmkörning* eller för *individuella parametrar*.

- *Default output raster layer extension* är som standard `tif`
- *Default output vector layer extension* är som standard `gpkg`
- *Invalid features filtering* vid exekvering av algoritm:
  - *Filtrera inte (bättre prestanda)*: alla funktioner (med giltiga och ogiltiga geometrier) bearbetas, men resultatet kan bli felaktigt beroende på hur geometrins ogiltighet påverkar operationerna
  - *Skippa (ignorera) funktioner med ogiltig geometri*, vilket innebär att endast en delmängd av din dataset (funktioner med giltig geometri) kommer att bearbetas
  - *Stop algorithm execution when a geometry is invalid*: du måste spåra och fixa de ogiltiga geometrierna om du vill att algoritmen ska bearbeta hela lagret. Algoritmer som *Kontrollera giltighet* eller *Fixa geometrier* kan hjälpa dig att uppnå detta.

  Inställningen *Invalid features filtering* kan åsidosättas för varje inmatning, vid algoritmens körtid.
- *Låt dialogrutan vara öppen efter att algoritmen körts*. När en algoritm har körts färdigt och dess utdatalager har laddats in i QGIS-projektet stängs algoritmdialogen. Om du vill hålla den öppen (för att köra algoritmen igen med andra parametrar eller för att bättre kontrollera utdata som skrivs till loggfliken) markerar du det här alternativet.
- *Max antal trådar*

- *Output folder* för icke temporära utdata: Om ingen mappsökväg anges för utdata från bearbetningskörningen är detta den mapp där de sparas. Standard är `processing/outputs` under den aktiva *användarprofilen* katalogen.
- *Overskriv sökvägen till mappen för temporära utdata*: Temporära utdata sparas som standard i mappen `tmp` på maskinen. Detta alternativ hjälper dig att ange en annan plats för lagring.
- *Pre-execution script* och *Post-execution script*. Dessa parametrar pekar på filer som innehåller skript som skrivits med hjälp av skriptfunktionen för bearbetning, som beskrivs i avsnittet om skript och konsolen.
- *Företräda utdatafilnamn för lagernamn*. Namnet på varje resulterande lager som skapas av en algoritm definieras av algoritmen själv. I vissa fall kan ett fast namn användas, vilket innebär att samma utdatanamn kommer att användas oavsett vilket inmatningslager som används. I andra fall kan namnet bero på namnet på inmatningslagret eller på några av de parametrar som används för att köra algoritmen. Om den här kryssrutan är markerad kommer namnet i stället att hämtas från utdatafilens namn. Observera att om utdata sparas i en temporär fil, är filnamnet för denna temporära fil vanligtvis långt och meningslöst för att undvika kollision med andra redan existerande filnamn.
- *Resultats gruppnamn*. Om du vill få fram alla bearbetningsresultatskikt i en grupp i panelen *Layers*, anger du ett gruppnamn för denna parameter. Gruppen kan redan existera eller inte. QGIS kommer att lägga till alla utmatningslager i en sådan grupp. Som standard är denna parameter tom, så alla utmatningslager läggs till på olika platser i panelen *Layers*, beroende på vilket objekt som är aktivt när en algoritm körs. Observera att utdatalager laddas till panelen *Layers* endast om *Open output file after running algorithm* är markerad i algoritmdialogen.
- *Visa algoritmer med kända problem*: Som standard undviker QGIS att visa trasiga algoritmer (vanligtvis från tredjepartsleverantörer). Om det är markerat kommer de att finnas tillgängliga i verktygslådan Processing, med en varningsikon och en verktygstips som förklarar att de har problem. Använd på egen risk.
- *Visa antal funktioner för utdatavektorlager*. Eftersom det kan ta lite tid att beräkna antalet funktioner i vissa dataformat är det här alternativet avaktiverat som standard.
- *Visa CRS-definition för lager i urvalsrutor*
- *Visa verktygstips när det finns inaktiverade leverantörer*
- *Style for line layers*, *Style for point layers*, *Style for polygons layers* och *Style for raster layers* används för att ange standardrenderingsstilen för utmatningslager (dvs. lager som genereras av bearbetningsalgoritmer). Skapa bara den stil du vill ha med QGIS, spara den i en fil och ange sedan sökvägen till den filen i inställningarna så att algoritmerna kan använda den. När ett lager laddas av Processing och läggs till i QGIS-rutan, kommer det att återges med den stilen.

  Renderingsstilar kan konfigureras individuellt för varje algoritm och var och en av dess utgångar. Högerklicka bara på algoritmens namn i verktygslådan och välj *Edit rendering styles for outputs*. Du kommer att se en dialogruta som den som visas här intill.

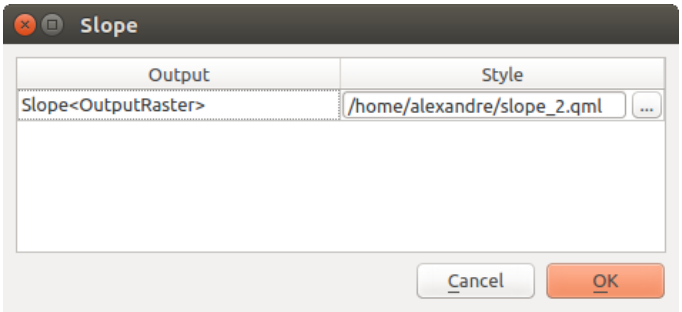


Fig. 23.5: Renderingsstilar

  Välj den stilfil (`.qml`) som du vill ha för varje utdata och tryck på *OK*.
- *Varning före exekvering om parameterns CRS inte matchar*: Som standard omprojicerar QGIS inbyggda algoritmer (dvs. de som listas i gruppen *Menus* ► *QGIS (native C++)*) de ingående lagren till det första

lagrets CRS före exekvering. Markera detta alternativ för att få ett meddelande från de andra verktygen som inte stöder omprojicering, när CRS för indata inte är identiska. Tredjepartsleverantörer berörs inte.

23.2.2 Menyer

Meny-blocket styr om en algoritm, ett skript eller en modell (inbyggd eller tillhandahållen av tillägg) ska göras tillgänglig via en särskild meny eller ett verktygsfält (tillsammans med Processing Toolbox). För varje objekt i varje leverantör kan du:

- *Add button in toolbar*, vilket gör den tillgänglig i verktygsfältet *Processing Algorithms*
- tilldela en *Icon* till algoritmen
- ange en *Menu path*: algoritmen kommer då att vara tillgänglig via en befintlig eller en anpassad meny, t.ex. `Vect&or/MyTopAlgorithms` kommer att lägga till en genväg i en *MyTopAlgorithms* undermeny i standardmenyn *Vector*.

Starta om QGIS för att tillämpa inställningarna. När som helst kan dina ändringar *Återställas till standardvärden*.

23.2.3 Modeller och manuskript

I blocken *Models* och *Scripts* anges en standardsökväg, som anges under den aktiva mappen *User profile*, för lagring av modeller respektive skript. Precis som för de andra alternativen kan du ändra sökvägen, t.ex. så att den leder till en fjärrdator eller delad mapp för alla dina användare.

23.2.4 Leverantörer

Du hittar också ett block för algoritmen *Leverantörer*. Det här är platsen där installerade leverantörer exponerar sina inställningar.

Som standard installeras QGIS med två algoritmleverantörer:

- *GDAL* vars algoritmer du kan *Aktivera* (eller inte) i den här dialogen. Läs mer om *GDAL-algoritmer*.
- *GRASS* vars algoritmer kräver installation av *GRASS GIS Processing Provider* från *Tillägg Manager*. Du kan sedan ställa in några dedikerade inställningar som t.ex:
  - *För rasterlager, använd r.external (snabbare) istället för r.in.gdal*
  - *För vektorlager, använd v.external (snabbare) istället för v.in.ogr*
  - *Location of GRASS docs*: som standard, och beroende på ditt operativsystem, använder QGIS en hårdkodad lista med sökvägar för att hitta den lokala GRASS-dokumentationen och öppna den när du trycker på *Help*-knappen i en GRASS-algoritmdialog. Om du fyller i detta alternativ åsidosätts standardsökvägen, så att du kan använda en anpassad plats eller fjärrdokumentation.
  - *Logga konsolutdata*
  - *Logga exekveringskommandon*

Installerade tillägg som tillhandahåller bearbetningsalgoritmer kan också ha sin leverantör listad i den här gruppen, med anpassade inställningar.

23.3 Verktygslådan

Processing Toolbox är huvudelementet i bearbetnings-GUI:t och det som du sannolikt kommer att använda i ditt dagliga arbete. Den visar listan över alla tillgängliga **algoritmer** grupperade i olika block som kallas *Providers*, och anpassade **modeller** och **skript** som du kan lägga till för att utöka verktygsuppsättningen. Verktygslådan är alltså en åtkomstpunkt för att köra dem, antingen som en enda process eller som en batchprocess som omfattar flera körningar av samma algoritm på olika uppsättningar indata.

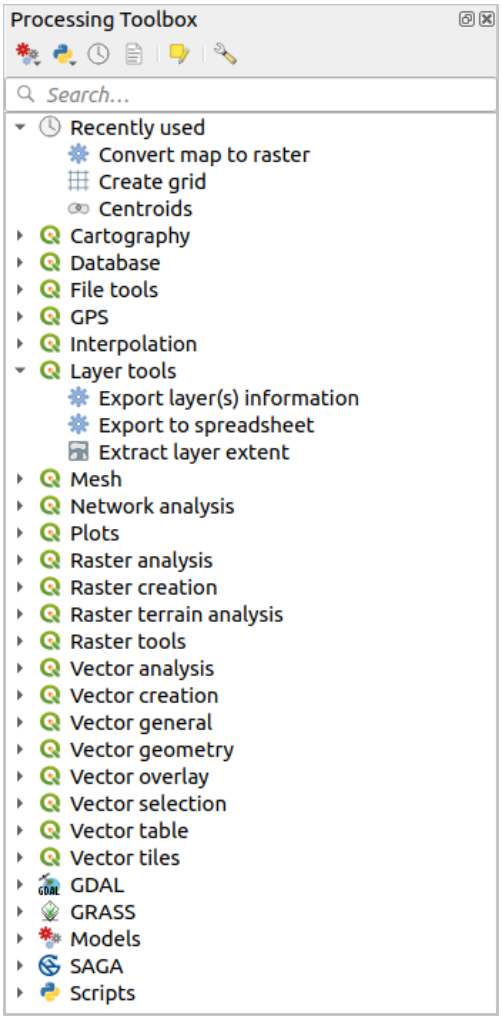


Fig. 23.6: Geobearbetning verktygslåda

Providers kan (av)aktiveras i dialogrutan *Processing settings*. Som standard är endast leverantörer som inte förlitar sig på tredjepartsprogram (dvs. de som endast kräver att QGIS-element körs) aktiva. Algoritmer som kräver externa program kan behöva ytterligare konfiguration. Konfigurering av providers förklaras i ett *längre kapitel* i den här handboken.

I den övre delen av verktygslådan hittar du en uppsättning verktyg för att:

- arbeta med Modeller: *Skapa ny modell…*, *Öppna befintlig modell…* och *Add Model to Toolbox…*;
- arbeta med Skript: *Skapa nytt skript…*, *Skapa nytt skript från mall…*, *Öppna befintligt skript…* och *Add Script to Toolbox…*;
- öppna panelen Historik;
- öppna panelen Resultatvisare;

- växla verktygslådan till *modifieringsläget på plats* med knappen Edit Features In-Place: endast de algoritmer som är lämpliga att utföras på det aktiva lagret utan att ett nytt lager matas ut visas;
- öppna dialogrutan Alternativ.

Under verktygsfältet finns en sökruta *Sök…* som hjälper dig att enkelt hitta de verktyg du behöver. Du kan skriva in valfritt ord eller fras i textrutan. Lägg märke till att antalet algoritmer, modeller eller skript i verktygslådan reduceras när du skriver till dem som innehåller den text du har skrivit i deras namn eller nyckelord.

Överst i algoritmlistan visas de mest använda verktygen Recently Used och verktygen Favorites, vilket gör det bekvämt att köra dem igen. Listan Recently Used finns där som standard och visar de algoritmer som du nyligen har kört. Listan Favorites skapas när du manuellt lägger till en algoritm i den.

Du kan högerklicka på valfri algoritm i verktygslådan och välja från den kontextuella menyn att:

- *Execute…* för att köra algoritmen.
- *Execute as Batch Process…* för att köra algoritmen i batch-läge.
- *Edit Rendering Styles for Outputs…* för att anpassa renderingsstilarna för algoritmens utdata.
- *Add to Favorites* för att lägga till algoritmen i favoritlistan.
- *Remove from Favorites* för att ta bort algoritmen från favoritlistan.

Till skillnad från listan ”Recently Used” är listan ”Favorites” statisk och påverkas inte av vilka algoritmer som körts.

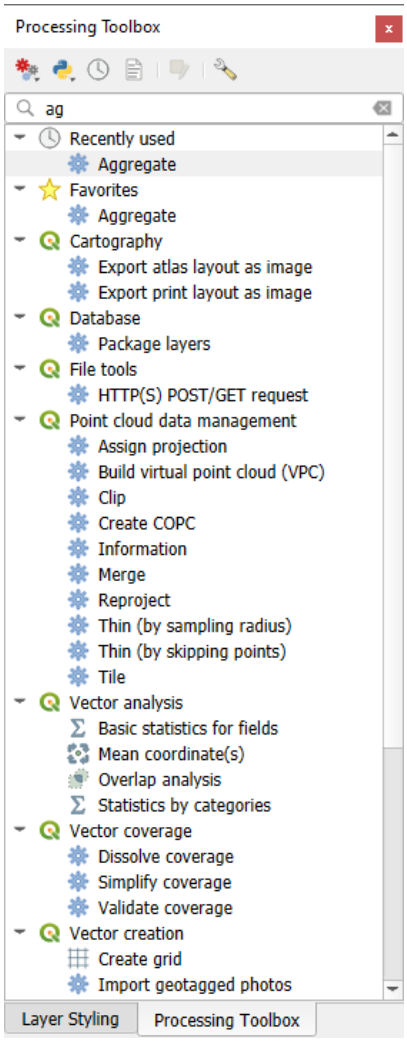


Fig. 23.7: Processing Toolbox visar sökresultat

För att köra ett verktyg dubbelklickar du bara på dess namn i verktygslådan.

23.3.1 Dialogrutan för algoritmen

När du dubbelklickar på namnet på den algoritm som du vill köra visas en dialog som liknar den i Fig. 23.8 nedan (i det här fallet motsvarar dialogen algoritmen `Centroids`). Dialogens titel kommer att innehålla gruppnamnet från vilket algoritmen härstammar.

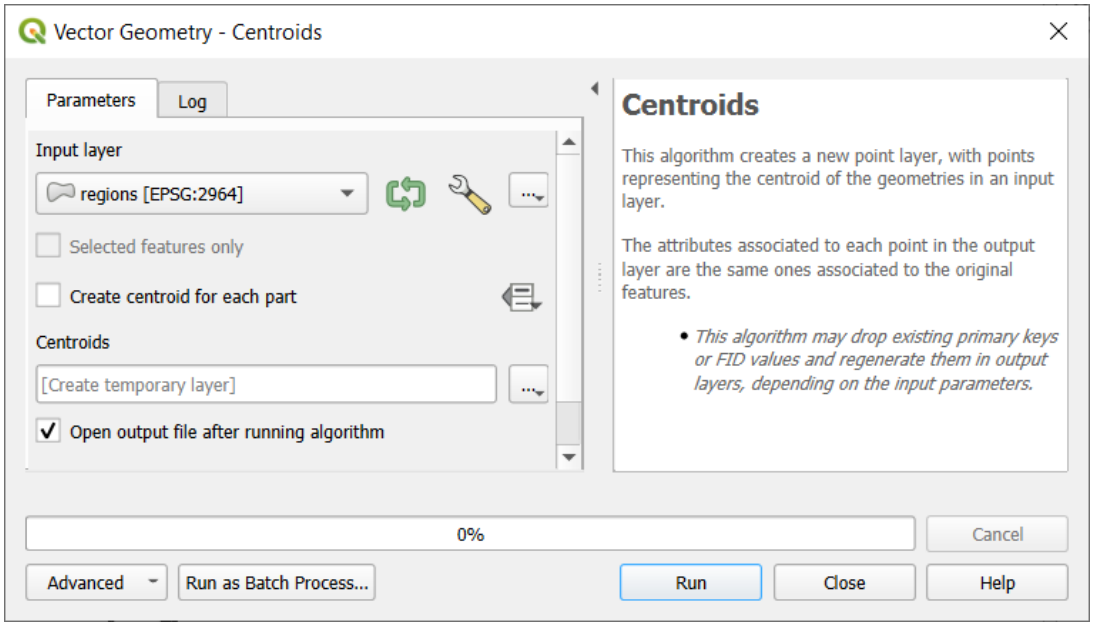


Fig. 23.8: Algoritmdialog - Parametrar

Dialogrutan har två flikar (*Parameters* och *Log*) till vänster, en algoritmbeskrivning till höger och en uppsättning knappar längst ned.

Parametertyper

Fliken *Parameters* används för att ställa in de ingångsvärden som algoritmen behöver för att kunna köras. Den visar en lista över de ingångsvärden och konfigurationsparametrar som ska ställas in. Den har naturligtvis olika innehåll beroende på kraven för den algoritm som ska köras, och skapas automatiskt utifrån dessa krav.

Tips

Ange dina egna standardvärden för algoritmparametrar

Algoritmdialogrutorna öppnas med vissa parametrar förfyllda med värden från QGIS-installationen. Det är dock möjligt att ställa in *Dina egna standardvärden* för specifika algoritmparametrar så att de används vid algoritmens start.

Även om antalet och typen av parametrar beror på algoritmens egenskaper, är strukturen likartad för dem alla. De parametrar som finns i tabellen kan vara av någon av följande typer.

- Ett **vektorlager**, att välja från en lista över alla tillgängliga (för närvarande öppnade) vektorlager i QGIS. Du kan också använda oladdade lager: tryck på …-knappen på widgetens högra sida och välj:
  - *Välj fil…*: väljer fil på disken med hjälp av operativsystemets filutforskare
  - *Browse for layer…*: öppnar *Browser panel*, vilket gör det möjligt att ta lager direkt från databaskällor (PostgreSQL, SQL Server, Oracle, …), webbtjänster (WFS, AFS, …) eller filer på disken.

Fig. 23.9: Widget för vektorinmatning

> **Observera**
>
> Som standard visar lagerwidgeten lagrets CRS tillsammans med dess namn. Om du inte vill se den här extra informationen kan du inaktivera den här funktionen i dialogrutan Processing Settings genom att avmarkera alternativet *General* ► *Show layer CRS definition in selection boxes*.

Widgeten för vektorinmatning har också följande funktioner:

- en iterator -knapp: Om den är aktiverad kommer algoritmen att köras iterativt på var och en av dess funktioner, istället för bara en gång för hela lagret, vilket ger lika många utdata som gånger algoritmen körs. Detta gör det möjligt att automatisera processen när alla funktioner i ett lager måste bearbetas separat. Om algoritmen innehåller flera ingångsvektorer som du kan iterera över, kommer iterationen endast att bearbetas på den första växlade parametern, i den ordning som parametrarna deklareras i algoritmen.
- Avancerade alternativ-knappen för att justera inställningarna som ska användas för den specifika parametern. Dessa inställningar gäller:
  - *Invalid feature filtering*: gör att *standardmetoden* för hantering av objekt med ogiltiga geometrier kan åsidosättas
  - *Limit features processed*: valfri gräns för antalet funktioner som bearbetas från källan
  - *Feature filter*: gör det möjligt att ange ett uttryck för att undergruppera lagret dynamiskt när verktyget körs, vilket undviker behovet av separata steg för att ställa in lagerfilter eller skapa lagerundergrupper.

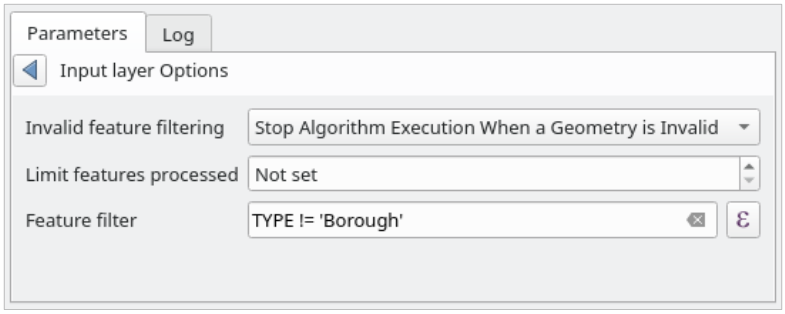


Fig. 23.10: Avancerade alternativ för widget för vektorinmatning

  - Det är också möjligt att begränsa algoritmkörningen på vektorlagret till dess *Selected features only*.
- En **tabell**, att välja från en lista över alla tillgängliga i QGIS. Icke-spatiala tabeller laddas in i QGIS som vektorlager och använder *same widget*.
- Ett **rasterlager**, för att välja från en lista över alla rasterlager som finns tillgängliga i QGIS. Väljaren innehåller också en …-knapp på höger sida, så att du kan välja filnamn som representerar lager som för närvarande inte är inlästa i QGIS.

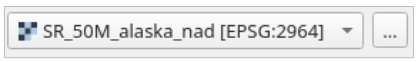


Fig. 23.11: Widget för inmatning av raster

- Ett **alternativ**, att välja från en lista med möjliga alternativ.
- Ett **numeriskt värde** som ska anges i en spinnruta. I vissa sammanhang (när parametern gäller på objektnivå och inte på lagernivå) finns det en knapp Data-defined override vid sidan om, så att du kan öppna *expression builder* och ange ett matematiskt uttryck för att generera variabelvärden för parametern. Vissa användbara variabler som är relaterade till data som laddats in i QGIS kan läggas till i uttrycket, så att du kan välja ett värde

som härrör från någon av dessa variabler, t.ex. cellstorleken för ett lager eller den nordligaste koordinaten för ett annat.

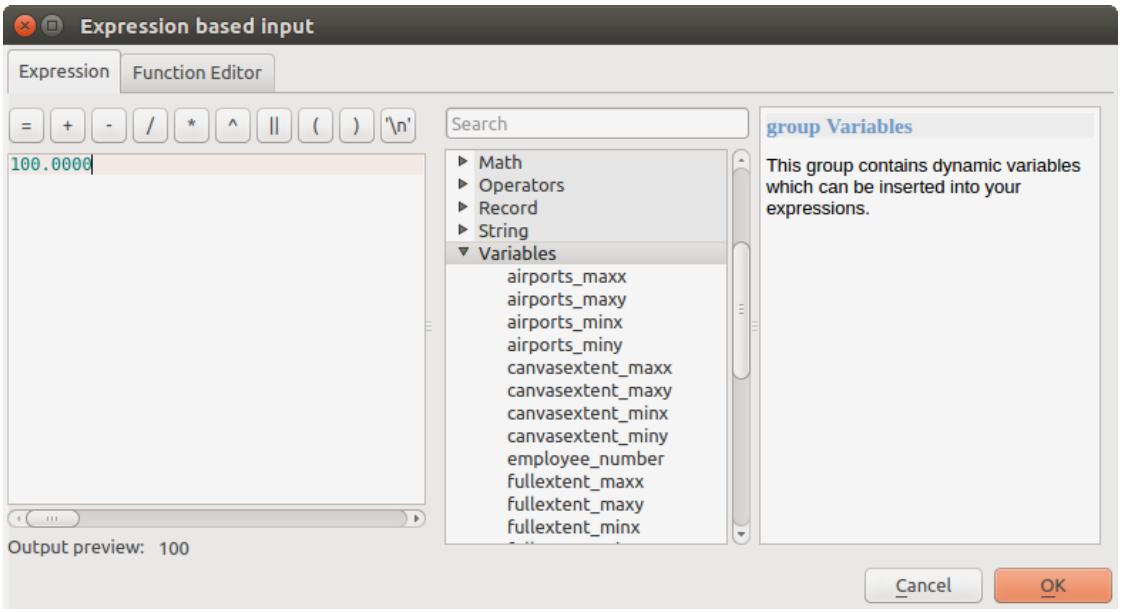


Fig. 23.12: Uttrycksbaserad indata

- Ett **intervall**, med min- och maxvärden som ska anges i två textrutor.
- En **textsträng** som ska läggas in i en textruta.
- Ett **fält**, för att välja mellan attributtabellen för ett vektorlager eller en enskild tabell som valts i en annan parameter.
- Ett **koordinatreferenssystem**. Du kan välja det bland de senast använda i listrutan eller i dialogrutan *CRS selection* som visas när du klickar på knappen till höger.
- En **extent**, en textruta som definierar en rektangel genom dess hörnkoordinater i formatet `xmin, xmax, ymin, ymax`. Tryck på knappen :sup:`Set to current map canvas extent` för att använda kartans utbredning. Om du klickar på pilen till höger om värdeväljaren visas en popup-meny som ger dig möjlighet att:
  - *Beräkna från lager* ►: fyller textrutan med koordinaterna för begränsningsrutan för ett lager som ska väljas bland de inlästa
  - *Beräkna från layoutkarta* ►: fyller textrutan med koordinaterna för ett kartobjekt som valts från en layout i det aktuella projektet
  - *Beräkna från bokmärke* ►: fyller textrutan med koordinaterna för ett sparat bokmärke
  - *Använd den aktuella kartans omfattning*
  - *Draw on canvas*: parameterfönstret döljer sig så att du kan klicka och dra på duken. När du har definierat utbredningens rektangel visas dialogrutan igen och innehåller värdena i textrutan för utbredning.

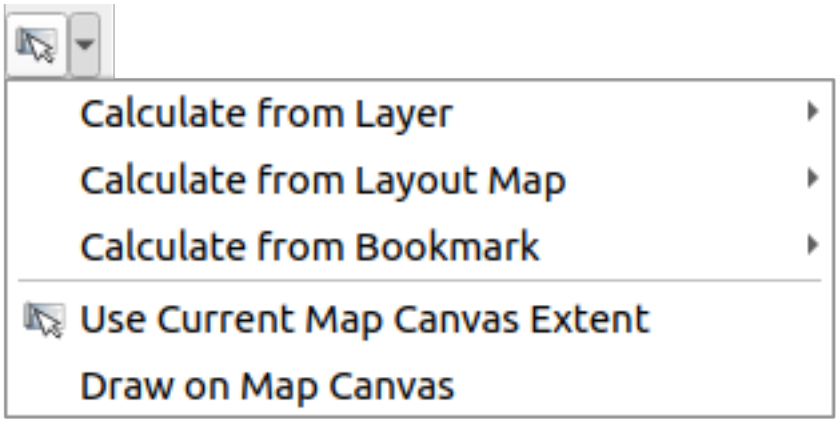


Fig. 23.13: Extentväljare

- En **lista med element** (raster- eller vektorlager, tabeller, fält) att välja mellan. Klicka på …-knappen till vänster om alternativet för att se en dialogruta som den följande. Flera val är tillåtna och när dialogrutan stängs visas antalet valda objekt i widgeten för parametertextrutan.

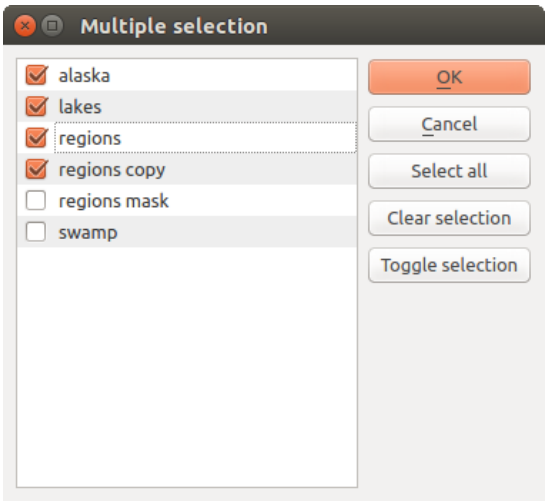


Fig. 23.14: Flervalsfält

- En **liten tabell** som kan redigeras av användaren. Dessa används bland annat för att definiera parametrar som lookup-tabeller eller convolution kernels.

  Klicka på knappen till höger för att se tabellen och redigera dess värden.

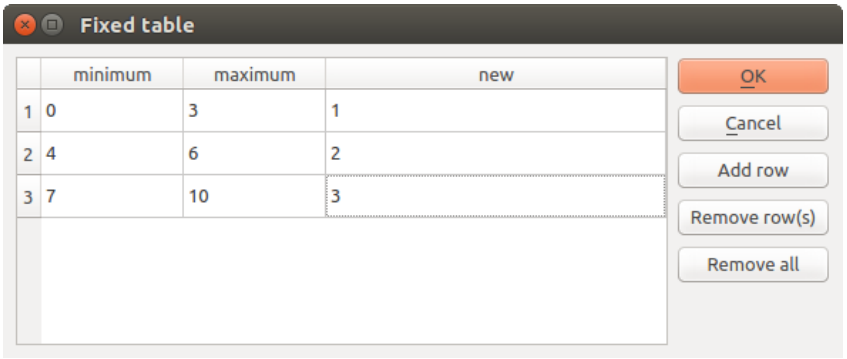


Fig. 23.15: Fast tabell

Beroende på algoritmen kan antalet rader ändras eller inte genom att använda knapparna till höger i fönstret.

Observera

Vissa algoritmer kräver många parametrar för att köras, t.ex. i *Rasterkalkylator* måste du manuellt ange cellstorlek, utsträckning och CRS. Du kan undvika att välja alla parametrar manuellt när algoritmen har parametern `Reference layers`. Med denna parameter kan du välja referenslager och alla dess egenskaper (cellstorlek, utsträckning, CRS) kommer att användas.

Loggning av utförandet

Tillsammans med fliken *Parameters* finns det en annan flik som heter *Log* (se Fig. 23.16 nedan). Information som tillhandahålls av algoritmen under dess exekvering skrivs i denna flik, så att du kan spåra exekveringen samt vara medveten om och få mer information om algoritmen när den körs. Du kan klicka direkt på namnen på utdatafiler, mappar eller HTML-filer som listas i loggen. Då öppnas den mapp som innehåller den genererade filen och den väljs automatiskt. Information om algoritmkörning visas också i :menuselection:`Visa --> Paneler --> Panel för loggmeddelanden`.

Observera att inte alla algoritmer skriver information till fliken *Log*, och många av dem kan köras tyst utan att producera någon annan utdata än de slutliga filerna. Kontrollera i så fall *Log Messages Panel*.

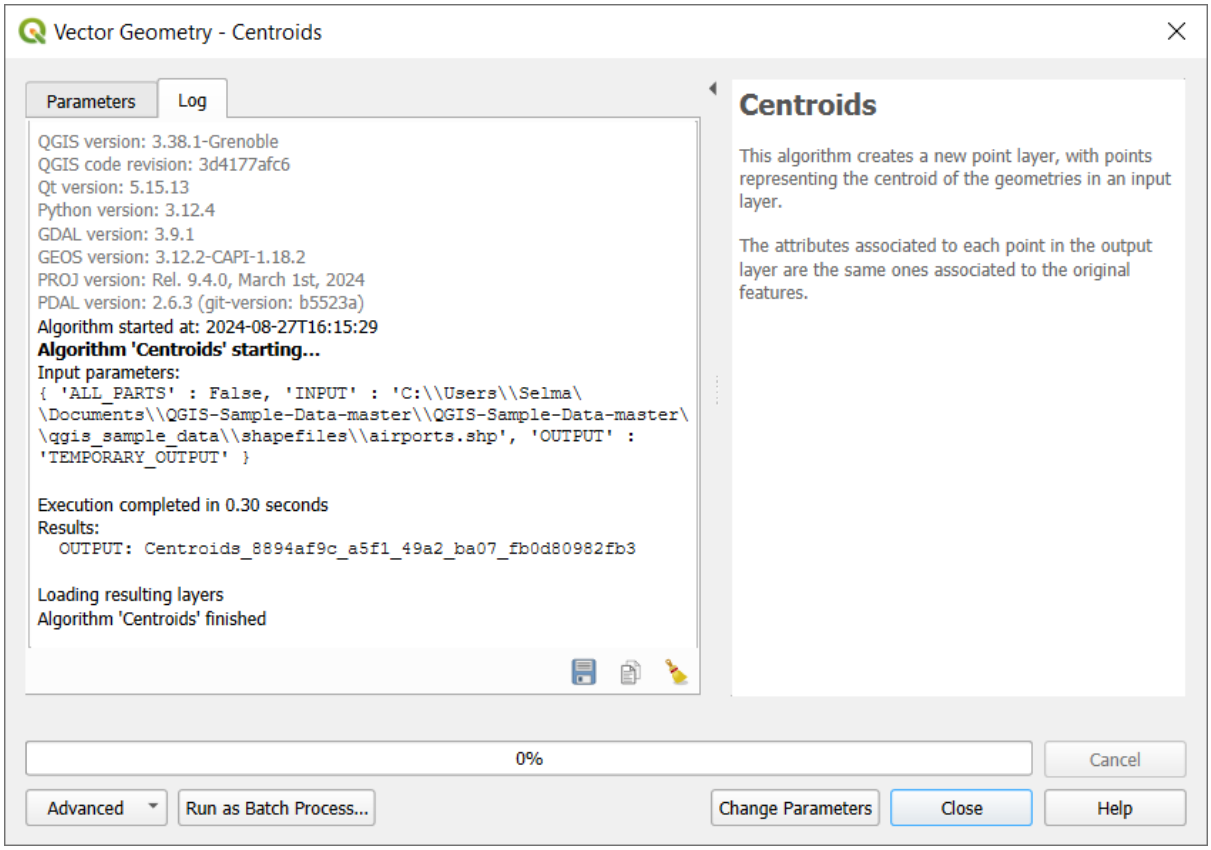


Fig. 23.16: Algoritmdialog - Logg

Längst ner på fliken *Logg* hittar du knapparna *Spara logg till fil*, *Kopiera logg till urklipp* och *Töm logg*. Dessa är särskilt praktiska när du har markerat *Keep dialog open after running algorithm* i *Allmänt*-delen av Processing-alternativen.

Andra verktyg

På höger sida av dialogrutan finns en kort beskrivning av algoritmen, som hjälper dig att förstå dess syfte och grundläggande idéer. Om en sådan beskrivning inte finns tillgänglig visas inte beskrivningspanelen.

För en mer detaljerad hjälpfil, som kan innehålla en beskrivning av varje parameter som används eller exempel, hittar du en *Help*-knapp längst ner i dialogrutan som tar dig till *Processing algorithms documentation* eller till leverantörsdokumentationen (för vissa tredjepartsleverantörer).

Menyn *Advanced* ► innehåller funktioner för att återanvända den konfiguration som definierats i dialogrutan utan att köra algoritmen:

- *Algoritminställningar…*: gör det möjligt att åsidosätta bearbetningsinställningar för den aktuella algoritmkörningen. Mer information finns i *Åsidosätta algoritminställningar*.
- *Kopiera som Python-kommando*: gör det enkelt att kopiera motsvarande *PyQGIS-kommando* för att köra verktyget med de parametrar som definieras i dialogrutan
- *Copy as qgis_process Command*: möjliggör enkel generering av *qgis_process command*, inklusive dess miljöinställningar som avståndsenheter, ytenheter, ellipsoid och eventuella knepiga parametervärden som GeoPackage-utdata med specifika lager
- *Kopiera som JSON*: alla inställningar för kommandot kopieras i ett `JSON`-format, redo att användas av qgis_process. Detta är ett bekvämt sätt att se det förväntade formatet för kommandona, även för komplexa parametrar (som TIN-interpolationsparametrar). Du kan enkelt lagra dessa och sedan återställa dem senare genom att klistra in värdena.
- *Klistra in inställningar* i ett `JSON`-format

Knappen *Run as Batch Process…* aktiverar *batch processing mode* som gör det möjligt att konfigurera och köra flera instanser av algoritmen med en mängd olika parametrar. En *Run as Single Process…*-knapp hjälper dig att växla tillbaka från batch-läget.

När en algoritmkörning avslutas (antingen framgångsrikt eller inte) visas en ny knapp *Change Parameters* så länge som fliken *Log* är aktiv.

Åsidosätta algoritminställningar

Utlöst från rullgardinsmenyn *Avancerat* längst ner i en algoritmdialog, *Algoritminställningar…* visar en panel där användare kan styra allmänna bearbetningsinställningar som endast gäller för den algoritmen. Det är tänkt att vara en plats där en användare kan åsidosätta sina *globala bearbetningsinställningar* på en ad hoc-basis utan att behöva ändra sina vanliga standardinställningar.

Inställningar som kan åsidosättas är:

- *Invalid feature filtering*: till skillnad från det befintliga åsidosättandet av inställningen per parameter för detta, kommer inställningen av hanteringsmetoden här att gälla för **ALLA indata** för algoritmen
- *Beräkningsinställningar*, t.ex. *Avståndsenheter* och *Areaenheter* att använda för mätning av avstånd/yta
- *Miljöinställningar*, till exempel *Temporär mapp* och *Antal trådar att använda*

En anmärkning om prognoser

Exekvering av bearbetningsalgoritmer utförs alltid i det ingående lagrets koordinatreferenssystem (CRS). På grund av QGIS omprojiceringsfunktioner i farten kan det hända att även om två lager verkar överlappa och matcha varandra, så är det inte sant om deras ursprungliga koordinater används utan att de omprojiceras till ett gemensamt koordinatsystem. När du använder mer än ett lager som indata till en *QGIS native-algoritm*, oavsett om det är vektor eller raster, kommer alla lager att omprojiceras för att matcha koordinatreferenssystemet för det första inmatningslagret.

Detta gäller dock i mindre utsträckning för de flesta externa applikationer vars algoritmer exponeras via bearbetningsramverket, eftersom de förutsätter att alla lager redan befinner sig i ett gemensamt koordinatsystem och är redo att analyseras.

Som standard visas en beskrivning av CRS för varje lager i dialogrutan Parametrar tillsammans med lagrets namn, vilket gör det enkelt att välja lager som har samma CRS och som ska användas som indatalager. Om du inte vill se denna ytterligare information kan du inaktivera funktionen i dialogrutan Processing settings genom att avmarkera alternativet *Show layer CRS definition in selection boxes*.

Om du försöker exekvera en algoritm med två eller flera lager med CRS som inte matchar varandra som indata visas en varningsdialog. Detta sker tack vare alternativet *Varna före exekvering om lagrens CRS inte matchar*.

Du kan fortfarande köra algoritmen, men tänk på att det i de flesta fall kommer att ge fel resultat, t.ex. tomma lager på grund av att inmatningslagren inte överlappar varandra.

Tips

Använd bearbetningsalgoritmer för att göra mellanliggande reprojektion

När en algoritm inte kan utföras på flera inmatningslager på grund av att CRS inte matchar, använd QGIS interna algoritm som *Återprojicera lagret* för att utföra lagrens återprojicering till samma CRS innan du utför algoritmen med dessa utdata.

23.3.2 Dataobjekt som genereras av algoritmer

Dataobjekt som genereras av en algoritm kan vara av olika typer. De kan vara ett lager (vektor med eller utan geometri, raster, mesh, punktmoln, …), lagrat som en vanlig fil på disken eller i en databas, textbaserade filer som t.ex. `qml` style file eller grafik `.html` file, en mapp, …

Parameterdialogen innehåller en widget som motsvarar den önskade utdata där du kan ange vilken utdatakanal som ska användas för att spara den. En utdatakanal innehåller den information som behövs för att spara det resulterande objektet någonstans. I det vanligaste fallet sparar du det i en fil, men när det gäller vektorlager och när de genereras av inbyggda algoritmer (algoritmer som inte använder externa program) kan du också spara i en PostGIS-, GeoPackage- eller SpatiaLite-databas, eller i ett minneslager.

Parametern **output** för att ställa in resultatet av algoritmkörningen ger en textruta för att fylla i sökvägen till utdataresultatet och en rullgardinsmeny … för ytterligare alternativ (tillgängligheten beror på vilken algoritm du kör):

- *Skip Output*: om du inte är intresserad av en viss output för denna parameter
- *Create Temporary Layer* (`TEMPORARY_OUTPUT`): utdata lagras i en vektor *temporary scratch layer*.
- *Save to File…*: du får upp en dialogruta för att spara filen, där du kan välja önskad filsökväg. Filtillägg som stöds visas i dialogrutans filformatväljare, beroende på vilken typ av utdata och algoritm som används.

  Utdatans format definieras av filnamnstillägget. De format som stöds beror på vad som stöds av själva algoritmen. För att välja ett format väljer du bara motsvarande filnamnstillägg (eller lägger till det om du skriver in filsökvägen direkt i stället). Om filtillägget för den filsökväg du angav inte matchar något av de format som stöds, läggs ett standardtillägg till filsökvägen och det filformat som motsvarar det tillägget används för att spara lagret eller tabellen. Standardtillägget är `.dbf` för tabeller, `.tif` för rasterlager och `.gpkg` för vektorlager. Dessa kan ändras i inställningsdialogen genom att välja något annat av de format som stöds av QGIS.

Du kan ange en standardmapp för utdatadataobjekt. Gå till inställningsdialogen (du kan öppna den från menyn *Inställningar* ► *Alternativ* ► *Processing*) och i gruppen *General* hittar du en parameter med namnet *Output folder*. Denna utdatamapp används som standardsökväg om du bara skriver ett filnamn utan sökväg (t.ex. `myfile.shp`) när du kör en algoritm.

- *Spara till en temporär fil*: om du inte anger något filnamn i textrutan för utdata (eller väljer motsvarande alternativ i snabbmenyn) sparas resultatet som en temporär fil i motsvarande standardfilformat och raderas när du avslutar QGIS (var försiktig med detta om du sparar ditt projekt och det innehåller temporära lager). Filen sparas i standardmappen``*TEMP*` på maskinen, om *not overridden*.
- *Save to Geopackage…*: utdatalagret sparas som en namngiven tabell i en ny eller befintlig GeoPackage-databas
- *Save to Database Table…*: utdatalagret sparas som en ny tabell i en databas som GeoPackage, SpatiaLite, PostgreSQL eller MS SQL Server. En anslutning bör redan finnas. Du måste ange namnet på den nya tabellen och beroende på vilken typ av måldatabas du har, välja ett schema,
- *Append to Layer…*: lägger till funktioner från algoritmen i ett befintligt lager. När du väljer mållagret öppnas en panel för fältmappning, där du manuellt kan ställa in hur fält från utdatalagret ska mappas till mållagrets fält.

- *Spara till katalog*: Använd en ny eller befintlig katalog för att lagra algoritmens utdata
- *Save to a Temporary Directory*: skapar en katalog i den inställda mappen `TEMP` för lagring av algoritmens utdata. Filerna kommer att raderas när du avslutar QGIS.

När du kör en algoritm som använder ett vektorlager i iterativt läge används den angivna filsökvägen som bassökväg för alla genererade filer, som namnges med hjälp av basnamnet och ett tillägg av ett nummer som representerar iterationens index. Filtillägget (och formatet) används för alla sådana genererade filer.

Algoritmer genererar också grafik och text som HTML-filer. Dessa resultat visas i slutet av algoritmkörningen i dialogrutan *Results Viewer*. Denna dialog behåller resultaten som producerats av en algoritm under den aktuella sessionen och kan visas när som helst genom att välja *Processing* ► *Results Viewer* från QGIS huvudmeny.

För lagerbaserade utdata är det möjligt att markera kryssrutan *Öppna utdatafil efter körning av algoritm* direkt i QGIS, som en ny post i panelen *Layers*. Som standard öppnas filerna.

23.4 Den historiskt ansvarige

23.4.1 Bearbetningshistorik

Varje gång du kör en algoritm lagras information om processen i historikhanteraren. Datum och tid för exekveringen sparas tillsammans med de parametrar som används, vilket gör det enkelt att spåra och kontrollera allt arbete som har utvecklats med hjälp av Processing-ramverket och att reproducera det.

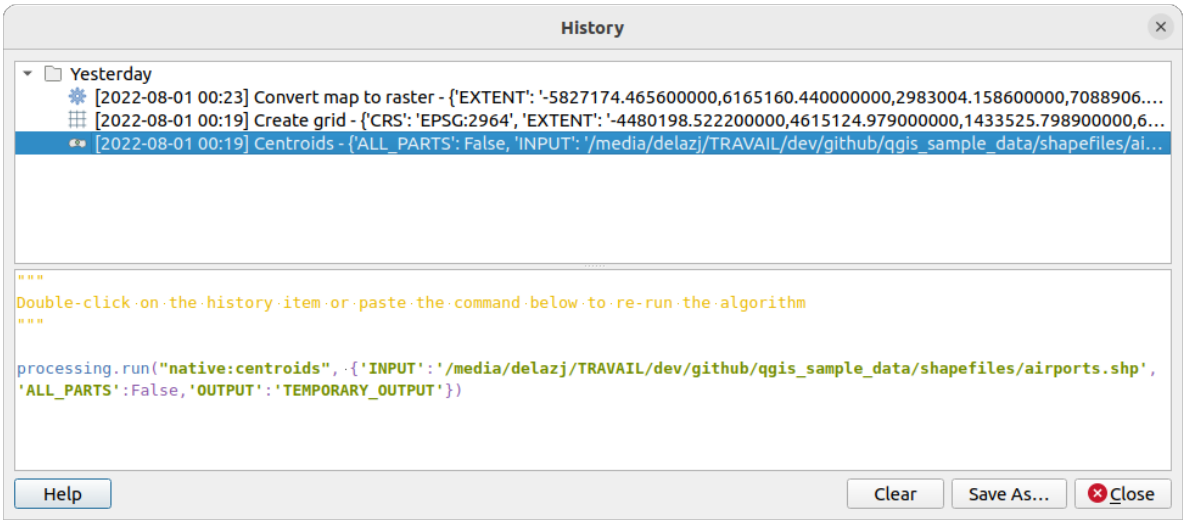


Fig. 23.17: Historik

Processinformationen sparas som ett kommandoradsuttryck, även om algoritmen startades från verktygslådan. Detta gör det användbart för dem som lär sig att använda kommandoradsgränssnittet, eftersom de kan anropa en algoritm med hjälp av verktygslådan och sedan kontrollera historikhanteraren för att se hur den skulle kunna anropas från kommandoraden.

Högerklicka på en rad så ska du kunna göra det:

- *Copy as Python Command*: gör det enkelt att kopiera det motsvarande *PyQGIS-kommandot* som körs från dialogrutan. Samma som koden som visas under kommandolistan.
- *Kopiera som qgis_process-kommando*: gör det enkelt att generera *qgis_process-kommando*, inklusive dess miljöinställningar som avståndsenheter, ytenheter, ellipsoid och eventuella knepiga parametervärden som GeoPackage-utdata med specifika lager
- *Copy as JSON*: alla inställningar för kommandot kopieras i ett `JSON`-format, redo att användas av qgis_process. Detta är ett bekvämt sätt att se det förväntade formatet för kommandona, även för komplexa parametrar (som TIN-interpolationsparametrar). Du kan enkelt lagra dessa och sedan återställa dem senare genom att klistra in värdena i en algoritmdialog.
- *Skapa test…* med hjälp av den aktuella algoritmen och parametrarna, enligt instruktionerna i Processing README-filen.

Förutom att bläddra bland posterna i registret kan du också köra processer igen genom att helt enkelt dubbelklicka på posten. Algoritmdialogen öppnas då med redan inställda parametrar, och du kan ändra någon av dem så att de passar dina behov och köra algoritmen igen.

Dialogrutan *History* ger också ett bekvämt sätt att bidra till konsolideringen av testinfrastrukturen för QGIS Processing-algoritmer och skript.

23.4.2 Behandlingsloggen

Historikdialogen innehåller endast exekveringsanropen, men inte den information som algoritmen producerar när den exekveras. Den informationen skrivs till QGIS-loggen (:menuselection:`View --> Panels --> Log Messages Panel`).

Algoritmer från tredje part körs vanligtvis med hjälp av deras kommandoradsgränssnitt, som kommunicerar med användaren via konsolen. Även om konsolen inte visas skrivs vanligtvis en fullständig dumpning av den till loggen varje gång du kör en av dessa algoritmer. För att undvika att loggen fylls med den informationen kan du inaktivera den för varje leverantör i inställningsdialogen.

Vissa algoritmer, även om de kan producera ett resultat med de givna indata, skickar ut kommentarer eller ytterligare information till loggen när de upptäcker potentiella problem med data, för att varna dig. Se till att kontrollera dessa meddelanden i loggen om du får oväntade resultat.

23.5 Modellkonstruktören

Med *model designer* kan du skapa komplexa modeller med hjälp av ett enkelt och lättanvänt gränssnitt. När man arbetar med ett GIS är de flesta analysoperationer inte isolerade, utan ingår i en kedja av operationer. Med hjälp av modelldesignern kan denna kedja av operationer förpackas i en enda process, vilket gör det bekvämt att köra senare med en annan uppsättning ingångar. Oavsett hur många steg och olika algoritmer det handlar om körs en modell som en enda algoritm, vilket sparar tid och ansträngning.

Modellkonstruktören kan öppnas från bearbetningsmenyn (*Bearbetning* ► *Modelldesigner*).

23.5.1 Gränssnittet för modellkonstruktören

Fig. 23.18: Modelldesigner

I huvudsak har modelleraren en arbetsyta där modellens struktur och det arbetsflöde som den representerar kan konstrueras.

Längst upp i dialogrutan finns olika menyer och verktygsfältet *Navigation* som ger tillgång till en mängd olika verktyg.

Meny för modell

Etikett	Genväg	Verktygsfält för navigering	Beskrivning
Validera modell			Kontrollerar om de algoritmer och indata som används i modellen finns. Praktiskt innan du släpper en modell.
Kör modell…	F5	☑	Exekverar modellen
Kör valda steg…	Shift+F5		Kör endast de valda stegen i en modell, vilket gör det möjligt att köra en delmängd av modellen
Ombeställ modellens indata…			Anger i vilken ordning indata ska presenteras för användaren i algoritmdialogen.
Omordna utmatningslagren…			Ställer in en specifik ordning som utdata från modellen måste använda när resultaten laddas in i ett projekt.
Öppna modell…	Ctrl+O	☑	Öppnar en `.model3`-fil för redigering eller exekvering
Spara modell	Ctrl+S	☑	Sparar modellen på hårddisken som en `.model3`-fil
Spara modell som…	Ctrl+Shi	☑	Sparar modellen på hårddisken som en ny `.model3`-fil
Spara modell i projekt		☑	Bäddar in modellfilen i projektfilen, så att den blir tillgänglig när du delar projektfilen.
Redigera modellhjälp…		☑	Ett gränssnitt för att dokumentera modellen, algoritmerna, parametrarna och utdata, samt författare och versionshantering
Exportera ►			
► *Exportera som bild…*		☑	Sparar modellens grafiska design i ett bildfilformat (för illustrationsändamål)
► *Exportera som PDF…*			Sparar modellens grafiska design till filformatet `PDF` (för illustrationsändamål)
► *Exportera som SVG…*			Sparar modellens grafiska design i filformatet `SVG` (för illustrationsändamål)
► *Exportera som skriptalgoritm…*		☑	Genererar en pythonskriptfil med instruktioner för modellen

Redigera meny

Etikett	Genväg	Verktygsfält för navigering	Beskrivning
Markera alla	`Ctrl+A`		Markerar alla modellkomponenter i designern
Snapna valda komponenter till rutnätet			snäpper och riktar in elementen i ett rutnät
Gör om	`Ctrl+Y`	☑	Återställer den senast avbrutna åtgärden. Se även panelen *Undo/Redo*.
Ångra	`Ctrl+Z`	☑	Avbryt den föregående ändringen. Se även panelen *Undo/Redo*.
Klipp ut	`Ctrl+X`		Skär ut ett urval av komponenter från modellen.
Kopiera	`Ctrl+C`		Kopierar ett urval av komponenter från modellen.
Klistra in	`Ctrl+V`		Klistrar in ett utskuret eller kopierat urval av komponenter från en modell till en annan eller inom samma modell. De valda komponenterna behåller sina ursprungliga egenskaper och kommentarer.
Ta bort markerade komponenter	`Del`		Tar bort en komponent från modellen.
Add Group Box			Lägger till en ruta i bakgrunden på relaterade komponenter för att gruppera dem visuellt. Särskilt användbart i stora modeller för att hålla arbetsflödet rent.

Visa meny

Etikett	Genväg	Verktygsfält för navigering	Beskrivning
Zooma till ►			Zoomar till den valda gruppboxens omfattning
Zooma in	`Ctrl++`	☑	
Zooma ut	`Ctrl+-`	☑	
Zoom till 100%	`Ctrl+1`	☑	
Zoom Full	`Ctrl+0`	☑	Visar alla komponenter i designerns aktuella arbetsyta
Visa kommentarer			Visar kommentarer som är kopplade till varje algoritm eller indata i modellkonstruktören
Aktivera fästning			
Växla panelsynlighet	`Ctrl+Tab`		Slår PÅ eller AV *panelerna* i designern

Paneler

Den vänstra delen av fönstret är en sektion med fem paneler som kan användas för att lägga till nya element i modellen:

1. *Model Properties*: ange namnet (obligatoriskt) på modellen och i vilken grupp den ska visas i *Processing Toolbox*
2. *Input*: alla *inputparametrar* som kan forma din modell
3. *Algoritmer*: tillgängliga *Processing-algoritmer*
4. *Variabler*: Modeller kan innehålla dedikerade *variabler* som är unika och endast tillgängliga för dem. Dessa variabler kan nås av alla uttryck som används i modellen. De är användbara för att styra algoritmer inom en modell och styra flera aspekter av modellen genom att ändra en enda variabel. Variablerna kan visas och ändras i panelen *Variables*.
5. *Undo History*: den här panelen registrerar allt som händer i modelleraren, vilket gör det enkelt att avbryta saker du gjort fel.

Om tillgängliga algoritmer

Vissa algoritmer som kan köras från verktygslådan visas inte i listan över tillgängliga algoritmer när du utformar en modell. För att en algoritm ska kunna ingå i en modell måste den ha rätt semantik. Om en algoritm inte har en sådan väldefinierad semantik (t.ex. om antalet utmatningslager inte är känt i förväg) går det inte att använda den i en modell och den visas inte i listan över algoritmer som du hittar i dialogrutan för modellering. Å andra sidan är vissa algoritmer specifika för modelleraren. Dessa algoritmer finns i gruppen ”Modelleringsverktyg”.

23.5.2 Skapa en modell

Att skapa en modell innebär två grundläggande steg:

1. *Definition av nödvändiga indata*. Dessa indata kommer att läggas till i parameterfönstret, så att användaren kan ställa in deras värden när modellen körs. Modellen i sig är en algoritm, så parameterfönstret genereras automatiskt som för alla algoritmer som är tillgängliga i Processing-ramverket.
2. *Definition av arbetsflödet*. Med hjälp av modellens indata definieras arbetsflödet genom att lägga till algoritmer och välja hur de använder de definierade indata eller de utdata som genereras av andra algoritmer i modellen.

Definition av ingångar

Det första steget är att definiera indata för modellen. De finns i *Inputs*-panelen till vänster i modelleringsfönstret. Om du håller muspekaren över indata visas ett verktygstips med ytterligare information. För en fullständig lista över tillgängliga parametrar i modeler och deras motsvarighet för skriptning, läs *Inmatnings- och utmatningstyper för bearbetningsalgoritmer*.

När du dubbelklickar på ett element visas en dialogruta där du kan definiera dess egenskaper. Beroende på parametern innehåller dialogrutan minst ett element (beskrivningen, som är det som användaren ser när modellen körs). Om du t.ex. lägger till ett numeriskt värde, som visas i nästa figur, måste du förutom beskrivningen av parametern ange ett standardvärde och ett intervall för giltiga värden.

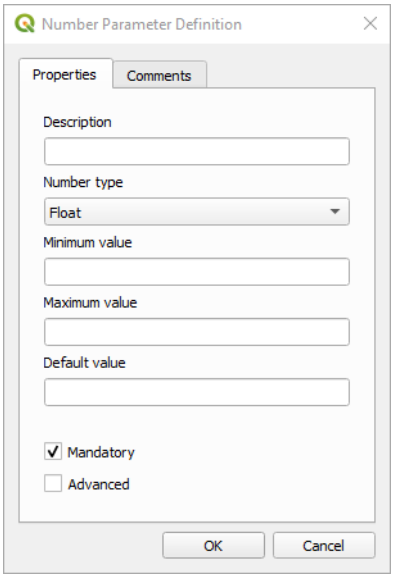


Fig. 23.19: Definition av modellparametrar

Du kan definiera dina indata som obligatoriska för din modell genom att markera alternativet `Mandatory` och genom att markera kryssrutan `Advanced` kan du ange att indata ska finnas i avsnittet `Advanced`. Detta är särskilt användbart när modellen har många parametrar och några av dem inte är triviala, men du vill ändå välja dem.

För varje ny inmatning läggs ett nytt element till i modelleringspanelen.

Fig. 23.20: Modell Parametrar

Du kan också lägga till ingångar genom att dra ingångstypen från listan och släppa den på den plats där du vill ha den i modelleringspanelen. Om du vill ändra en parameter för en befintlig ingång dubbelklickar du bara på den, så visas samma dialogruta.

När du använder en modell i en annan modell visas de nödvändiga in- och utmatningarna i Canvas.

Definition av arbetsflödet

I följande exempel lägger vi till två indata och två algoritmer. Syftet med modellen är att kopiera höjdvärdena från ett DEM-rasterlager till ett linjelager med hjälp av algoritmen ”Drape” och sedan beräkna linjeskiktets totala stigning med hjälp av algoritmen ”Climb Along Line”.

På fliken *Inputs* väljer du de två indata som `Vector Layer` för linjen och `Raster Layer` för DEM. Vi är nu redo att lägga till algoritmerna i arbetsflödet.

Algoritmer finns i panelen *Algorithms*, grupperade på ungefär samma sätt som i verktygslådan i Processing.

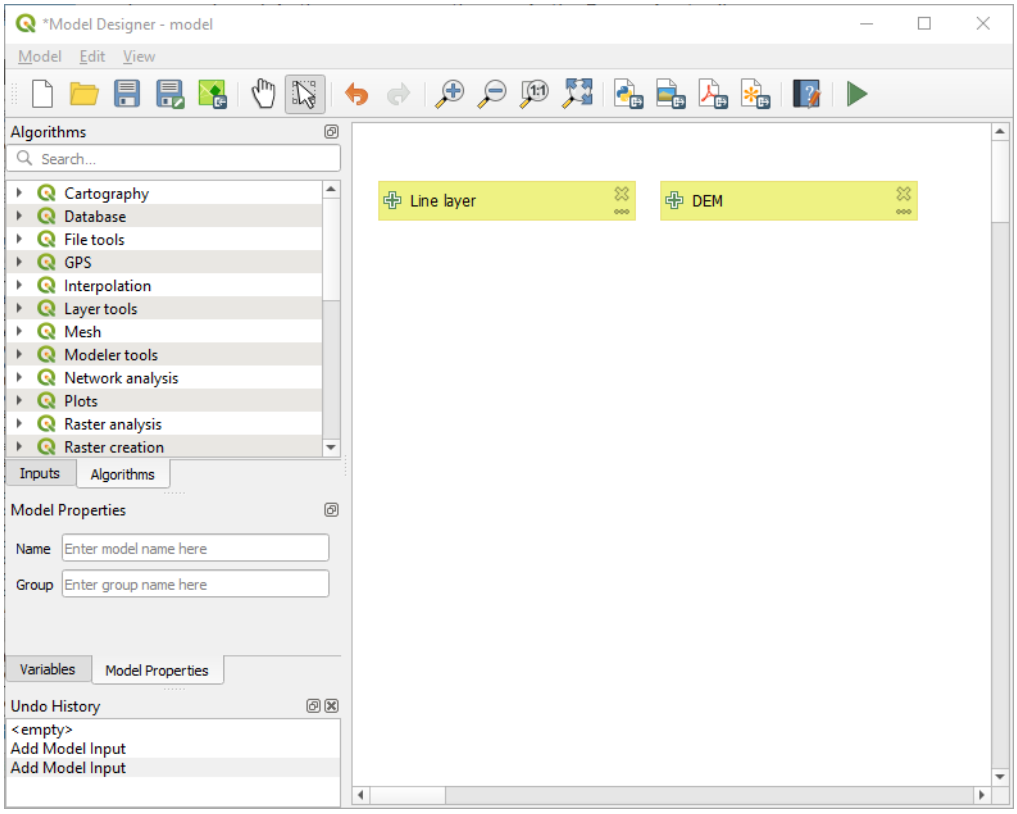


Fig. 23.21: Inmatning av modell

För att lägga till en algoritm i en modell dubbelklickar du på dess namn eller drar och släpper den, precis som för indata. På samma sätt som för indata kan du ändra beskrivningen av algoritmen och lägga till en kommentar. När du lägger till en algoritm visas en exekveringsdialog med ett innehåll som liknar det som finns i exekveringspanelen som visas när du exekverar algoritmen från verktygslådan. Följande bild visar både algoritmdialogerna `Drape (set Z value from raster)` och `Climb along line`.

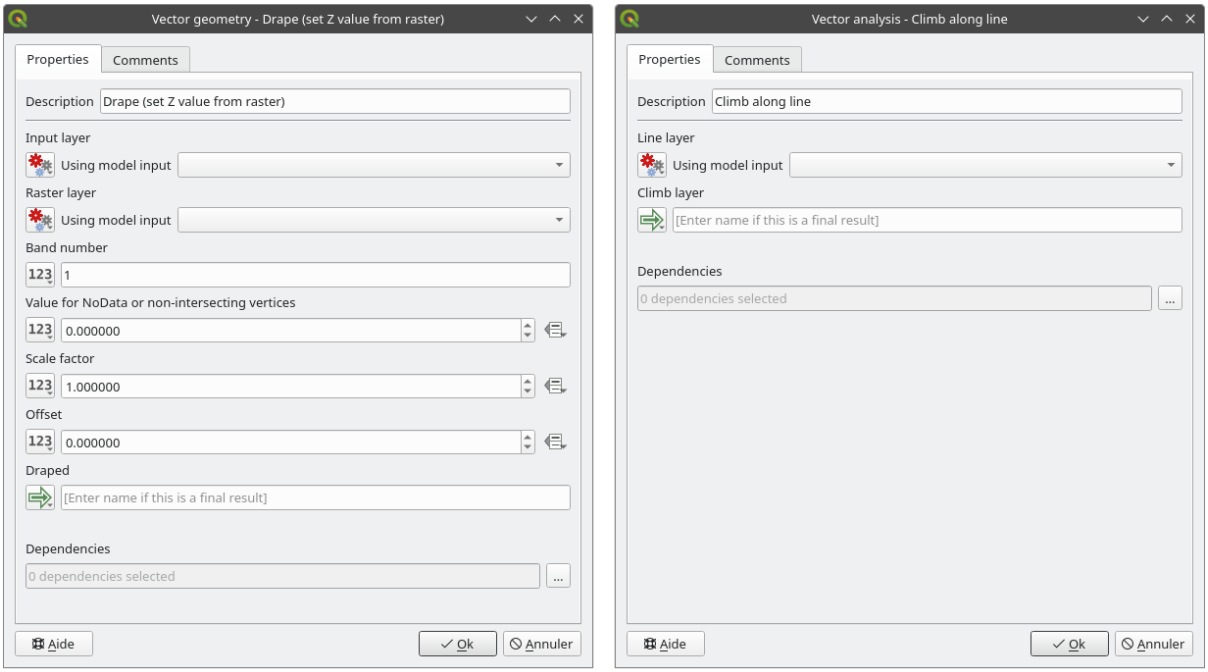


Fig. 23.22: Modell Algoritm parametrar

Som du kan se finns det dock vissa skillnader. Varje parameter har en rullgardinsmeny bredvid sig som gör det möjligt att styra hur den ska serveras under arbetsflödet:

- Value: gör att du kan tilldela parametern ett statiskt värde. Beroende på parametertyp kan du i widgeten ange ett tal (`5.0`), en sträng (`mytext`), välja lager som laddats i QGIS-projektet eller från en mapp, välja objekt från en lista, …
- Pre-calculated Value: öppnar dialogrutan *Expression Builder* och låter dig definiera ett uttryck för att fylla i parametern. Modellinmatningar tillsammans med viss annan lagerstatistik är tillgängliga som **variabler** och listas högst upp i dialogrutan Search i Expression Builder. Uttrycket utvärderas en gång innan den underordnade algoritmen körs och används under körningen av den algoritmen.
- Model Input: gör det möjligt att använda en indata som lagts till i modellen som en parameter. När du klickar på detta alternativ visas en lista över alla lämpliga indata för parametern.
- Algorithm Output: gör det möjligt att använda utdata från en annan algoritm som indata för den aktuella algoritmen. När det gäller modellinmatningar kommer detta alternativ att lista alla lämpliga inmatningar för parametern.
- Även **output parameter** har ovanstående alternativ i sin rullgardinsmeny:
  - lägga till statiska utdata för underordnade algoritmer, t.ex. alltid spara en underordnad algoritms utdata till ett fördefinierat geopaket eller postgres-lager
  - använda ett uttrycksbaserat utdatavärde för underordnade algoritmer, t.ex. generera ett automatiskt filnamn baserat på dagens datum och spara utdata i den filen
  - använda en modellinmatning, t.ex. modellinmatningen *File/Folder* för att ange en utdatafil eller -mapp
  - använda en annan algoritms utdata, t.ex. utdata från algoritmen *Create directory* (från *Modeler tools*)
  - ett extra alternativ Model Output gör algoritmens utdata tillgängliga i modellen. Om ett lager som genereras av algoritmen endast ska användas som indata till en annan algoritm, redigera inte textrutan.

  I följande bild kan du se de två ingångsparametrarna som definieras som ”Model Input” och det tillfälliga utmatningslagret:

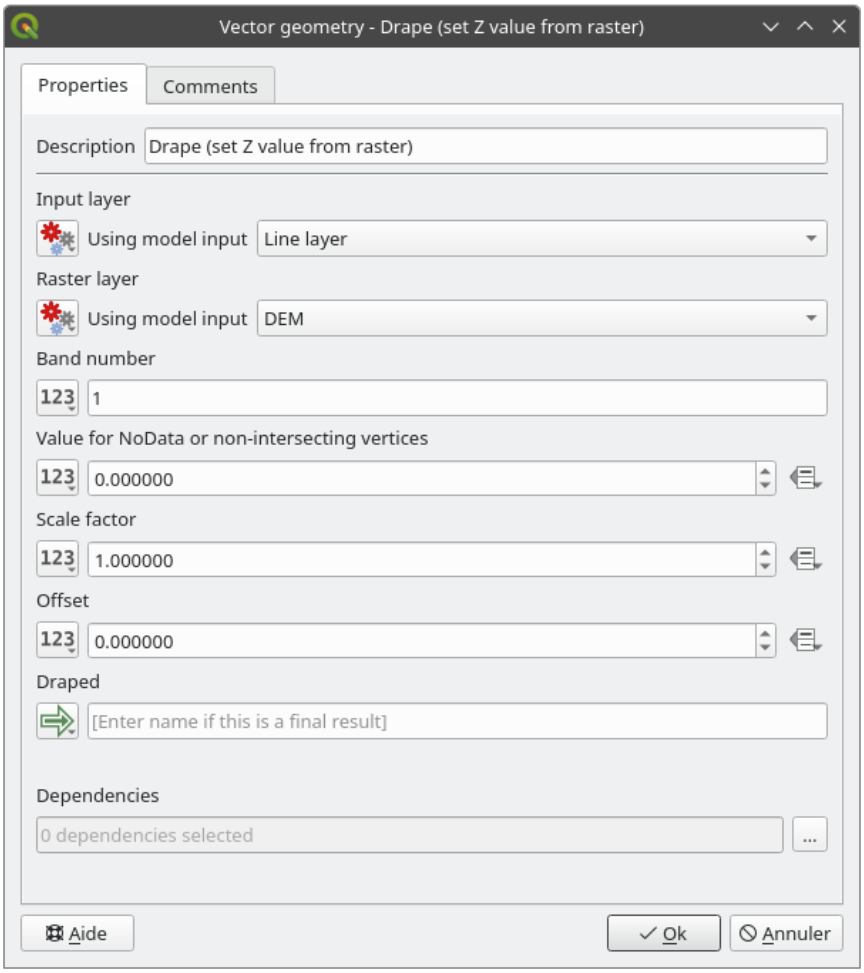


Fig. 23.23: Algoritm Parametrar för inmatning och utmatning

Du hittar också en ytterligare parameter med namnet *Dependencies* som inte är tillgänglig när du anropar algoritmen från verktygslådan. Med denna parameter kan du definiera i vilken ordning algoritmerna exekveras genom att uttryckligen definiera en algoritm som en *förälder* till den aktuella algoritmen. Detta tvingar *föräldra*-algoritmen att exekveras före den aktuella algoritmen.

När du använder utdata från en tidigare algoritm som indata för din algoritm, innebär det implicit att den tidigare algoritmen blir överordnad den aktuella algoritmen (och att motsvarande pil placeras i modelleringspanelen). I vissa fall kan dock en algoritm vara beroende av en annan algoritm även om den inte använder något utdataobjekt från den (t.ex. en algoritm som kör en SQL-sats på en PostGIS-databas och en annan som importerar ett lager till samma databas). I så fall väljer du bara den föregående algoritmen i parametern *Dependencies* så kommer de att köras i rätt ordning.

När alla parametrar har tilldelats giltiga värden klickar du på *OK* och algoritmen läggs till i kanvasen. Den kommer att länkas till de element i Canvas (algoritmer eller indata) som tillhandahåller objekt som används som indata för algoritmen.

Element kan dras till en annan position på duken med hjälp av verktyget Select/Move Item. Detta är användbart för att göra modellens struktur tydligare och mer intuitiv. Du kan också ändra storlek på elementen genom att ta tag i deras kant. Detta är särskilt användbart om beskrivningen av indata eller algoritm är lång. Om alternativet *View* ► *Enable snapping* är markerat kan element som ändras i storlek eller förflyttas bindas till ett virtuellt rutnät, vilket ger en mer visuellt strukturerad algoritmdesign.

Länkar mellan element uppdateras automatiskt och du kan se en +-knapp högst upp och längst ned i varje algoritm. Om du klickar på knappen listas alla algoritmens in- och utgångar så att du snabbt kan få en överblick.

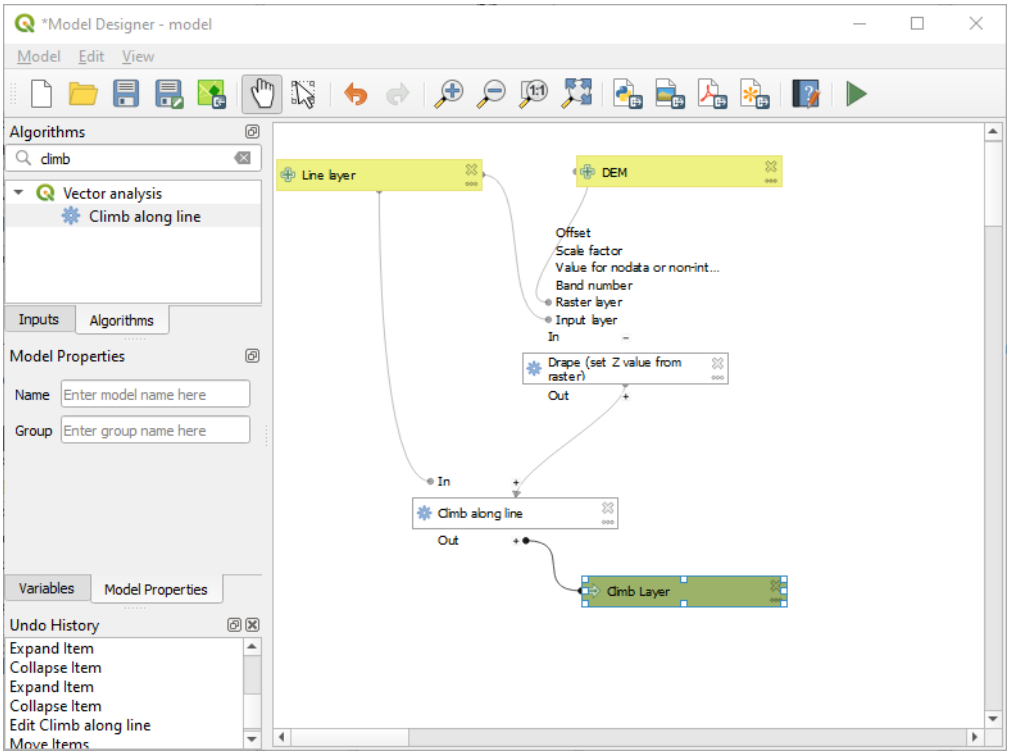


Fig. 23.24: En komplett modell

Med verktyget *Edit* ► *Add Group Box* kan du lägga till en dragbar *låda* i Canvas. Denna funktion är mycket användbar i stora modeller för att gruppera relaterade element i modelleringskanvasen och för att hålla arbetsflödet rent. Vi kan t.ex. gruppera alla ingångar i exemplet:

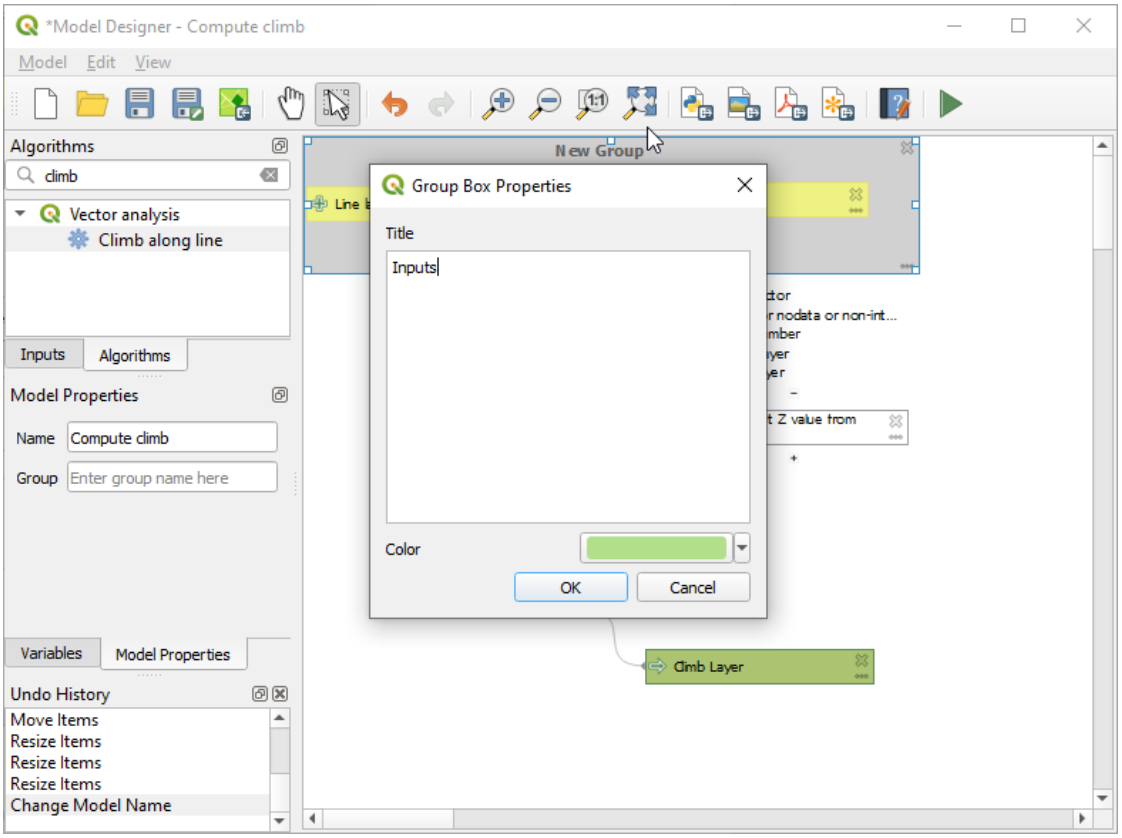


Fig. 23.25: Modell Grupp Box

Du kan ändra namn och färg på boxarna. Gruppboxar är mycket användbara när de används tillsammans med :menuselection:`View --> Zoom To -->`, så att du kan zooma till en viss del av modellen. Du kan också zooma in och ut med hjälp av mushjulet.

Du kanske vill ändra ordningen på indata och hur de listas i huvudmodellens dialogruta. Längst ned i panelen `Input` hittar du knappen `Reorder Model Inputs...` och genom att klicka på den öppnas en ny dialogruta där du kan ändra ordningen på indata:

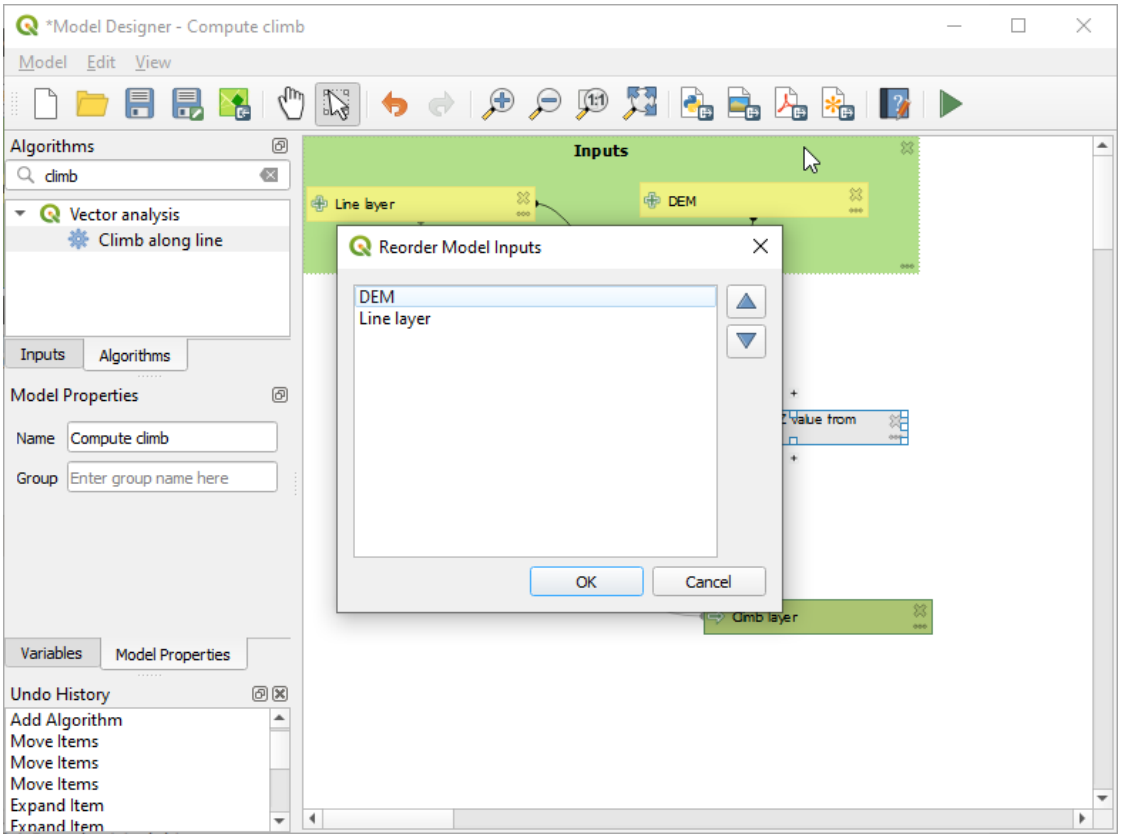


Fig. 23.26: Omordna modellens ingångar

Det finns också möjlighet att ange en specifik ordning som utdata från modellen måste använda när resultaten laddas in i ett projekt. Detta ger modellskaparen ett sätt att se till att lagren är logiskt ordnade på duken när en modell körs, t.ex. genom att placera en vektorlagerutgång över en rasterlagerutgång eller ett punktlager över ett polygonlager. Modellskaparen kan också ange ett valfritt ”Gruppnamn” för utdata för att automatiskt gruppera utdata i lagerträdet med hjälp av ett nytt gruppnamn eller genom att lägga till dem i en befintlig grupp. I menyn `Model` hittar du posten `Reorder Output Layers...` och genom att klicka på den öppnas en ny dialogruta där du kan ändra ordningen på utdatalagren:

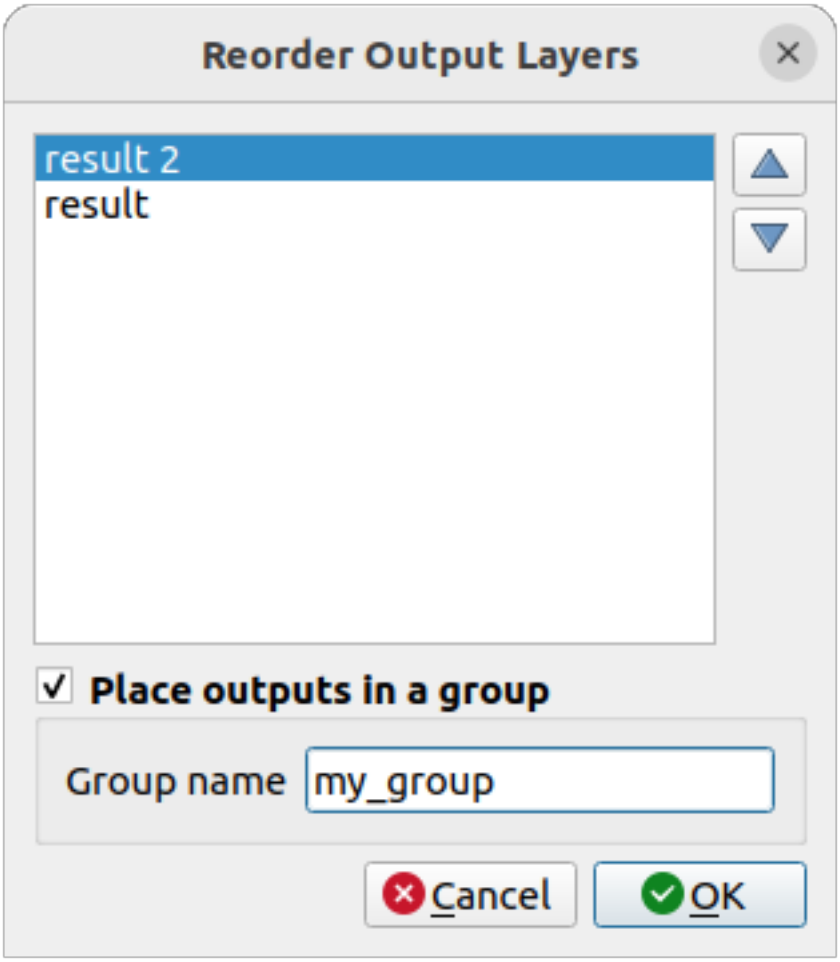


Fig. 23.27: Ordna om utmatningslagren

Kommentarer kan också läggas till indata eller algoritmer som finns i modelleraren. Detta kan göras genom att gå in på fliken *Comment* i objektet eller genom att högerklicka. I samma flik kan en färg ställas in manuellt för enskilda modellkommentarer. Kommentarer är endast synliga i modelleringspanelen och inte i dialogen för den slutliga algoritmen; de kan döljas genom att avaktivera *View* ► *Show Comments*.

Din modell kan köras på olika sätt:

- Du kan köra hela modellen genom att klicka på Kör modell… i verktygsfältet eller *Modell* ► *Kör modell…*: När du använder editorn för att köra en modell sparas alla värden som inte är standardvärden i indata. Detta innebär att om modellen körs vid ett senare tillfälle från redigeraren kommer dialogrutan att fyllas i med dessa värden vid varje efterföljande körning.
- Du kan välja ut delar av modellen och köra endast den delmängden av modellen: tryck på alternativet *Run Selected Steps…* från menyn *Models* ► eller från den kontextuella menyn för en vald algoritm. Initialtillståndet hämtas från alla tidigare körningar av modellen via editorn, så resultat från tidigare steg i modellen är tillgängliga för de valda stegen. Detta gör det möjligt att fixa delar av en stor modell utan att hela tiden behöva köra hela modellen för att testa. Detta är särskilt användbart när tidigare steg i modellen är tidskrävande.
- Du kan köra en delmängd av modellen med början från en viss algoritm: högerklicka på algoritmen och välj *Kör härifrån…*. På samma sätt hämtas värden från tidigare körningar.
- För att kunna använda algoritmen från verktygslådan Processing måste den sparas och dialogrutan för modellering stängas, så att verktygslådan kan uppdatera sitt innehåll.

Dokumentera din modell

Du måste dokumentera din modell, och detta kan göras från själva modelleraren. Klicka på knappen

så visas en dialogruta som den som visas här intill.

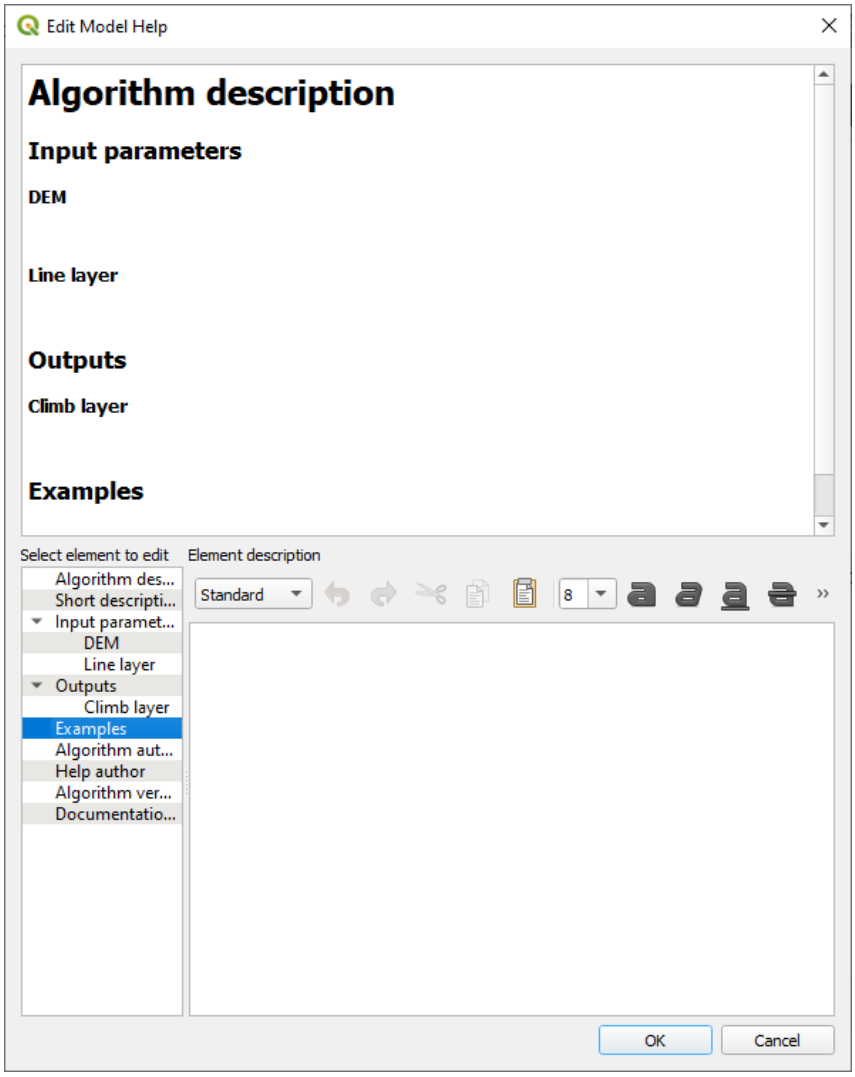


Fig. 23.28: Hjälp med redigering

På höger sida ser du en enkel HTML-sida som skapats med hjälp av beskrivningen av algoritmens ingångsparametrar och utdata, tillsammans med några ytterligare objekt som en allmän beskrivning av modellen eller dess författare. Det finns också ett avsnitt med exempel där du kan skriva in dina egna anpassade exempel för att förklara hur modellen används. Första gången du öppnar hjälpredigeraren är alla dessa beskrivningar tomma, men du kan redigera dem med hjälp av elementen på vänster sida av dialogrutan. Välj ett element i den övre delen och skriv sedan dess beskrivning i textrutan nedan.

Modellhjälp sparas som en del av själva modellen.

23.5.3 Spara och ladda modeller

Spara modeller

Använd knappen

för att spara den aktuella modellen och knappen

för att öppna en tidigare sparad modell. Modeller sparas med filnamnstillägget `.model3`. Om modellen redan har sparats från modelleringsfönstret kommer du inte att uppmanas att ange något filnamn. Eftersom det redan finns en fil kopplad till modellen kommer den filen att användas för efterföljande sparningar.

Innan du sparar en modell måste du ange ett namn och en grupp för den i textrutorna i den övre delen av fönstret.

Modeller som sparats i mappen `models` (standardmappen när du uppmanas att ange ett filnamn för att spara modellen) visas i verktygslådan i motsvarande gren. När verktygslådan startas söker den i mappen `models` efter filer med filnamnstillägget `.model3` och laddar de modeller de innehåller. Eftersom en modell i sig är en algoritm kan den läggas till i verktygslådan precis som vilken annan algoritm som helst.

Modeller kan också sparas i projektfilen med hjälp av knappen

. Modeller som sparas med den här metoden skrivs inte ut som `.model3`-filer på disken utan bäddas in i projektfilen.

Projektmodeller finns tillgängliga i menyn
Projektmodeller i verktygslådan och i menyalternativet *Projekt* ► *Modeller*.

Modellmappen kan ställas in i dialogrutan Processing configuration, under gruppen *Modeler*.

Modeller som laddas från mappen `models` visas inte bara i verktygslådan utan även i algoritmträdet på fliken *Algorithms* i modelleringsfönstret. Det innebär att du kan integrera en modell som en del av en större modell, precis som andra algoritmer.

Modellerna kommer att visas i *Browser*-panelen och kan köras därifrån.

Exportera en modell som ett Python-skript

Som vi kommer att se i ett senare kapitel kan Processing-algoritmer anropas från QGIS Python-konsolen, och nya Processing-algoritmer kan skapas med hjälp av Python. Ett snabbt sätt att skapa ett sådant Python-skript är att skapa en modell och sedan exportera den som en Python-fil.

För att göra det, klicka på

i modelleringsrutan eller högerklicka på namnet på modellen i Processing Toolbox och välj

.

Exportera en modell som en bild, PDF eller SVG

En modell kan också exporteras som en bild, SVG eller PDF (för illustrationssyften) genom att klicka på

,

eller

.

23.5.4 Redigera en modell

Du kan redigera den modell du håller på att skapa och omdefiniera arbetsflödet och relationerna mellan de algoritmer och indata som definierar modellen.

Om du högerklickar på en algoritm i Canvas visas en kontextmeny som den som visas här intill:

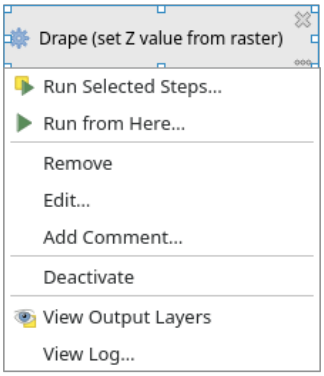


Fig. 23.29: Modellerare Högerklicka

- Alternativ för *kör en del av modellen* har exponerats tidigare.
- Om du väljer alternativet *Remove* kommer den valda algoritmen att tas bort. En algoritm kan bara tas bort om det inte finns några andra algoritmer som är beroende av den. Det vill säga om ingen utdata från algoritmen används som indata i en annan algoritm.
- Om du väljer alternativet *Edit…* visas parameterdialogen så att du kan ändra indata och parametervärden. Alla indataelement som är tillgängliga i modellen visas inte som tillgängliga indata. Lager eller värden som genereras i ett mer avancerat steg i det arbetsflöde som definieras av modellen kommer inte att vara tillgängliga om de orsakar cirkulära beroenden.

  Välj de nya värdena och klicka på knappen *OK* som vanligt. Anslutningarna mellan modellelementen kommer att ändras i modelleringspanelen i enlighet med detta.
- Med *Add comment…* kan du lägga till en kommentar till algoritmen för att bättre beskriva beteendet.

Observera

Om du högerklickar på en inmatningsparameter kan du också *Remove*, *Edit…* och *Add comment…*

- En modell kan köras delvis genom att avaktivera några av dess algoritmer. Det gör du genom att välja alternativet *Deaktivera* i snabbmenyn som visas när du högerklickar på ett algoritmelement. Den valda algoritmen och alla de algoritmer i modellen som är beroende av den visas i grått och körs inte som en del av modellen.

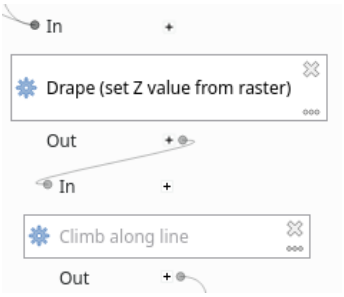


Fig. 23.30: Modell med avaktiverad algoritm

  När du högerklickar på en algoritm som inte är aktiv visas ett *Activate*-menyalternativ som du kan använda för att aktivera den igen.
- När du redigerar en modell via designern (och efter att ha kört modellen) kan du högerklicka på ett underordnat steg i modellen och välja *Visa utmatningslager*. Detta lägger till utmatningslagren från det steget som nya lager i det aktuella QGIS-projektet.

  Den här åtgärden är tillgänglig för alla underordnade algoritmer i modellen, även om modellen inte är konfigurerad för att använda utdata från dessa underordnade algoritmer som modellutdata. Denna åtgärd är

avsedd att vara till hjälp vid felsökning. Om en modell misslyckas (eller ger oväntade resultat) kan du spåra den genom modellen och visa utdata för misstänkta problematiska steg. Det gör att man inte behöver lägga till tillfälliga utdata i en modell och köra om den för att testa.

Dessutom är denna åtgärd alltid tillgänglig efter att modellen har körts, även om modellen i sig misslyckades (t.ex. på grund av ett felkonfigurerat steg senare i modellen).

- Åtgärden *View Log* hjälper dig att se loggen för varje underordnad algoritm efter att du har stängt algoritmdialogen.

23.6 Gränssnittet för batchbehandling

23.6.1 Introduktion

Alla algoritmer (inklusive modeller) kan exekveras som en batchprocess. Det innebär att de inte bara kan exekveras med en enda uppsättning indata, utan med flera, och att algoritmen kan exekveras så många gånger som behövs. Detta är användbart vid bearbetning av stora datamängder, eftersom det inte är nödvändigt att starta algoritmen många gånger från verktygslådan.

Om du vill köra en algoritm som en batchprocess högerklickar du på dess namn i verktygslådan och väljer alternativet *Execute as batch process* i popup-menyn som visas.

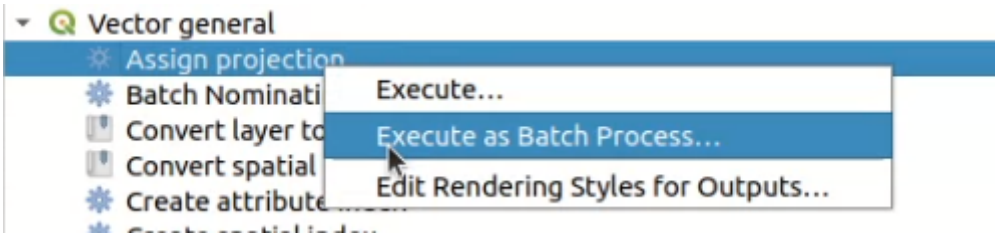


Fig. 23.31: Batchbehandling från högerklick

Om du har algoritmens exekveringsdialog öppen kan du också starta batchbehandlingsgränssnittet därifrån genom att klicka på knappen *Run as batch process…*.

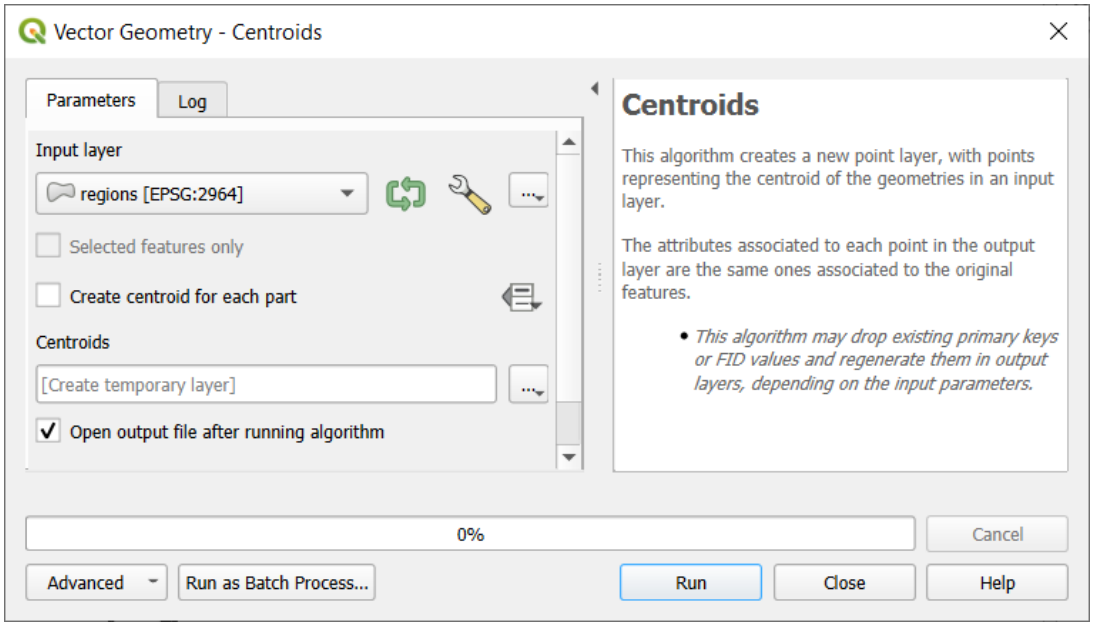


Fig. 23.32: Batchbehandling från algoritmdialog

23.6.2 Parametertabellen

Att köra en batchprocess liknar att köra en enda exekvering av en algoritm. Parametervärdena måste definieras, men i det här fallet behöver vi inte bara ett enda värde för varje parameter, utan en uppsättning av dem, ett för varje gång algoritmen måste köras. Värdena anges med hjälp av en tabell som den som visas här intill, där varje rad är en iteration och kolumnerna är algoritmens parametrar.

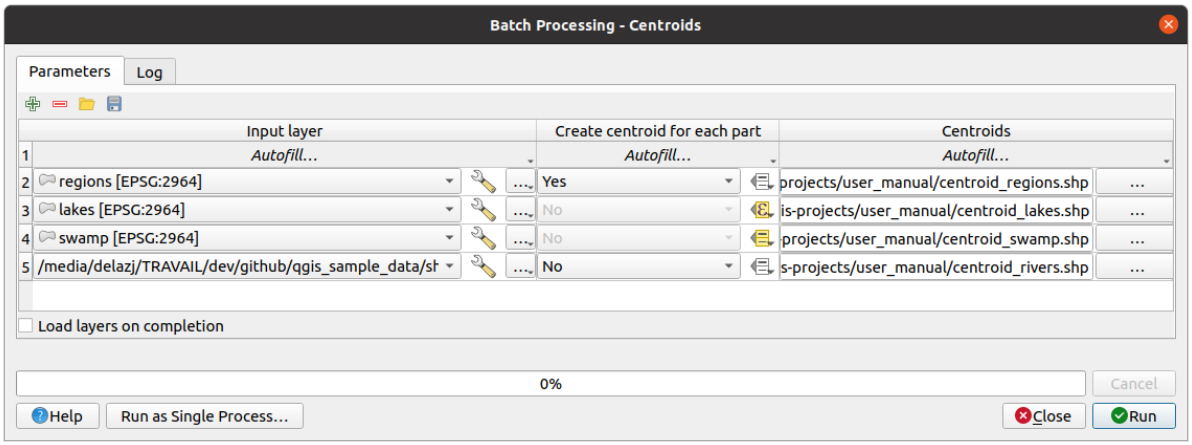


Fig. 23.33: Batchbehandling

Från det övre verktygsfältet kan du:

- Skifta till avancerat läge: Denna knapp är endast tillgänglig när algoritmen har parametrar som är markerade som avancerade och gör det möjligt att visa eller dölja sådana parametrar i batchdialogen.
- Lägg till rad: lägger till en ny bearbetningspost för konfiguration
- Ta bort rad(er): ta bort valda rader från tabellen. Radval görs genom att klicka på numret till vänster och tillåter *tangentbordskombination* för flerval.

- Öppna en konfigurationsfil för batchbehandling
- Spara konfigurationen för batchbearbetning till en `.JSON`-fil som kan köras i efterhand

Som standard innehåller tabellen bara två rader:

- Den första raden visar i varje cell en *Autofill…* ► rullgardinsmeny med *options* för att snabbt fylla cellerna nedanför. Vilka alternativ som är tillgängliga beror på parametertypen.
- Den andra raden (liksom varje efterföljande rad) representerar en enda exekvering av algoritmen, och varje cell innehåller värdet för en av parametrarna. Den liknar dialogrutan Parametrar som du ser när du kör en algoritm från verktygslådan, men med ett annat arrangemang.

Längst ner i tabellen kan du ställa in om du vill att *Load layers on completion*.

När tabellens storlek har fastställts måste det fyllas med de önskade värdena.

23.6.3 Fylla i parametertabellen

För de flesta parametrar är det trivialt att ställa in värdet. Den lämpliga widgeten, samma som i *single process dialog*, tillhandahålls, så att du bara kan skriva in värdet eller välja det från en lista med möjliga värden, beroende på parametertypen. Detta inkluderar även data-definieringswidget, när den är kompatibel.

För att automatisera definitionen av batchprocessen och undvika att fylla tabellen cell för cell, kan du trycka ned *Autofill…*-menyn för en parameter och välja något av följande alternativ för att ersätta värden i kolumnen:

- *Fill Down* tar inmatningen för den första processen och matar in den för alla andra processer.
- *Calculate by Expression…* gör att du kan skapa ett nytt QGIS-uttryck som används för att uppdatera alla befintliga värden i kolumnen. Befintliga parametervärden (inklusive de från andra kolumner) är tillgängliga för användning i uttrycket via *variables*. T.ex. inställning av antalet segment baserat på buffertavståndet för varje lager:

```
CASE WHEN @DISTANCE > 20 THEN 12 ELSE 8 END
```

- *Add Values by Expression…* lägger till nya rader med värdena från ett uttryck som returnerar en matris (i motsats till *Calculate by Expression…*, som bara fungerar på befintliga rader). Det avsedda användningsfallet är att göra det möjligt att fylla i batchdialogrutan med komplexa numeriska serier. Om du t.ex. lägger till rader för en batchbuffert med uttrycket `generate_series(100, 1000, 50)` för avståndsparametern resulterar det i nya rader med värdena 100, 150, 200, …. 1000.
- När du ställer in en fil- eller lagerparameter finns det fler alternativ:
  - *Lägg till filer efter mönster…*: lägger till nya rader i tabellen för filer som matchar ett *Filmönster* i en mapp att *Titta i*. T.ex. kommer `*.shp` att lägga till alla `SHP`-filer i mappen i listan. Markera *Sök rekursivt* för att även bläddra i undermappar.
  - *Välj filer…* individuellt på hårddisken
  - *Lägg till alla filer från en katalog…*
  - *Välj från Öppna lager…* i det aktiva projektet

Output data parameter exponerar samma funktioner som när algoritmen exekveras som en enskild process. Beroende på algoritmen kan utdata vara:

- hoppas över, om cellen lämnas tom
- sparas som ett tillfälligt lager: fyll cellen med `TEMPORARY_OUTPUT` och kom ihåg att kryssa i kryssrutan *Load layers on completion*.
- sparas som en vanlig fil (`.SHP`, `.GPKG`, `.XML`, `.PDF`, `.JPG`,…) vars sökväg kan ställas in med *Autofill*-alternativen som exponerats i förväg. Använd t.ex. *Calculate by Expression…* för att ställa in utdatafilnamn till komplexa uttryck som:

```
'/home/me/stuff/buffer_' || left(@INPUT, 30) || '_' || @DISTANCE || '.shp'
```

Du kan också skriva in filsökvägen direkt eller använda den dialogruta för filval som visas när du klickar på den tillhörande …-knappen. När du har valt fil visas en ny dialogruta för automatisk komplettering av andra celler i samma kolumn (samma parameter).

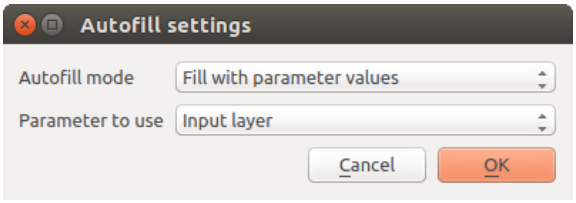


Fig. 23.34: Batchbehandling Spara

Om standardvärdet (*Do not autofill*) väljs kommer det valda filnamnet bara att placeras i den valda cellen i parametertabellen. Om något av de andra alternativen väljs kommer alla celler **under** den valda att fyllas i automatiskt baserat på ett definierat kriterium:

- *Fyll med siffror*: lägger stegvis till en siffra till filnamnet
- *Fyll med parametervärden*: du kan välja en parameter vars värde i samma rad läggs till filnamnet. Detta är särskilt användbart för att namnge utdataobjekt enligt indataobjekt.

- sparas som ett lager i en databasbehållare:

```
# 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 Exekvering av batchprocessen

För att köra batchprocessen när du har angett alla nödvändiga värden klickar du bara på *Run*. Panelen *Log* aktiveras och visar detaljer och steg i exekveringsprocessen. Förloppet för den globala batchuppgiften visas i förloppsindikatorn i den nedre delen av dialogrutan.

23.7 Använda bearbetningsalgoritmer från konsolen

Konsolen gör det möjligt för avancerade användare att öka sin produktivitet och utföra komplexa operationer som inte kan utföras med hjälp av något av de andra grafiska gränssnittselementen i bearbetningsramverket. Modeller som innehåller flera algoritmer kan definieras med hjälp av kommandoradsgränssnittet och ytterligare operationer som loopar och villkorliga meningar kan läggas till för att skapa mer flexibla och kraftfulla arbetsflöden.

Det finns ingen bearbetningskonsol i QGIS, men alla bearbetningskommandon är istället tillgängliga från QGIS inbyggda *Python-konsol*. Det innebär att du kan införliva dessa kommandon i ditt konsolarbete och ansluta bearbetningsalgoritmer till alla andra funktioner (inklusive metoder från QGIS API) som finns tillgängliga därifrån.

Den kod som du kan exekvera från Python-konsolen, även om den inte anropar någon specifik bearbetningsmetod, kan konverteras till en ny algoritm som du senare kan anropa från verktygslådan, den grafiska modelleraren eller någon annan komponent, precis som du gör med vilken annan algoritm som helst. Faktum är att vissa algoritmer som du kan hitta i verktygslådan är enkla skript.

I det här avsnittet ska vi se hur man använder bearbetningsalgoritmer från QGIS Python-konsolen, och även hur man skriver algoritmer med Python.

23.7.1 Anropa algoritmer från Python-konsolen

Det första du måste göra är att importera bearbetningsfunktionerna med följande rad:

```
>>> from qgis import processing
```

Nu finns det i princip bara en (intressant) sak du kan göra med det från konsolen: exekvera en algoritm. Det görs med metoden `run()`, som tar namnet på den algoritm som ska köras som sin första parameter och sedan ett varierande antal ytterligare parametrar beroende på algoritmens krav. Så det första du behöver veta är namnet på den algoritm som ska köras. Det är inte det namn som du ser i verktygslådan, utan snarare ett unikt kommandoradsnamn. För att hitta rätt namn för din algoritm kan du använda `processingRegistry`. Skriv följande rad i din konsol:

```
>>> for alg in QgsApplication.processingRegistry().algorithms():
        print(alg.id(), "->", alg.displayName())
```

Du kommer att se något liknande detta (med några extra streck tillagda för att förbättra läsbarheten).

```
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
...
```

Det är en lista över alla tillgängliga algoritm-ID:n, sorterade efter leverantörsnamn och algoritmnamn, tillsammans med deras motsvarande namn.

När du vet kommandoradsnamnet på algoritmen är nästa sak att bestämma rätt syntax för att exekvera den. Det innebär att du måste veta vilka parametrar som behövs när du anropar metoden `run()`.

Det finns en metod för att beskriva en algoritm i detalj, som kan användas för att få en lista över de parametrar som en algoritm kräver och de utdata som den kommer att generera. För att få den här informationen kan du använda metoden `algorithmHelp(id_of_the_algorithm)`. Använd algoritmens ID, inte det fullständiga beskrivande namnet.

Om man anropar metoden med `native:buffer` som parameter (`qgis:buffer` är ett alias för `native:buffer` och fungerar också) får man följande beskrivning:

```
>>> 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.
```

(continues on next page)

(fortsättning från föregående sida)

```
The join style parameter specifies whether round, miter or
beveled joins should be used when offsetting corners in a line.

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
```

(continues on next page)

(fortsättning från föregående sida)

```
   Available values:
           - 0: Round
           - 1: Miter
           - 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 PostGIS table
           - QgsProcessingOutputLayerDefinition
           - QgsProperty

----------------
Outputs
----------------

OUTPUT:  <QgsProcessingOutputVectorLayer>
   Buffered
```

Nu har du allt du behöver för att köra en algoritm. Som vi redan har nämnt kan algoritmer köras med hjälp av: `run()`. Dess syntax är som följer:

```
>>> processing.run(name_of_the_algorithm, parameters)
```

Där parametrar är en ordlista med parametrar som beror på den algoritm du vill köra, och är exakt den lista som metoden `algorithmHelp()` ger dig.

```
>>> processing.run("native:buffer", {'INPUT': '/data/lines.shp',
              'DISTANCE': 100.0,
              'SEGMENTS': 10,
              'DISSOLVE': True,
              'END_CAP_STYLE': 0,
```

(continues on next page)

(fortsättning från föregående sida)

```
                'JOIN_STYLE': 0,
                'MITER_LIMIT': 10,
                'OUTPUT': '/data/buffers.shp'})
```

Om en parameter är valfri och du inte vill använda den, ska du inte ta med den i ordlistan.

Om en parameter inte anges kommer standardvärdet att användas.

Beroende på typ av parameter introduceras värdena på olika sätt. Nästa lista ger en snabb genomgång av hur man introducerar värden för varje typ av inmatningsparameter:

- Rasterlager, vektorlager eller tabell. Använd helt enkelt en sträng med namnet som identifierar dataobjektet som ska användas (namnet som det har i QGIS innehållsförteckning) eller ett filnamn (om motsvarande lager inte är öppnat kommer det att öppnas men inte läggas till i kartbilden). Om du har en instans av ett QGIS-objekt som representerar lagret kan du också skicka det som parameter.
- Uppräkning. Om en algoritm har en uppräkningsparameter ska värdet för denna parameter anges med ett heltalsvärde. Om du vill veta vilka alternativ som finns kan du använda kommandot `algorithmHelp()`, enligt ovan. Till exempel har algoritmen `native:buffer` en uppräkning som heter 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
  ```

  I det här fallet har parametern tre alternativ. Observera att ordningen är nollbaserad.
- Booleansk. Använd `True` eller `False`.
- Flera inmatningar. Värdet är en sträng med indatabeskrivningar åtskilda med semikolon (`;`). Precis som för enskilda lager eller tabeller kan varje indatabeskrivare vara dataobjektets namn eller dess filsökväg.
- Tabellfält från XXX. Använd en sträng med namnet på det fält som ska användas. Denna parameter är skiftlägeskänslig.
- Fast tabell. Skriv in listan över alla tabellvärden separerade med kommatecken (`,`) och inneslutna mellan citattecken (`"`). Värdena börjar på den övre raden och går från vänster till höger. Du kan också använda en 2D-array med värden som representerar tabellen.
- CRS. Ange EPSG-kodnumret för önskad CRS.
- Utsträckning. Du måste använda en sträng med värdena `xmin`, `xmax`, `ymin` och `ymax` åtskilda med kommatecken (`,`).

Booleska, fil-, sträng- och numeriska parametrar behöver inga ytterligare förklaringar.

Inmatningsparametrar som strängar, booleaner eller numeriska värden har standardvärden. Standardvärdet används om motsvarande parameterinmatning saknas.

För objekt med utdatadata skriver du den filsökväg som ska användas för att spara dem, precis som i verktygslådan. Om utdataobjektet inte anges sparas resultatet i en tillfällig fil (eller hoppas över om det är en valfri utdata). Filtillägget avgör vilket filformat som används. Om du anger en filändelse som inte stöds av algoritmen används standardfilformatet för den utdatatypen och motsvarande filändelse läggs till i den angivna filsökvägen.

Till skillnad från när en algoritm exekveras från verktygslådan läggs inte utdata till i kartbilden om du exekverar samma algoritm från Python-konsolen med metoden `run()`. Den metoden returnerar en ordbok med ett eller flera

utdatanamn (de som visas i algoritmbeskrivningen) som nycklar och filsökvägarna för dessa utdata som värden:

```
>>> myresult = 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'})
>>> myresult['OUTPUT']
/data/buffers.shp
```

Du kan sedan ladda utdata i projektet som vilket vanligt lager som helst:

```
>>> buffered_layer = myresult['OUTPUT']
>>> QgsProject.instance().addMapLayer(buffered_layer)
```

För att omedelbart ladda bearbetningsresultaten i projektet kan du använda metoden `runAndLoadResults()` i stället för `run()`.

```
>>> processing.runAndLoadResults("native:buffer", {parameters:values})
```

Om du vill öppna en algoritmdialog från konsolen kan du använda metoden `createAlgorithmDialog`. Den enda obligatoriska parametern är algoritmens namn, men du kan också definiera en ordlista med parametrar så att dialogrutan fylls i automatiskt:

```
>>> 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()
```

Metoden `execAlgorithmDialog` öppnar dialogrutan omedelbart:

```
>>> 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 Skapa skript och köra dem från verktygslådan

Du kan skapa dina egna algoritmer genom att skriva Python-kod. Bearbetningsskript utökar `QgsProcessingAlgorithm`, så du måste lägga till några extra rader kod för att implementera obligatoriska funktioner. Du hittar *Create new script* (rent skript) och *Create New Script from Template* (mall som innehåller kod för obligatoriska funktioner i `QgsProcessingAlgorithm`) under rullgardinsmenyn *Scripts* högst upp i verktygslådan Processing. Processing Script Editor öppnas och det är där du ska skriva din kod. Om du sparar skriptet därifrån i mappen `scripts` (standardmappen när du öppnar dialogrutan för att spara filer) med tillägget `.py` skapas motsvarande algoritm.

Namnet på algoritmen (den som du ser i verktygslådan) definieras i koden.

Låt oss ta en titt på följande kod, som definierar en Processing-algoritm som utför en buffertoperation med ett användardefinierat buffertavstånd på ett vektorlager som anges av användaren, efter att först ha utjämnat lagret.

```
from qgis.core import (QgsProcessingAlgorithm,
       QgsProcessingParameterNumber,
       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}
```

Efter att ha gjort de nödvändiga importerna specificeras följande `QgsProcessingAlgorithm` funktioner:

- `name()`: Algoritmens id (gemener).

- `displayName()`: Ett mänskligt läsbart namn för algoritmen.
- `createInstance()`: Skapa en ny instans av algoritmklassen.
- `initAlgorithm()`: Konfigurera parameterDefinitioner och outputDefinitioner.

  Här beskriver du algoritmens parametrar och utdata. I det här fallet en funktionskälla för indata, en funktionssänka för resultatet och ett tal för buffertavståndet.
- `processAlgorithm()`: Gör arbetet.

  Här kör vi först algoritmen `smoothgeometry` för att jämna ut geometrin, och sedan kör vi algoritmen `buffer` på det utjämnade resultatet. För att kunna köra algoritmer inifrån en annan algoritm måste vi sätta argumentet `is_child_algorithm` till `True`. Du kan se hur in- och utparametrar används som parametrar för algoritmerna `smoothgeometry` och `buffer`.

Det finns ett antal olika parametertyper tillgängliga för in- och utdata. Du kan hitta hela listan på *Inmatnings- och utmatningstyper för bearbetningsalgoritmer*.

Den första parametern till konstruktörerna är namnet på parametern och den andra är beskrivningen av parametern (för användargränssnittet). Resten av konstruktörsparametrarna är parametertypspecifika.

Indata kan omvandlas till QGIS-klasser med hjälp av funktionerna `parameterAs` i `QgsProcessingAlgorithm`. Till exempel för att få det nummer som anges för buffertavståndet som en dubbel:

```
self.parameterAsDouble(parameters, self.INPUT_BUFFERDIST, context)).
```

Funktionen `processAlgorithm` bör returnera en ordbok som innehåller värden för varje utdata som definieras av algoritmen. Detta gör det möjligt att få tillgång till dessa utdata från andra algoritmer, inklusive andra algoritmer som ingår i samma modell.

Välskötta algoritmer bör definiera och returnera så många utdata som är rimligt. Utdata som inte är funktioner, t.ex. siffror och strängar, är mycket användbara när du kör din algoritm som en del av en större modell, eftersom dessa värden kan användas som inparametrar för efterföljande algoritmer inom modellen. Överväg att lägga till numeriska utdata för saker som antalet bearbetade funktioner, antalet ogiltiga funktioner som påträffats, antalet funktioner som matats ut osv. Ju fler utdata du returnerar, desto mer användbar blir din algoritm!

Feedback

Objektet `feedback` som skickas till `processAlgorithm()` bör användas för användarens återkoppling/interaktion. Du kan använda funktionen `setProgress()` i objektet `feedback` för att uppdatera förloppsindikatorn (0 till 100) för att informera användaren om algoritmens framsteg. Detta är mycket användbart om din algoritm tar lång tid att slutföra.

Objektet `feedback` tillhandahåller en metod `isCanceled()` som bör övervakas för att användaren ska kunna avbryta algoritmen. Metoden `pushInfo()` i `feedback` kan användas för att skicka information till användaren, och `reportError()` är praktisk för att skicka icke-fatala fel till användare.

Algoritmer bör undvika att använda andra former av återkoppling till användare, t.ex. utskriftssatser eller loggning till `QgsMessageLog`, och bör alltid använda feedback-objektet istället. Detta möjliggör utförlig loggning för algoritmen och är även trådsäkert (vilket är viktigt eftersom algoritmer vanligtvis körs i en bakgrundstråd).

Hantering av fel

Om din algoritm stöter på ett fel som hindrar den från att köras, t.ex. ogiltiga indatavärden eller något annat tillstånd som den inte kan eller bör återhämta sig från, bör du skapa en `QgsProcessingException`. T.ex.:

```
if feature['value'] < 20:
  raise QgsProcessingException('Invalid input value {}, must be >= 20'.
↪format(feature['value']))
```

Försök att undvika `QgsProcessingException` för icke-fatala fel (t.ex. när en funktion har en null-geometri), och rapportera istället bara dessa fel via `feedback.reportError()` och hoppa över funktionen. Detta bidrar till att göra din algoritm ”modellvänlig”, eftersom den undviker att stoppa exekveringen av en hel algoritm när ett icke-fatalt fel påträffas.

Dokumentera dina skript

På samma sätt som för modeller kan du skapa ytterligare dokumentation för dina skript för att förklara vad de gör och hur de ska användas.

`QgsProcessingAlgorithm` tillhandahåller `helpString()`, `shortHelpString()` och `helpUrl()` funktioner för detta ändamål. Ange / åsidosätt dessa för att ge mer hjälp till användaren.

`shortDescription()` används i verktygstipset när du håller muspekaren över algoritmen i verktygslådan.

23.7.3 Skriptkrokar före och efter utförandet

Skript kan också användas som pre- och post-exekveringskrokar som körs före respektive efter att en algoritm körs. Detta kan användas för att automatisera uppgifter som ska utföras när en algoritm körs.

Syntaxen är identisk med den som förklaras ovan, men det finns ytterligare en global variabel med namnet `alg` som representerar den algoritm som just har körts (eller ska köras).

I gruppen *General* i dialogen processing options finns två poster med namnen *Pre-execution script* och *Post-execution script* där filnamnen på de skript som ska köras i varje enskilt fall kan anges.

23.8 Använda bearbetning från kommandoraden

QGIS levereras med ett verktyg som heter `QGIS Processing Executor` som gör att du kan köra Processing-algoritmer och modeller (inbyggda eller tillhandahållna av tillägg) direkt från kommandoraden utan att starta QGIS Desktop själv.

Från ett kommandoradsverktyg, kör `qgis_process` och du bör få:

```
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)
```

(continues on next page)

(fortsättning från föregående sida)

```
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,
                      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.
```

Observera

Endast installerade tillägg som annonserar `hasProcessingProvider=yes` i sin `metadata.txt`-fil känns igen och kan aktiveras eller laddas av qgis_process-verktyget.

Råd

Innan du anropar qgis_process på ett system utan fönsterhanterare (t.ex. en huvudlös server) bör du ställa in:

```
export QT_QPA_PLATFORM=offscreen
```

Kommandot `list` kan användas för att få en lista över alla tillgängliga leverantörer och algoritmer.

```
qgis_process list
```

Kommandot `help` kan användas för att få ytterligare information om kommandon eller algoritmer.

```
qgis_process help qgis:regularpoints
```

Kommandot `run` kan användas för att köra en algoritm eller modell. Ange namnet på algoritmen eller en sökväg till en modell som första parameter.

```
qgis_process run native:buffer -- INPUT=source.shp DISTANCE=2 OUTPUT=buffered.shp
```

Om en parameter accepterar en lista med värden kan du ange samma variabel flera gånger.

```
qgis_process run native:mergevectorlayers -- LAYERS=input1.shp LAYERS=input2.shp␣
↪OUTPUT=merged.shp
```

När en algoritm körs visas ett textbaserat feedbackfält och operationen kan avbrytas via `CTRL+C`.

Kommandot `run` har även stöd för ytterligare parametrar.

- `--json` formaterar utdata från stdout på ett JSON-strukturerat sätt.
- `--ellipsoid` kommer att ställa in ellipsoiden till den angivna.
- `--distance_units` kommer att använda de angivna avståndsenheterna.
- `--area_units` kommer att använda de angivna areaenheterna.
- `--project_path` kommer att ladda det angivna projektet för att köra algoritmen.

Komplexa inmatningsparametrar, dvs. parametertyper som själva specificeras som ett objekt av typen dictionary för algoritmer, stöds av qgis_process. För att ange att parametrar kommer att anges via stdin måste kommandot qgis_process följa formatet (med en efterföljande – i stället för den vanliga argumentlistan).

```
qgis_process run algorithmId –
```

JSON-objektet måste innehålla en ”inputs”-nyckel, som är en karta över värdena för inparametrarna. T.ex.

```
echo "{'inputs': {'INPUT': 'my_shape.shp', 'DISTANCE': 5}}" | qgis_process run␣
↪native:buffer –
```

Dessutom kan extra inställningar som avståndsenheter, ytenheter, ellipsoid och projektväg inkluderas i detta JSON-objekt:

```
{
 'ellipsoid': 'EPSG:7019',
 'distance_units': 'feet',
 'area_units': 'ha',
 'project_path': 'C:/temp/my_project.qgs'
 'inputs': {'DISTANCE': 5, 'SEGMENTS': 8 ... }
}
```

Om du anger inparametrar via stdin innebär det automatiskt att `JSON` används som utdataformat för resultaten.

23.9 Skriva nya Processing-algoritmer som Python-skript

Det finns två alternativ för att skriva Processing-algoritmer med hjälp av Python.

- *Extending* `QgsProcessingAlgorithm <qgis.core.QgsProcessingAlgorithm>`
- *Använder dekoratorn @alg*

I QGIS kan du använda *Create new script* i menyn *Scripts* högst upp i verktygslådan *Processing Toolbox* för att öppna *Processing Script Editor* där du kan skriva din kod. För att förenkla uppgiften kan du börja med en skriptmall genom att använda *Create new script from template* från samma meny. Detta öppnar en mall som utökar `QgsProcessingAlgorithm`.

Om du sparar skriptet i mappen `scripts` (standardplatsen) med tillägget `.py` blir algoritmen tillgänglig i verktygslådan *Processing Toolbox*.

23.9.1 Förlängning av QgsProcessingAlgorithm

Följande kod

1. tar ett vektorlager som indata
2. räknar antalet funktioner
3. gör en buffertoperation
4. skapar ett rasterlager från resultatet av buffertoperationen
5. returnerar buffertlager, rasterlager och antal funktioner

```
from qgis.PyQt.QtCore import QCoreApplication
from qgis.core import (QgsProcessing,
                       QgsProcessingAlgorithm,
                       QgsProcessingException,
                       QgsProcessingOutputNumber,
                       QgsProcessingParameterDistance,
                       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.
```

(continues on next page)

(fortsättning från föregående sida)

```
        """
        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(
                '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):
```

(continues on next page)

(fortsättning från föregående sida)

```
        """
        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)
        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.
```

(continues on next page)

(fortsättning från föregående sida)

```
                '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}
```

Standardfunktioner för bearbetningsalgoritm:

- **createInstance (obligatorisk)**
  Måste returnera en ny kopia av din algoritm. Om du ändrar namnet på klassen, se till att du också uppdaterar det värde som returneras här för att matcha!
- **namn (obligatoriskt)**
  Returnerar det unika algoritmnamnet som används för att identifiera algoritmen.
- **displayName (obligatoriskt)**
  Returnerar det översatta algoritmnamnet.
- **group**
  Returnerar namnet på den grupp som den här algoritmen tillhör.
- **groupId**
  Returnerar det unika ID:t för den grupp som den här algoritmen tillhör.
- **shortHelpString**
  Returnerar en lokaliserad kort hjälpsträng för algoritmen.
- **initAlgorithm (obligatorisk)**
  Här definierar vi algoritmens in- och utgångar.

  `INPUT` och `OUTPUT` är rekommenderade namn för huvudparametrarna för inmatning respektive utmatning.

  Om en parameter är beroende av en annan parameter används `parentParameterName` för att ange detta förhållande (kan vara fältet/bandet för ett lager eller avståndsenheterna för ett lager).
- **processAlgorithm (obligatorisk)**
  Det är här som bearbetningen sker.

  Parametrar hämtas med hjälp av specialfunktioner, t.ex. `parameterAsSource` och `parameterAsDouble`.

  `processing.run` kan användas för att köra andra bearbetningsalgoritmer från en bearbetningsalgoritm. Den första parametern är namnet på algoritmen, den andra är en ordbok med parametrarna för algoritmen. `is_child_algorithm` sätts normalt till `True` när en algoritm körs från en annan algoritm. `context` och `feedback` informerar algoritmen om den miljö den ska köras i och kanalen för att kommunicera med användaren (fånga upp begäran om avbrytande, rapportera framsteg, ge textuell feedback). När (föräldra)algoritmens parametrar används som parametrar för ”underordnade” algoritmer, skall de ursprungliga parametervärdena användas (t.ex. `parameters['OUTPUT']`).

  Det är god praxis att kontrollera feedbackobjektet för annullering så mycket som det rimligen är möjligt! Detta gör det möjligt att avbryta på ett smidigt sätt, istället för att tvinga användarna att vänta på att en oönskad behandling ska ske.

  Algoritmen ska returnera värden för alla utdataparametrar som den har definierat som en ordbok. I det här fallet är det buffert- och rastrerade utdatalager samt antalet bearbetade funktioner. Ordbokens nycklar

måste matcha de ursprungliga parameter-/utdatanamnen.

23.9.2 @alg-dekoratorn

Med hjälp av @alg-dekoratorn kan du skapa dina egna algoritmer genom att skriva Python-koden och lägga till några extra rader för att tillhandahålla ytterligare information som behövs för att göra den till en riktig Processing-algoritm. Detta förenklar skapandet av algoritmer och specifikationen av in- och utgångar.

En viktig begränsning med dekoratormetoden är att algoritmer som skapas på detta sätt alltid kommer att läggas till i användarens Processing Scripts-provider - det är inte möjligt att lägga till dessa algoritmer i en anpassad provider, t.ex. för användning i tillägg.

Följande kod använder @alg-dekoratorn för att

1. använda ett vektorlager som indata
2. räkna antalet funktioner
3. göra en buffertoperation
4. skapa ett rasterlager från resultatet av buffertoperationen
5. returnerar buffertlager, rasterlager och antal funktioner

```
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,
```

(continues on next page)

(fortsättning från föregående sida)

```
                                    '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'],
            'NUMBEROFFEATURES': numfeatures}
```

Som du kan se handlar det om två algoritmer (’native:buffer’ och ’qgis:rasterize’). Den sista (’qgis:rasterize’) skapar ett rasterlager från det buffertlager som genererades av den första (’native:buffer’).

Den del av koden där denna bearbetning sker är inte svår att förstå om du har läst föregående kapitel. De första raderna kräver dock lite extra förklaring. De innehåller den information som behövs för att omvandla din kod till en algoritm som kan köras från någon av de grafiska gränssnittskomponenterna, t.ex. verktygslådan eller modelldesignern.

Dessa rader är alla anrop till dekoratorfunktionerna `@alg` som hjälper till att förenkla kodningen av algoritmen.

- Dekoratorn @alg används för att definiera algoritmens namn och plats i verktygslådan.
- Dekoratorn @alg.input används för att definiera algoritmens ingångar.
- Dekoratorn @alg.output används för att definiera algoritmens utdata.

För befintliga parametrar och deras motsvarighet, läs *Inmatnings- och utmatningstyper för bearbetningsalgoritmer*.

23.9.3 Utmatning av algoritm för överlämning

När du deklarerar en utdata som representerar ett lager (raster eller vektor) kommer algoritmen att försöka lägga till det i QGIS när den är klar.

- Utmatning av rasterlager: QgsProcessingParameterRasterDestination / alg.RASTER_LAYER_DEST.
- Vektorskiktets utgång: QgsProcessingParameterVectorDestination / alg.VECTOR_LAYER_DEST.

Så även om metoden `processing.run()` inte lägger till de lager som skapas i användarens aktuella projekt, kommer de två utdatalagren (buffert och rasterbuffert) att laddas, eftersom de sparas till de destinationer som användaren anger (eller till tillfälliga destinationer om användaren inte anger några destinationer).

Om ett lager skapas som resultat av en algoritm bör det deklareras som sådant. Annars kommer du inte att kunna använda algoritmen på rätt sätt i modelleraren, eftersom det som deklareras inte kommer att stämma överens med vad algoritmen verkligen skapar.

Du kan returnera strängar, tal och annat genom att ange dem i resultatordlistan (som visas för ”NUMBEROFFEATURES”), men de ska alltid uttryckligen definieras som utdata från din algoritm. Vi uppmuntrar algoritmer att ge ut så många användbara värden som möjligt, eftersom dessa kan vara värdefulla för användning i senare algoritmer när din algoritm används som en del av en modell.

23.9.4 Kommunicera med användaren

Om din algoritm tar lång tid att bearbeta är det en bra idé att informera användaren om framstegen. Du kan använda `feedback` (`QgsProcessingFeedback`) för detta.

Förloppstexten och förloppsfältet kan uppdateras med två metoder: `setProgressText(text)` och `setProgress(percent)`.

Du kan ge mer information genom att använda `pushCommandInfo(text)`, `pushDebugInfo(text)`, `pushInfo(text)` och `reportError(text)`.

Om ditt skript har ett problem är det korrekta sättet att hantera det att skapa ett `QgsProcessingException`. Du kan skicka ett meddelande som ett argument till konstruktören för undantaget. Processing tar hand om hanteringen och kommunikationen med användaren, beroende på varifrån algoritmen körs (verktygslåda, modellerare, Python-konsol, …)

23.9.5 Dokumentera dina skript

Du kan dokumentera dina skript genom att överbelasta metoderna `helpString()` och `helpUrl()` i `QgsProcessingAlgorithm`.

23.9.6 Flaggor

Du kan åsidosätta metoden `flags()` i `QgsProcessingAlgorithm` för att berätta mer för QGIS om din algoritm. Du kan t.ex. tala om för QGIS att skriptet ska döljas för modelleraren, att det kan avbrytas, att det inte är trådsäkert och mycket mer.

Tips

Som standard kör Processing algoritmer i en separat tråd för att hålla QGIS responsivt medan bearbetningsuppgiften körs. Om din algoritm kraschar regelbundet använder du förmodligen API-anrop som inte är säkra att göra i en bakgrundstråd. Försök att returnera flaggan QgsProcessingAlgorithm.FlagNoThreading från din algoritms flags()-metod för att tvinga Processing att köra din algoritm i huvudtråden istället.

23.9.7 Bästa praxis för att skriva skriptalgoritmer

Här är en snabb sammanfattning av idéer att tänka på när du skapar dina skriptalgoritmer och, särskilt, om du vill dela dem med andra QGIS-användare. Genom att följa dessa enkla regler säkerställs enhetlighet mellan de olika bearbetningselementen, t.ex. verktygslådan, modelleraren eller gränssnittet för batchbearbetning.

- Ladda inte resulterande lager. Låt Processing hantera dina resultat och ladda dina lager om det behövs.
- Deklarera alltid de utdata som din algoritm skapar.
- Visa inte meddelanderutor eller använd något grafiskt gränssnittselement från skriptet. Om du vill kommunicera med användaren använder du metoderna i feedback-objektet (`QgsProcessingFeedback`) eller kastar en `QgsProcessingException`.

Det finns redan många bearbetningsalgoritmer tillgängliga i QGIS. Du kan hitta kod på QGIS repo.

23.10 Inmatnings- och utmatningstyper för bearbetningsalgoritmer

Här är listan över in- och utdatatyper som stöds i Processing med deras motsvarande alg-dekoratorkonstanter, deras namn i modelleringsdesignern om de exponeras.

23.10.1 Inmatningstyper

Namn på inmatning	Klass	Alg konstant	Beskrivning
Lager för annoteringar	QgsProcessingParame	alg. ANNOTATION_LAYE	Ett lager med anteckningar
Område	QgsProcessingParame		En numerisk parameter som representerar ett ytmått
Konfiguration av autentisering	QgsProcessingParame	alg.AUTH_CFG	Låter användare välja bland tillgängliga autentiseringskonfigurationer eller skapa nya autentiseringskonfigurationer
Rasterband	QgsProcessingParame	alg.BAND	Ett band av ett rasterlager
Boolean	QgsProcessingParame	alg.BOOL	Ett booleanskt värde
Kontaktens namn	QgsProcessingParame		Ett urval av tillgängliga registrerade anslutningar för en viss dataleverantör
Färg	QgsProcessingParame	alg.COLOR	En färg
Koordinera driften	QgsProcessingParame	alg. COORDINATE_OPER	En koordinatoperation (för CRS-transformationer)
CRS	QgsProcessingParame	alg.CRS	Ett koordinatbaserat referenssystem
Schema för databas	QgsProcessingParame	alg. DATABASE_SCHEMA	Ett databasschema
Databastabell	QgsProcessingParame	alg. DATABASE_TABLE	En databastabell
Datum	ProcessingDateTimeP Date	alg.DATE	En ren träff
Datum Tid	ProcessingDateTimeP DateTime	alg.DATETIME	En datatid
Avstånd	QgsProcessingParame	alg.DISTANCE	En dubbel numerisk parameter för avståndsvärden
DXF-lager	QgsProcessingParame		En lista över vektorlager som ska exporteras som DXF-fil
Enum	QgsProcessingParame	alg.ENUM	En uppräkning som gör det möjligt att välja från en uppsättning fördefinierade värden
Uttryck	QgsProcessingParame	alg. EXPRESSION	Ett uttryck
Utsträckning	QgsProcessingParame	alg.EXTENT	En spatial utsträckning definierad av xmin, xmax, ymin, ymax
	QgsProcessingParame	alg.FIELD	Ett fält i attributtabellen för ett vektorlager
Fältaggregat	QgsProcessingParame		En parameter för ”aggregat”-konfigurationer, som består av en definition av önskade utdatafält, typer och aggregat som används för att fylla dem

continues on next page

Table 23.1 – fortsättning från föregående sida

Namn på inmatning	Klass	Alg konstant	Beskrivning
Fält Mapper	QgsProcessingParame		En parameter för konfigurationer av ”fältmappning”, som består av en definition av önskade utdatafält, typer och uttryck som används för att fylla dem
Fil/mapp	QgsProcessingParame	alg.FILE	Ett filnamn för en befintlig fil
	QgsProcessingParame	alg.FILE_DEST	Ett filnamn för en nyskapad utdatafil
	QgsProcessingParame	alg.FOLDER_DEST	En mapp (målmapp)
Geometri	QgsProcessingParame	alg.GEOMETRY	En geometri
	ProcessingNumberPar Integer	alg.INT	Ett heltal
Layout för utskrift	QgsProcessingParame	alg.LAYOUT	En layout
Objekt för utskriftslayout	QgsProcessingParame	alg.LAYOUT_ITEM	Ett layoutobjekt
Kartlager	QgsProcessingParame	alg.MAPLAYER	Ett kartlager
Karttema	QgsProcessingParame	alg.MAP_THEME	Ett tema för projektkartan
Matris	QgsProcessingParame	alg.MATRIX	En matris
Grupper av nätdataset	QgsProcessingParame		Datasetgrupper av nätlager
Mesh Dataset Tid	QgsProcessingParame		Datasetindex från ett nätlager med tidsinställning
Nätlager	QgsProcessingParame	alg.MESH_LAYER	Ett nätlager
Flera inmatningar	QgsProcessingParame	alg.MULTILAYER	En uppsättning lager
Siffra	ProcessingNumberPar Double	alg.NUMBER	Ett numeriskt värde
Punkt	QgsProcessingParame	alg.POINT	En punkt
Punktmolnsattribut	QgsProcessingParame		Ett attribut från ett punktmolnslager
	QgsProcessingParame	alg.POINTCLOUD_LAYE	En destinationsparameter för ett punktmolnslager, för att specificera destinationsvägen för ett punktmolnslager som skapats av algoritmen
Punktmolnslager	QgsProcessingParame	alg.POINTCLOUD_LAYE	Ett punktmolnslager
	QgsProcessingParame	alg.PROVIDER_CONNEC	En tillgänglig anslutning för en databasleverantör
Intervall	QgsProcessingParame	alg.RANGE	Ett nummerintervall
Rasterlager	QgsProcessingParame	alg.RASTER_LAYER	Ett rasterlager
	QgsProcessingParame	alg.RASTER_LAYER_DE	En destinationsparameter för rasterlager, för att specificera destinationsvägen för ett rasterlager som skapats av algoritmen
Skala	QgsProcessingParame	alg.SCALE	En kartskala
	QgsProcessingParame	alg.SINK	En funktionssänka
Text	QgsProcessingParame	alg.STRING	En textsträng
TIN-skapande lager	QgsProcessingParame		Val av flera lager för att skapa en TIN med hörnpunkter och/eller brytlinjer

continues on next page

Table 23.1 – fortsättning från föregående sida

Namn på inmatning	Klass	Alg konstant	Beskrivning
Tid	ProcessingDateTimeP Time	alg.TIME	En ren tid
Vektorfunktioner	QgsProcessingParame	alg.SOURCE	En funktionskälla
Vektorlager	QgsProcessingParame	alg. VECTOR_LAYER	Ett vektorlager
	QgsProcessingParame	alg. VECTOR_LAYER_DE	En destinationsparameter för vektorlager, för att specificera destinationsvägen för ett vektorlager som skapats av algoritmen
Vektor Tile Writer-lager	QgsProcessingParame		En destinationsparameter för ett vektorskiktlager, för att specificera destinationsvägen för ett vektorskiktlager som skapats av algoritmen
Volym	QgsProcessingParame		En numerisk parameter som representerar ett volymmått

23.10.2 Typer av utdata

Klass	Alg konstant	Beskrivning
QgsProcessingOutputBoolean	alg.BOOL	Ett booleanskt värde
QgsProcessingOutputNumber	alg.DISTANCE	En dubbel numerisk parameter för avståndsvärden
QgsProcessingOutputFile	alg.FILE	Ett filnamn för en befintlig fil
QgsProcessingOutputFolder	alg.FOLDER	Videomapp
QgsProcessingOutputHtml	alg.HTML	HTML
QgsProcessingOutputNumber	alg.INT	Ett heltal
QgsProcessingOutputLayerDefinition	alg.LAYERDEF	En definition av lager
QgsProcessingOutputMapLayer	alg.MAPLAYER	Ett kartlager
QgsProcessingOutputMultipleLayers	alg.MULTILAYER	En uppsättning lager
QgsProcessingOutputNumber	alg.NUMBER	Ett numeriskt värde
QgsProcessingOutputPointCloudLayer	alg. POINTCLOUD_LAYER	Ett punktmolnslager
QgsProcessingOutputRasterLayer	alg.RASTER_LAYER	Ett rasterlager
QgsProcessingOutputString	alg.STRING	En textsträng
QgsProcessingOutputVectorLayer	alg.VECTOR_LAYER	Ett vektorlager
QgsProcessingOutputVectorTileLayer		Ett lager med vektorplattor

23.11 Konfigurera externa applikationer

Ramverket för bearbetning kan utökas med ytterligare applikationer. Algoritmer som förlitar sig på externa applikationer hanteras av sina egna algoritmleverantörer. Ytterligare leverantörer finns som separata tillägg och installeras med QGIS Tilläggshanterare.

I det här avsnittet visas hur du konfigurerar Processing-ramverket så att det omfattar dessa ytterligare applikationer, och några speciella egenskaper hos de algoritmer som baseras på dem förklaras. När du har konfigurerat systemet på rätt sätt kan du köra externa algoritmer från valfri komponent, t.ex. verktygslådan eller modelldesignern, precis som du gör med alla andra algoritmer.

Som standard är algoritmer som förlitar sig på en extern applikation som inte levereras med QGIS inte aktiverade. Du kan aktivera dem i dialogrutan Bearbetningsinställningar om de är installerade på ditt system.

23.11.1 En anmärkning för Windows-användare

Om du inte är en avancerad användare och du kör QGIS på Windows, kanske du inte är intresserad av att läsa resten av detta kapitel. Se till att du installerar QGIS i ditt system med hjälp av den fristående installationsprogrammet. Detta kommer automatiskt att installera SAGA och GRASS i ditt system och konfigurera dem så att de kan köras från QGIS. Alla algoritmer från dessa leverantörer kommer att vara redo att köras utan att behöva någon ytterligare konfiguration. Om du installerar med OSGeo4W-programmet, se till att du också väljer SAGA och GRASS för installation.

23.11.2 En anmärkning om filformat

När du använder extern programvara betyder inte öppnandet av en fil i QGIS att den kan öppnas och bearbetas i den andra programvaran. I de flesta fall kan andra program läsa vad du har öppnat i QGIS, men i vissa fall kanske det inte är sant. När du använder databaser eller ovanliga filformat, oavsett om det gäller raster- eller vektorlager, kan problem uppstå. Om det händer, försök att använda välkända filformat som du är säker på att båda programmen förstår, och kontrollera konsolutmatningen (i loggpanelen) för att ta reda på vad som går fel.

Du kan t.ex. få problem och inte kunna slutföra ditt arbete om du anropar en extern algoritm med ett GRASS-rasterlager som indata. Av denna anledning kommer sådana lager inte att visas som tillgängliga för algoritmer.

Du bör dock inte ha några problem med vektorlager, eftersom QGIS automatiskt konverterar från det ursprungliga filformatet till ett format som accepteras av det externa programmet innan lagret skickas till det. Detta ger extra bearbetningstid, som kan vara betydande för stora lager, så bli inte förvånad om det tar längre tid att bearbeta ett lager från en DB-anslutning än ett lager från en dataset i Shapefile-format av liknande storlek.

Leverantörer som inte använder externa program kan bearbeta alla lager som du kan öppna i QGIS, eftersom de öppnar det för analys genom QGIS.

Alla raster- och vektorformat som produceras av QGIS kan användas som inmatningslager. Vissa leverantörer stöder inte vissa format, men alla kan exportera till vanliga format som senare kan omvandlas automatiskt av QGIS. När det gäller inmatningslager kan det öka bearbetningstiden om det krävs en konvertering.

23.11.3 En anteckning om val av vektorlager

Externa applikationer kan också göras medvetna om de urval som finns i vektorlager inom QGIS. Detta kräver dock att alla ingående vektorlager skrivs om, precis som om de ursprungligen hade ett format som inte stöds av det externa programmet. Endast när inget urval finns, eller om alternativet *Use only selected features* inte är aktiverat i den allmänna konfigurationen för bearbetning, kan ett lager skickas direkt till ett externt program.

I andra fall behöver endast utvalda funktioner exporteras, vilket medför längre exekveringstider.

23.11.4 Använda tredjepartsleverantörer

SAGA

SAGA-algoritmer kan köras från QGIS om SAGA ingår i QGIS-installationen.

Om du kör Windows ingår SAGA i både det fristående installationsprogrammet och installationsprogrammet för OSGeo4W.

Om begränsningar i SAGA:s gridsystem

De flesta SAGA-algoritmer som kräver flera rasterlager som indata kräver att de har samma gridsystem. Det vill säga att de måste täcka samma geografiska område och ha samma cellstorlek, så att deras motsvarande rutnät matchar. När du anropar SAGA-algoritmer från QGIS kan du använda vilket lager som helst, oavsett cellstorlek och utsträckning. När flera rasterlager används som indata för en SAGA-algoritm, samplar QGIS om dem till ett gemensamt rutnätssystem och skickar dem sedan till SAGA (såvida inte SAGA-algoritmen kan arbeta med lager från olika rutnätssystem).

Definitionen av det gemensamma rutnätssystemet styrs av användaren, och du hittar flera parametrar i SAGA-gruppen i inställningsfönstret för att göra detta. Det finns två sätt att ställa in målrastersystemet:

- Ställer in den manuellt. Du definierar omfattningen genom att ställa in värdena för följande parametrar:
  - *Återprovtagning min X*
  - *Återprovtagning max X*
  - *Återhämtning min Y*
  - *Återhämtning max Y*
  - *Återhämtning av cellstorlek*

  Observera att QGIS kommer att omsampla inmatningslagren i den utsträckningen, även om de inte överlappar med den.
- Ställer in det automatiskt från inmatningslager. För att välja detta alternativ markerar du bara alternativet *Use min covering grid system for resampling*. Alla andra inställningar ignoreras och den minsta utsträckning som täcker alla inmatningslagren används. Cellstorleken i mållagret är den maximala av alla cellstorlekar i inmatningslagren.

För algoritmer som inte använder flera rasterlager, eller för algoritmer som inte behöver ett unikt grid-system för indata, utförs ingen omsampling innan SAGA anropas, och dessa parametrar används inte.

Begränsningar för flerbandslager

Till skillnad från QGIS har SAGA inget stöd för flerbandslager. Om du vill använda ett flerbandslager (t.ex. en RGB- eller multispektralbild) måste du först dela upp det i enbandsbilder. För att göra detta kan du använda algoritmen ”SAGA/Grid - Tools/Split RGB image” (som skapar tre bilder från en RGB-bild) eller algoritmen ”SAGA/Grid - Tools/Extract band” (för att extrahera ett enda band).

Begränsningar i cellstorlek

SAGA förutsätter att rasterlager har samma cellstorlek i X- och Y-axeln. Om du arbetar med ett lager med olika värden för horisontell och vertikal cellstorlek kan du få oväntade resultat. I så fall kommer en varning att läggas till i bearbetningsloggen, vilket indikerar att ett inmatningslager kanske inte är lämpligt att bearbetas av SAGA.

Loggning

När QGIS anropar SAGA görs det med hjälp av kommandoradsgränssnittet, vilket innebär att en uppsättning kommandon skickas för att utföra alla nödvändiga operationer. SAGA visar sina framsteg genom att skriva information till konsolen, vilket inkluderar procentandelen av bearbetningen som redan har gjorts, tillsammans med ytterligare innehåll. Den här informationen filtreras och används för att uppdatera förloppsindikatorn medan algoritmen körs.

Både de kommandon som skickas av QGIS och den ytterligare information som skrivs ut av SAGA kan loggas tillsammans med andra loggar för bearbetning, och du kan ha nytta av dem för att spåra vad som händer när QGIS kör en SAGA-algoritm. Du hittar två inställningar, nämligen *Logga konsolutdata* och *Logga exekveringskommandon*, för att aktivera denna loggning.

De flesta andra leverantörer som använder externa program och anropar dem via kommandoraden har liknande alternativ, så du hittar dem också på andra ställen i listan med bearbetningsinställningar.

R-skript och bibliotek

För att aktivera R i Processing måste du installera tillägget **Processing R Provider** och konfigurera R för QGIS.

Konfigurationen görs i *Provider* ► *R* på fliken *Processing* i *Inställningar* ► *Alternativ*.

Beroende på ditt operativsystem kan du behöva använda *R folder* för att ange var dina R-binärfiler finns.

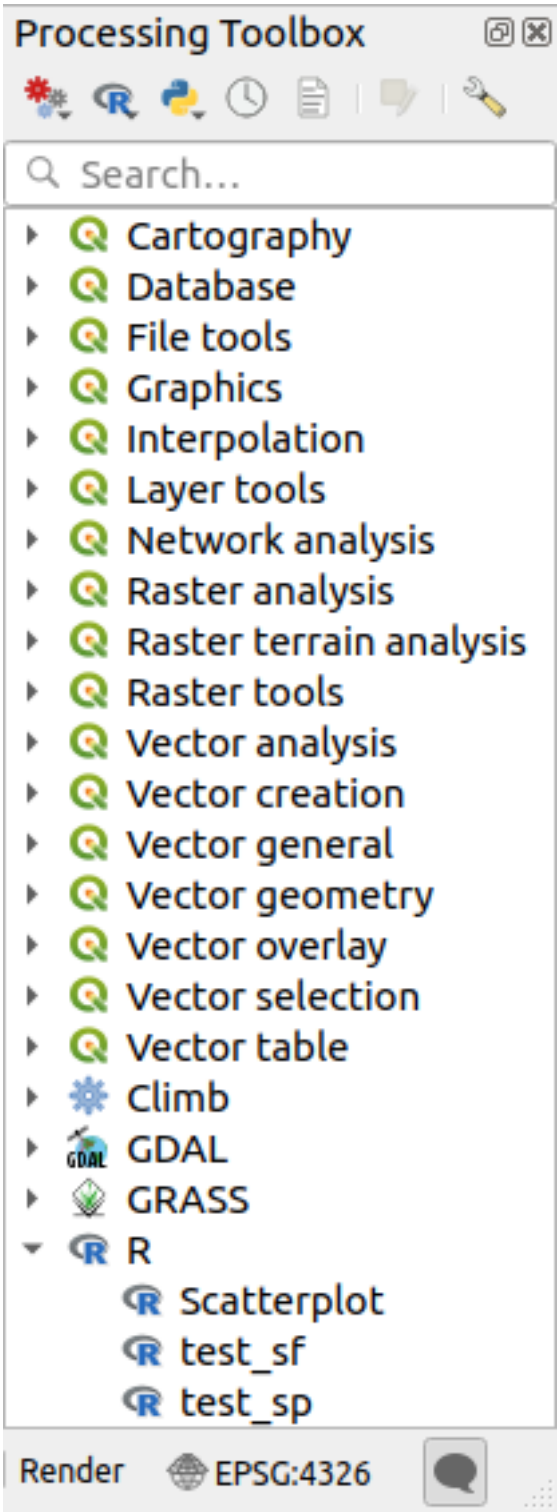
Observera

På **Windows** finns den körbara R-filen normalt i en mapp (`R-<version>`) under `C:\Program Files\R\`. Ange mappen och **INTE** den binära filen!

På **Linux** behöver du bara se till att mappen R finns i miljövariabeln PATH. Om `R` i ett terminalfönster startar R, är du redo att köra.

När du har installerat **Processing R Provider** plugin, hittar du några exempelskript i *Processing Toolbox*:

- *Scatterplot* kör en R-funktion som producerar ett spridningsdiagram från två numeriska fält i det angivna vektorlagret.
- *test_sf* gör vissa operationer som är beroende av `sf`-paketet och kan användas för att kontrollera om R-paketet `sf` är installerat. Om paketet inte är installerat kommer R att försöka installera det (och alla paket som det är beroende av) åt dig, med hjälp av *Paketarkiv* som anges i *Datakälla* ► *R* i Processing options. Standard är https://cran.r-project.org/. Installationen kan ta lite tid…
- *test_sp* kan användas för att kontrollera om R-paketet `sp` är installerat. Om paketet inte är installerat kommer R att försöka installera det åt dig.



Om du har konfigurerat R korrekt för QGIS bör du kunna köra dessa skript.

Lägga till R-skript från QGIS-samlingen

R-integrationen i QGIS skiljer sig från den i SAGA genom att det inte finns någon fördefinierad uppsättning algoritmer som du kan köra (förutom några exempelskript som medföljer tillägget *Processing R Provider*).

En uppsättning exempel på R-skript finns tillgängliga i QGIS Repository. Utför följande steg för att ladda och aktivera dem med hjälp av tillägget *QGIS Resource Sharing*.

1. Lägg till tillägget *QGIS Resource Sharing* (du kan behöva aktivera *Visa även experimentella tillägg* i Tilläggshanterare *Inställningar*)
2. Öppna den (*Tillägg* ► *Resursdelning* ► *Resursdelning*)
3. Välj fliken *Inställningar*
4. Klicka på *Reload repositories*
5. Välj fliken *Alla*
6. Välj *QGIS R script collection* i listan och klicka på *Install*-knappen
7. Samlingen bör nu visas i fliken *Installed*
8. Stäng tillägget
9. Öppna *Processing Toolbox*, och om allt är OK kommer exempelskripten att finnas under R, i olika grupper (endast några av grupperna är expanderade i skärmdumpen nedan).

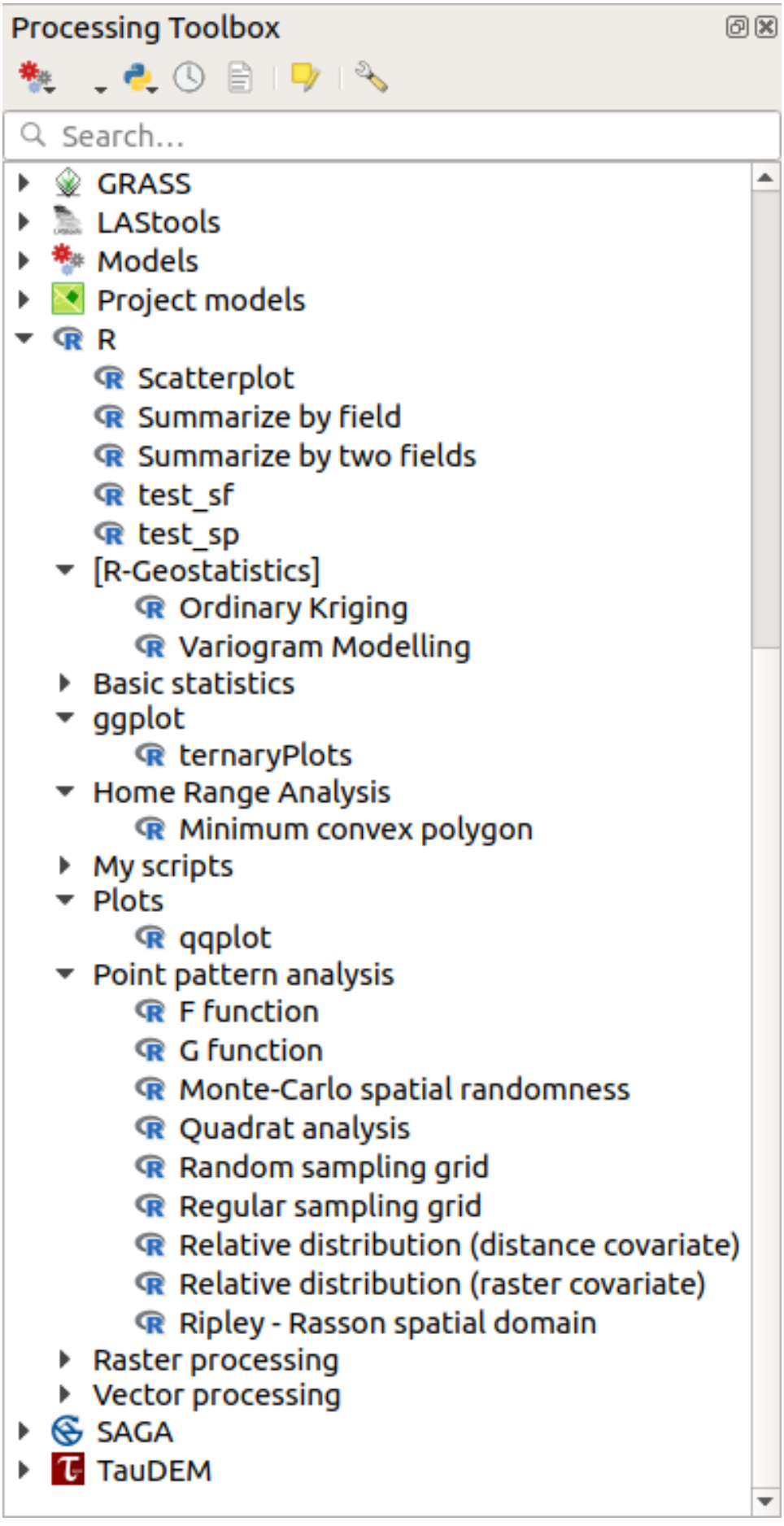
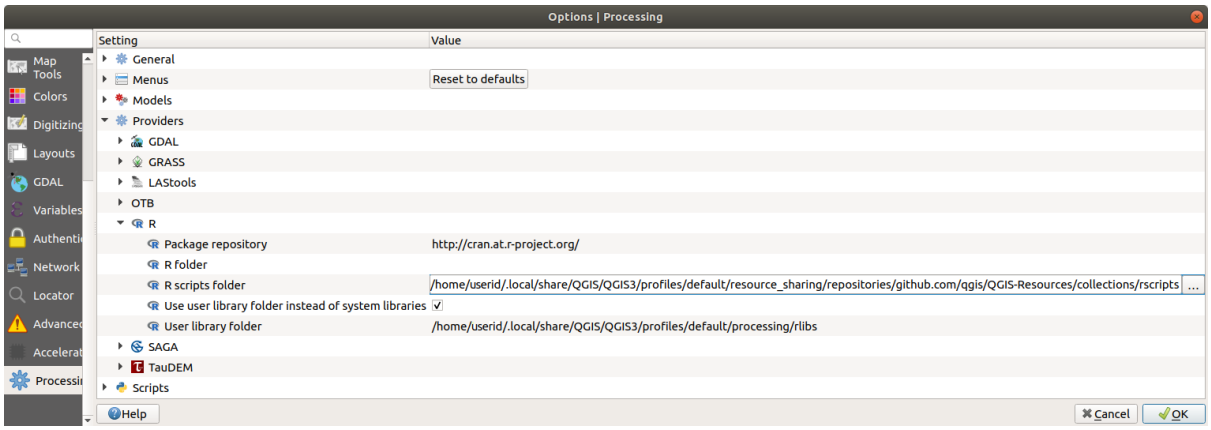


Fig. 23.35: Verktygslådan *Processing Toolbox* med några R-skript som visas

Skripten längst upp är exempelskript från tillägget *Processing R Provider*.

10. Om skripten av någon anledning inte finns tillgängliga i *Processing Toolbox* kan du försöka:
    1. Öppna bearbetningsinställningarna (*Inställningar* ► *Alternativ* ► *Bearbetning* flik)
    2. Gå till *Providers* ► *R* ► *R scripts folder*
        - På Ubuntu ställer du in sökvägen till (eller, ännu bättre, inkluderar i sökvägen):

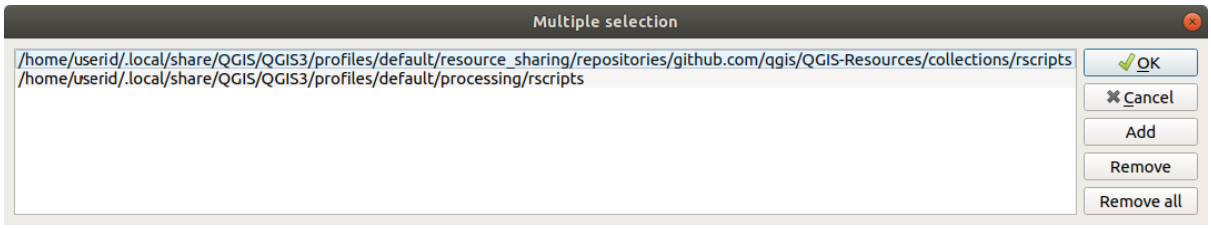
          /home/<user>/.local/share/QGIS/QGIS3/profiles/default/resource_sharing/repositories/github.com/qgis/QGIS-Resources/collections/rscripts



        - I Windows anger du sökvägen till (eller, ännu bättre, inkluderar i sökvägen):

          C:\Users\<user>\AppData\Roaming\QGIS\QGIS3\profiles\default\resource_sharing\repositories\github.com\qgis\QGIS-Resources\collections\rscripts

    Dubbelklicka för att redigera. Du kan sedan välja att bara klistra in/skriva sökvägen, eller så kan du navigera till katalogen med hjälp av …-knappen och trycka på *Add*-knappen i dialogen som öppnas. Det är möjligt att ange flera kataloger här. De kommer att separeras med ett semikolon (";").



Om du vill hämta alla R-skript från QGIS 2 online-samlingen kan du välja *QGIS R script collection (from QGIS 2)* istället för *QGIS R script collection*. Du kommer förmodligen att upptäcka att skript som är beroende av indata eller utdata för vektordata inte kommer att fungera.

Skapa R-skript

Du kan skriva skript och anropa R-kommandon, precis som du skulle göra från R. I det här avsnittet visas syntaxen för att använda R-kommandon i QGIS, och hur du använder QGIS-objekt (lager, tabeller) i dem.

För att lägga till en algoritm som anropar en R-funktion (eller ett mer komplext R-skript som du har utvecklat och som du vill ska vara tillgängligt från QGIS) måste du skapa en skriptfil som utför R-kommandona.

R-skriptfiler har tillägget `.rsx`, och det är ganska enkelt att skapa dem om du bara har grundläggande kunskaper om R-syntax och R-skript. De bör lagras i mappen R scripts. Du kan ange mappen (*R scripts folder*) i inställningsgruppen *R* i dialogrutan Processing settings).

Låt oss titta på en mycket enkel skriptfil, som anropar R-metoden `spsample` för att skapa ett slumpmässigt rutnät inom polygonernas gränser i ett givet polygonlager. Denna metod hör till paketet `maptools`. Eftersom nästan alla algoritmer som du kanske vill införliva i QGIS kommer att använda eller generera spatiala data, är kunskap om spatiala paket som `maptools` och `sp`/`sf` mycket användbar.

```
##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)
```

De första raderna, som börjar med ett dubbelt Python-kommentartecken (`##`), definierar skriptets visningsnamn och grupp och berättar för QGIS om dess in- och utdata.

Observera

Om du vill veta mer om hur du skriver dina egna R-skript kan du läsa avsnittet R Intro i utbildningsmanualen och avsnittet *QGIS R Syntax*.

När du deklarerar en inmatningsparameter använder QGIS den informationen för två saker: att skapa användargränssnittet för att be användaren om värdet på parametern och att skapa en motsvarande R-variabel som kan användas som inmatning för R-funktionen.

I exemplet ovan har vi deklarerat en indata av typen `vektor` med namnet `Vector_layer`. När algoritmen körs kommer QGIS att öppna det lager som användaren valt och lagra det i en variabel med namnet `Vector_layer`. Namnet på en parameter är alltså namnet på den variabel som du använder i R för att komma åt värdet på den parametern (du bör därför undvika att använda reserverade R-ord som parameternamn).

Spatiala parametrar som vektor- och rasterlager läses med hjälp av kommandona `st_read()` (eller `readOGR`) och `brick()` (eller `readGDAL`) (du behöver inte tänka på att lägga till dessa kommandon i din beskrivningsfil - QGIS gör det), och de lagras som `sf` (eller `Spatial*DataFrame`) objekt.

Tabellfälten lagras som strängar som innehåller namnet på det valda fältet.

Vektorfiler kan läsas med kommandot `readOGR()` istället för `st_read()` genom att ange `##load_vector_using_rgdal`. Detta kommer att producera ett `Spatial*DataFrame`-objekt istället för ett `sf`-objekt.

Rasterfiler kan läsas med kommandot `readGDAL()` i stället för `brick()` genom att ange `##load_raster_using_rgdal`.

Om du är en avancerad användare och inte vill att QGIS ska skapa objektet för lagret, kan du använda `##pass_filenames` för att ange att du föredrar en sträng med filnamnet. I detta fall är det upp till dig att öppna filen innan du utför någon åtgärd på de data som den innehåller.

Med ovanstående information är det möjligt att förstå de första raderna i R-skriptet (den första raden som inte börjar med ett Python-kommentartecken).

```
library(sp)
spatpoly = as(Vector_layer, "Spatial")
pts=spsample(polyg,numpoints,type="random")
```

Funktionen `spsample` tillhandahålls av biblioteket *sp*, så det första vi gör är att ladda det biblioteket. Variabeln `Vector_layer` innehåller ett `sf`-objekt. Eftersom vi ska använda en funktion (`spsample`) från *sp*-biblioteket måste vi konvertera `sf`-objektet till ett `SpatialPolygonsDataFrame`-objekt med hjälp av funktionen `as`.

Sedan anropar vi funktionen `spsample` med detta objekt och inparametern `numpoints` (som anger antalet punkter som ska genereras).

Eftersom vi har deklarerat en vektorutgång med namnet `Output` måste vi skapa en variabel med namnet `Output` som innehåller ett `sf`-objekt.

Vi gör detta i två steg. Först skapar vi ett `SpatialPolygonsDataFrame`-objekt från resultatet av funktionen med hjälp av funktionen *SpatialPointsDataFrame*, och sedan konverterar vi det objektet till ett `sf`-objekt med hjälp av funktionen `st_as_sf` (i biblioteket *sf*).

Du kan använda vilka namn du vill för dina mellanliggande variabler. Se bara till att variabeln som lagrar ditt slutresultat har det definierade namnet (i det här fallet `Output`) och att den innehåller ett lämpligt värde (ett `sf`-objekt för vektorlagerutdata).

I det här fallet måste resultatet från metoden `spsample` uttryckligen konverteras till ett `sf`-objekt via ett `SpatialPointsDataFrame`-objekt, eftersom det i sig är ett objekt av klassen `ppp`, som inte kan returneras till QGIS.

Om din algoritm genererar rasterlager beror sättet de sparas på om du har använt alternativet `##dontuserasterpackage` eller inte. Om du har använt det sparas lagren med hjälp av metoden `writeGDAL()`. Om inte, kommer metoden `writeRaster()` från paketet `raster` att användas.

Om du har använt alternativet `##pass_filenames` genereras utdata med hjälp av paketet `raster` (med `writeRaster()`).

Om din algoritm inte genererar ett lager, utan ett textresultat i konsolen, måste du ange att du vill att konsolen ska visas när exekveringen är klar. Det gör du genom att inleda de kommandorader som ger de resultat som du vill skriva ut med tecknet > (”större än”). Endast utdata från rader med prefixet > visas. Här är t.ex. beskrivningsfilen för en algoritm som utför ett normalitetstest på ett givet fält (kolumn) av attributen i ett vektorlager:

```
##layer=vector
##field=field layer
##nortest=group
library(nortest)
>lillie.test(layer[[field]])
```

Utdata från den sista raden skrivs ut, men inte utdata från den första (och inte heller utdata från andra kommandorader som QGIS lägger till automatiskt).

Om din algoritm skapar någon form av grafik (med hjälp av metoden `plot()`), lägg till följande rad (`output_plots_to_html` brukade vara `showplots`):

```
##output_plots_to_html
```

Detta gör att QGIS omdirigerar alla grafiska utdata från R till en temporär fil, som öppnas när R-körningen har avslutats.

Både grafik- och konsolresultat kommer att finnas tillgängliga via resultathanteraren för bearbetning.

För mer information, se R-skript i den officiella QGIS-samlingen (du hämtar och installerar dem med hjälp av tillägget *QGIS Resource Sharing*, som förklaras i *Lägga till R-skript från QGIS-samlingen*). De flesta av dem är ganska enkla och kommer att hjälpa dig att förstå hur du skapar dina egna skript.

Observera

Biblioteken `sf`, `rgdal` och `raster` laddas som standard, så du behöver inte lägga till motsvarande kommandon `library()`. Andra bibliotek som du kan behöva måste dock laddas explicit genom att skriva: `library(ggplot2)` (för att ladda biblioteket `ggplot2`). Om paketet inte redan finns installerat på din dator kommer Processing att försöka hämta och installera det. På detta sätt kommer paketet också att bli tillgängligt i R Standalone. **Tänk på** att om paketet måste laddas ner kan det ta lång tid att köra skriptet första gången.

R-bibliotek installerade när sf_test körs

R-skriptet `sp_test` försöker ladda R-paketen `sp` och `raster`.

R-skriptet *sf_test* försöker ladda `sf` och `raster`. Om dessa två paket inte är installerade kan R försöka läsa in och installera dem (och alla bibliotek som de är beroende av).

Följande R-bibliotek hamnar i `~/.local/share/QGIS/QGIS3/profiles/default/processing/rscripts` efter att `sf_test` har körts från Processing Toolbox på Ubuntu med version 2.0 av tillägget *Processing R Provider* och en ny installation av *R* 3.4.4 (endast *apt*-paketet `r-base-core`):

```
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

Konfigurationen av GRASS skiljer sig inte mycket från konfigurationen av SAGA. Först måste sökvägen till GRASS-mappen definieras, men bara om du kör Windows.

Som standard försöker Processing-ramverket konfigurera sin GRASS-koppling så att den använder GRASS-distributionen som levereras tillsammans med QGIS. Detta bör fungera utan problem för de flesta system, men om du upplever problem kan du behöva konfigurera GRASS-kopplingen manuellt. Om du vill använda en annan GRASS-installation kan du också ändra inställningen så att den pekar på den mapp där den andra versionen är installerad. GRASS 7 behövs för att algoritmerna ska fungera korrekt.

Om du kör Linux behöver du bara se till att GRASS är korrekt installerat och att det kan köras utan problem från ett terminalfönster.

GRASS-algoritmer använder en region för beräkningar. Denna region kan definieras manuellt med hjälp av värden liknande de som finns i SAGA-konfigurationen, eller automatiskt, genom att ta den minsta utsträckning som täcker alla inmatningslager som används för att exekvera algoritmen varje gång. Om det är det senare tillvägagångssättet du föredrar, markera bara alternativet *Use min covering region* i GRASS konfigurationsparametrar.

LAStools

För att använda LAStools i QGIS måste du ladda ner och installera LAStools på din dator och installera LAStools plugin (tillgängligt från det officiella arkivet) i QGIS.

På Linux-plattformar behöver du Wine för att kunna köra några av verktygen.

LAStools aktiveras och konfigureras i Processing options (*Inställningar* ► *Alternativ*, *Processing* tab, *Leverantörer* ► *LAStools*), där du kan ange platsen för LAStools (*LAStools folder*) och Wine (*Wine folder*). På Ubuntu är standardmappen för Wine `/usr/bin`.

OTB-applikationer

OTB-applikationer stöds fullt ut inom ramen för QGIS Processing.

OTB (Orfeo ToolBox) är ett bildbehandlingsbibliotek för fjärranalysdata. Det innehåller också applikationer som tillhandahåller funktioner för bildbehandling. Listan över applikationer och deras dokumentation finns i OTB CookBook

Observera

Observera att OTB inte distribueras med QGIS och måste installeras separat. Binära paket för OTB finns på nedladdningssidan <https://www.orfeo-toolbox.org/download>`_.

För att konfigurera QGIS-bearbetning för att hitta OTB-biblioteket:

1. Öppna bearbetningsinställningarna: *Inställningar* ► *Alternativ* ► *Bearbetning*
2. Du kan se `OTB` under *Leverantörer*-menyn:
    1. Utöka *OTB*-posten
    2. Ställ in *OTB-mapp*. Detta är platsen för din OTB-installation.
    3. Ställ in *OTB applikationsmapp*. Detta är platsen för dina OTB-applikationer (`<PATH_TO_OTB_INSTALLATION>/lib/otb/applications`)
    4. Klicka på *OK* för att spara inställningarna och stänga dialogrutan.

Om inställningarna är korrekta kommer OTB-algoritmer att finnas tillgängliga i *Processing Toolbox*.

Dokumentation av OTB-inställningar tillgängliga i QGIS Processing

- **OTB-mapp**: Detta är den katalog där OTB finns tillgängligt.
- **OTB applikationsmapp**: Detta är platsen/platserna för OTB-applikationer.

  Flera sökvägar är tillåtna.
- **Loggningsnivå** (valfritt): Loggarens nivå som ska användas av OTB-applikationer.

  Loggningsnivån styr hur mycket detaljer som skrivs ut under algoritmens exekvering. Möjliga värden för loggningsnivå är `INFO`, `WARNING`, `CRITICAL`, `DEBUG`. Detta värde är `INFO` som standard. Detta är en avancerad användarkonfiguration.
- **Maximalt RAM-minne att använda** (valfritt): som standard använder OTB-program allt tillgängligt RAM-minne i systemet.

  Du kan dock instruera OTB att använda en viss mängd RAM (i MB) med hjälp av detta alternativ. Ett värde på 256 ignoreras av OTB:s bearbetningsleverantör. Detta är en avancerad användarkonfiguration.
- **Geoid-fil** (valfritt): Sökväg till geoidfilen.

  Detta alternativ anger värdet för parametrarna elev.dem.geoid och elev.geoid i OTB-applikationer. Genom att ställa in detta värde globalt kan användare dela det mellan flera bearbetningsalgoritmer. Tomt som standard.
- **SRTM tiles folder** (valfritt): Katalog där SRTM-plattor finns tillgängliga.

  SRTM-data kan lagras lokalt för att undvika nedladdning av filer under bearbetningen. Detta alternativ anger värdet för parametrarna elev.dem.path och elev.dem i OTB-applikationer. Genom att ställa in detta värde globalt kan användare dela det mellan flera bearbetningsalgoritmer. Tomt som standard.

Kompatibilitet och felsökning

Från och med OTB 6.6.1 görs nya versioner av OTB kompatibla med åtminstone den senaste QGIS-versionen som är tillgänglig vid den tidpunkten.

Om du har problem med OTB-applikationer i QGIS Processing, vänligen öppna ett ärende på OTB bug tracker, med etiketten `qgis`.

Ytterligare information om OTB och QGIS finns i OTB Cookbook.

KAPITEL 24

Bearbetningsleverantörer och algoritmer

Bearbetningsalgoritmer och deras parametrar (som presenteras i användargränssnittet) dokumenteras här.

24.1 QGIS algoritmleverantör

QGIS algoritmleverantör implementerar olika analys- och geobearbetningsoperationer med hjälp av mestadels endast QGIS API. Så nästan alla algoritmer från den här leverantören fungerar ”direkt” utan någon ytterligare konfiguration.

Denna leverantör införlivar vissa algoritmer från tillägg och lägger också till sina egna algoritmer.

24.1.1 3D-plattor

Konvertera B3DM till GLTF

Konverterar filer från det äldre formatet `.B3DM` till `.GLTF` eller `.GLB`.

Parametrar

Etikett	Namn	Typ	Beskrivning
Ingång B3DM	`INPUT`	[3D Tile]	Inmatningsfil som ska konverteras.
Utmatningsfil	`OUTPUT`	[3D Tile] Standard: `[Spara till temporär fil]`	Ange den utgående 3D-kakelfilen. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Utmatningsfil	`OUTPUT`	[3D Tile]	Utmatningslager i `.GLTF`-format.

Python-kod

Algoritm-ID: `original:b3dmtogltf`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Konvertera GLTF till vektorfunktioner

Konverterar filinnehållet i `.GLTF` eller `.GLB` till standardformat för vektorlager.

Parametrar

Etikett	Namn	Typ	Beskrivning
Input GLTF	`INPUT`	[3D Tile]	Inmatningsfil som ska konverteras.
Polygoner för utdata Valfritt	`OUTPUT_POLYGON`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange utdatavektorlagret för att lagra polygonfunktioner. *Ett av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Utmatningslinjer Valfritt	`OUTPUT_LINES`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange utdatavektorlagret för att lagra linjefunktioner. *Ett av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Polygoner för utdata	`OUTPUT_POLYGON`	[vektor: polygon]	Ett vektorlager som lagrar polygonfunktionerna i inmatningslagret.
Utmatningslinjer	`OUTPUT_LINES`	[vektor: linje]	Ett vektorlager som lagrar linjefunktionerna i inmatningslagret.

Python-kod

Algoritm-ID: `native:gltftovector`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.2 Kartografi

Rikta in punkter mot funktioner

Beräknar den rotation som krävs för att rikta in punktelement med deras närmaste element från ett annat referenslager. Ett nytt fält läggs till i utdatalagret som fylls med vinkeln (i grader, medurs) till närmaste referensobjekt.

Alternativt kan utdataskiktets symbologi ställas in så att det beräknade rotationsfältet automatiskt används för att rotera markörsymboler. Om så önskas kan ett maximalt avstånd som ska användas när punkter riktas in ställas in för att undvika att isolerade punkter riktas in mot avlägsna objekt.

Råd

Denna algoritm är utformad för användningsfall som att rikta in byggnaders punktsymboler för att följa närmaste vägriktning.

Tillåter *features in-place modification* av punktfeatures

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: punkt]	Punktfunktioner för att beräkna rotationen för
Reference layer	`REFERENCE_LAYE`	[vektor: geometri]	Lager för att hitta den närmaste funktionen för rotationsberäkning
Maximalt avstånd att beakta Valfritt	`MAX_DISTANCE`	[numerisk: dubbel] Standard: Ej inställd	Om ingen referensfunktion hittas inom detta avstånd tilldelas ingen rotation till punktfunktionen.
Namn på vinkelfält	`FIELD_NAME`	[sträng] Standard: ”rotation	Fält där rotationsvärdet ska lagras.
Automatiskt tillämpa symbologi	`APPLY_SYMBOLOG`	[boolean] Standard: Sann	Roterar symbolmarkören för funktionerna med hjälp av fältvärdet för vinkel
Justerat lager	`OUTPUT`	[vektor: punkt] Standard: `[Spara till temporär fil]`	Ange det roterade utdatavektorlagret. *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Justerat lager	`OUTPUT`	[vektor: punkt]	Punktlagret med ett rotationsfält som tillägg. Om det laddas till QGIS tillämpas som standard symbologin för inmatningslagret, med en datadefinierad rotation av dess markörsymbol.

Python-kod

Algoritm-id: `native:angletonearest`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Kombinera stildatabaser

Kombinerar flera QGIS-stildatabaser till en enda stildatabas. Om objekt av samma typ med samma namn finns i olika källdatabaser kommer dessa att döpas om så att de får unika namn i den kombinerade utdatadatabasen.

Se även

Skapa stildatabas från projekt

Parametrar

Etikett	Namn	Typ	Beskrivning
Ingångsdatabaser	`INPUT`	[fil] [lista]	Filer som innehåller QGIS-stilobjekt
Objekt att kombinera	`OBJECTS`	[uppräkning] [lista]	Typer av stilobjekt i indatabaserna som du vill lägga till i den nya databasen. Dessa kan vara: • 0 — *Symboler* • 1 — *Färgramper* • 2 — *Textformat* • 3 — *Labelinställningar*
Databas för utmatningsstil	`OUTPUT`	&Arkiv Standard: `[Spara till temporär fil]`	Utdata `.XML`-fil som kombinerar de valda stilobjekten. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Antal färgramper	`COLORRAMPS`	[numerisk: heltal]	
Etikettinställningar räknas	`LABELSETTINGS`	[numerisk: heltal]	
Databas för utmatningsstil	`OUTPUT`	&Arkiv	Utdata `.XML`-fil som kombinerar de valda stilelementen
Symbolantal	`SYMBOLS`	[numerisk: heltal]	
Textformat räkning	`TEXTFORMATS`	[numerisk: heltal]	

Python-kod

Algoritm-ID: `native:kombinationer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa kategoriserad renderare från stilar

Ställer in ett vektorlagers renderare till en kategoriserad renderare som använder matchande symboler från en stildatabas. Om ingen stilfil anges används istället symboler från användarens aktuella *symbolbibliotek*.

Ett angivet uttryck eller fält används för att skapa kategorier för renderingsverktyget. Varje kategori matchas individuellt med de symboler som finns i den angivna QGIS XML-stildatabasen. När ett matchande symbolnamn hittas kommer kategorins symbol att ställas in på den matchande symbolen.

Om så önskas kan utdata också vara tabeller som innehåller listor över de kategorier som inte kunde matchas med symboler och symboler som inte kunde matchas med kategorier.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Vektorlager för att tillämpa en kategoriserad stil på
Kategorisera med hjälp av uttryck	`FIELD`	[uttryck]	Fält eller uttryck för att kategorisera funktionerna
Stildatabas (lämna tomt om du vill använda sparade symboler)	`STYLE`	&Arkiv	Fil (`.XML`) som innehåller de symboler som ska tillämpas på kategorierna i inmatningslagret. Filen kan hämtas från verktyget Style Manager *Share symbols*. Om ingen fil anges används QGIS lokala symbolbibliotek.
Använd skiftlägeskänslig matchning av symbolnamn	`CASE_SENSITIVE`	[boolean] Standard: Falsk	Om True (markerad) tillämpas en skiftlägeskänslig jämförelse mellan kategorierna och symbolernas namn
Ignorera icke-alfanumeriska tecken vid matchning	`TOLERANT`	[boolean] Standard: Falsk	Om True (markerad) ignoreras icke-alfanumeriska tecken i namnen på kategorierna och symbolerna, vilket ger större tolerans under matchningen.
Kategorier som inte stämmer överens Valfritt	`NON_MATCHING_C`	[vektor: tabell] Standard: `[Hoppa över utdata]`	Utmatningstabell för kategorier som inte matchar någon symbol i databasen. *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Symbolnamn som inte stämmer överens Valfritt	`NON_MATCHING_S`	[vektor: tabell] Standard: `[Hoppa över utdata]`	Utmatningstabell för symboler från den angivna stildatabasen som inte matchar någon kategori. *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Kategorier som inte stämmer överens	`NON_MATCHING_C`	[vektor: tabell]	Listar kategorier som inte kunde matchas mot någon symbol i den angivna stildatabasen
Symbolnamn som inte stämmer överens	`NON_MATCHING_S`	[vektor: tabell]	Listar symboler från den tillhandahållna stildatabasen som inte kunde matcha någon kategori
Kategoriserat lager	`OUTPUT`	[samma som inmatning]	Det ingående vektorlagret med den kategoriserade stilen tillämpad. Inget nytt lager matas ut.

Python-kod

Algoritm-id: `native:categorizeusingstyle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa stildatabas från projekt

Extraherar alla stilobjekt (symboler, färgramper, textformat och etikettinställningar) från ett QGIS-projekt.

De extraherade symbolerna sparas i en QGIS-stildatabas (`XML`-format), som kan hanteras och importeras via dialogrutan *Style Manager*.

Se även

Kombinera stildatabaser

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsprojekt (lämna tomt för att använda nuvarande) Valfritt	`INPUT`	&Arkiv	En QGIS-projektfil för att extrahera stilobjekten från
Objekt att extrahera	`OBJECTS`	[uppräkning] [lista]	Typer av stilobjekt i inmatningsprojektet som du vill lägga in i den nya databasen. Dessa kan vara: • 0 — *Symboler* • 1 — *Färgramper* • 2 — *Textformat* • 3 — *Labelinställningar*
Databas för utmatningsstil	`OUTPUT`	&Arkiv Standard: `[Spara till temporär fil]`	Ange utdata `.XML`-filen för de valda stilobjekten. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Antal färgramper	`COLORRAMPS`	[numerisk: heltal]	Antal färgramper
Etikettinställningar räknas	`LABELSETTINGS`	[numerisk: heltal]	Antal etikettinställningar
Databas för utmatningsstil	`OUTPUT`	&Arkiv	Utdata `.XML`-fil för de valda stilobjekten
Symbolantal	`SYMBOLS`	[numerisk: heltal]	Antal symboler
Textformat räkning	`TEXTFORMATS`	[numerisk: heltal]	Antal textformat

Python-kod

Algoritm ID: `native:stylefromproject`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exportera atlaslayout som bild

Exporterar atlasen för en utskriftslayout som bildfiler (t.ex. PNG- eller JPEG-bilder).

Om ett täckningslager har angetts skrivs den valda layoutens atlasinställningar som visas i denna algoritm över. I detta fall kommer ett tomt filter eller ett uttryck för sortering efter att stänga av dessa inställningar.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Atlas layout	`LAYOUT`	[layout]	Layout för export
Täckningslager Valfritt	`COVERAGE_LAYER`	[vektor: geometri]	Lager som ska användas för att generera atlasen
Filteruttryck	`FILTER_EXPRESS`	[uttryck]	Uttryck att använda för att filtrera bort atlasfunktioner
Sortera uttryck Valfritt	`SORTBY_EXPRESS`	[uttryck]	Uttryck att använda för att sortera atlasfunktionerna
Omvänd sorteringsordning	`SORTBY_REVERSE`	[boolean] Standard: Falsk	Avgör om sorteringen ska vara inverterad. Används när ett sorteringsuttryck anges.
Uttryck för utdatafilnamn	`FILENAME_EXPRE`	[uttryck] Standard: 'output_'||@atlas_feat	Uttryck som används för att generera filnamn
Utmatningsmapp	`FOLDER`	mapp	Destinationsmapp där bilderna kommer att genereras

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Kartlager som ska tilldelas till olåsta kartobjekt Valfritt	`LAYERS`	[uppräkning] [lager]	Lager som ska visas i kartan objekt vars innehåll inte är låst
Bildformat	`EXTENSION`	[uppräkning] Standard: png	Filformat för den eller de genererade utdata. Listan över tillgängliga format varierar beroende på operativsystem och installerade drivrutiner.
DPI Valfritt	`DPI` Standard: Ej inställd	[numerisk: dubbel]	DPI för utdatafil(er). Om detta inte anges kommer värdet i inställningarna för utskriftslayouten att användas.
Generera världsfil	`GEOREFERENCE`	[boolean] Standard: Sann	Avgör om en världsfil ska genereras
Exportera RDF-metadata	`INCLUDE_METADA`	[boolean] Standard: Sann	Bestämmer om RDF-metadata (titel, författare, …) ska genereras
Aktivera antialiasing	`ANTIALIAS`	[boolean] Standard: Sann	Avgör om antialiasing ska vara aktiverat

Utgångar

Etikett	Namn	Typ	Beskrivning
Bildfil	`OUTPUT`	&Arkiv	Bildfiler som genereras av atlasens layout

Python-kod

Algoritm-ID: `native:atlaslayouttoimage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exportera atlaslayout som PDF (flera filer)

Exporterar atlasen för en utskriftslayout till flera PDF-filer.

Om ett täckningslager har angetts skrivs den valda layoutens atlasinställningar som visas i denna algoritm över. I detta fall kommer ett tomt filter eller ett uttryck för sortering efter att stänga av dessa inställningar.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Atlas layout	`LAYOUT`	[layout]	Layout för export
Täckningslager Valfritt	`COVERAGE_LAYER`	[vektor: geometri]	Lager som ska användas för att generera atlasen
Filteruttryck	`FILTER_EXPRESS`	[uttryck]	Uttryck att använda för att filtrera bort atlasfunktioner
Sortera uttryck Valfritt	`SORTBY_EXPRESS`	[uttryck]	Uttryck att använda för att sortera atlasfunktionerna
Omvänd sorteringsordning	`SORTBY_REVERSE`	[boolean] Standard: Falsk	Avgör om sorteringen ska vara inverterad. Används när ett sorteringsuttryck anges.
Filnamn för utdata Valfritt	`OUTPUT_FILENAM`	[uttryck]	Namnmönster för PDF-utdatafilerna.
Utmatningsmapp	`OUTPUT_FOLDER`	mapp	Målmapp för de PDF-filer som ska skrivas ut.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Kartlager som ska tilldelas till olåsta kartobjekt Valfritt	`LAYERS`	[uppräkning] [lager]	Lager som ska visas i kartan objekt vars innehåll inte är låst
DPI Valfritt	`DPI` Standard: Ej inställd	[numerisk: dubbel]	DPI för utdatafil(er). Om detta inte anges kommer värdet i inställningarna för utskriftslayouten att användas.
Alltid exportera som vektorer	`FORCE_VECTOR`	[boolean] Standard: Falsk	Bestämmer om vektoriell data ska lämnas som vektorer
Alltid exportera som raster	`FORCE_RASTER`	[boolean] Standard: Falsk	Tvingar alla objekt i kartan att rasteriseras. Denna parameter har företräde framför parametern `FORCE_VECTOR`.
Lägga till information om georeferens	`GEOREFERENCE`	[boolean] Standard: Sann	Avgör om en världsfil ska genereras
Exportera RDF-metadata	`INCLUDE_METADA`	[boolean] Standard: Sann	Bestämmer om RDF-metadata (titel, författare, …) ska genereras
Avaktivera export av rasterlager med kaklade lager	`DISABLE_TILED`	[boolean] Standard: Falsk	Avgör om rastret ska vara kaklat
Förenkla geometrierna för att minska storleken på utdatafilen	`SIMPLIFY`	[boolean] Standard: Sann	Avgör om geometrier ska förenklas för att minska storleken på utdatafilen
Textexport	`TEXT_FORMAT`	[uppräkning] Standard: 0	Bestämmer om text ska exporteras som sökväg eller textobjekt. *Möjliga alternativ för* är: • 0 - Exportera alltid text som sökvägar (rekommenderas) • 1 - Exportera alltid texter som textobjekt • 2 - Föredrar att exportera texter som textobjekt
Bildkomprimering	`IMAGE_COMPRESS`	[uppräkning] Standard: 0	Bestämmer bildens komprimeringsnivå och hur lämplig filen kan vara för utskrift eller efterbehandling i externa program. Möjliga alternativ är: • 0 - Med förlust (JPEG) • 1 - Förlustfri

Utgångar

Etikett	Namn	Typ	Beskrivning
PDF-fil	`OUTPUT`	&Arkiv	PDF-fil som motsvarar den exporterade atlaslayouten

Python-kod

Algoritm-ID: `native:atlaslayouttomultiplepdf`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exportera atlaslayout som PDF (enstaka fil)

Exporterar atlasen för en utskriftslayout som en enda PDF-fil.

Om ett täckningslager har angetts skrivs den valda layoutens atlasinställningar som visas i denna algoritm över. I detta fall kommer ett tomt filter eller ett uttryck för sortering efter att stänga av dessa inställningar.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Atlas layout	`LAYOUT`	[layout]	Layout för export
Täckningslager Valfritt	`COVERAGE_LAYER`	[vektor: geometri]	Lager som ska användas för att generera atlasen
Filteruttryck	`FILTER_EXPRESS`	[uttryck]	Uttryck att använda för att filtrera bort atlasfunktioner
Sortera uttryck Valfritt	`SORTBY_EXPRESS`	[uttryck]	Uttryck att använda för att sortera atlasfunktionerna
Omvänd sorteringsordning	`SORTBY_REVERSE`	[boolean] Standard: Falsk	Avgör om sorteringen ska vara inverterad. Används när ett sorteringsuttryck anges.
PDF-fil	`OUTPUT`	&Arkiv Standard: [Spara till temporär fil]	Namn (inklusive sökväg) på utdatafilen. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Kartlager som ska tilldelas till olåsta kartobjekt Valfritt	`LAYERS`	[uppräkning] [lager]	Lager som ska visas i kartan objekt vars innehåll inte är låst
DPI Valfritt	`DPI` Standard: Ej inställd	[numerisk: dubbel]	DPI för utdatafil(er). Om detta inte anges kommer värdet i inställningarna för utskriftslayouten att användas.
Alltid exportera som vektorer	`FORCE_VECTOR`	[boolean] Standard: Falsk	Bestämmer om vektoriell data ska lämnas som vektorer
Alltid exportera som raster	`FORCE_RASTER`	[boolean] Standard: Falsk	Tvingar alla objekt i kartan att rasteriseras. Denna parameter har företräde framför parametern `FORCE_VECTOR`.
Lägga till information om georeferens	`GEOREFERENCE`	[boolean] Standard: Sann	Avgör om en världsfil ska genereras
Exportera RDF-metadata	`INCLUDE_METADA`	[boolean] Standard: Sann	Bestämmer om RDF-metadata (titel, författare, …) ska genereras
Avaktivera export av rasterlager med kaklade lager	`DISABLE_TILED`	[boolean] Standard: Falsk	Avgör om rastret ska vara kaklat
Förenkla geometrierna för att minska storleken på utdatafilen	`SIMPLIFY`	[boolean] Standard: Sann	Avgör om geometrier ska förenklas för att minska storleken på utdatafilen
Textexport	`TEXT_FORMAT`	[uppräkning] Standard: 0	Bestämmer om text ska exporteras som sökväg eller textobjekt. *Möjliga alternativ för* är: • 0 - Exportera alltid text som sökvägar (rekommenderas) • 1 - Exportera alltid texter som textobjekt • 2 - Föredrar att exportera texter som textobjekt
Bildkomprimering	`IMAGE_COMPRESS`	[uppräkning] Standard: 0	Bestämmer bildens komprimeringsnivå och hur lämplig filen kan vara för utskrift eller efterbehandling i externa program. Möjliga alternativ är: • 0 - Med förlust (JPEG) • 1 - Förlustfri

Utgångar

Etikett	Namn	Typ	Beskrivning
PDF-fil	`OUTPUT`	&Arkiv	PDF-fil som motsvarar den exporterade atlaslayouten

Python-kod

Algoritm-ID: `native:atlaslayouttopdf`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exportera utskriftslayout som bild

Exporterar en utskriftslayout som en bildfil (t.ex. PNG- eller JPEG-bilder)

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Print layout	`LAYOUT`	[layout]	Layout för export
Bildfil	`OUTPUT`	&Arkiv Standard: [Spara till temporär fil]	Namn (inklusive sökväg) på utdatafilen. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Kartlager som ska tilldelas till olåsta kartobjekt Valfritt	`LAYERS`	[uppräkning] [lager]	Lager som ska visas i kartan objekt vars innehåll inte är låst
DPI Valfritt	`DPI` Standard: Ej inställd	[numerisk: dubbel]	DPI för utdatafil(er). Om detta inte anges kommer värdet i inställningarna för utskriftslayouten att användas.
Generera världsfil	`GEOREFERENCE`	[boolean] Standard: Sann	Avgör om en världsfil ska genereras
Exportera RDF-metadata	`INCLUDE_METADA`	[boolean] Standard: Sann	Bestämmer om RDF-metadata (titel, författare, …) ska genereras
Aktivera antialiasing	`ANTIALIAS`	[boolean] Standard: Sann	Avgör om antialiasing ska vara aktiverat

Utgångar

Etikett	Namn	Typ	Beskrivning
Bildfil	`OUTPUT`	&Arkiv	Bildfil som motsvarar den exporterade utskriftslayouten

Python-kod

Algoritm-ID: `native:utskriftlayouttillbild`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exportera utskriftslayouten som PDF

Exporterar en utskriftslayout som en PDF-fil.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Print Layout	`LAYOUT`	[layout]	Layout för export
PDF-fil	`OUTPUT`	&Arkiv Standard: [Spara till temporär fil]	Namn (inklusive sökväg) på utdatafilen. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Kartlager som ska tilldelas till olåsta kartobjekt Valfritt	LAYERS	[uppräkning] [lager]	Lager som ska visas i kartan objekt vars innehåll inte är låst
DPI Valfritt	DPI Standard: Ej inställd	[numerisk: dubbel]	DPI för utdatafil(er). Om detta inte anges kommer värdet i inställningarna för utskriftslayouten att användas.
Alltid exportera som vektorer	FORCE_VECTOR	[boolean] Standard: Falsk	Bestämmer om vektoriell data ska lämnas som vektorer
Alltid exportera som raster	FORCE_RASTER	[boolean] Standard: Falsk	Tvingar alla objekt i kartan att rasteriseras. Denna parameter har företräde framför parametern FORCE_VECTOR.
Lägga till information om georeferens	GEOREFERENCE	[boolean] Standard: Sann	Avgör om en världsfil ska genereras
Exportera RDF-metadata	INCLUDE_METADA	[boolean] Standard: Sann	Bestämmer om RDF-metadata (titel, författare, …) ska genereras
Avaktivera export av rasterlager med kaklade lager	DISABLE_TILED	[boolean] Standard: Falsk	Avgör om rastret ska vara kaklat
Förenkla geometrierna för att minska storleken på utdatafilen	SIMPLIFY	[boolean] Standard: Sann	Avgör om geometrier ska förenklas för att minska storleken på utdatafilen
Textexport	TEXT_FORMAT	[uppräkning] Standard: 0	Bestämmer om text ska exporteras som sökväg eller textobjekt. *Möjliga alternativ för* är: • 0 - Exportera alltid text som sökvägar (rekommenderas) • 1 - Exportera alltid texter som textobjekt • 2 - Föredrar att exportera texter som textobjekt
Bildkomprimering	IMAGE_COMPRESS	[uppräkning] Standard: 0	Bestämmer bildens komprimeringsnivå och hur lämplig filen kan vara för utskrift eller efterbehandling i externa program. Möjliga alternativ är: • 0 - Med förlust (JPEG) • 1 - Förlustfri
Exportera lager som separata PDF-filer	SEPARATE_LAYER	[boolean] Standard: Falsk	Om True anges skapas en separat PDF-fil per lager och per kartobjekt i layouten. Dessutom kan separata PDF-filer skapas för andra komplexa layoutobjekt, vilket resulterar i en uppsättning PDF-filer som innehåller logiska atomkomponenter i layouten.

Utgångar

Etikett	Namn	Typ	Beskrivning
PDF-fil	`OUTPUT`	&Arkiv	PDF-fil(er) som motsvarar den exporterade utskriftslayouten

Python-kod

Algoritm-ID: `` `native:printlayouttopdf` ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Extrahera etiketter

Extraherar etikettinformation från en renderad karta i en given omfattning och skala.

Om ett karttema anges kommer den renderade kartan att matcha synligheten och symbologin för det temat. Om det lämnas tomt kommer alla synliga lager från projektet att användas. Extraherad etikettinformation inkluderar: position (serveras som punktgeometrier), det associerade lagernamnet och funktions-ID, etikettext, rotation (i grader, medurs), multilinjejustering och teckensnittsdetaljer.

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Karta omfattning	`EXTENT`	[omfattning]	Kartans utbredning för att extrahera etiketterna från Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Kartskala	`SCALE`	[skala]	Extraherade etiketter kommer att renderas med hjälp av deras egenskaper som ställts in i denna skala.
Karta tema Valfritt	`MAP_THEME`	[maptheme]	Ett karttema som visar de lager som etiketterna ska extraheras från. Om inställningen inte ändras extraheras etiketterna för de lager som är synliga för tillfället.
Inkludera ej placerade etiketter	`INCLUDE_UNPLAC`	[boolean] Standard: Sann	Ange om alla överlappande etiketter ska extraheras, även de som strider mot varandra (och därmed inte är placerade).
Extraherade etiketter	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange utdatavektorlagret för omfattningen/omfattningarna. *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Kartans upplösning (i DPI) Valfritt	`DPI` Standard: 96,0	[numerisk: dubbel]	

Utgångar

Etikett	Namn	Typ	Beskrivning
Extraherade etiketter	`OUTPUT`	[vektor: punkt]	Punktvektorlager som representerar de hämtade etiketterna. Varje objekt har attribut som identifierar källan (lager, objekt-ID) och de tilldelade märkningsegenskaperna (text, teckensnitt, storlek, rotation, …). En standardstil med märkning och nollsymbol tillämpas också på lagret. **Varning** Eftersom vissa av de genererade fälten har namn med mer än 10 tecken kan användning av ESRI shapefile-format (`.SHP`) för att lagra utdata leda till oväntad rendering när lagret laddas i QGIS.

Python-kod

Algoritm-ID: `native:extrahera etiketter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skriv ut layoutkartans utsträckning till lager

Skapar ett polygonlager som innehåller omfattningen av ett (eller flera) kartobjekt för utskriftslayouten, med attribut som anger kartans storlek (i layoutenheter, dvs. *reference map*-enheterna), skala och rotation.

Om parametern map item anges kommer endast den matchande kartutbredningen att exporteras. Om den inte anges exporteras alla kartutsnitt från layouten.

Alternativt kan en specifik CRS för utdata anges. Om det inte anges kommer CRS för det ursprungliga kartobjektet att användas.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Print layout	`LAYOUT`	[uppräkning]	En utskriftslayout i det aktuella projektet
Karta objekt Valfritt	`MAP`	[uppräkning] Standard: *Alla objekt på kartan*	Det eller de kartobjekt vars information du vill extrahera. Om ingen anges bearbetas alla kartobjekt.
Extent	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange utdatavektorlagret för omfattningen/omfattningarna. *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Överskrider CRS Valfritt	`CRS`	[crs] Standard: *Layouten CRS*	Välj CRS för det lager i vilket informationen ska rapporteras.

Utgångar

Etikett	Namn	Typ	Beskrivning
Kartans höjd	`HEIGHT`	[numerisk: dubbel]	
Extent	`OUTPUT`	[vektor: polygon]	Utgående polygonvektorskikt som innehåller utbredning av alla inmatade layoutkartobjekt
Rotation av karta	`ROTATION`	[numerisk: dubbel]	
Kartskala	`SCALE`	[numerisk: dubbel]	
Kartbredd	`WIDTH`	[numerisk: dubbel]	

Python-kod

Algoritm-ID: `native:printlayoutmapextenttolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Ange stil för lager

Tillämpar en angiven stil på ett lager. Stilen måste vara definierad i en QML-fil.

Inga nya utdata skapas: stilen tilldelas omedelbart till lagret.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	INPUT	[lager]	Inmatningslager som du vill använda stilen på
Style-fil	STYLE	&Arkiv	Sökväg till filen `.qml` för stilen

Utgångar

Etikett	Namn	Typ	Beskrivning
	OUTPUT	[samma som inmatning]	Ingångslagret med den nya stilen tilldelas. Inget nytt lager skapas.

Python-kod

Algoritm-ID: `native:setlayerstyle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Topologisk färgläggning

Tilldelar ett färgindex till polygonfunktioner på ett sådant sätt att inga intilliggande polygoner delar samma färgindex, samtidigt som antalet färger som krävs minimeras.

Algoritmen gör det möjligt att välja vilken metod som ska användas för att tilldela färger.

Ett minsta antal färger kan anges om så önskas. Färgindexet sparas i ett nytt attribut med namnet **color_id**.

I följande exempel visas algoritmen med fyra olika färger valda; som du kan se har varje färgklass samma antal funktioner.

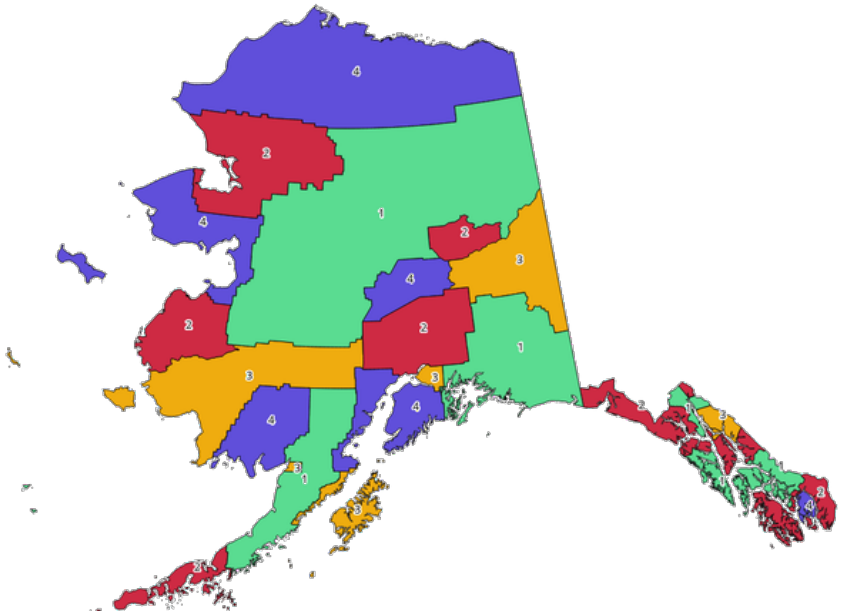


Fig. 24.1: Exempel på topologiska färger

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: polygon]	Det ingående polygonlagret
Minsta antal färger	`MIN_COLORS`	[numerisk: heltal] Standard: 4	Det minsta antalet färger som ska tilldelas. Minst 1, högst 1000.
Minsta avstånd mellan funktioner	`MIN_DISTANCE`	[numerisk: dubbel] Standard: 0,0	Förhindra att funktioner i närheten (men som inte rör vid varandra) tilldelas samma färg. Minimum 0,0.
Balansera färgtilldelningen	`BALANCE`	[uppräkning] Standard: 0	Menyalternativ: *Lager ► Lägg till ett rasterlager* eller *Inställningar ► Verktygsrader ► Digitalisering* • 0 — Enligt antal funktioner Försök att tilldela färger så att antalet funktioner som tilldelas varje enskilt färgindex är balanserat. • 1 — Per tilldelat område Tilldelar färger så att den totala ytan av objekt som tilldelats varje färg är balanserad. Det här läget kan vara användbart för att undvika att stora objekt resulterar i att en av färgerna blir mer dominerande på en färgkarta. • 2 — Avståndet mellan färgerna Tilldelar färger i syfte att maximera avståndet mellan objekt med samma färg. Detta läge bidrar till att skapa en mer enhetlig fördelning av färger över en karta.

continues on next page

Table 24.4 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Färgad	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Färgad	`OUTPUT`	[vektor: polygon]	Polygonvektorlager med en tillagd kolumn `color_id`

Python-kod

Algoritm-ID: `qgis:topologisk kolorering`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Överför anteckningar från huvudlagret

Överför alla *annotationer* från huvudannoteringslagret i ett projekt till ett nytt annoteringslager. Objektens placering kan sedan justeras inom lagerstapeln.

Parametrar

Etikett	Namn	Typ	Beskrivning
Nytt namn på lagret	`LAYER_NAME`	[sträng] Standard: ”Anteckningar	Namn på det annoteringslager som ska skapas

Utgångar

Etikett	Namn	Typ	Beskrivning
Nytt namn på lagret	`OUTPUT`	[lager]	Ett lager med objekt från det huvudsakliga annoteringslagret

Python-kod

Algoritm-ID: `` `native:överförannotationerfrånhuvudet ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.3 Databas

Exportera till PostgreSQL

Exporterar ett vektorlager till en PostgreSQL-databas och skapar en ny relation. Om det finns en relation med samma namn kan den tas bort innan den nya relationen skapas. Innan detta måste en anslutning mellan QGIS och PostgreSQL-databasen skapas (se t.ex. *Skapa en lagrad anslutning*).

Parametrar

Etikett	Namn	Typ	Beskrivning
Lager att importera	`INPUT`	[vektor: vilken som helst]	Vektorlager att lägga till i databasen
Databas (anslutningsnamn)	`DATABASE`	[sträng]	Namnet på databasanslutningen (inte databasnamnet). Befintliga anslutningar visas i kombinationsrutan.
Schema (schemats namn) Valfritt	`SCHEMA`	[sträng] Standard: ”offentlig	Namn på det schema där data ska lagras. Det kan vara ett nytt schema eller ett som redan finns.
Tabell att importera till (lämna tomt för att använda lagernamn) Valfritt	`TABLENAME`	[sträng] Standard: ”	Definierar ett tabellnamn för den importerade vektorfilen. Om inget läggs till kommer lagernamnet att användas.
Primärt nyckelfält Valfritt	`PRIMARY_KEY`	[tabellfält: alla]	Ställer in fältet för primärnyckel från ett befintligt fält i vektorlagret. En kolumn med **unika** värden kan användas som primärnyckel för databasen.

continues on next page

Table 24.6 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Geometri kolumn	`GEOMETRY_COLUM`	[sträng] Standard: 'geom'	Definierar namnet på geometrikolumnen i den nya PostGIS-tabellen. Geometriinformation för objekten lagras i denna kolumn.
Kodning Valfritt	`ENCODING`	[sträng] Standard: "UTF-8	Definierar kodningen av utmatningslagret
Skriv över	`OVERWRITE`	[boolean] Standard: Sann	Om den angivna tabellen existerar kommer detta alternativ att raderas och en ny tabell skapas innan funktionerna läggs till om det här alternativet anges till "True". Om detta alternativ är `False` och tabellen existerar, kommer algoritmen att göra ett undantag ("relationen existerar redan").
Skapa spatialt index	`CREATEINDEX`	[boolean] Standard: Sann	Anger om ett spatialt index ska skapas eller inte
Konvertera fältnamn till gemener	`LOWERCASE_NAME`	[boolean] Standard: Sann	Konverterar fältnamnen i inmatningsvektorlagret till gemener
Ta bort längdbegränsningen för teckenfält	`DROP_STRING_LE`	[boolean] Standard: Falsk	Bör längdbegränsningar för teckenfält tas bort eller inte
Skapa endelade geometrier istället för flerdelade	`FORCE_SINGLEPA`	[boolean] Standard: Falsk	Ska funktionerna i utdatalagret vara endelade istället för flerdelade. Som standard bevaras den befintliga geometriinformationen.

Utgångar

Algoritmen har ingen utdata.

Python-kod

Algoritm ID: `qgis:importintopostgis`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exportera till SpatiaLite

Exporterar ett vektorlager till en SpatiaLite-databas. Innan detta kan göras måste en anslutning mellan QGIS och SpatiaLite-databasen skapas (se t.ex. *SpatiaLite-lager*).

Parametrar

Etikett	Namn	Typ	Beskrivning
Lager att importera	`INPUT`	[vektor: vilken som helst]	Vektorlager att lägga till i databasen
Fildatabas	`DATABASE`	[vektor: vilken som helst]	SQLite/SpatiaLite-databasfilen att ansluta till
Tabell att importera till (lämna tomt för att använda lagernamn) Valfritt	`TABLENAME`	[sträng] Standard: ”	Definierar tabellnamnet för den importerade vektorfilen. Om inget anges kommer lagernamnet att användas.
Primärt nyckelfält Valfritt	`PRIMARY_KEY`	[tabellfält: alla]	Använd ett fält i inputvektorlagret som primär nyckel
Geometri kolumn	`GEOMETRY_COLUM`	[sträng] Standard: ’geom’	Definierar namnet på geometrikolumnen i den nya SpatiaLite-tabellen. Geometriinformation för objekten lagras i denna kolumn.
Kodning Valfritt	`ENCODING`	[sträng] Standard: ”UTF-8	Definierar kodningen av utmatningslagret
Skriv över	`OVERWRITE`	[boolean] Standard: Sann	Om den angivna tabellen finns, kommer detta alternativ att raderas om det sätts till ”True” och en ny tabell kommer att skapas innan lagrets funktioner läggs till. Om detta alternativ är `False` och tabellen finns, kommer algoritmen att göra ett undantag (”tabellen finns redan”).
Skapa spatialt index	`CREATEINDEX`	[boolean] Standard: Sann	Anger om ett spatialt index ska skapas eller inte
Konvertera fältnamn till gemener	`LOWERCASE_NAME`	[boolean] Standard: Sann	Konvertera fältnamnen i inputvektorlagret till gemener
Ta bort längdbegränsningen för teckenfält	`DROP_STRING_LE`	[boolean] Standard: Falsk	Bör längdbegränsningar för teckenfält tas bort eller inte
Skapa endelade geometrier istället för flerdelade	`FORCE_SINGLEPA`	[boolean] Standard: Falsk	Ska funktionerna i utdatalagret vara endelade istället för flerdelade. Som standard bevaras den befintliga geometriinformationen.

Utgångar

Algoritmen har ingen utdata.

Python-kod

Algoritm-ID: `qgis:importintospatialite`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Paketlager

Lägger till lager till ett GeoPackage.

Om GeoPackage finns och `Overwrite existing GeoPackage` är markerat kommer det att skrivas över (tas bort och återskapas). Om GeoPackage finns och `Overwrite existing GeoPackage` inte är markerat, kommer lagret att läggas till.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`LAYERS`	[vektor: valfri] [lista]	De (vektor-)lager som ska importeras till GeoPackage. Rasterlager stöds inte. Om ett rasterlager läggs till kommer ett `QgsProcessingException` att kastas.
Overwrite existing GeoPackage (Skriv över befintligt GeoPackage)	`OVERWRITE`	[boolean] Standard: Falsk	Om det angivna GeoPackage finns kommer detta alternativ att raderas om det är inställt på ”True” och ett nytt skapas innan lagren läggs till. Om alternativet är inställt på `False` kommer lagren att läggas till.
Spara lagerformat i GeoPackage	`SAVE_STYLES`	[boolean] Standard: Sann	Spara lagerstilarna
Spara metadata för lager i GeoPackage	`SPARA_METADATA`	[boolean] Standard: Sann	Om markerat kopieras källagrets metadata till GeoPackage så att det blir standardmetadata för lagret.
Spara endast utvalda funktioner	`SELECTED_FEATU`	[boolean] Standard: Falsk	Om ett lager har ett urval kommer endast valda funktioner att sparas om du ställer in detta alternativ på `True`. För lager utan urval sparas alla funktioner.
Exportrelaterade lager enligt de relationer som definieras i projektet	`EXPORT_RELATED`	[boolean] Standard: Falsk	Om ett inmatningslager har *relations* inställt i projektet, kommer detta alternativ att exportera även dess relaterade lager om det är inställt på `True`. Om lagret har valda funktioner exporteras endast deras relaterade funktioner, såvida inte det relaterade lagret också var ett inmatningslager.
Destination GeoPackage	`OUTPUT`	&Arkiv Standard: `[Spara till temporär fil]`	Ange var GeoPackage-filen ska lagras. En av • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Lager inom nytt paket	`OUTPUT_LAYERS`	[sträng] [lista]	Listan över lager som lagts till i GeoPackage.

Python-kod

Algoritm-ID: `native:paket`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

PostgreSQL kör och ladda SQL

Tillåter att en SQL-databasfråga utförs på en PostgreSQL-databas som är ansluten till QGIS och laddar resultatet. Algoritmen **skapar** inte ett nytt lager: den är utformad för att köra frågor på själva lagret.

Exempel

1. Ställ in alla värden i ett befintligt fält till ett fast värde. SQL-frågesträngen kommer att vara:

   ```
   UPDATE your_table SET field_to_update=20;
   ```

   I exemplet ovan kommer värdena för fältet `field_to_update` i tabellen `your_table` att sättas till `20`.

2. Skapa en ny kolumn `area` och beräkna arean för varje objekt med PostGIS-funktionen `ST_AREA`.

   ```
   -- 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);
   ```

Se även

PostgreSQL kör SQL, *Exekvera SQL*, *SpatiaLite kör SQL*

Parametrar

Etikett	Namn	Typ	Beskrivning
Databas (anslutningsnamn)	`DATABASE`	[sträng]	Databasanslutningen (inte databasnamnet). Befintliga anslutningar visas i kombinationsrutan.
SQL-förfrågan	`SQL`	[sträng]	Definierar SQL-frågan, t.ex. `'UPDATE my_table SET field=10'`.
Unikt namn på ID-fält	`ID_FIELD`	[sträng] Standard: id	Anger fältet för primärnyckel (en kolumn i resultattabellen)
Namn på geometriskt fält Valfritt	`GEOMETRY_FIELD`	[sträng] Standard: 'geom'	Namn på geometrikolumnen (en kolumn i resultattabellen)

Utgångar

Etikett	Namn	Typ	Beskrivning
SQL-lager	`OUTPUT`	[vektor: vilken som helst]	Det resulterande vektorlagret som ska laddas in i QGIS.

Python-kod

Algoritm ID: `qgis:postgisexecuteandloadsql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

PostgreSQL kör SQL

Gör det möjligt att utföra en SQL-databasfråga på en PostgreSQL-databas som är ansluten till QGIS. Algoritmen **skapar** inte ett nytt lager: den är utformad för att köra frågor på själva lagret.

Exempel

1. Ställ in alla värden i ett befintligt fält till ett fast värde. SQL-frågesträngen kommer att vara:

   ```
   UPDATE your_table SET field_to_update=20;
   ```

   I exemplet ovan kommer värdena för fältet `field_to_update` i tabellen `your_table` att sättas till `20`.

2. Skapa en ny kolumn `area` och beräkna arean för varje objekt med PostGIS-funktionen `ST_AREA`.

   ```
   -- 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);
   ```

Se även

PostgreSQL kör och ladda SQL, *Exekvera SQL*, *SpatiaLite kör SQL*

Parametrar

Etikett	Namn	Typ	Beskrivning
Databas (anslutningsnamn)	`DATABASE`	[sträng]	Databasanslutningen (inte databasnamnet). Befintliga anslutningar visas i kombinationsrutan.
SQL-förfrågan	`SQL`	[sträng]	Definierar SQL-frågan, t.ex. `'UPDATE my_table SET field=10'`.

Utgångar

Ingen utdata skapas. SQL-frågan exekveras på plats.

Python-kod

Algoritm ID: `native:postgisexecutesql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

SpatiaLite kör SQL

Gör det möjligt att utföra en SQL-databasfråga på en SpatiaLite-databas. Databasen bestäms av ett inmatningslager eller en fil. Algoritmen **skapar** inte ett nytt lager: den är utformad för att köra frågor på själva lagret.

Se även

PostgreSQL kör SQL, *Exekvera SQL*

För några exempel på SQL-frågor, se *PostGIS SQL Query Examples*.

Parametrar

Etikett	Namn	Typ	Beskrivning
Databaslager (eller fil)	`DATABASE`	&Arkiv	SQLite/SpatiaLite-databasfilen att ansluta till
SQL-förfrågan	`SQL`	[sträng] Standard: ”	Definierar SQL-frågan, t.ex. `'UPDATE my_table SET field=10'`.

Utgångar

Ingen utdata skapas. SQL-frågan exekveras på plats.

Python-kod

Algoritm-ID: `native:spatialiteexecutesql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

SpatiaLite exekverar SQL (registrerad DB)

Gör det möjligt att utföra en SQL-databasfråga på en SpatiaLite-databas som är ansluten till QGIS. Algoritmen **skapar** inte ett nytt lager: den är utformad för att köra frågor på själva lagret.

Se även

PostgreSQL kör SQL, *Exekvera SQL*

För några exempel på SQL-frågor, se *PostGIS SQL Query Examples*.

Parametrar

Etikett	Namn	Typ	Beskrivning
Spara i databas	`DATABASE`	[uppräkning] Standard: ej inställd	Välj en SQLite/SpatiaLite-databas som är ansluten till den aktuella sessionen
SQL-förfrågan	`SQL`	[sträng] Standard: ”	Definierar SQL-frågan, t.ex. `'UPDATE my_table SET field=10'`.

Utgångar

Ingen utdata skapas. SQL-frågan exekveras på plats.

Python-kod

Algoritm-ID: `` `native:spatialiteexecutesqlregistered ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.4 Verktyg för filer

Ladda ner fil via HTTP(S)

Laddar ner en URL till filsystemet med en HTTP(S) GET- eller POST-begäran.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
URL	`URL`	[sträng]	URL-adressen till filen som ska laddas ner.
Filens destination	`OUTPUT`	[sträng] Standard: `[Spara till temporär fil]`	Specifikation av filens destination. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Metod	`METHOD`	[uppräkning] Standard: 0	Den HTTP-metod som ska användas för begäran. Alternativen är: • 0 — GET • 1 — POST
★☆☆ (Grundläggande nivå) Följ med: Sammanfoga data om skogsbestånd Valfritt	`DATA`	[sträng]	De data som ska läggas till i brödtexten om begäran är en POST.

Utgångar

Etikett	Namn	Typ	Beskrivning
Filens destination	`OUTPUT`	[sträng]	Platsen för den nedladdade filen

Python-kod

Algoritm-ID: `qgis:filedownloader`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

HTTP(S) POST/GET-begäran

Utför en HTTP(S) POST/GET-begäran och returnerar HTTP-statuskoden och svarsdata. Om ett fel inträffar returneras felkoden och meddelandet. Eventuellt kan resultatet skrivas till en fil på disken. Som standard kommer algoritmen att varna vid fel. Alternativt kan algoritmen ställas in så att HTTP-fel behandlas som misslyckanden.

Parametrar

Etikett	Namn	Typ	Beskrivning
URL eller sökväg till fil	`URL`	[sträng]	URL eller sökväg till den fil som ska öppnas.
Metod	`METHOD`	[uppräkning] Standard: 0	Den HTTP-metod som ska användas för begäran. Alternativen är: • 0 — GET • 1 — POST
POST Data Valfritt	`DATA`	[sträng]	De data som ska läggas till i brödtexten om begäran är en POST.
Autenticering Valfritt	`AUTH_CONFIG`	[authconfig] Standard: Ingen autentisering	En autentiseringskonfiguration att skicka
Betrakta HTTP-fel som misslyckanden	`FAIL_ON_ERROR`	[boolean] Standard: Falsk	Om den är inställd kommer algoritmen att misslyckas om den stöter på ett HTTP-fel.
Filens destination Valfritt	`OUTPUT`	[sträng] Standard: `[Hoppa över utdata]`	Resultatet kan skrivas till en fil i stället för att returneras som en sträng. Specifikation av filens destination. *En av*: • Hoppa över utmatning • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Filens destination	`OUTPUT`	&Arkiv	Den nedladdade filen med det returnerade resultatet
HTTP-status	`STATUS_CODE`	[sträng]	HTTP-status
Felkod för nätverk	`ERROR_CODE`	[sträng]	Felkoden när det misslyckas (som vid 404 HTTP Status etc.)
Nätverksfelmeddela	`ERROR_MESSAGE`	[sträng]	En sträng som innehåller felmeddelandet i händelse av fel
Resultatdata	`RESULT_DATA`	[sträng]	En sträng som innehåller resultatdata i händelse av framgång

Python-kod

Algoritm ID: `native:httprequest`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Öppna fil eller URL

Öppnar filer i deras associerade standardprogram eller webbadresser i användarens standardwebbläsare.

Parametrar

Etikett	Namn	Typ	Beskrivning
URL eller sökväg till fil	`URL`	[sträng]	URL eller sökväg till den fil som ska öppnas.

Utgångar

Algoritmen har ingen utdata.

Python-kod

Algoritm ID: `native:openurl`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.5 GPS

Konvertera GPS-data

Använder verktyget GPSBabel för att konvertera en GPS-datafil från en rad olika format till GPX-standardformatet.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsfil	`INPUT`	&Arkiv	Fil som innehåller data som ska konverteras
Format	`FORMAT`	[uppräkning]	Format på filen som ska konverteras, från den här listan.
Funktionstyp	`FEATURE_TYPE`	[uppräkning] Standard: 0	Typ av data som ska konverteras • 0 — Waypoints • 1 — Rutter • 2 — Spår
Utdata	`OUTPUT`	[vektor: vilken som helst] Standard: `[Spara till temporär fil]`	Specifikation av GPX-filen för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Utdatalager	`OUTPUT_LAYER`	[vektor: vilken som helst]	Utmatningslager med data i GPX-standardformat

Python-kod

Algoritm-ID: `native:convertgpsdata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Konvertera GPX-funktionstyp

Använder verktyget GPSBabel för att konvertera GPX-funktioner från en typ till en annan (t.ex. konvertera alla waypoint-funktioner till en rutt).

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsfil	`INPUT`	&Arkiv	Fil som innehåller data som ska konverteras

continues on next page

Table 24.10 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Omvandling	`CONVERSION`	[uppräkning] Standard: 0	Vilken typ av konvertering som ska tillämpas • 0 — Waypoints från en rutt • 1 — Waypoints från ett spår • 2 — Rutter från waypoints • 3 — Spår från waypoints
Utdata	`OUTPUT`	[vektor: punkt eller linje] Standard: `[Spara till temporär fil]`	Specifikation av utdatafilen. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Utdata	`OUTPUT`	[vektor: vilken som helst]	Utgående lager med konverterade GPX-funktioner

Python-kod

Algoritm ID: `` `native:convertgpxfeaturetype ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Hämta GPS-data från enheten

Använder verktyget GPSBabel för att ladda ner data från en GPS-enhet till GPX-standardformatet.

Parametrar

Etikett	Namn	Typ	Beskrivning
Enhet	`DEVICE`	[uppräkning] Standard:*Garmin serie*	Den GPS-enhet som används för att skapa data. Måste deklareras i dialogen *GPS Settings*.
Port	`PORT`	[uppräkning]	Den port som enheten är ansluten till. Tillgängliga portar beror på operativsystemet.

continues on next page

Table 24.12 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Funktionstyp	`FEATURE_TYPE`	[uppräkning] Standard: 0	Typ av data som ska konverteras • 0 — Waypoints • 1 — Rutter • 2 — Spår
Utdata	`OUTPUT`	[vektor: vilken som helst] Standard: `[Spara till temporär fil]`	Specifikation av utdatafilen. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Utdatalager	`OUTPUT_LAYER`	[vektor: vilken som helst]	Utmatningslager med data i GPX-standardformat

Python-kod

Algoritm-ID: `native:nedladdninggpsdata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Ladda upp GPS-data till enheten

Använder verktyget GPSBabel för att ladda upp data till en GPS-enhet från GPX-standardformatet.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsfil	`INPUT`	&Arkiv	`.GPX`-fil som innehåller de data som ska laddas upp
Enhet	`DEVICE`	[uppräkning] Standard:*Garmin serie*	Den GPS-enhet som du vill ladda upp data till. Måste deklareras i dialogrutan *GPS-inställningar*.
Port	`PORT`	[uppräkning]	Den port som enheten är ansluten till. Tillgängliga portar beror på operativsystemet.

continues on next page

Table 24.14 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Funktionstyp	`FEATURE_TYPE`	[uppräkning] Standard: 0	Typ av data som ska laddas upp • 0 — Waypoints • 1 — Rutter • 2 — Spår

Utgångar

Ingen utmatning sker. Om det lyckas laddas data till enheten.

Python-kod

Algoritm ID: `native:uploadgpsdata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.6 Interpolation

Värmekarta (uppskattning av kärntäthet)

Skapar ett täthetsraster (värmekarta) av ett inmatat punktvektorlager med hjälp av kärntäthetsuppskattning.

Densiteten beräknas utifrån antalet punkter på en plats, där ett större antal klustrade punkter ger större värden. Värmekartor gör det enkelt att identifiera *hotspots* och klustring av punkter.

Parametrar

Etikett	Namn	Typ	Beskrivning
Punktlager	`INPUT`	[vektor: punkt]	Punktvektorlager som ska användas för värmekartan
Radius	`RADIUS`	[numerisk: dubbel] Standard: 100.0	Heatmap-sökradie (eller kärnbandbredd) i kartenheter. Radien anger det avstånd runt en punkt där punktens påverkan kommer att märkas. Större värden resulterar i större utjämning, men mindre värden kan visa finare detaljer och variation i punkttäthet.

continues on next page

Table 24.15 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Storlek på utmatningsraster	`PIXEL_SIZE`	[numerisk: dubbel] Standard: 0,1	Pixelstorlek för det utgående rasterlagret i lagerenheter. I GUI kan storleken anges med antalet rader (`Antal rader`) / kolumner (`Antal kolumner`) **eller** pixelstorleken(`Pixelstorlek X` / `Pixelstorlek Y`). Om du ökar antalet rader eller kolumner minskar cellstorleken och filstorleken på utdatarastret ökar. Värdena i `Rows`, `Columns`, `Pixel Size X` och `Pixel Size Y` uppdateras samtidigt - en fördubbling av antalet rader innebär en fördubbling av antalet kolumner och en halvering av cellstorleken. Utbredningen av utdatarastret förblir densamma (ungefär).
Radie från fältet Valfritt	`RADIUS_FIELD`	[tabellfält: numerisk]	Ställer in sökradien för varje funktion från ett attributfält i inmatningslagret.
Vikt från fältet Valfritt	`WEIGHT_FIELD`	[tabellfält: numerisk]	Tillåter att indatafunktioner viktas med ett attributfält. Detta kan användas för att öka det inflytande som vissa funktioner har på den resulterande värmekartan.
Kärnans form	`KERNEL`	[uppräkning] Standard: *0*	Styr hur snabbt en punkts inflytande minskar när avståndet från punkten ökar. Olika kärnor avtar i olika takt, så en triviktskärna ger funktioner större vikt för avstånd närmare punkten än vad Epanechnikov-kärnan gör. Följaktligen resulterar triweight i ”skarpare” hotspots och Epanechnikov i ”mjukare” hotspots. Det finns många olika former (se Wikipediasidan <https://en.wikipedia.org/wiki/Kernel_(statistik)#Kernel_functions_in_common_use>`_ för mer information): • 0 — Kvadratisk • 1 — Triangelformad • 2 — Uniform • 3 — Trivikt • 4 — Epanechnikov

continues on next page

Table 24.15 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Sönderfallskvot (endast triangulära kärnor) Valfritt	`DECAY`	[numerisk: dubbel] Standard: *0.0*	Kan användas med triangulära kärnor för att ytterligare kontrollera hur värmen från ett objekt minskar med avståndet från objektet. • Ett värde på 0 (=minimum) innebär att värmen koncentreras till mitten av den givna radien och släcks helt vid kanten. • Ett värde på 0,5 innebär att pixlar i utkanten av radien får hälften så mycket värme som pixlar i mitten av sökradien. • Ett värde på 1 innebär att värmen sprids jämnt över hela cirkeln med sökradien. (Detta motsvarar kärnan ”Uniform”) • Ett värde som är större än 1 indikerar att värmen är högre mot kanten av sökradien än i mitten.
Skalning av utmatningsvärde	`OUTPUT_VALUE`	[uppräkning] Standard: *0*	Gör det möjligt att ändra värdena för den utgående värmekartans raster. En av: • 0 — Obearbetad • 1 — Skalad
Värmekarta	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange det utgående rasterlagret med kärndensitetsvärden. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Värmekarta	`OUTPUT`	[raster]	Rasterlager med kärndensitetsvärden

Exempel: Skapa en värmekarta

I följande exempel använder vi vektorpunktskiktet `airports` från QGIS exempeldatauppsättning (se *Nedladdning av exempeldata*). En annan utmärkt QGIS-handledning om hur man gör värmekartor finns på http://qgistutorials.com.

I Fig. 24.2 visas flygplatserna i Alaska.

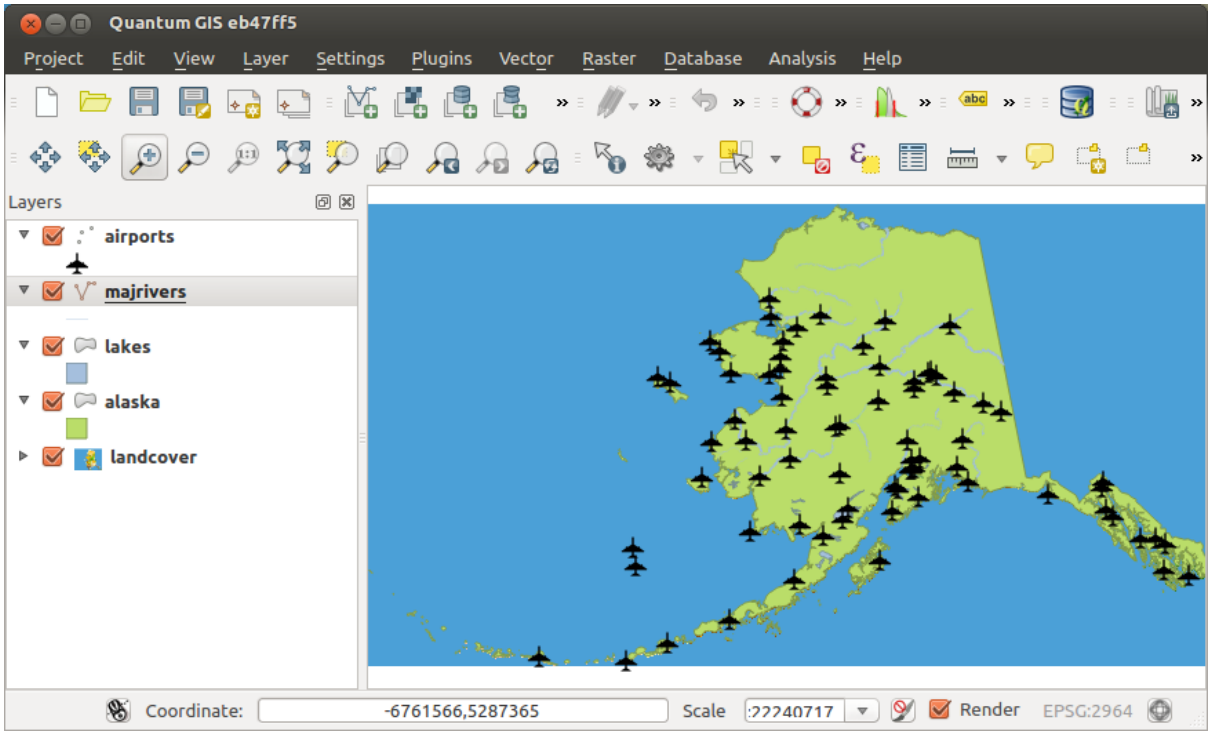


Fig. 24.2: Flygplatser i Alaska

1. Öppna algoritmen *Heatmap (Kernel Density Estimation)* från QGIS *Interpolation*-gruppen
2. I fältet *Point layer* väljer du `airports` från listan över punktlager som laddats i det aktuella projektet.
3. Ändra *Radius* till `1000000` meter.
4. Ändra *Pixel size X* till `1000`. *Pixel size Y*, *Rows* och *Columns* uppdateras automatiskt.
5. Klicka på *Run* för att skapa och ladda flygplatsens värmekarta (se Fig. 24.4).

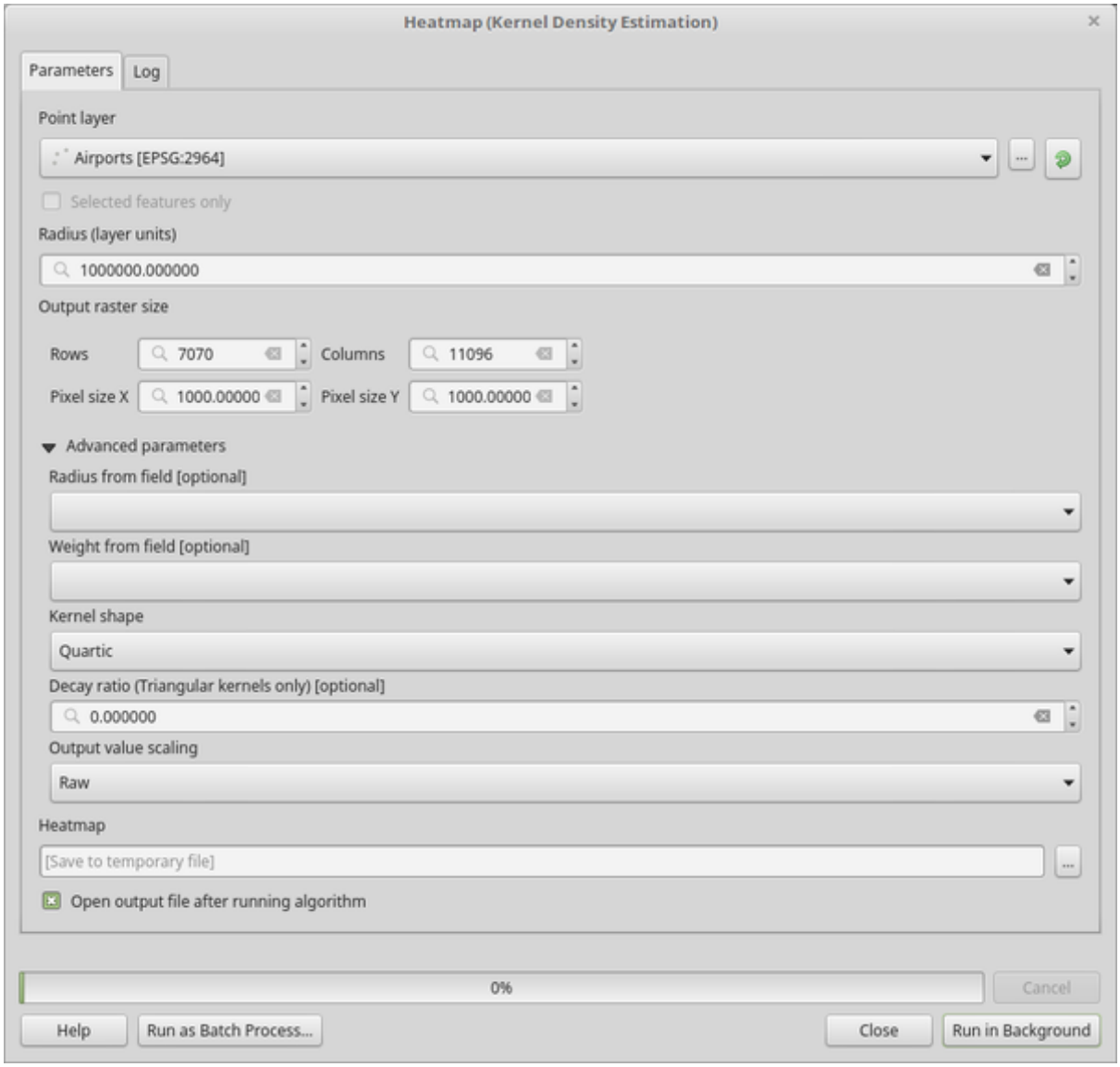


Fig. 24.3: Dialogrutan för värmekarta

QGIS genererar värmekartan och lägger till den i kartfönstret. Som standard är värmekartan skuggad i gråskala, där ljusare områden visar högre koncentrationer av flygplatser. Värmekartan kan nu stylas i QGIS för att förbättra dess utseende.

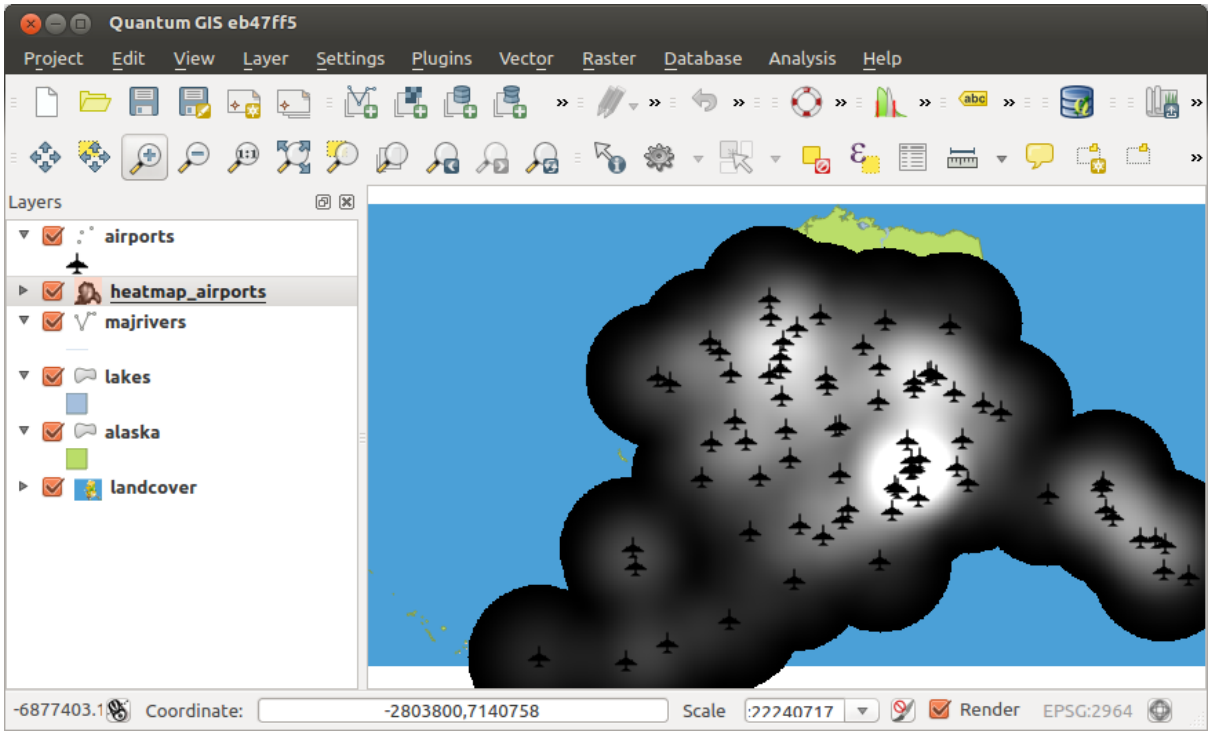


Fig. 24.4: Värmekartan efter laddning ser ut som en grå yta

1. Öppna dialogrutan för egenskaper för lagret `heatmap_airports` (markera lagret `heatmap_airports`, öppna snabbmenyn med höger musknapp och välj *Properties*).
2. Välj fliken *Symbologi*.
3. Ändra *Render type* till ”Singleband pseudocolor”.
4. Välj en lämplig *Color ramp* , till exempel `YlOrRd`.
5. Klicka på knappen *Classify*.
6. Tryck på *OK* för att uppdatera lagret.

Det slutliga resultatet visas i Fig. 24.5.

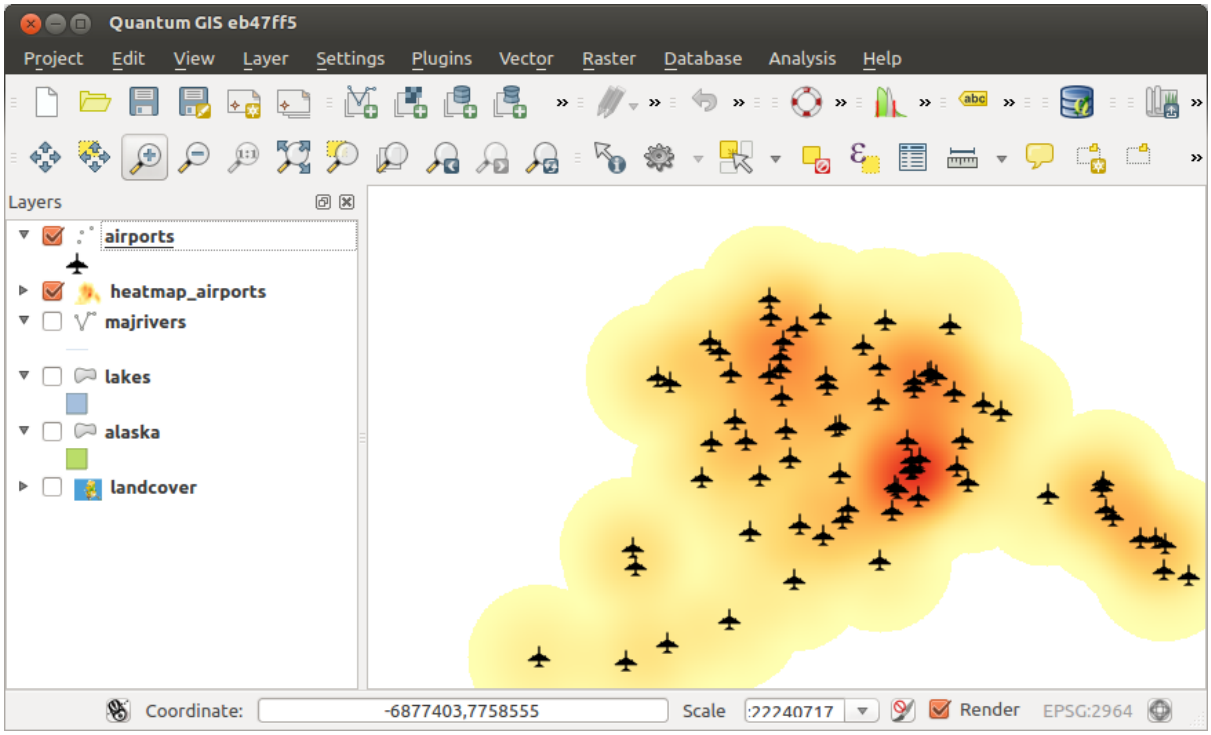


Fig. 24.5: Stylad värmekarta över flygplatser i Alaska

Python-kod

Algoritm-ID: `qgis:heatmapkerneldensityestimation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

IDW-interpolering

Skapar en IDW-interpolation (Inverse Distance Weighted) av ett punktvektorlager.

Provpunkterna viktas under interpoleringen så att påverkan från en punkt i förhållande till en annan minskar med avståndet från den okända punkt som du vill skapa.

IDW-interpolationsmetoden har också vissa nackdelar: kvaliteten på interpolationsresultatet kan minska om fördelningen av datapunkterna i urvalet är ojämn.

Dessutom kan maximi- och minimivärden i den interpolerade ytan endast förekomma vid datapunkter i urvalet.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INTERPOLATION_`	[sträng]	Vektorlager och fält som ska användas för interpoleringen, kodade i en sträng (se klassen `ParameterInterpolationData` i InterpolationWidgets för mer information). Följande GUI-element finns för att komponera strängen med interpoleringsdata: • **Vektorlager** [vektor: valfri] • **Interpoleringsattribut** [tabellfält: numeriskt]: Attribut som ska användas i interpoleringen • **Använd Z-koordinat för interpolering** [boolean]: Använder lagrets lagrade Z-värden (standard: False) För var och en av de tillagda kombinationerna av lager och fält kan en typ väljas: • *Poäng* • *Strukturerade linjer* • *Brytande linjer* I strängen separeras lagerfältselementen med `'::|::'`. Underelementen i lagerfältselementen separeras med `'::~::'`.
Avståndskoefficient P	`DISTANCE_COEFF`	[numerisk: dubbel] Standard: 2,0	Ställer in avståndskoefficienten för interpoleringen. Minsta värde: 0.0, högsta: 100.0.
Extent (xmin, xmax, ymin, ymax)	`EXTENT`	[omfattning]	Utsträckningen av det utgående rasterlagret. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`

continues on next page

Table 24.17 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Storlek på utmatningsraster	`PIXEL_SIZE`	[numerisk: dubbel] Standard: 0,1	Pixelstorlek för det utgående rasterlagret i lagerenheter. I GUI kan storleken anges med antalet rader (`Antal rader`) / kolumner (`Antal kolumner`) **eller** pixelstorleken(`Pixelstorlek X` / `Pixelstorlek Y`). Om du ökar antalet rader eller kolumner minskar cellstorleken och filstorleken på utdatarastret ökar. Värdena i `Rows`, `Columns`, `Pixel Size X` och `Pixel Size Y` uppdateras samtidigt - en fördubbling av antalet rader innebär en fördubbling av antalet kolumner och en halvering av cellstorleken. Utbredningen av utdatarastret förblir densamma (ungefär).
Interpolerad	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Rasterlager med interpolerade värden. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Interpolerad	`OUTPUT`	[raster]	Rasterlager med interpolerade värden

Python-kod

Algoritm-ID: `qgis:idwinterpolation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Linjetäthet

Beräknar för varje rastercell täthetsmåttet för linjära objekt inom ett cirkulärt grannskap. Detta mått erhålls genom att summera alla linjesegment som skär det cirkulära grannskapet och dividera denna summa med grannskapets area. En viktningsfaktor kan tillämpas på linjesegmenten.

Observera

Den här algoritmen använder ellipsoidbaserade mätningar och respekterar de aktuella *ellipsoidinställningarna*.

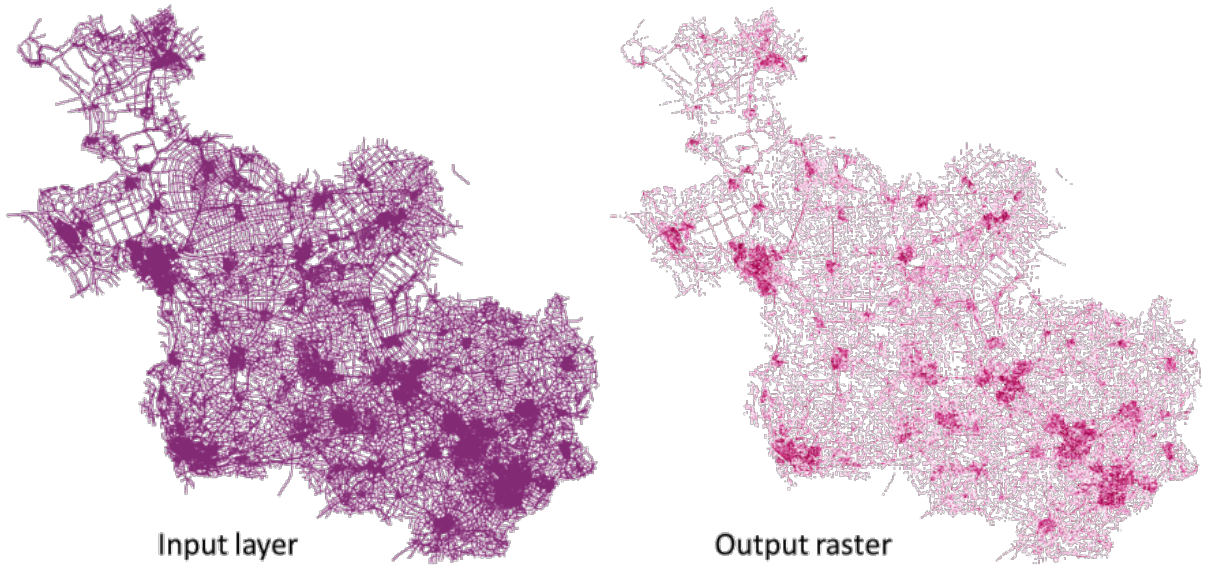


Fig. 24.6: Exempel på linjetäthet. Källa för inmatningslager: Vägar Overijssel - Nederländerna (OSM).

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslinjelage	`INPUT`	[vektor: linje]	Vektorlager för indata som innehåller linjefunktioner
Viktfält	`WEIGHT`	[tabellfält: numerisk]	Fält för det lager som innehåller den viktfaktor som ska användas vid beräkningen
Sökradie	`RADIUS`	[numerisk: dubbel] Standard: 10.0	Radie för det cirkulära grannskapet. Enheter kan anges här.
Pixelstorlek	`PIXEL_SIZE`	[numerisk: dubbel] Standard: 10.0	Pixelstorlek för det utgående rasterlagret i lagerenheter. Rastret har kvadratiska pixlar.
** Linjetäthet raster**	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Utdata som ett rasterlager. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
** Linjetäthet raster**	`OUTPUT`	[raster]	Det utgående rasterlagret för linjetäthet.

Python-kod

Algoritm-id: `native:linedensity`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

TIN-interpolering

Skapar en TIN-interpolation (Triangulated Irregular Network) av ett punktvektorlager.

Med TIN-metoden kan du skapa en yta som bildas av trianglar av närmaste grannpunkter. För att göra detta skapas omkretsar runt utvalda provpunkter och deras skärningspunkter ansluts till ett nätverk av icke överlappande och så kompakta trianglar som möjligt. De resulterande ytorna är inte släta.

Algoritmen skapar både rasterlagret med de interpolerade värdena och vektorlinjeskiktet med trianguleringsgränserna.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INTERPOLATION_`	[sträng]	Vektorlager och fält som ska användas för interpoleringen, kodade i en sträng (se klassen `ParameterInterpolationData` i InterpolationWidgets för mer information). Följande GUI-element finns för att komponera strängen med interpoleringsdata: • **Vektorlager** [vektor: valfri] • **Interpoleringsattribut** [tabellfält: numeriskt]: Attribut som ska användas i interpoleringen • **Använd Z-koordinat för interpolering** [boolean]: Använder lagrets lagrade Z-värden (standard: False) För var och en av de tillagda kombinationerna av lager och fält kan en typ väljas: • *Poäng* • *Strukturerade linjer* • *Brytande linjer* I strängen separeras lagerfältselementen med `'::|::'`. Underelementen i lagerfältselementen separeras med `'::~::'`.
Interpolationsmetod	`METHOD`	[uppräkning] Standard: 0	Ställ in vilken interpoleringsmetod som ska användas. En av: • *Lineär* • *Clough-Toucher (kubik)*
Extent (xmin, xmax, ymin, ymax)	`EXTENT`	[omfattning]	Utsträckningen av det utgående rasterlagret. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`

continues on next page

Table 24.22 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Storlek på utmatningsraster	`PIXEL_SIZE`	[numerisk: dubbel] Standard: 0,1	Pixelstorlek för det utgående rasterlagret i lagerenheter. I GUI kan storleken anges med antalet rader (`Antal rader`) / kolumner (`Antal kolumner`) **eller** pixelstorleken(`Pixelstorlek X` / `Pixelstorlek Y`). Om du ökar antalet rader eller kolumner minskar cellstorleken och filstorleken på utdatarastret ökar. Värdena i `Rows`, `Columns`, `Pixel Size X` och `Pixel Size Y` uppdateras samtidigt - en fördubbling av antalet rader innebär en fördubbling av antalet kolumner och en halvering av cellstorleken. Utbredningen av utdatarastret förblir densamma (ungefär).
Interpolerad	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Den utgående TIN-interpolationen som ett rasterlager. *En av*: • Spara till en temporär fil • Spara till fil…
Triangulering	`TRIANGULATION`	[vektor: linje] Standard: `[Hoppa över utdata]`	Utdata TIN som ett vektorlager. *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Interpolerad	`OUTPUT`	[raster]	Den utgående TIN-interpolationen som ett rasterlager
Triangulering	`TRIANGULATION`	[vektor: linje]	Utdata TIN som ett vektorlager.

Python-kod

Algoritm-ID: `qgis:tininterpolation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.7 Verktyg för lager

Exportera information om lager

Skapar ett polygonlager med egenskaper som motsvarar omfattningen av det/de valda lagret/lagren.

Ytterligare skiktdetaljer (CRS, leverantörsnamn, filsökväg, lagernamn, subsetfilter, abstract och attribution) bifogas som attribut till varje funktion.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`LAYERS`	[lager] [lista]	Inmatningslager för att få information om.
Utdata	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Specifikation av utdatalagret med information. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Utdata	`OUTPUT`	[vektor: polygon]	Polygonvektorskikt som visar utbredningen av ingående lager och tillhörande information i attribut.

Python-kod

Algoritm-ID: `native:exportlayersinformation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exportera till kalkylblad

Exporterar attributen för ett urval av vektorlager till ett kalkylbladsdokument eller lägger till dem i ett befintligt kalkylblad som ytterligare ark.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`LAYERS`	[vektor: valfri] [lista]	Vektorlager för inmatning. Utdatakalkylbladet kommer att bestå av ett ark för varje lager som innehåller attributen för detta lager.
Använd fältalias som kolumnrubriker	`USE_ALIAS`	[boolean] Standard: Falsk	Använd fältaliasen från attributtabellen för kalkylbladet.
Exportera formaterade värden istället för råa värden	`FORMATTED_VALU`	[boolean] Standard: Falsk	Om `True` exporteras de formaterade, läsbara värdena (t.ex. från en *värdemapp eller värderelation*) till kalkylbladet.
Overwrite existing spreadsheet (Skriv över befintligt kalkylblad)	`OVERWRITE`	[boolean] Standard: Sann	Om det angivna kalkylbladet finns, kommer detta alternativ att skriva över det befintliga kalkylbladet om det sätts till `True`. Om alternativet är `False` och kalkylbladet finns, kommer lagren att läggas till som ytterligare ark.
Kalkylblad för destination	`OUTPUT`	&Arkiv Standard: `[Spara till temporär fil]`	Utmatningskalkylblad med ett ark för varje lager. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Kalkylblad för destination	`OUTPUT`	&Arkiv	Kalkylblad med ett blad för varje lager.
Lager inom kalkylblad	`OUTPUT_LAYERS`	Konfigurationens id %1 finns redan i listan	Listan över ark som lagts till i kalkylbladet.

Python-kod

Algoritm-ID: `native:exporttospreadsheet`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Extrahera lagrets utsträckning

Genererar ett vektorlager med den minsta avgränsningsboxen (rektangel med N-S-orientering) som täcker alla indatafunktioner.

Utmatningslagret innehåller en enda avgränsande box för hela inmatningslagret.

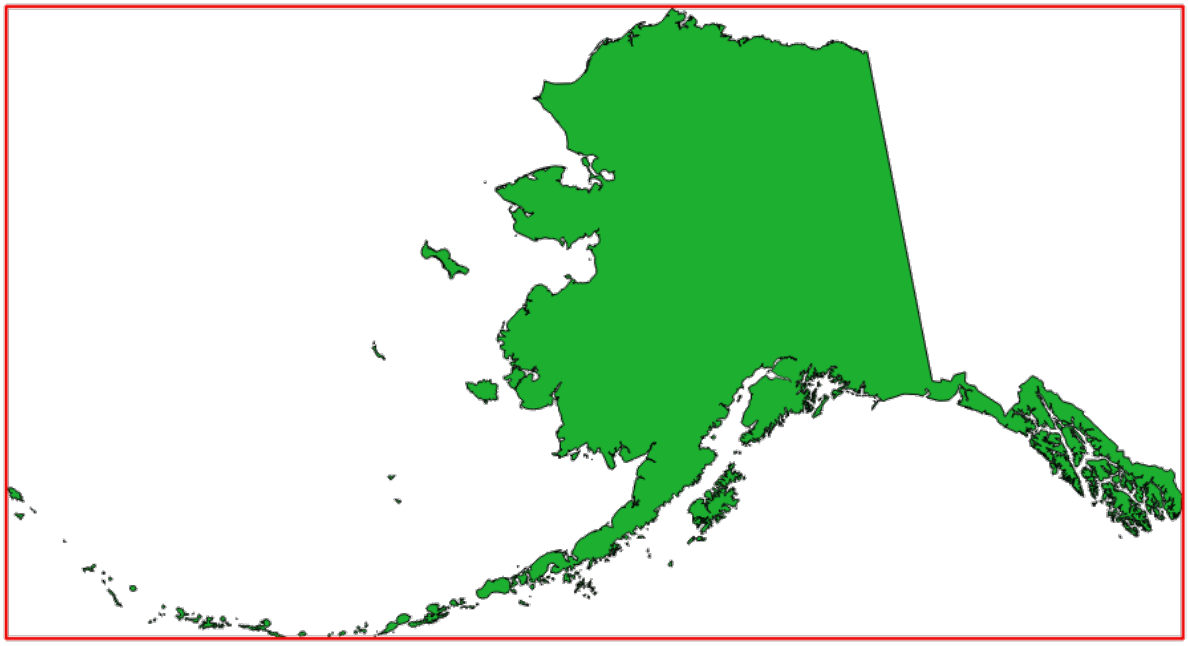


Fig. 24.7: I rött avgränsningsrutan för källagret

Standardmeny: *Vektor* ► *Forskningsverktyg*

Parametrar

Etikett	Namn	Typ	Beskrivning
Lager	`INPUT`	[lager]	Inmatningslager
Extent	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange polygonvektorlagret för utdataomfånget. *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Extent	`OUTPUT`	[vektor: polygon]	Vektorlager för utdata (polygon) med utsträckning (minsta avgränsande box)

Python-kod

Algoritm-ID: `qgis:polygonfrånlayerextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.8 Nät

Exportera konturer

Skapar konturer som ett vektorlager från en skalär dataset för mesh.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsnätlager	`INPUT`	[mesh]	Mesh-lagret att exportera data från
Dataset-grupper	`DATASET_GROUPS`	[lager] [lista]	Datasetet består av följande grupper
Dataset-tid	`DATASET_TIME`	[datetime]	Tidsintervall att ta hänsyn till • 0 — Aktuell canvastid • 1 — Definierat datum/tid • 2 — Dataset grupp tidssteg
Inkrement mellan konturnivåer Valfritt	`INCREMENT`	[numerisk: dubbel] Standard: Ej inställd	Intervall mellan genererade nivåer.
Minsta konturnivå Valfritt	`MINIMUM`	[numerisk: dubbel] Standard: Ej inställd	Startnivåvärden för konturer. Inga genererade nivåer kommer att vara lägre än detta värde.
Maximal konturnivå Valfritt	`MAXIMUM`	[numerisk: dubbel] Standard: Ej inställd	Konturernas maxvärden, dvs. inga genererade nivåer kommer att vara större än detta värde.
Lista över konturer nivå Valfritt	`CONTOUR_LEVEL_`	[sträng] Standard: Ej inställd	Lista över önskade konturnivåer (åtskilda med kommatecken). Om fälten increment, minimum och maximum är ifyllda kommer de inte att beaktas.
Koordinatsystem för utdata Valfritt	`CRS_OUTPUT`	[crs]	Koordinatreferenssystem som ska tilldelas utmatningen

continues on next page

Table 24.24 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Exporterade konturlinjer	`OUTPUT_LINES`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange det utgående linjelagret som representerar konturerna av mesh-lagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Exporterade konturpolygoner	`OUTPUT_POLYGON`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående polygonlagret som representerar mesh-lagrets konturer. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Exporterade konturlinjer	`OUTPUT_LINES`	[vektor: linje]	Linjelager som representerar konturerna av mesh-lagret.
Exporterade konturpolygoner	`OUTPUT_POLYGON`	[vektor: polygon]	Polygonlager som representerar konturerna av meshlagret.

Python-kod

Algoritm-ID: `native:meshkonturer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exportera värden för tvärsnittsdataset på linjer från mesh

Extraherar värden för en mesh-datauppsättning från linjer i ett vektorlager.

Varje linje är diskretiserad med en parameter för upplösningsavstånd för extraktion av värden på dess hörn.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsnätlager	`INPUT`	[mesh]	Mesh-lagret att exportera data från
Dataset-grupper	`DATASET_GROUPS`	[lager] [lista]	Datasetet består av följande grupper
Dataset-tid	`DATASET_TIME`	[datetime]	Tidsintervall att ta hänsyn till • 0 — Aktuell canvastid • 1 — Definierat datum/tid • 2 — Dataset grupp tidssteg
Rader för dataexport	`INPUT_LINES`	[vektor: linje]	Rader där data kommer att extraheras från datasetets nät
Upplösning av linjesegmentering	`RESOLUTION`	[numerisk: dubbel] Standard: 10.0	Avståndet mellan punkter på de linjer där data kommer att extraheras från datasetets mesh. Enheter kan väljas.
Digits count for coordinates	`COORDINATES_DI`	[numerisk: heltal] Standard: 2	Antal siffror för avrundning av koordinatvärdena
Antal siffror för datasetets värde	`DATASET_DIGITS`	[numerisk: heltal] Standard: 2	Antal siffror för avrundning av datasetets värden
CSV-fil med exporterade data	`OUTPUT`	&Arkiv Standard: `[Spara till temporär fil]`	Specifikation av utdatafilen. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
CSV-fil med exporterade data	`OUTPUT`	&Arkiv	

Python-kod

Algoritm-ID: `native:meshexportcrosssection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exportera nätkanter

Exporterar ett mesh-lagers kanter till ett linjevektorlager, med datasetvärdena på kanterna som attributvärden.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsnätlager	`INPUT`	[mesh]	Mesh-lagret att exportera data från
Dataset-grupper	`DATASET_GROUPS`	[lager] [lista]	Datasetet består av följande grupper
Dataset-tid	`DATASET_TIME`	[datetime]	Tidsintervall att ta hänsyn till • 0 — Aktuell canvastid • 1 — Definierat datum/tid • 2 — Dataset grupp tidssteg
Koordinatsystem för utdata Valfritt	`CRS_OUTPUT`	[crs]	Koordinatreferenssystem som ska tilldelas utmatningen
Export vektor alternativ	`VECTOR_OPTION`	[uppräkning]	Koordinattyp för export av vektorvärde. • 0 — Kartesisk (x,y) • 1 — Polar (magnitud, grad) • 2 — Cartesiansk och polär
Vektorlager för utdata	`OUTPUT`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Specifikation av utdatafilen. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Vektorlager för utdata	`OUTPUT`	[vektor: linje]	Utgående vektorlinjeskikt som innehåller kanterna på det ingående nätlagret med tillhörande datasetvärden

Python-kod

Algoritm-ID: `` `native:exportmeshedges` ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exportera nätytor

Exporterar ett nätlagers ytor till ett polygonvektorskikt, med datasetvärdena på ytorna som attributvärden.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsnätlager	`INPUT`	[mesh]	Mesh-lagret att exportera data från
Dataset-grupper	`DATASET_GROUPS`	[lager] [lista]	Datasetet består av följande grupper
Dataset-tid	`DATASET_TIME`	[datetime]	Tidsintervall att ta hänsyn till • 0 — Aktuell canvastid • 1 — Definierat datum/tid • 2 — Dataset grupp tidssteg
Koordinatsystem för utdata Valfritt	`CRS_OUTPUT`	[crs]	Koordinatreferenssystem som ska tilldelas utmatningen
Export vektor alternativ	`VECTOR_OPTION`	[uppräkning]	Koordinattyp för export av vektorvärde. • 0 — Kartesisk (x,y) • 1 — Polar (magnitud, grad) • 2 — Cartesiansk och polär
Vektorlager för utdata	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Specifikation av utdatafilen. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Vektorlager för utdata	`OUTPUT`	[vektor: polygon]	Vektorpolygonlager för utdata som innehåller ytorna i meshlagret för inmatning med tillhörande datasetvärden

Python-kod

Algoritm-ID: `native:exportmeshfaces`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exportera mesh på rutnät

Exporterar ett mesh-lagers datasetvärden till ett vektorlager med rutnätspunkter, med datasetvärdena på denna punkt som attributvärden.

För volymdata (staplade 3D-datasetvärden) medelvärdesberäknas de exporterade datasetvärdena på ytor med den metod som definieras i *meshlagrets egenskaper* (standard är Multi level averaging method). 1D-mesh stöds inte.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsnätlager	`INPUT`	[mesh]	Mesh-lagret att exportera data från
Dataset-grupper	`DATASET_GROUPS`	[lager] [lista]	Datasetet består av följande grupper
Dataset-tid	`DATASET_TIME`	[datetime]	Tidsintervall att ta hänsyn till • 0 — Aktuell canvastid • 1 — Definierat datum/tid • 2 — Dataset grupp tidssteg
Extent Valfritt	`EXTENT`	[omfattning]	Ange den spatiala omfattning som data ska bearbetas på. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Avstånd mellan rutorna Valfritt	`GRID_SPACING`	[numerisk: dubbel] Standard: 10.0	Avstånd mellan de provpunkter som ska användas
Koordinatsystem för utdata Valfritt	`CRS_OUTPUT`	[crs]	Koordinatreferenssystem som ska tilldelas utmatningen
Export vektor alternativ	`VECTOR_OPTION`	[uppräkning]	Koordinattyp för export av vektorvärde. • 0 — Kartesisk (x,y) • 1 — Polar (magnitud, grad) • 2 — Cartesiansk och polär
Vektorlager för utdata	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Specifikation av utdatafilen. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Vektorlager för utdata	`OUTPUT`	[vektor: punkt]	Utgående vektorpunktlager med datasetvärden som beräknats från det överlagrade ansiktet.

Python-kod

Algoritm-ID: `` `native:exportmeshongrid` ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exportera nätets hörnpunkter

Exporterar ett mesh-lagers hörn till ett punktvektorskikt, med datasetvärdena på hörnen som attributvärden.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsnätlager	`INPUT`	[mesh]	Mesh-lagret att exportera data från
Dataset-grupper	`DATASET_GROUPS`	[lager] [lista]	Datasetet består av följande grupper
Dataset-tid	`DATASET_TIME`	[datetime]	Tidsintervall att ta hänsyn till • 0 — Aktuell canvastid • 1 — Definierat datum/tid • 2 — Dataset grupp tidssteg
Koordinatsystem för utdata Valfritt	`CRS_OUTPUT`	[crs]	Koordinatreferenssystem som ska tilldelas utmatningen
Export vektor alternativ	`VECTOR_OPTION`	[uppräkning]	Koordinattyp för export av vektorvärde. • 0 — Kartesisk (x,y) • 1 — Polar (magnitud, grad) • 2 — Cartesiansk och polär
Vektorlager för utdata	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Specifikation av utdatafilen. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Vektorlager för utdata	`OUTPUT`	[vektor: punkt]	Utgående vektorpunktlager som innehåller hörnen i det ingående meshlagret med tillhörande datasetvärden

Python-kod

Algoritm-ID: `` `native:exportmeshvertices` ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exportera tidsserievärden från punkter i en mesh-datauppsättning

Extraherar tidsserievärden för en mesh-datauppsättning från punkter i ett vektorlager.

Om tidssteget behålls till sitt standardvärde (0 timmar) används tidssteget för de två första dataseten i den först valda datasetgruppen.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsnätlager	`INPUT`	[mesh]	Mesh-lagret för att extrahera data från
Dataset-grupper	`DATASET_GROUPS`	[lager] [lista]	Datasetet består av följande grupper
Starttid	`STARTING_TIME`	[datetime]	Början på det tidsintervall som ska tas med i beräkningen • 0 — Aktuell canvastid • 1 — Definierat datum/tid • 2 — Dataset grupp tidssteg
** Sluttid**	`FINISHING_TIME`	[datetime]	Slutet på tidsintervallet att ta hänsyn till • 0 — Aktuell canvastid • 1 — Definierat datum/tid • 2 — Dataset grupp tidssteg
Tidssteg (timmar) Valfritt	`TIME_STEP`	[numerisk: dubbel] Standard: 0	Tid mellan två på varandra följande steg för att extrahera. Behåll `0` för att använda tidssteget för den första valda datasetgruppen.
Poäng för dataexport	`INPUT_POINTS`	[vektor: punkt]	Vektorlager som innehåller punkter där data kommer att extraheras från datasetet mesh
Digits count for coordinates	`COORDINATES_DI`	[numerisk: heltal]	Antal siffror för avrundning av koordinatvärden Standard: 2
Antal siffror för datasetets värde	`DATASET_DIGITS`	[numerisk: heltal] Standard: 2	Antal siffror för avrundning av dataset-värden

continues on next page

Table 24.30 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
CSV-fil med exporterade data	`OUTPUT`	&Arkiv Standard: `[Spara till temporär fil]`	Specifikation av utdatafilen. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
CSV-fil med exporterade data	`OUTPUT`	&Arkiv	`.CSV`-fil som innehåller tidsserievärden för nätdatasetet vid de överlagrade punktfunktionerna

Python-kod

Algoritm-ID: `native:meshexporttimeseries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rastrera nätdataset

Skapar ett rasterlager från en mesh-datauppsättning.

För volymdata (staplade 3D-datasetvärden) medelvärdesberäknas de exporterade datasetvärdena på ytor med den metod som definieras i *meshlagrets egenskaper* (standard är Multi level averaging method). 1D-mesh stöds inte.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsnätlager	`INPUT`	[mesh]	Mesh-lagret att exportera data från
Dataset-grupper	`DATASET_GROUPS`	[lager] [lista]	Datasetet består av följande grupper
Dataset-tid	`DATASET_TIME`	[datetime]	Tidsintervall att ta hänsyn till • 0 — Aktuell canvastid • 1 — Definierat datum/tid • 2 — Dataset grupp tidssteg

continues on next page

Table 24.31 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Extent Valfritt	`EXTENT`	[omfattning]	Ange den spatiala omfattning som data ska bearbetas på. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Pixelstorlek	`PIXEL_SIZE`	[numerisk: dubbel] Standard: 1,0	Pixelstorlek för det utgående rasterlagret.
Koordinatsystem för utdata Valfritt	`CRS_OUTPUT`	[crs]	Koordinatreferenssystem som ska tilldelas utmatningen
Rasterlager för utdata	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utdatafilen. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Rasterlager för utdata	`OUTPUT`	[raster]	Utgående rasterlager med datasetvärden som beräknats från meshlagret.

Python-kod

Algoritm-ID: `native:meshrasterize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapande av TIN-nät

Skapar ett TIN-meshlager från vektorlager. TIN-nätet skapas med hjälp av en Delaunay-triangulering.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`SOURCE_DATA`	[vektor: valfri] [lista]	Vektorlager som ska kombineras för att generera mesh-lagret
Ett vektorlager	ENDAST GUI	[vektor: valfri] [lista]	En väljare för de vektorlager som ska kombineras för att generera mesh-lagret
Värde på toppunkt	ENDAST GUI	[tabellfält: alla]	En väljare av det fält som ska användas från det valda lagret. Varje toppunkt tilldelas motsvarande värde för sin ursprungliga funktion.
Använd Z-koordinat för värde på toppunkt	ENDAST GUI	[boolean] Standard: Falsk	Om det är markerat används Z-värdet för vektorlagrets punkter eller polygoner/linjers hörn för att tilldela Z-värdet för vertexmeshlagret. Endast tillgängligt om inmatningslagren är 3D.
Utmatningsformat	`MESH_FORMAT`	[uppräkning] Standard: 2DM	Utdataformat för det genererade lagret • 0 — 2DM • 1 — SELAFIN • 2 — PLY • 3 — Ugrid • 4 — Mike21
Koordinatsystem för utdata Valfritt	`CRS_OUTPUT`	[crs]	Koordinatreferenssystem som ska tilldelas utmatningen
Utmatningsfil	`OUTPUT_MESH`	[mesh] Standard: `[Spara till temporär fil]`	Specifikation av utdatafilen. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Utmatningsfil	`OUTPUT_MESH`	[mesh]	Utgående mesh-lager med datasetvärden som beräknats från vektorlagren.

Python-kod

Algoritm-ID: `native:tinmeshcreation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.9 Verktyg för modellerare

Varning

Dessa verktyg är endast tillgängliga i modelldesignern. De finns inte tillgängliga i verktygslådan Processing Toolbox.

Beräkna uttryck

Den beräknar resultatet av ett QGIS-uttryck och eliminerar behovet av att använda samma uttryck flera gånger i en modell när samma resultat krävs mer än en gång. Dessutom möjliggör den användningsfall som annars inte skulle vara möjliga. Du kan t.ex. generera ett tidsstämpelvärde en gång och använda det flera gånger i modellen, om tidsstämpeln skulle beräknas om varje gång skulle värdena variera under modellens körtid.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatning	`INPUT`	[uttryck]	Uttryck för att beräkna

Utgångar

Etikett	Namn	Typ	Beskrivning
Värde	`OUTPUT`	[Resultatvärde]	Beräknat resultatvärde, datatypen för utdata varierar beroende på det specifika uttryck som används i algoritmen.

Python-kod

Algoritm-ID: `native:beräknauttryck`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Villkorlig gren

Lägger till en villkorlig gren i en modell, vilket gör att delar av modellen kan köras baserat på resultatet av en uttrycksutvärdering. Oftast genom att använda verktygsberoenden för att styra flödet i en modell.

Parametrar

Etikett	Namn	Typ	Beskrivning
Textfält	`BRANCH`	[sträng]	Namn på tillståndet
Textfält	`CONDITION`	[uttryck]	Uttryck som ska utvärderas

Utgångar

Ingen.

Python-kod

Algoritm-ID: `` `native:villkor` ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa katalog

Skapar en ny katalog i ett filsystem. Kataloger skapas rekursivt och skapar alla nödvändiga överordnade kataloger för att konstruera hela den angivna katalogsökvägen. Inga felmeddelanden skickas om katalogen redan finns.

Parametrar

Etikett	Namn	Typ	Beskrivning
Direktory sökväg	`PATH`	[sträng]	Mappsökväg för att skapa

Utgångar

Etikett	Namn	Typ	Beskrivning
Utdata	`OUTPUT`	mapp	Skapad mapp

Python-kod

Algoritm ID: `native:skapadkatalog`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Funktion filter

Filtrerar funktioner från inmatningslagret och omdirigerar dem till en eller flera utgångar. Om du inte känner till några attributnamn som är gemensamma för alla möjliga inmatningslager, är filtrering endast möjlig på objektets geometri och allmänna registreringsmekanismer, t.ex. `$id` och `uuid`.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Det ingående lagret.
Utgångar och filter (en eller flera)	`OUTPUT_<name of the filter>`	[samma som inmatning]	Utmatningslagren med filter (lika många som det finns filter).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utdata (en eller flera)	`original:filter of filter>`	[samma som inmatning]	Utmatningslagren med filtrerade funktioner (lika många som det finns filter).

Python-kod

Algoritm-ID: `native:filter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Filterlager efter typ

Filtrerar lager efter deras typ. Inkommande lager styrs till olika utgångar beroende på om de är vektor- eller rasterlager.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[lager]	Generiskt kartlager

Utgångar

Etikett	Namn	Typ	Beskrivning
Vektorfunktioner Valfritt	`VECTOR`	[vektor]	Ett vektorlager av indata, om kompatibelt
Ett rasterlager Valfritt	`RASTER`	[raster]	En Raster Layer av inmatningen, om kompatibel

Python-kod

Algoritm ID: `native:filterlayersbytype`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Ladda lagret till projektet

Laddar ett lager till det aktuella projektet.

Parametrar

Etikett	Namn	Typ	Beskrivning
Lager	`INPUT`	[lager]	Lager som ska laddas i förklaringen
Laddade lagernamn	`NAME`	[sträng]	Namn på det laddade lagret

Utgångar

Etikett	Namn	Typ	Beskrivning
Lager	`OUTPUT`	[samma som inmatning]	Det (omdöpta) laddade lagret

Python-kod

Algoritm-ID: `native:loadlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Utlös undantag

Utlöser ett undantag och avbryter en modells exekvering. Undantagsmeddelandet kan anpassas och som tillval kan ett uttrycksbaserat villkor anges. Om ett uttrycksbaserat villkor används kommer undantaget endast att utlösas om uttryckets resultat är sant. Ett falskt resultat indikerar att inget undantag kommer att uppstå och att modellkörningen kan fortsätta utan avbrott.

Parametrar

Etikett	Namn	Typ	Beskrivning
Meddelande	`MESSAGE`	[sträng]	Meddelande att visa
Villkor Valfritt	`CONDITION`	[uttryck]	Uttryck som ska utvärderas om det är sant

Utgångar

Ett meddelande i loggpanelen.

Python-kod

Algoritm ID: `native:raiseexception`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Lyft fram meddelandet

Visar ett informationsmeddelande i loggen. Meddelandet kan anpassas och som tillval kan ett uttrycksbaserat villkor anges. Om ett uttrycksvillkor används loggas meddelandet endast om uttrycksresultatet är sant. Ett falskt resultat indikerar att inget meddelande kommer att loggas.

Parametrar

Etikett	Namn	Typ	Beskrivning
Informationsmeddel	`MESSAGE`	[sträng]	Meddelande att visa
Villkor Valfritt	`CONDITION`	[uttryck]	Uttryck som ska utvärderas om det är sant

Utgångar

Ett meddelande i loggpanelen.

Python-kod

Algoritm ID: `native:raisemessage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Höj varningen

Visar ett varningsmeddelande i loggen. Varningsmeddelandet kan anpassas och som tillval kan ett uttrycksbaserat villkor anges. Om ett uttrycksvillkor används loggas varningen endast om uttrycksresultatet är sant. Ett falskt resultat indikerar att ingen varning kommer att loggas.

Parametrar

Etikett	Namn	Typ	Beskrivning
Meddelande	`MESSAGE`	[sträng]	Meddelande att visa
Villkor Valfritt	`CONDITION`	[uttryck]	Uttryck som ska utvärderas om det är sant

Utgångar

Ett meddelande i loggpanelen.

Python-kod

Algoritm-ID: `native:raisewarning`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Byt namn på lager

Byter namn på ett lager.

Parametrar

Etikett	Namn	Typ	Beskrivning
Lager	`INPUT`	[lager]	Lager att byta namn på
Nytt namn	`NAME`	[sträng]	Det nya namnet på lagret

Utgångar

Etikett	Namn	Typ	Beskrivning
Lager	`OUTPUT`	[samma som inmatning]	Det (omdöpta) utgående lagret

Python-kod

Algoritm-ID: `native:renamelayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Spara logg till fil

Sparar modellens exekveringslogg i en fil. Som tillval kan loggen sparas i en HTML-formaterad version.

Parametrar

Etikett	Namn	Typ	Beskrivning
Använd HTML	`USE_HTML`	[Boolean] Standard: Falsk	Använd HTML-formatering

Utgångar

Etikett	Namn	Typ	Beskrivning
&Arkiv	`OUTPUT`	[sträng]	Destination för loggen

Python-kod

Algoritm-ID: `native:savelog`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Ställ in projektvariabel

Ställer in en uttrycksvariabel för det aktuella projektet.

Parametrar

Etikett	Namn	Typ	Beskrivning
Variabelns namn	`NAME`	[sträng]	Variabelns namn
Variabelvärde	`VALUE`	[sträng]	Värde som ska lagras

Utgångar

Ingen.

Python-kod

Algoritm-ID: `native:setprojectvariable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Sammankoppling av strängar

Sammanfogar två strängar till en enda i bearbetningsmodelleraren.

Parametrar

Etikett	Namn	Typ	Beskrivning
Ingång 1	`INPUT_1`	[sträng]	Första strängen
Ingång 2	`INPUT_2`	[sträng]	Andra strängen

Utgångar

Etikett	Namn	Typ	Beskrivning
Sammanlänkning	`CONCATENATION`	[sträng]	Den konkatenerade strängen

Python-kod

Algoritm-ID: `native:strängkonkatenering`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Buffert med variabelt avstånd

Varning

Denna algoritm är föråldrad och kan tas bort när som helst. Använd hellre *Buffra*-algoritmen istället.

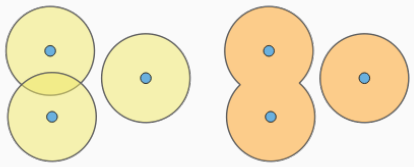
Beräknar ett buffertområde för alla funktioner i ett inmatningslager.

Storleken på bufferten för en viss funktion definieras av ett attribut, vilket gör det möjligt för olika funktioner att ha olika buffertstorlekar.

Se även

Buffra

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlager
Distansfält	`DISTANCE`	[tabellfält: numerisk]	Attribut för buffertens avståndsradie
Segment	`SEGMENTS`	[numerisk: heltal] Standard: *5*	Styr antalet linjesegment som ska användas för att approximera en kvartscirkel när man skapar avrundade offsets.
Lös upp resultatet	`DISSOLVE`	[boolean] Standard: *Falskt*	Välj att lösa upp den sista bufferten, vilket resulterar i en enda funktion som täcker alla ingående funktioner.  Fig. 24.8: Normal och upplöst buffert
Typ av slutlock	`END_CAP_STYLE`	[uppräkning] Standard: Rund	Styr hur radavslut ska hanteras i bufferten. Fig. 24.9: Runda, platta och fyrkantiga lock
Join style	`JOIN_STYLE`	[uppräkning] Standard: Rund	Anger om runda, geringade eller avfasade skarvar ska användas vid förskjutning av hörn i en linje.
Mitergräns	`MITER_LIMIT`	[numerisk: dubbel] Standard: 2,0	Gäller endast för gjutna skarvar och styr det maximala avståndet från offsetkurvan som ska användas när en gjuten skarv skapas.

Utgångar

Etikett	Namn	Typ	Beskrivning
Buffert	`OUTPUT`	[vektor: polygon]	Vektorlager med buffertpolygon.

Python-kod

Algoritm ID: `qgis:variabledistancebuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.10 Analys av nätverk

Serviceområde (från lager)

Returnerar alla kanter eller delar av kanter i ett nätverk som kan nås inom ett visst avstånd eller en viss tid, med utgångspunkt från ett punktlager. Detta möjliggör utvärdering av tillgängligheten i ett nätverk, t.ex. vilka platser jag kan navigera till i ett vägnät utan att kostnaden överstiger ett givet värde (kostnaden kan vara avstånd eller tid).

Observera

Den här algoritmen använder ellipsoidbaserade mätningar och respekterar de aktuella *ellipsoidinställningarna*.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Vektorlager som representerar nätverket	`INPUT`	[vektor: linje]	Linjevektorlager som representerar det nätverk som ska täckas
Vektorlager med startpunkter	`START_POINTS`	[vektor: punkt]	Punktvektorlager vars egenskaper används som startpunkter för att generera serviceområdena
Stigtyp att beräkna	`STRATEGY`	[uppräkning] Standard: 0	Den typ av sökväg som ska beräknas. En av: • 0 — Kortast • 1 — Snabbast
Resekostnad (avstånd för ”Kortast”, tid för ”Snabbast”)	`TRAVEL_COST`	[numerisk: dubbel] Standard: 0,0	Värdet beräknas som ett avstånd (i nätverkslagerenheter) när man letar efter den *kortaste* vägen och som tid (i timmar) för den *snabbaste* vägen.

continues on next page

Table 24.35 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Serviceområde (linjer)	`OUTPUT_LINES`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange det utgående linjelagret för serviceområdet. *Ett av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Serviceområde (gränsnoder)	`OUTPUT`	[vektor: punkt] Standard: `[Hoppa över utdata]`	Ange utmatningspunktskiktet för tjänsteområdets gränsnoder. *Ett av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Funktioner som inte går att routa	`UTGÅNG_INTE_RO`	[vektor: punkt] Standard: `[Hoppa över utdata]`	Ange den utgång som ska användas för att lagra alla ingångsfunktioner som inte kunde routas (t.ex. de som ligger för långt från nätverkslagret). *One of*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Riktningsfält Valfritt	`DIRECTION_FIEL`	[tabellfält: alla]	Det fält som används för att ange riktningar för nätverkets kanter. De värden som används i detta fält anges med de tre parametrarna `Value for forward direction`, `Value for backward direction` och `Value for both directions`. Framåt- och bakåtriktning motsvarar en enkelriktad kant, ”båda riktningarna” anger en dubbelriktad kant. Om en egenskap inte har något värde i detta fält, eller om inget fält har angetts, används standardinställningen för riktning (som anges i parametern `Default direction`).
Värde för riktning framåt Valfritt	`VALUE_FORWARD`	[sträng] Standard: ” (tom sträng)	Värde som anges i fältet direction för att identifiera kanter med en riktning framåt

continues on next page

Table 24.36 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Värde för bakåtriktad riktning Valfritt	VALUE_BACKWARD	[sträng] Standard: ” (tom sträng)	Värde som anges i fältet direction för att identifiera kanter med en bakåtriktad riktning
Värde för båda riktningarna Valfritt	VALUE_BOTH	[sträng] Standard: ” (tom sträng)	Värde som anges i fältet direction för att identifiera dubbelriktade kanter
Förvald riktning	DEFAULT_DIRECT	[uppräkning] Standard: 2	Om en funktion inte har något värde angivet i fältet för riktning eller om inget fält för riktning har angetts, används detta riktningsvärde. Ett av följande: • 0 — Riktning framåt • 1 — Bakåtriktad riktning • 2 — Båda riktningarna
Hastighetsfält Valfritt	SPEED_FIELD	[tabellfält: numerisk]	Fält som anger hastighetsvärdet (i km/h) för nätverkets kanter när man letar efter den snabbaste vägen. Om en funktion inte har något värde i detta fält, eller om inget fält har angetts, används standardhastighetsvärdet (som anges med parametern ”Standardhastighet”).
Förvald hastighet (km/h)	DEFAULT_SPEED	[numerisk: dubbel] Standard: 50,0	Värde som ska användas för att beräkna restiden om inget hastighetsvärde anges för en kant i det angivna fältet
Topologisk tolerans	TOLERANCE	[numerisk: dubbel] Standard: 0,0	Två linjer med noder som ligger närmare varandra än den angivna toleransen anses vara anslutna
Inkludera övre/nedre gränspunkter	INCLUDE_BOUNDS	[boolean] Standard: Falsk	Skapar en punktlagerutgång med två punkter för varje kant vid gränserna för serviceområdet. Den ena punkten är början på kanten, den andra är slutet.
Maximalt punktavstånd från nätverket Valfritt	POINT_TOLERANC	[numerisk: dubbel] Standard: Ej inställd	Anger en valfri gräns för avståndet från punkterna till nätverkslagret. Om en punkt ligger längre bort från nätverket än detta avstånd kommer den att behandlas som icke-routerbar. Om detta inte anges kommer slutpunkterna att knäppas till den närmaste punkten på nätverkslagret, oavsett hur långt bort från nätverket de faktiskt befinner sig.

Utgångar

Etikett	Namn	Typ	Beskrivning
Serviceområde (gränsnoder)	`OUTPUT`	[vektor: punkt]	Utgångspunktens lager med serviceområdets gränsnoder.
Serviceområde (linjer)	`OUTPUT_LINES`	[vektor: linje]	Linjelager som representerar de delar av nätverket som kan betjänas av startpunkterna, för den givna kostnaden.
Funktioner som inte kan dirigeras	`UTGÅNG_INTE_RO`	[vektor: punkt]	En valfri utgång som kommer att användas för att lagra alla ingångsfunktioner som inte kunde routas (t.ex. de som ligger för långt från nätverkslagret).

Python-kod

Algoritm ID: `qgis:serviceareafromlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Serviceområde (från punkt)

Returnerar alla kanter eller delar av kanter i ett nätverk som kan nås inom ett givet avstånd eller en given tid, med utgångspunkt från en punktfunktion. Detta gör det möjligt att utvärdera tillgängligheten i ett nätverk, t.ex. vilka platser jag kan navigera till i ett vägnät utan att kostnaden överstiger ett visst värde (kostnaden kan vara avstånd eller tid).

Observera

Den här algoritmen använder ellipsoidbaserade mätningar och respekterar de aktuella *ellipsoidinställningarna*.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Vektorlager som representerar nätverket	`INPUT`	[vektor: linje]	Linjevektorlager som representerar det nätverk som ska täckas
Stigtyp att beräkna	`STRATEGY`	[uppräkning] Standard: 0	Den typ av sökväg som ska beräknas. En av: • 0 — Kortast • 1 — Snabbast

continues on next page

Table 24.37 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Startpunkt	`START_POINT`	[koordinater]	Koordinat för punkten för att beräkna serviceområdet runt. Tryck på knappen … bredvid alternativet och klicka på duken för att fylla parametern med den klickade punktkoordinaten.
Resekostnad (avstånd för ”Kortast”, tid för ”Snabbast”)	`TRAVEL_COST`	[numerisk: dubbel] Standard: 0,0	Värdet beräknas som ett avstånd (i nätverkslagerenheter) när man letar efter den *kortaste* vägen och som tid (i timmar) för den *snabbaste* vägen.
Serviceområde (linjer) Valfritt	`OUTPUT_LINES`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange det utgående linjelagret för serviceområdet. *Ett av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Serviceområde (gränsnoder) Valfritt	`OUTPUT`	[vektor: punkt] Standard: `[Hoppa över utdata]`	Ange utmatningspunktskiktet för tjänsteområdets gränsnoder. *Ett av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Riktningsfält Valfritt	`DIRECTION_FIEL`	[tabellfält: alla]	Det fält som används för att ange riktningar för nätverkets kanter. De värden som används i detta fält anges med de tre parametrarna `Value for forward direction`, `Value for backward direction` och `Value for both directions`. Framåt- och bakåtriktning motsvarar en enkelriktad kant, ”båda riktningarna” anger en dubbelriktad kant. Om en egenskap inte har något värde i detta fält, eller om inget fält har angetts, används standardinställningen för riktning (som anges i parametern `Default direction`).
Värde för riktning framåt Valfritt	`VALUE_FORWARD`	[sträng] Standard: ” (tom sträng)	Värde som anges i fältet direction för att identifiera kanter med en riktning framåt
Värde för bakåtriktad riktning Valfritt	`VALUE_BACKWARD`	[sträng] Standard: ” (tom sträng)	Värde som anges i fältet direction för att identifiera kanter med en bakåtriktad riktning

continues on next page

Table 24.38 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Värde för båda riktningarna Valfritt	VALUE_BOTH	[sträng] Standard: ” (tom sträng)	Värde som anges i fältet direction för att identifiera dubbelriktade kanter
Förvald riktning	DEFAULT_DIRECT	[uppräkning] Standard: 2	Om en funktion inte har något värde angivet i fältet för riktning eller om inget fält för riktning har angetts, används detta riktningsvärde. Ett av följande: • 0 — Riktning framåt • 1 — Bakåtriktad riktning • 2 — Båda riktningarna
Hastighetsfält Valfritt	SPEED_FIELD	[tabellfält: numerisk]	Fält som anger hastighetsvärdet (i km/h) för nätverkets kanter när man letar efter den snabbaste vägen. Om en funktion inte har något värde i detta fält, eller om inget fält har angetts, används standardhastighetsvärdet (som anges med parametern ”Standardhastighet”).
Förvald hastighet (km/h)	DEFAULT_SPEED	[numerisk: dubbel] Standard: 50,0	Värde som ska användas för att beräkna restiden om inget hastighetsvärde anges för en kant i det angivna fältet
Topologisk tolerans	TOLERANCE	[numerisk: dubbel] Standard: 0,0	Två linjer med noder som ligger närmare varandra än den angivna toleransen anses vara anslutna
Maximalt punktavstånd från nätverket Valfritt	POINT_TOLERANC	[numerisk: dubbel] Standard: 0,0	Anger en valfri gräns för avståndet från startpunkten till nätverkslagret. Om punkten ligger längre bort från nätverket än detta avstånd kommer ett fel att uppstå. Om den inte anges kommer punkten att knäppas till den närmaste punkten på nätverkslagret, oavsett hur långt bort från nätverket den faktiskt är.
Inkludera övre/nedre gränspunkter	INCLUDE_BOUNDS	[boolean] Standard: Falsk	Skapar en punktlagerutgång med två punkter för varje kant vid gränserna för serviceområdet. Den ena punkten är början på kanten, den andra är slutet.

Utgångar

Etikett	Namn	Typ	Beskrivning
Serviceområde (gränsnoder)	OUTPUT	[vektor: punkt]	Utgångspunktens lager med serviceområdets gränsnoder.
Serviceområde (linjer)	OUTPUT_LINES	[vektor: linje]	Linjelager som representerar de delar av nätverket som kan betjänas av startpunkten, för den givna kostnaden.

Python-kod

Algoritm-ID: `native:tjänsteleverantörfrånpunkt`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Kortaste vägen (lager till punkt)

Beräknar de optimala (kortaste eller snabbaste) rutterna från flera startpunkter som definieras av ett vektorlager och en given slutpunkt.

Observera

Den här algoritmen använder ellipsoidbaserade mätningar och respekterar de aktuella *ellipsoidinställningarna*.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Vektorlager som representerar nätverket	`INPUT`	[vektor: linje]	Linjevektorlager som representerar det nätverk som ska täckas
Stigtyp att beräkna	`STRATEGY`	[uppräkning] Standard: 0	Den typ av sökväg som ska beräknas. En av: • 0 — Kortast • 1 — Snabbast
Vektorlager med startpunkter	`START_POINTS`	[vektor: punkt]	Punktvektorlager vars egenskaper används som startpunkter för rutterna
Slutpunkt (x, y)	`END_POINT`	[koordinater]	Punktfunktion som representerar slutpunkten för rutterna Tryck på knappen … bredvid alternativet och klicka på duken för att fylla parametern med den klickade punktkoordinaten.
Kortaste vägen	`OUTPUT`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange utmatningslinjeskiktet för de kortaste vägarna. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

continues on next page

Table 24.39 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Funktioner som inte går att routa Valfritt	`UTGÅNG_INTE_RO`	[vektor: punkt] Standard: `[Hoppa över utdata]`	Ange den utgång som ska användas för att lagra alla ingångsfunktioner som inte kunde routas (t.ex. de som ligger för långt från nätverkslagret). *One of*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Riktningsfält Valfritt	`DIRECTION_FIEL`	[tabellfält: alla]	Det fält som används för att ange riktningar för nätverkets kanter. De värden som används i detta fält anges med de tre parametrarna `Value for forward direction`, `Value for backward direction` och `Value for both directions`. Framåt- och bakåtriktning motsvarar en enkelriktad kant, ”båda riktningarna” anger en dubbelriktad kant. Om en egenskap inte har något värde i detta fält, eller om inget fält har angetts, används standardinställningen för riktning (som anges i parametern `Default direction`).
Värde för riktning framåt Valfritt	`VALUE_FORWARD`	[sträng] Standard: ” (tom sträng)	Värde som anges i fältet direction för att identifiera kanter med en riktning framåt
Värde för bakåtriktad riktning Valfritt	`VALUE_BACKWARD`	[sträng] Standard: ” (tom sträng)	Värde som anges i fältet direction för att identifiera kanter med en bakåtriktad riktning
Värde för båda riktningarna Valfritt	`VALUE_BOTH`	[sträng] Standard: ” (tom sträng)	Värde som anges i fältet direction för att identifiera dubbelriktade kanter
Förvald riktning	`DEFAULT_DIRECT`	[uppräkning] Standard: 2	Om en funktion inte har något värde angivet i fältet för riktning eller om inget fält för riktning har angetts, används detta riktningsvärde. Ett av följande: • 0 — Riktning framåt • 1 — Bakåtriktad riktning • 2 — Båda riktningarna

continues on next page

Table 24.40 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Hastighetsfält Valfritt	`SPEED_FIELD`	[tabellfält: numerisk]	Fält som anger hastighetsvärdet (i `km/h`) för nätverkets kanter när man letar efter den snabbaste vägen. Om en funktion inte har något värde i detta fält, eller om inget fält har angetts, används standardhastighetsvärdet (som anges med parametern ”Standardhastighet”).
Förvald hastighet (km/h)	`DEFAULT_SPEED`	[numerisk: dubbel] Standard: 50,0	Värde som ska användas för att beräkna restiden om inget hastighetsvärde anges för en kant i det angivna fältet
Topologisk tolerans	`TOLERANCE`	[numerisk: dubbel] Standard: 0,0	Två linjer med noder som ligger närmare varandra än den angivna toleransen anses vara anslutna
Maximalt punktavstånd från nätverket Valfritt	`POINT_TOLERANC`	[numerisk: dubbel] Standard: Ej inställd	Anger en valfri gräns för avståndet från start- och slutpunkterna till nätverkslagret. Om en startpunkt ligger längre bort från nätverket än detta avstånd kommer den att behandlas som icke-routerbar. Om slutpunkten ligger längre bort från nätverket än detta avstånd kommer ett fel att uppstå. Om detta inte anges kommer punkterna att knäppas till den närmaste punkten på nätverkslagret, oavsett hur långt bort från nätverket de faktiskt befinner sig.

Utgångar

Etikett	Namn	Typ	Beskrivning
Kortaste vägen	`OUTPUT`	[vektor: linje]	Linjelager med den kortaste eller snabbaste vägen från var och en av startpunkterna till slutpunkten
Funktioner som inte kan dirigeras	`UTGÅNG_INTE_RO`	[vektor: punkt]	Ett valfritt utdatalager som används för att lagra alla inmatningsfunktioner som inte kunde dirigeras (t.ex. de som ligger för långt från nätverkslagret).

Python-kod

Algoritm-ID: `original:kortastepathlayertopoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Kortaste vägen (punkt till lager)

Beräknar de optimala (kortaste eller snabbaste) rutterna mellan en given startpunkt och flera slutpunkter som definieras av ett punktvektorlager.

Observera

Den här algoritmen använder ellipsoidbaserade mätningar och respekterar de aktuella *ellipsoidinställningarna*.

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Vektorlager som representerar nätverket	`INPUT`	[vektor: linje]	Linjevektorlager som representerar det nätverk som ska täckas
Stigtyp att beräkna	`STRATEGY`	[uppräkning] Standard: 0	Den typ av sökväg som ska beräknas. En av: • 0 — Kortast • 1 — Snabbast
Startpunkt	`START_POINT`	[koordinater]	Punktfunktion som representerar startpunkten för rutterna Tryck på knappen … bredvid alternativet och klicka på duken för att fylla parametern med den klickade punktkoordinaten.
Vektorlager med ändpunkter	`END_POINTS`	[vektor: punkt]	Punktvektorlager vars egenskaper används som slutpunkter för rutterna
Kortaste vägen	`OUTPUT`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange utmatningslinjeskiktet för de kortaste vägarna. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Funktioner som inte går att routa	`UTGÅNG_INTE_RO` Valfritt	[vektor: punkt] Standard: `[Hoppa över utdata]`	Ange den utgång som ska användas för att lagra alla ingångsfunktioner som inte kunde routas (t.ex. de som ligger för långt från nätverkslagret). *One of*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Riktningsfält Valfritt	`DIRECTION_FIEL`	[tabellfält: alla]	Det fält som används för att ange riktningar för nätverkets kanter. De värden som används i detta fält anges med de tre parametrarna `Value for forward direction`, `Value for backward direction` och `Value for both directions`. Framåt- och bakåtriktning motsvarar en enkelriktad kant, ”båda riktningarna” anger en dubbelriktad kant. Om en egenskap inte har något värde i detta fält, eller om inget fält har angetts, används standardinställningen för riktning (som anges i parametern `Default direction`).
Värde för riktning framåt Valfritt	`VALUE_FORWARD`	[sträng] Standard: ” (tom sträng)	Värde som anges i fältet direction för att identifiera kanter med en riktning framåt
Värde för bakåtriktad riktning Valfritt	`VALUE_BACKWARD`	[sträng] Standard: ” (tom sträng)	Värde som anges i fältet direction för att identifiera kanter med en bakåtriktad riktning
Värde för båda riktningarna Valfritt	`VALUE_BOTH`	[sträng] Standard: ” (tom sträng)	Värde som anges i fältet direction för att identifiera dubbelriktade kanter
Förvald riktning	`DEFAULT_DIRECT`	[uppräkning] Standard: 2	Om en funktion inte har något värde angivet i fältet för riktning eller om inget fält för riktning har angetts, används detta riktningsvärde. Ett av följande: • 0 — Riktning framåt • 1 — Bakåtriktad riktning • 2 — Båda riktningarna
Hastighetsfält Valfritt	`SPEED_FIELD`	[tabellfält: numerisk]	Fält som anger hastighetsvärdet (i `km/h`) för nätverkets kanter när man letar efter den snabbaste vägen. Om en funktion inte har något värde i detta fält, eller om inget fält har angetts, används standardhastighetsvärdet (som anges med parametern ”Standardhastighet”).
Förvald hastighet (km/h)	`DEFAULT_SPEED`	[numerisk: dubbel] Standard: 50,0	Värde som ska användas för att beräkna restiden om inget hastighetsvärde anges för en kant i det angivna fältet
Topologisk tolerans	`TOLERANCE`	[numerisk: dubbel] Standard: 0,0	Två linjer med noder som ligger närmare varandra än den angivna toleransen anses vara anslutna

Maximalt punktavstånd från nätverket Valfritt	POINT_TOLERANC	[numerisk: dubbel] Standard: Ej inställd	Anger en valfri gräns för avståndet från start- och slutpunkterna till nätverkslagret. Om startpunkten ligger längre bort från nätverket än detta avstånd kommer ett fel att uppstå. Om en slutpunkt ligger längre bort från nätverket än detta avstånd kommer den att behandlas som icke-routerbar. Om detta inte anges kommer punkterna att knäppas till den närmaste punkten på nätverkslagret, oavsett hur långt bort från nätverket de faktiskt befinner sig.

Utgångar

Etikett	Namn	Typ	Beskrivning
Kortaste vägen	OUTPUT	[vektor: linje]	Linjelager med den kortaste eller snabbaste vägen från var och en av startpunkterna till slutpunkten
Funktioner som inte kan dirigeras	UTGÅNG_INTE_RO	[vektor: punkt]	Ett valfritt utdatalager som används för att lagra alla inmatningsfunktioner som inte kunde dirigeras (t.ex. de som ligger för långt från nätverkslagret).

Python-kod

Algoritm-ID: `native:kortastevägspinttolager`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Kortaste vägen (punkt till punkt)

Beräknar den optimala (kortaste eller snabbaste) rutten mellan en given startpunkt och en given slutpunkt.

Observera

Den här algoritmen använder ellipsoidbaserade mätningar och respekterar de aktuella *ellipsoidinställningarna*.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Vektorlager som representerar nätverket	`INPUT`	[vektor: linje]	Linjevektorlager som representerar det nätverk som ska täckas
Stigtyp att beräkna	`STRATEGY`	[uppräkning] Standard: 0	Den typ av sökväg som ska beräknas. En av: • 0 — Kortast • 1 — Snabbast
Startpunkt (x, y)	`START_POINT`	[koordinater]	Punktfunktion som representerar startpunkten för rutterna. Tryck på knappen … bredvid alternativet och klicka på duken för att fylla parametern med den klickade punktkoordinaten.
Slutpunkt (x, y)	`END_POINT`	[koordinater]	Punktfunktion som representerar slutpunkten för rutterna. Tryck på knappen … bredvid alternativet och klicka på duken för att fylla parametern med den klickade punktkoordinaten.
Kortaste vägen	`OUTPUT`	[vektor: linje]	Ange utmatningslinjeskiktet för de kortaste vägarna. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Riktningsfält Valfritt	`DIRECTION_FIEL`	[tabellfält: alla]	Det fält som används för att ange riktningar för nätverkets kanter. De värden som används i detta fält anges med de tre parametrarna `Value for forward direction`, `Value for backward direction` och `Value for both directions`. Framåt- och bakåtriktning motsvarar en enkelriktad kant, ”båda riktningarna” anger en dubbelriktad kant. Om en egenskap inte har något värde i detta fält, eller om inget fält har angetts, används standardinställningen för riktning (som anges i parametern `Default direction`).
Värde för riktning framåt Valfritt	`VALUE_FORWARD`	[sträng] Standard: ” (tom sträng)	Värde som anges i fältet direction för att identifiera kanter med en riktning framåt

continues on next page

Table 24.44 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Värde för bakåtriktad riktning Valfritt	VALUE_BACKWARD	[sträng] Standard: ” (tom sträng)	Värde som anges i fältet direction för att identifiera kanter med en bakåtriktad riktning
Värde för båda riktningarna Valfritt	VALUE_BOTH	[sträng] Standard: ” (tom sträng)	Värde som anges i fältet direction för att identifiera dubbelriktade kanter
Förvald riktning	DEFAULT_DIRECT	[uppräkning] Standard: 2	Om en funktion inte har något värde angivet i fältet för riktning eller om inget fält för riktning har angetts, används detta riktningsvärde. Ett av följande: • 0 — Riktning framåt • 1 — Bakåtriktad riktning • 2 — Båda riktningarna
Hastighetsfält Valfritt	SPEED_FIELD	[tabellfält: numerisk]	Fält som anger hastighetsvärdet (i km/h) för nätverkets kanter när man letar efter den snabbaste vägen. Om en funktion inte har något värde i detta fält, eller om inget fält har angetts, används standardhastighetsvärdet (som anges med parametern ”Standardhastighet”).
Förvald hastighet (km/h)	DEFAULT_SPEED	[numerisk: dubbel] Standard: 50,0	Värde som ska användas för att beräkna restiden om inget hastighetsvärde anges för en kant i det angivna fältet
Topologisk tolerans	TOLERANCE	[numerisk: dubbel] Standard: 0,0	Två linjer med noder som ligger närmare varandra än den angivna toleransen anses vara anslutna
Maximalt punktavstånd från nätverket Valfritt	POINT_TOLERANC	[numerisk: dubbel] Standard: Ej inställd	Anger en valfri gräns för avståndet från start- och slutpunkterna till nätverkslagret. Om någon av punkterna ligger längre bort från nätverket än detta avstånd kommer ett fel att uppstå. Om detta inte anges kommer punkterna att snäppas till den närmaste punkten på nätverkslagret, oavsett hur långt bort från nätverket de faktiskt är.

Utgångar

Etikett	Namn	Typ	Beskrivning
Kortaste vägen	OUTPUT	[vektor: linje]	Linjelager med den kortaste eller snabbaste vägen från varje startpunkt till slutpunkten

Python-kod

Algoritm-id: `native:shortestpathpointtopoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.11 Tomter

Bar plot

Denna algoritm skapar ett stapeldiagram från en kategori och lagerfält.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Namn på kategorifält	`NAME_FIELD`	[tabellfält: alla]	Kategoriskt fält att använda för gruppering av staplarna (X-axel)
Värdefält	`VALUE_FIELD`	[tabellfält: alla]	Värde som ska användas för diagrammet (Y-axeln).
** Stapeldiagram**	`OUTPUT`	[html] Standard: `[Spara till temporär fil]`	Ange HTML-filen för diagrammet. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
** Stapeldiagram**	`OUTPUT`	[html]	HTML-fil med plotten. Tillgänglig i *Processing* ► *Result Viewer*.

Python-kod

Algoritm-ID: `qgis:barplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Låddiagram

Skapar en boxplot från ett kategorifält och ett numeriskt lagerfält.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Fält för kategorinamn	`NAME_FIELD`	[tabellfält: alla]	Kategoriskt fält att använda för att gruppera boxarna (X-axel)
Värdefält	`VALUE_FIELD`	[tabellfält: alla]	Värde som ska användas för diagrammet (Y-axeln).
Ytterligare statistiska rader	`MSD`	[uppräkning] Standard: 0	Ytterligare statistikinformation att lägga till i handlingen. En av: • 0 — Visa medelvärde • 1 — Visa standardavvikelse • 2 — Visar inte medelvärde och standardavvikelse
Box plot	`OUTPUT`	[html] Standard: `[Spara till temporär fil]`	Ange HTML-filen för diagrammet. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Box plot	`OUTPUT`	[html]	HTML-fil med plotten. Tillgänglig i *Processing* ► *Result Viewer*.

Python-kod

Algoritm-ID: `qgis:boxplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Diagram över medelvärde och standardavvikelse

Skapar ett boxdiagram med medelvärden och standardavvikelser.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningstabell	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Fält för kategorinamn	`NAME_FIELD`	[tabellfält: alla]	Kategoriskt fält att använda för att gruppera boxarna (X-axel)
Värdefält	`VALUE_FIELD`	[tabellfält: numerisk]	Värde som ska användas för diagrammet (Y-axeln).
Plot	`OUTPUT`	[html] Standard: `[Spara till temporär fil]`	Ange HTML-filen för diagrammet. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Plot	`OUTPUT`	[html]	HTML-fil med plotten. Tillgänglig i *Processing* ► *Result Viewer*.

Python-kod

Algoritm-ID: `qgis:medelvärdes- och standardavvikelseplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Polär plot

Genererar en polär plot baserat på värdet i ett inputvektorlager.

Två fält måste anges som parametrar: ett som definierar kategorin för varje objekt (för att gruppera objekten) och ett annat med den variabel som ska plottas (detta måste vara en numerisk variabel).

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Fält för kategorinamn	`NAME_FIELD`	[tabellfält: alla]	Kategoriskt fält som ska användas för att gruppera funktionerna (X-axel)
Värdefält	`VALUE_FIELD`	[tabellfält: numerisk]	Värde som ska användas för diagrammet (Y-axeln).
Polar plot	`OUTPUT`	[html] Standard: `[Spara till temporär fil]`	Ange HTML-filen för diagrammet. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Polar plot	`OUTPUT`	[html]	HTML-fil med plotten. Tillgänglig i *Processing* ► *Result Viewer*.

Python-kod

Algoritm-ID: `qgis:polarplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Histogram för rasterlager

Genererar ett histogram med värdena i ett rasterlager.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatat rasterlager
Bandnummer	`BAND`	[rasterband]	Rasterband som ska användas för histogrammet
antal fack	`BINS`	[numerisk: heltal] Standard: 10	Antalet bins som ska användas i histogrammet (X-axeln). Minimum 2.
Histogram	`OUTPUT`	[html] Standard: `[Spara till temporär fil]`	Ange HTML-filen för diagrammet. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Histogram	`OUTPUT`	[html]	HTML-fil med plotten. Tillgänglig i *Processing* ► *Result Viewer*.

Python-kod

Algoritm ID: `qgis:rasterlayerhistogram`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Histogram för vektorlager

Skapar ett histogram med värdena för attributet i ett vektorlager.

Det attribut som ska användas för att beräkna histogrammet måste vara numeriskt.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlager
Attribut	`FIELD`	[tabellfält: numerisk]	Värde som ska användas för diagrammet (Y-axeln).
antal fack	`BINS`	[numerisk: heltal] Standard: 10	Antalet bins som ska användas i histogrammet (X-axeln). Minimum 2.
Histogram	`OUTPUT`	[html] Standard: `[Spara till temporär fil]`	Ange HTML-filen för diagrammet. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Histogram	`OUTPUT`	[html]	HTML-fil med plotten. Tillgänglig i *Processing* ► *Result Viewer*.

Python-kod

Algoritm-ID: `qgis:vektorlagerhistogram`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Vektorlager spridningsdiagram

Skapar ett enkelt `X` - `Y` spridningsdiagram för ett vektorlager.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
X-attribut	`XFIELD`	[tabellfält: alla]	Fält som ska användas för X-axeln
Y-attribut	`YFIELD`	[tabellfält: alla]	Fält som ska användas för Y-axeln
Scatterplot	`OUTPUT`	[html] Standard: `[Spara till temporär fil]`	Ange HTML-filen för diagrammet. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Scatterplot	`OUTPUT`	[html]	HTML-fil med plotten. Tillgänglig i *Processing* ► *Result Viewer*.

Python-kod

Algoritm-ID: `qgis:vectorlayerscatterplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Vektorlager spridningsdiagram 3D

Skapar ett 3D-spridningsdiagram för ett vektorlager.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
X-attribut	`XFIELD`	[tabellfält: alla]	Fält som ska användas för X-axeln
Y-attribut	`YFIELD`	[tabellfält: alla]	Fält som ska användas för Y-axeln
Z-attribut	`ZFIELD`	[tabellfält: alla]	Fält som ska användas för Z-axeln
Histogram	`OUTPUT`	[html] Standard: `[Spara till temporär fil]`	Ange HTML-filen för diagrammet. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Histogram	`OUTPUT`	[html]	HTML-fil med plotten. Tillgänglig i *Processing* ► *Result Viewer*.

Python-kod

Algoritm-ID: `qgis:scatter3dplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.12 Konvertering av punktmoln

Observera

Dessa algoritmer är endast tillgängliga om QGIS använder PDAL-biblioteket version 2.5.0 eller nyare.

Konvertera format

Konverterar ett punktmoln till ett annat filformat, t.ex. skapar en komprimerad `.LAZ`.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[punktmoln]	Inmatat punktmolnslager som ska konverteras
Konverterad	`OUTPUT`	[punktmoln] Standard: `[Spara till temporär fil]`	Ange den punktmolnsfil som ska användas som utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Konverterad	`OUTPUT`	[punktmoln]	Utmatning av punktmolnslager i ett modifierat filformat. Format som stöds för närvarande är `.LAS`, `.LAZ`, `.COPC.LAZ` och `.VPC`.

Python-kod

Algoritm-ID: `pdal:convertformat`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exportera till raster

Exporterar punktmolnsdata till ett 2D-rasterrutnät med cellstorlek med given upplösning och skriver värden från det angivna attributet.

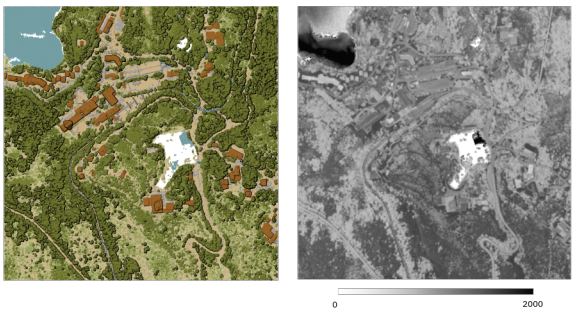


Fig. 24.10: Rasterutmatning med hjälp av punkternas attribut Intensity

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[punktmoln]	Inmatning av punktmolnslager för export
Attribut	`ATTRIBUTE`	[fält] [uppräkning]	Ett fält i punktmolnslagret för att extrahera värdena från
Upplösning av täthetsrastret	`RESOLUTION`	[numerisk: dubbel] Standard: 1,0	Cellstorlek på utmatningsrastret
Storlek på plattor för parallella körningar	`TILE_SIZE`	[numerisk: heltal] Standard: 1000	
Exporterad	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange den rasterfil som data ska exporteras till. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Filteruttryck Valfritt	`FILTER_EXPRESS`	[uttryck]	Ett *PDAL-uttryck* för att välja en delmängd av funktioner i punktmolnsdata
Klippningens omfattning Valfritt	`FILTER_EXTENT`	[omfattning]	En kartutsträckning för att välja en delmängd av funktioner i punktmolnsdata Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
X-origo för en bricka för parallella körningar Valfritt	`ORIGIN_X`	[numerisk: dubbel]	
Y-origo för en kakelplatta för parallella körningar Valfritt	`ORIGIN_Y`	[numerisk: dubbel]	

Utgångar

Etikett	Namn	Typ	Beskrivning
Exporterad	`OUTPUT`	[raster]	Utgående rasterlager som funktioner i punktmolnslagret exporteras till. Format som stöds för närvarande är `.TIF`.

Python-kod

Algoritm-ID: `pdal:exportraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exportera raster (med hjälp av triangulering)

Exporterar punktmolnsdata till ett 2D-rasterraster med hjälp av en triangulering av punkter och sedan interpolering av cellvärden från trianglar.

Observera

Den här algoritmen kan vara långsammare om du har att göra med en stor datamängd. Om ditt punktmoln är tätt kan du exportera dina markpunkter som ett raster med hjälp av algoritmen *Exportera till raster*.

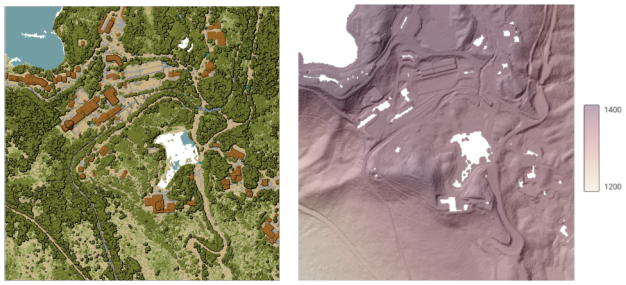


Fig. 24.11: Terrängraster som genereras genom triangulering av punktmoln

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[punktmoln]	Inmatning av punktmolnslager för export
Upplösning av täthetsrastret	`RESOLUTION`	[numerisk: dubbel] Standard: 1,0	Cellstorlek på utmatningsrastret
Storlek på plattor för parallella körningar	`TILE_SIZE`	[numerisk: heltal] Standard: 1000	
Exporterad	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange den rasterfil som data ska exporteras till. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Filteruttryck Valfritt	`FILTER_EXPRESS`	[uttryck]	Ett *PDAL-uttryck* för att välja en delmängd av funktioner i punktmolnsdata
Klippningens omfattning Valfritt	`FILTER_EXTENT`	[omfattning]	En kartutsträckning för att välja en delmängd av funktioner i punktmolnsdata Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
X-origo för en bricka för parallella körningar Valfritt	`ORIGIN_X`	[numerisk: dubbel]	
Y-origo för en kakelplatta för parallella körningar Valfritt	`ORIGIN_Y`	[numerisk: dubbel]	

Utgångar

Etikett	Namn	Typ	Beskrivning
Exporterad (med hjälp av triangulering)	`OUTPUT`	[raster]	Utgående rasterlager som funktioner i punktmolnslagret exporteras till. Format som stöds för närvarande är `.TIF`.

Python-kod

Algoritm-id: `pdal:exportrastertin`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exportera till vektor

Exporterar punktmolnsdata till ett vektorlager med 3D-punkter (ett GeoPackage), eventuellt med extra attribut.

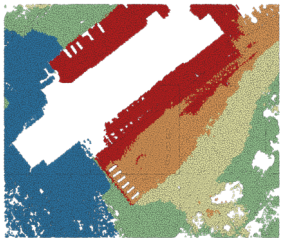


Fig. 24.12: Exportera punktmoln (markpunkter) till ett vektorlager som stylas baserat på höjden

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[punktmoln]	Inmatning av punktmolnslager för export
Attribut Valfritt	`ATTRIBUTE`	[fält] [lista]	Ett eller flera fält i punktmolnslagret som ska exporteras med punkterna.
Exporterad	`OUTPUT`	[vektor] Standard: `[Spara till temporär fil]`	Ange den vektorfil som data ska exporteras till. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Filteruttryck Valfritt	`FILTER_EXPRESS`	[uttryck]	Ett *PDAL-uttryck* för att välja en delmängd av funktioner i punktmolnsdata
Klippningens omfattning Valfritt	`FILTER_EXTENT`	[omfattning]	En kartutsträckning för att välja en delmängd av funktioner i punktmolnsdata Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`

Utgångar

Etikett	Namn	Typ	Beskrivning
Exporterad	`OUTPUT`	[vektor]	Utgångsvektorlager som funktioner i punktmolnslagret exporteras till. Format som stöds för närvarande är `.GPKG`.

Python-kod

Algoritm-id: `pdal:exportvector`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.13 Hantering av data från punktmoln

Observera

Dessa algoritmer är endast tillgängliga om QGIS använder PDAL-biblioteket version 2.5.0 eller nyare.

Tilldela projektion

Tilldelar ett koordinatreferenssystem till ett punktmolnslager, om det saknas eller är felaktigt. Ett nytt lager skapas.

Se även

Reprojicera

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[punktmoln]	Molnlager med inmatningspunkter som ska tilldelas en CRS
Önskad CRS	`CRS`	[crs]	CRS för att applicera på lagret
Utdatalager	`OUTPUT`	[punktmoln] Standard: `[Spara till temporär fil]`	Ange den punktmolnsfil som ska användas som utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Konverterad	`OUTPUT`	[punktmoln]	Utmatning av punktmolnslager med en ny CRS. Format som stöds för närvarande är `.LAS`, `.LAZ`, `.COPC.LAZ` och `.VPC`.

Python-kod

Algoritm-ID: `pdal:assignprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa ett virtuellt punktmoln (VPC)

Skapar ett *virtuellt punktmoln (VPC)* från ingående punktmolnsdata.

Om du låter de valfria parametrarna vara avmarkerade kommer VPC-filen att byggas mycket snabbt eftersom algoritmen endast läser metadata för inmatningsfiler. Om någon av de valfria parametrarna är inställd kommer algoritmen att läsa alla punkter, vilket kan ta lite tid.

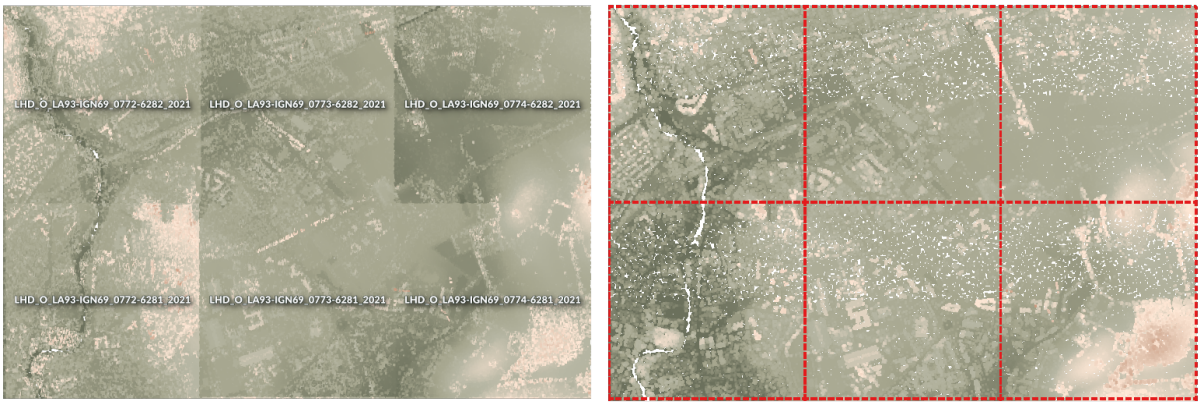


Fig. 24.13: Generering av virtuellt punktmoln med översikt från en uppsättning punktmolnsplattor

Se även

Sammanfoga

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`LAYERS`	[punktmoln] [lista]	Inmatade punktmolnslager som ska kombineras i ett virtuellt punktmolnslager
Beräkna polygon för avgränsning	`BOUNDARY`	[boolean] Standard: Falsk	Ställ in till True för att visa de exakta gränserna för data (snarare än bara rektangulär utsträckning)
Beräkna statistik	`STATISTICS`	[boolean] Standard: Falsk	Sätt till True för att förstå värdeintervallen för de olika attributen
Bygga ett översiktligt punktmoln	`OVERVIEW`	[boolean] Standard: Falsk	Genererar ett enda ”förtunnat” punktmoln av alla indata (använder endast var 1000:e punkt från originaldata). Det översiktliga punktmolnet skapas bredvid VPC-filen - t.ex. för `mydata.vpc` skulle det översiktliga punktmolnet få namnet `mydata-overview.copc.laz`.
Virtuellt punktmoln	`OUTPUT`	[punktmoln] Standard: `[Spara till temporär fil]`	Ange den punktmolnsfil som data ska byggas in i. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Virtuellt punktmoln	`OUTPUT`	[raster]	Utgående punktmolnslager som kombinerar alla indata, som en virtuell fil.

Python-kod

Algoritm-ID: `pdal:virtualpointcloud`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Klipp

Klipper ett punktmolnslager genom ett polygonlager så att det resulterande punktmolnet endast innehåller punkter inom polygonerna.

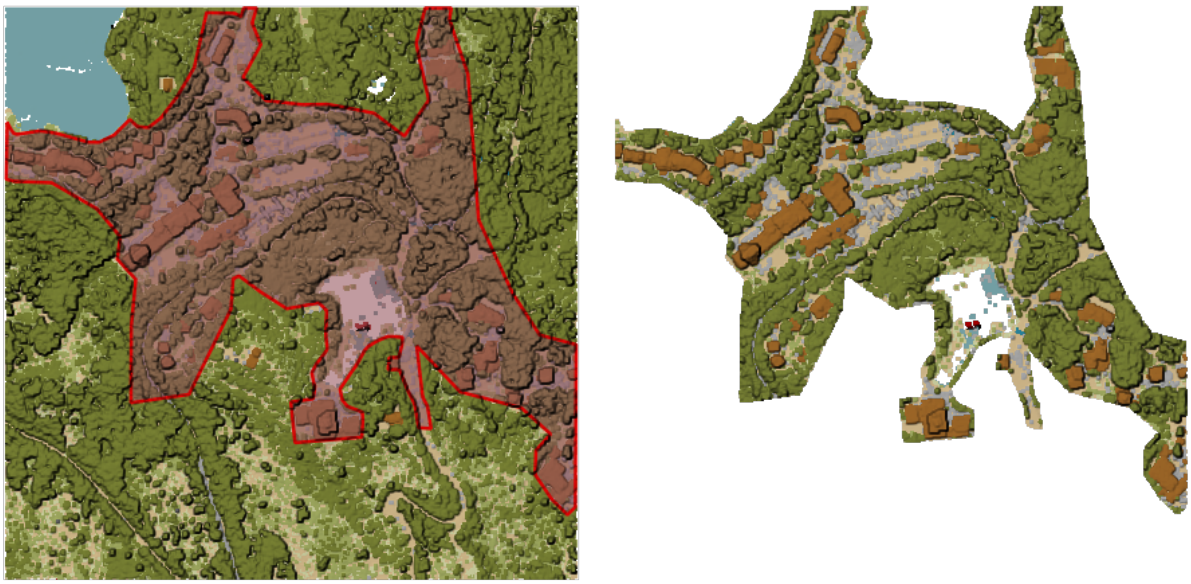


Fig. 24.14: Klippning av ett inmatat punktmolnslager med en polygontäckning

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[punktmoln]	Inmatat punktmolnslager som ska klippas
Klippning av polygoner	`OVERLAY`	[vektor: polygon]	Polygonvektorlager att använda som täckning för att klippa punkterna
Klippt	`OUTPUT`	[punktmoln] Standard: `[Spara till temporär fil]`	Ange den punktmolnsfil som de klippta punkterna ska exporteras till. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Filteruttryck Valfritt	`FILTER_EXPRESS`	[uttryck]	Ett *PDAL-uttryck* för att välja en delmängd av funktioner i punktmolnsdata
Klippningens omfattning Valfritt	`FILTER_EXTENT`	[omfattning]	En kartutsträckning för att välja en delmängd av funktioner i punktmolnsdata Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`

Utgångar

Etikett	Namn	Typ	Beskrivning
Klippt	`OUTPUT`	[raster]	Utmatat punktmoln vars funktioner är punkterna inom polygonlagret för täckning.

Python-kod

Algoritm-ID: `pdal:clip`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa COPC

Skapar indexet för alla ingående punktmolnsfiler i batch-läge.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`LAYERS`	[punktmoln] [lista]	Inmatning av punktmolnslager för att skapa ett index för
Utmatningskatalog Valfritt	`OUTPUT`	mapp Standard: `[Hoppa över utdata]`	Ange den mapp som de nya filerna ska skapas i. *En av*: • Hoppa över utmatning • Spara till en temporär katalog • Spara till katalog

Utgångar

Etikett	Namn	Typ	Beskrivning
Utmatningskatalog	`OUTPUT`	mapp	Utdatamapp som innehåller punktmolnslager med tillhörande COPC-indexfiler.

Python-kod

Algoritm ID: `pdal:createcopc`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Information

Utmatning av grundläggande metadata från en inmatad punktmolnsfil.

Exempel på utmatningsinformation:

```
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
```

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[punktmoln]	Punktmolnslager för att extrahera metadatainformation från
Lagerinformation	`OUTPUT`	&Arkiv Standard: `[Spara till temporär fil]`	Ange den fil där metadatainformationen ska lagras. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Lagerinformation	`OUTPUT`	[vektor]	`HTML`-fil för lagring av metadatainformation.

Python-kod

Algoritm ID: `pdal:info`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Sammanfoga

Sammanfogar flera punktmolnsfiler till en enda.

Se även

Skapa ett virtuellt punktmoln (VPC)

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`LAYERS`	[punktmoln] [lista]	Inmatade punktmolnslager som ska slås samman till ett enda
Sammanslagna	`OUTPUT`	[punktmoln] Standard: `[Spara till temporär fil]`	Ange utdatapunktmolnet som sammanfogar indatafiler. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Filteruttryck Valfritt	`FILTER_EXPRESS`	[uttryck]	Ett *PDAL-uttryck* för att välja en delmängd av funktioner i punktmolnsdata
Klippningens omfattning Valfritt	`FILTER_EXTENT`	[omfattning]	En kartutsträckning för att välja en delmängd av funktioner i punktmolnsdata Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`

Utgångar

Etikett	Namn	Typ	Beskrivning
Sammanslagna	`OUTPUT`	[punktmoln]	Utgående punktmolnslager som sammanfogar alla indatafiler.

Python-kod

Algoritm-ID: `pdal:merge`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Reprojicera

Återprojicerar ett punktmoln till ett annat koordinatreferenssystem (CRS).

Se även

Tilldela projektion

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[punktmoln]	Inmatning av punktmolnlager för att omprojicera till ett annat CRS
Mål CRS	`CRS`	[crs]	CRS för att applicera på lagret
Omprojekterad	`OUTPUT`	[punktmoln] Standard: `[Spara till temporär fil]`	Ange den reprojicerade punktmolnsfilen. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Koordinatoperation Valfritt	`OPERATION`	[datum]	Den *datumtransformation* som ska användas för att omprojicera data mellan ursprungs- och målsystem.

Utgångar

Etikett	Namn	Typ	Beskrivning
REPROJECTED	`OUTPUT`	[punktmoln]	Utgående punktmolnslager i mål-CRS.

Python-kod

Algoritm ID: `pdal:reproject`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Tunn (efter provtagningsradie)

Skapar en förtunnad version av punktmolnet genom att utföra sampling efter avståndspunkt (minskar antalet punkter inom en viss radie).

Se även

Tunn (genom att hoppa över punkter)

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[punktmoln]	Inmatat punktmolnslager att skapa en förtunnad version från
Provtagningsradie (i kartenheter)	`SAMPLING_RADIU`	[numerisk: dubbel] Standard: 1,0	Avstånd inom vilket punkter samplas till en unik punkt
Förtunnad (genom radie)	`OUTPUT`	[punktmoln] Standard: `[Spara till temporär fil]`	Ange det utgående punktmolnet med reducerade punkter. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Filteruttryck Valfritt	`FILTER_EXPRESS`	[uttryck]	Ett *PDAL-uttryck* för att välja en delmängd av funktioner i punktmolnsdata
Klippningens omfattning Valfritt	`FILTER_EXTENT`	[omfattning]	En kartutsträckning för att välja en delmängd av funktioner i punktmolnsdata Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`

Utgångar

Etikett	Namn	Typ	Beskrivning
Förtunnad (genom radie)	`OUTPUT`	[punktmoln]	Utgående punktmolnslager med reducerade punkter.

Python-kod

Algoritm-ID: `pdal:thinbyradius`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Tunn (genom att hoppa över punkter)

Skapar en förtunnad version av punktmolnet genom att endast behålla var N:e punkt (minskar antalet punkter genom att hoppa över närliggande punkter).

Se även

Tunn (efter provtagningsradie)

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[punktmoln]	Inmatat punktmolnslager att skapa en förtunnad version från
Antal punkter att hoppa över	`POINTS_NUMBER`	[numerisk: heltal] Standard: 1	Behåll endast varje N:e punkt i inmatningslagret
Förtunnad (genom decimering)	`OUTPUT`	[punktmoln] Standard: `[Spara till temporär fil]`	Ange det utgående punktmolnet med reducerade punkter. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Filteruttryck Valfritt	`FILTER_EXPRESS`	[uttryck]	Ett *PDAL-uttryck* för att välja en delmängd av funktioner i punktmolnsdata
Klippningens omfattning Valfritt	`FILTER_EXTENT`	[omfattning]	En kartutsträckning för att välja en delmängd av funktioner i punktmolnsdata Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`

Utgångar

Etikett	Namn	Typ	Beskrivning
Förtunnad (genom decimering)	`OUTPUT`	[punktmoln]	Utgående punktmolnslager med reducerade punkter.

Python-kod

Algoritm ID: `pdal:thinbydecimate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Platta

Skapar plattor från inmatade punktmolnsfiler, vilket rekommenderas för bästa prestanda (vid visning eller analys) med sådana dataset i QGIS.

Se även

Skapa ett virtuellt punktmoln (VPC), *Skapa COPC*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`LAYERS`	[punktmoln] [lista]	Inmatade punktmolnslager att skapa plattor från
Kaklets längd	`LENGTH`	[numerisk: dubbel] Standard: 1000,0	Storlek på kanten på varje genererad kakelplatta
Utmatningskatalog	`OUTPUT`	mapp Standard: `[Spara till tillfällig mapp]`	Ange den mapp där de genererade plattorna ska lagras. *En av*: • Spara till en temporär katalog • Spara till katalog

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Tilldela CRS Valfritt	`CRS`	[crs]	CRS för att applicera på lagret

Utgångar

Etikett	Namn	Typ	Beskrivning
Utmatningskatalog	`OUTPUT`	mapp	Utdatamapp som innehåller de plattor som genererats från inmatningsfilerna.

Python-kod

Algoritm-ID: `pdal:tile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.14 Extrahering av punktmoln

Observera

Dessa algoritmer är endast tillgängliga om QGIS använder PDAL-biblioteket version 2.5.0 eller nyare.

Begränsning

Exporterar en polygonfil som innehåller punktmolnets lagergräns. Den kan innehålla hål och vara en polygon i flera delar.

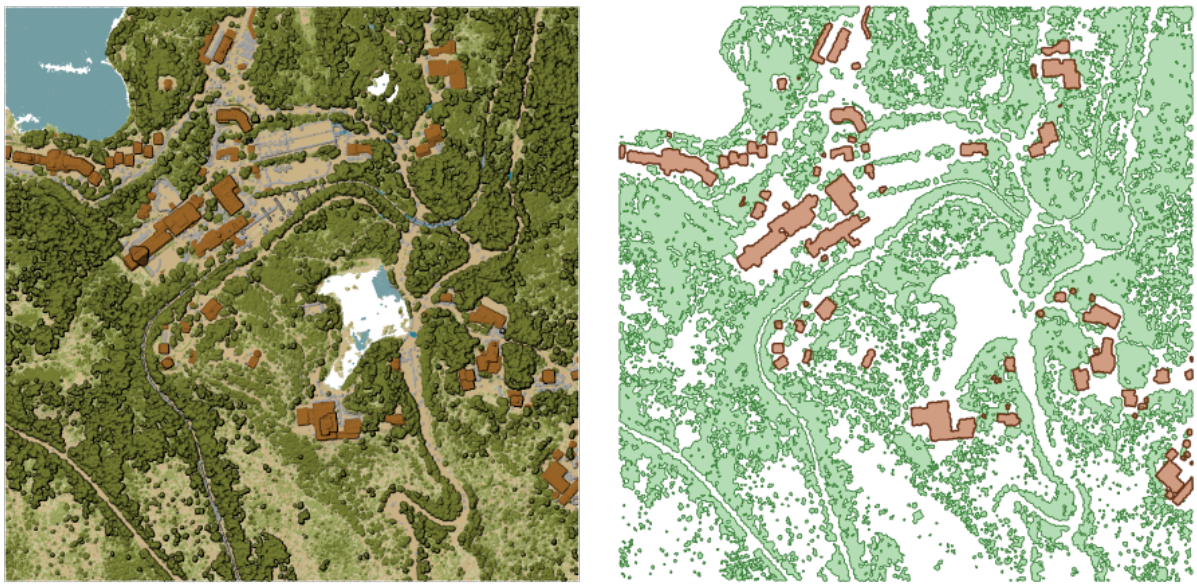


Fig. 24.15: Extrahering av polygoner för hög vegetation och byggnader från ett punktmolnslager

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[punktmoln]	Ingångspunktmolnlager för att beräkna gränsen för
Upplösning av celler som används för att beräkna gränsen Valfritt	`RESOLUTION`	[numerisk: dubbel]	Upplösning av celler som används för att beräkna gränsen
Minsta antal punkter i en cell för att cellen ska anses vara ockuperad Valfritt	`THRESHOLD`	[numerisk: heltal]	Minsta antal punkter i en cell för att cellen ska anses vara ockuperad
Begränsning	`OUTPUT`	[vektor: polygon] Standard: `[Spara till temporär fil]`	Ange den punktmolnsfil som ska användas som utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Filteruttryck Valfritt	`FILTER_EXPRESS`	[uttryck]	Ett *PDAL-uttryck* för att välja en delmängd av funktioner i punktmolnsdata
Klippningens omfattning Valfritt	`FILTER_EXTENT`	[omfattning]	En kartutsträckning för att välja en delmängd av funktioner i punktmolnsdata Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`

Utgångar

Etikett	Namn	Typ	Beskrivning
Begränsning	`OUTPUT`	[vektor: polygon]	Utmatning av polygonvektorlager med punktmolnets gräns. Format som stöds för närvarande är `.GPKG`.

Python-kod

Algoritm ID: `pdal:gräns`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Densitet

Exporterar en rasterfil baserat på antalet punkter inom varje rastercell - användbart för kvalitetskontroll av dataset med punktmoln.

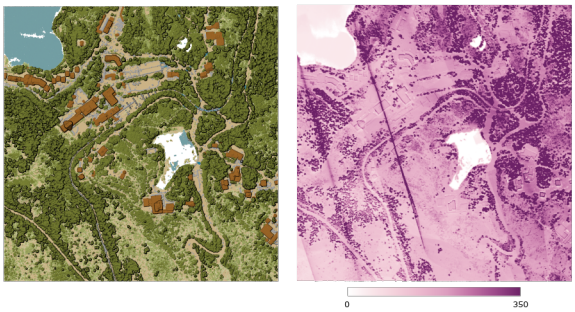


Fig. 24.16: Punkttäthet (antal punkter per 2x2 m) som raster

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[punktmoln]	Inmatning av punktmolnslager för export
Upplösning av täthetsrastret	`RESOLUTION`	[numerisk: dubbel] Standard: 1,0	Cellstorlek på utmatningsrastret
Storlek på plattor för parallella körningar	`TILE_SIZE`	[numerisk: heltal] Standard: 1000	Storlek på de rutor som data ska delas upp i för parallella körningar
Densitet	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange den rasterfil som data ska exporteras till. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Filteruttryck Valfritt	`FILTER_EXPRESS`	[uttryck]	Ett *PDAL-uttryck* för att välja en delmängd av funktioner i punktmolnsdata
Klippningens omfattning Valfritt	`FILTER_EXTENT`	[omfattning]	En kartutsträckning för att välja en delmängd av funktioner i punktmolnsdata Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
X-ursprung för en bricka för parallella körningar Valfritt	`ORIGIN_X`	[numerisk: dubbel]	
Y-ursprung för en kakelplatta för parallella körningar Valfritt	`ORIGIN_Y`	[numerisk: dubbel]	

Utgångar

Etikett	Namn	Typ	Beskrivning
Densitet	`OUTPUT`	[raster]	Utmatning av rasterlager med antal punkter inom varje cell. Format som stöds för närvarande är `.TIF`.

Python-kod

Algoritm-ID: `pdal:densitet`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Filtrera

Extraherar punkter från det ingående punktmolnet som matchar PDAL-uttryck och/eller ligger inom en beskärningsrektangel.

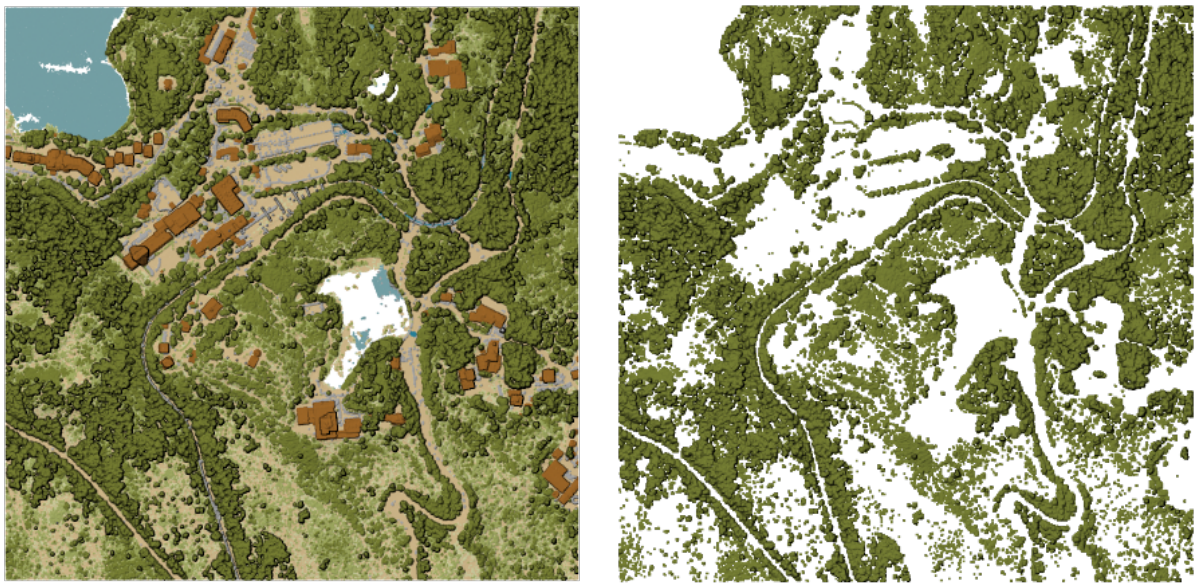


Fig. 24.17: Filtrering av hög vegetationsklass från ett ingående punktmolnslager

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[punktmoln]	Inmatning av punktmolnslager för export
Filteruttryck Valfritt	`FILTER_EXPRESS`	[uttryck]	Ett *PDAL-uttryck* för att välja en delmängd av funktioner i punktmolnsdata
Klippningens omfattning Valfritt	`FILTER_EXTENT`	[omfattning]	En kartutsträckning för att välja en delmängd av funktioner i punktmolnsdata Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Filtrerad	`OUTPUT`	[punktmoln] Standard: `[Spara till temporär fil]`	Ange den punktmolnsfil som data ska exporteras till. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Filtrerad	`OUTPUT`	[punktmoln]	Utgående punktmolnslager med de filtrerade funktionerna i det ingående punktmolnslagret. Format som stöds för närvarande är `.LAS`, `.LAZ`, `.COPC.LAZ` och `.VPC`.

Python-kod

Algoritm-ID: `pdal:filter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.15 Rasteranalys

Cellstapel procent rang från värde

Beräknar det cellvisa procentrankvärdet för en stapel raster baserat på ett enda indatavärde och skriver dem till ett utdataraster.

Vid varje cellplats rangordnas det angivna värdet bland respektive värden i stapeln med alla överlagrade och sorterade cellvärden från indatagrastren. För värden som ligger utanför stapelns värdefördelning returnerar algoritmen NoData eftersom värdet inte kan rangordnas bland cellvärdena.

Det finns två metoder för percentilberäkning:

- Inkluderande linjär interpolation (PERCENTRANK.INC)
- Exklusiv linjär interpolation (PERCENTRANK.EXC)

De linjära interpoleringsmetoderna returnerar den unika procentuella rangordningen för olika värden. Båda interpoleringsmetoderna följer sina motsvarigheter som implementeras av LibreOffice eller Microsoft Excel.

Utdatarasterns utsträckning och upplösning definieras av en referensraster. Rasterlager i indata som inte matchar cellstorleken i referensrasterlagret kommer att omsamplas med hjälp av omsampling med närmaste granne. NoData-värden i något av inmatningslagren kommer att resultera i en NoData-cellutmatning om parametern ”Ignore NoData values” inte är inställd. Rasterdatatypen för utdata kommer alltid att vara `Float32`.

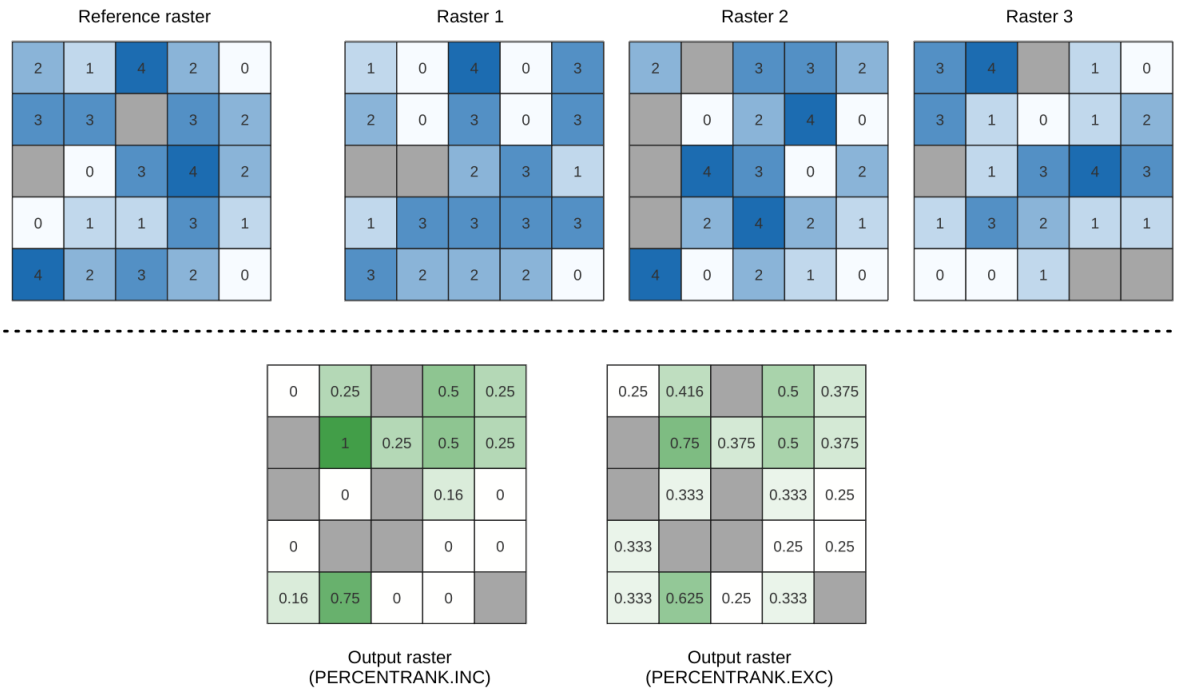


Fig. 24.18: Procentuell ranking Värde = 1. `NoData` celler (grå) ignoreras.

Se även

Percentil för cellstapel, *Cellstapel procentrank från rasterlager*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster] [lista]	Rasterlager som ska utvärderas. Om flerbandsraster används i datarasterstacken kommer algoritmen alltid att utföra analysen på det första bandet i rastren
Metod	`METHOD`	[uppräkning] Standard: 0	Metod för percentilberäkning: • 0 — Inkluderande linjär interpolation (PERCENTRANK.INC) • 1 — Exklusiv linjär interpolation (PERCENTRANK.EXC)
Värde	`VALUE`	[numerisk: dubbel] Standard: 10.0	Värde att rangordna bland de respektive värdena i stapeln av alla överlagrade och sorterade cellvärden från inmatningsrastrarna
Ignorera NoData-värden	`IGNORE_NODATA`	[boolean] Standard: Sann	Om det inte är markerat kommer alla NoData-celler i inmatningslagren att resultera i en NoData-cell i utmatningsrastret

continues on next page

Table 24.46 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Reference layer	`REFERENCE_LAYE`	[raster]	Referenslagret för skapandet av utdatalagret (utsträckning, CRS, pixeldimensioner)
Utdatalager	`OUTPUT`	[samma som inmatning] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Utmatning av NoData-värde	`OUTPUT_NODATA_`	[numerisk: dubbel] Standard: -9999.0	Värde som ska användas för NoData i utdatalagret
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utdatalager	`OUTPUT`	[raster]	Utgående rasterlager som innehåller resultatet
CRS auktoritetsidentifier	`CRS_AUTHID`	[sträng]	Koordinatreferenssystemet för det utgående rasterlagret
Extent	`EXTENT`	[sträng]	Den spatiala omfattningen av det utgående rasterlagret
Bredd i pixlar	`WIDTH_IN_PIXEL`	[numerisk: heltal]	Antalet kolumner i det utgående rasterlagret
Höjd i pixlar	`HEIGHT_IN_PIXE`	[numerisk: heltal]	Antalet rader i det utgående rasterlagret
Totalt antal pixlar	`TOTAL_PIXEL_CO`	[numerisk: heltal]	Antalet pixlar i det utgående rasterlagret

Python-kod

Algoritm-id: `native:cellstackpercentrankfromvalue`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Percentil för cellstapel

Beräknar det cellvisa percentilvärdet för en stack med raster och skriver resultatet till ett utdataraster. Percentilen som ska returneras bestäms av percentilinmatningsvärdet (varierar mellan 0 och 1). På varje cellplats erhålls den angivna percentilen med hjälp av respektive värde från stapeln med alla överlagrade och sorterade cellvärden i indatarastren.

Det finns tre metoder för percentilberäkning:

- Närmaste rang: returnerar det värde som ligger närmast den angivna percentilen
- Inkluderande linjär interpolation (PERCENTRANK.INC)
- Exklusiv linjär interpolation (PERCENTRANK.EXC)

De linjära interpoleringsmetoderna returnerar de unika värdena för olika percentiler. Båda interpoleringsmetoderna följer sina motsvarigheter som implementeras av LibreOffice eller Microsoft Excel.

Utdatarasterns utsträckning och upplösning definieras av en referensraster. Rasterlager i indata som inte matchar cellstorleken i referensrasterlagret kommer att omsamplas med hjälp av omsampling med närmaste granne. NoData-värden i något av inmatningslagren kommer att resultera i en NoData-cellutmatning om parametern ”Ignore NoData values” inte är inställd. Rasterdatatypen för utdata kommer alltid att vara `Float32`.

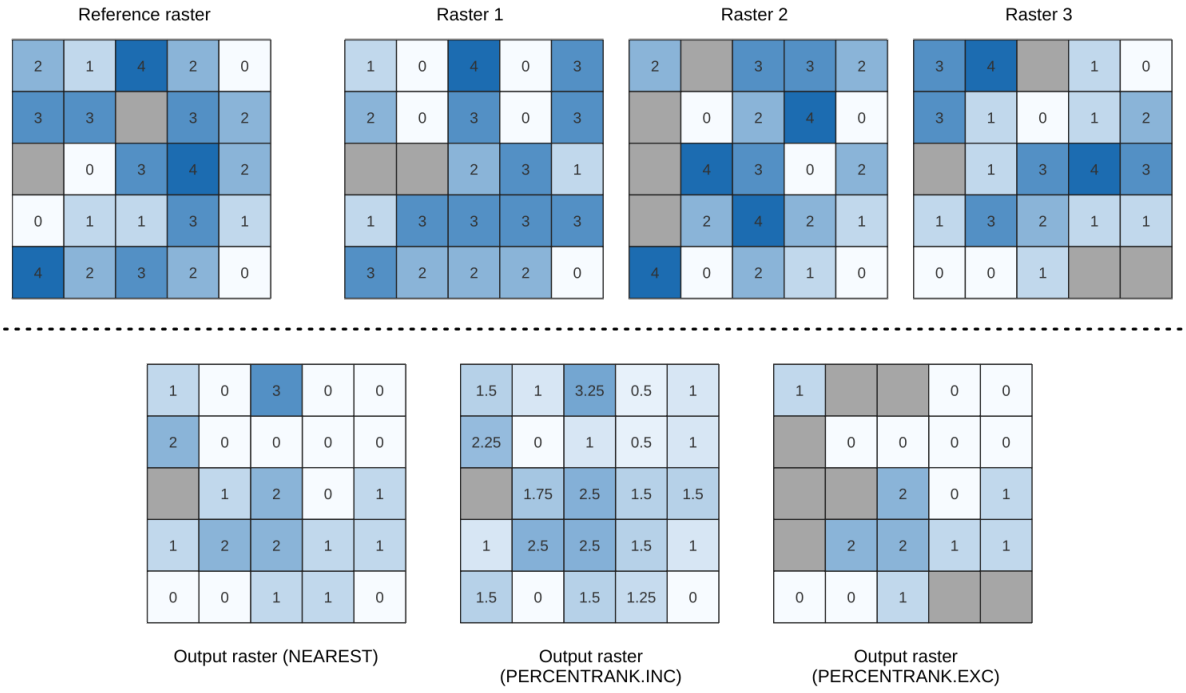


Fig. 24.19: Percentil = 0,25. `NoData`-celler (grå) ignoreras.

Se även

Percentil för cellstapel, *Cellstapel procentrank från rasterlager*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster] [lista]	Rasterlager som ska utvärderas. Om flerbandsraster används i datarasterstacken kommer algoritmen alltid att utföra analysen på det första bandet i rastren
Metod	`METHOD`	[uppräkning] Standard: 0	Metod för percentilberäkning: • 0 — Närmaste rang: returnerar det värde som ligger närmast den angivna percentilen • 1 — Inkluderande linjär interpolation (PERCENTILE.INC) • 2 — Exklusiv linjär interpolation (PERCENTILE.EXC)
Percentil	`VALUE`	[numerisk: dubbel] Standard: 0,25	Värde som ska rangordnas bland respektive värden i stapeln med alla överlagrade och sorterade cellvärden från indatarastren. Mellan 0 och 1.
Ignorera NoData-värden	`IGNORE_NODATA`	[boolean] Standard: Sann	Om det inte är markerat kommer alla NoData-celler i inmatningslagren att resultera i en NoData-cell i utmatningsrastret
Reference layer	`REFERENCE_LAYE`	[raster]	Referenslagret för skapandet av utdatalagret (utsträckning, CRS, pixeldimensioner)
Utdatalager	`OUTPUT`	[samma som inmatning] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Utmatning av NoData-värde	`OUTPUT_NODATA_`	[numerisk: dubbel] Standard: -9999.0	Värde som ska användas för NoData i utdatalagret
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utdatalager	`OUTPUT`	[raster]	Utgående rasterlager som innehåller resultatet
CRS auktoritetsidentifier	`CRS_AUTHID`	[sträng]	Koordinatreferenssystemet för det utgående rasterlagret
Extent	`EXTENT`	[sträng]	Den spatiala omfattningen av det utgående rasterlagret
Bredd i pixlar	`WIDTH_IN_PIXEL`	[numerisk: heltal]	Antalet kolumner i det utgående rasterlagret
Höjd i pixlar	`HEIGHT_IN_PIXE`	[numerisk: heltal]	Antalet rader i det utgående rasterlagret
Totalt antal pixlar	`TOTAL_PIXEL_CO`	[numerisk: heltal]	Antalet pixlar i det utgående rasterlagret

Python-kod

Algoritm-id: `native:cellstackpercentile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Cellstapel procentrank från rasterlager

Beräknar det cellvisa procentrankvärdet för en stack med raster baserat på ett indatavärdesraster och skriver dem till ett utdataraster.

Vid varje cellplats rangordnas det aktuella värdet i värderastret bland respektive värden i stapeln av alla överlagrade och sorterade cellvärden i indatarastren. För värden som ligger utanför stapelns värdefördelning returnerar algoritmen NoData eftersom värdet inte kan rangordnas bland cellvärdena.

Det finns två metoder för percentilberäkning:

- Inkluderande linjär interpolation (PERCENTRANK.INC)
- Exklusiv linjär interpolation (PERCENTRANK.EXC)

De linjära interpoleringsmetoderna returnerar de unika värdena för olika percentiler. Båda interpoleringsmetoderna följer sina motsvarigheter som implementeras av LibreOffice eller Microsoft Excel.

Utdatarasterns utsträckning och upplösning definieras av en referensraster. Rasterlager i indata som inte matchar cellstorleken i referensrasterlagret kommer att omsamplas med hjälp av omsampling med närmaste granne. NoData-värden i något av inmatningslagren kommer att resultera i en NoData-cellutmatning om parametern ”Ignore NoData values” inte är inställd. Rasterdatatypen för utdata kommer alltid att vara `Float32`.

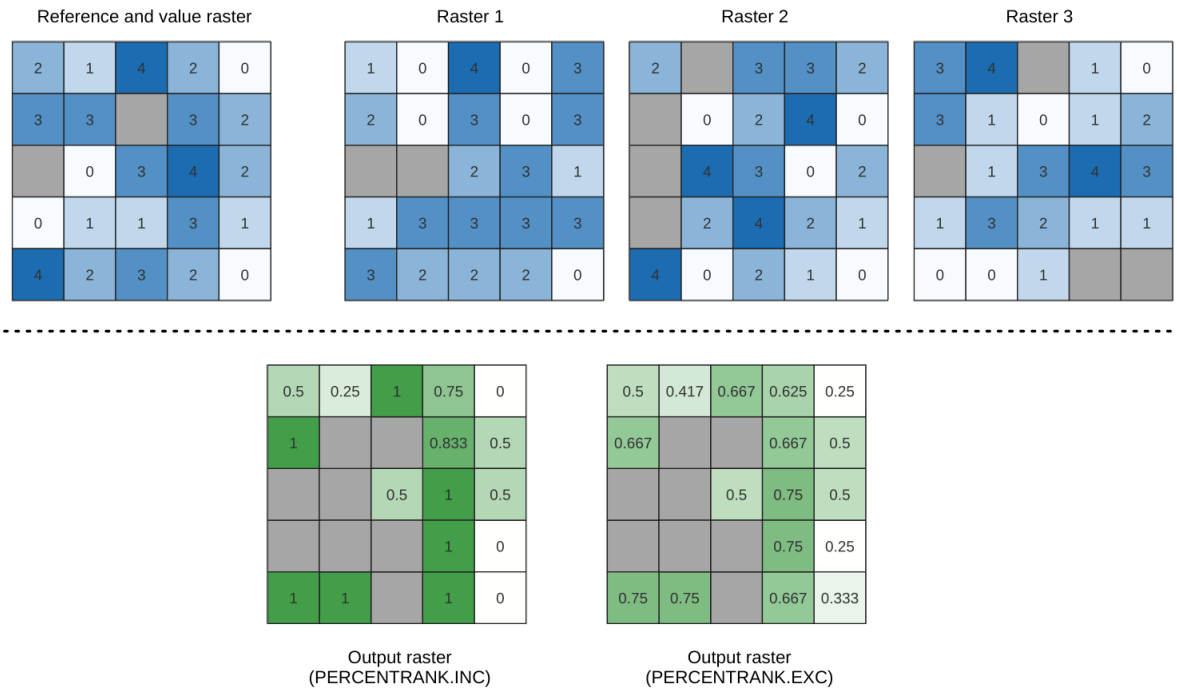


Fig. 24.20: Rangordnar rasterlagrets värdeceller. celler med `NoData` (grå) ignoreras.

Se även

Percentil för cellstapel, *Cellstapel procent rang från värde*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster] [lista]	Rasterlager som ska utvärderas. Om flerbandsraster används i datarasterstacken kommer algoritmen alltid att utföra analysen på det första bandet i rastren
Värde rasterlager	`INPUT_VALUE_RA`	[raster]	Det lager som ska rangordna värdena bland stapeln av alla överlagrade lager
Värde rasterband	`VALUE_RASTER_B`	[numerisk: heltal] Standard: 1	Band av ”value raster layer” att jämföra med
Metod	`METHOD`	[uppräkning] Standard: 0	Metod för percentilberäkning: • 0 — Inkluderande linjär interpolation (PERCENTRANK.INC) • 1 — Exklusiv linjär interpolation (PERCENTRANK.EXC)
Ignorera NoData-värden	`IGNORE_NODATA`	[boolean] Standard: Sann	Om det inte är markerat kommer alla NoData-celler i inmatningslagren att resultera i en NoData-cell i utmatningsrastret

continues on next page

Table 24.50 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Reference layer	`REFERENCE_LAYE`	[raster]	Referenslagret för skapandet av utdatalagret (utsträckning, CRS, pixeldimensioner)
Utdatalager	`OUTPUT`	[samma som inmatning] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Utmatning av NoData-värde	`OUTPUT_NODATA_`	[numerisk: dubbel] Standard: -9999.0	Värde som ska användas för NoData i utdatalagret
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utdatalager	`OUTPUT`	[raster]	Utgående rasterlager som innehåller resultatet
CRS auktoritetsidentifier	`CRS_AUTHID`	[sträng]	Koordinatreferenssystemet för det utgående rasterlagret
Extent	`EXTENT`	[sträng]	Den spatiala omfattningen av det utgående rasterlagret
Bredd i pixlar	`WIDTH_IN_PIXEL`	[numerisk: heltal]	Antalet kolumner i det utgående rasterlagret
Höjd i pixlar	`HEIGHT_IN_PIXE`	[numerisk: heltal]	Antalet rader i det utgående rasterlagret
Totalt antal pixlar	`TOTAL_PIXEL_CO`	[numerisk: heltal]	Antalet pixlar i det utgående rasterlagret

Python-kod

Algoritm-id: `native:cellstackpercentrankfromrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Cellstatistik

Beräknar statistik per cell baserat på ingående rasterlager och för varje cell skrivs den resulterande statistiken till ett utdataraster. På varje cellplats definieras utdatavärdet som en funktion av alla överlagrade cellvärden i indatarastren.

Som standard kommer en NoData-cell i NÅGOT av inmatningslagren att resultera i en NoData-cell i utdatarastret. Om alternativet *Ignore NoData values* är markerat ignoreras NoData-ingångar i den statistiska beräkningen. Detta kan resultera i NoData-utdata för platser där alla celler är NoData.

Parametern *Reference layer* anger ett befintligt rasterlager som ska användas som referens när utdatarastern skapas. Utdatarastret kommer att ha samma utsträckning, CRS och pixeldimensioner som detta lager.

Beräkningsdetaljer: Rasterlager i indata som inte matchar cellstorleken i referensrasterlagret kommer att omsamplas med hjälp av `närmaste granne omsampling`. Rasterdatatypen för utdata sätts till den mest komplexa datatypen i indatasetet, utom när funktionerna `Mean`, `Standardavvikelse` och `Varians` används (datatypen är alltid `Float32` eller `Float64` beroende på indatatyp) eller `Count` och `Variety` (datatypen är alltid `Int32`).

- `Count`: Count-statistiken resulterar alltid i antalet celler utan NoData-värden på den aktuella cellpositionen.
- `Median`: Om antalet inmatningslager är jämnt beräknas medianen som det aritmetiska medelvärdet av de två mellersta värdena i de ordnade cellinmatningsvärdena.
- `Minoritet/Majoritet`: Om ingen unik minoritet eller majoritet kunde hittas, blir resultatet NoData, förutom att alla indatacellvärden är lika.

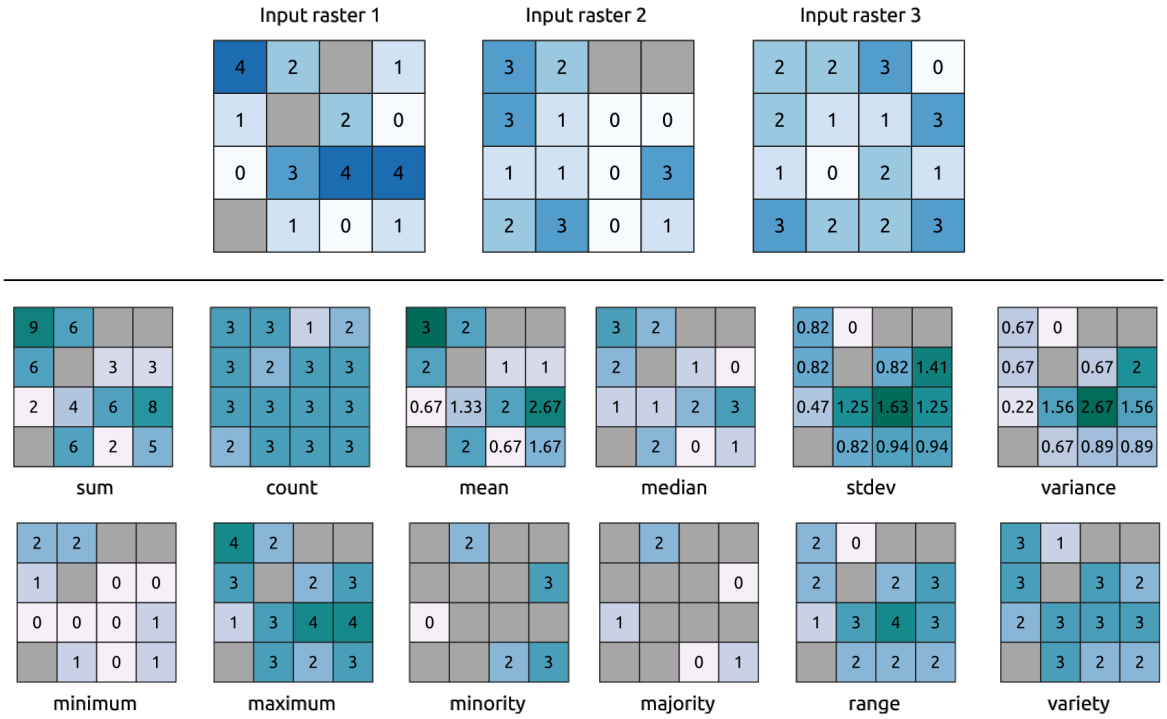


Fig. 24.21: Exempel med alla statistiska funktioner. `NoData`-celler (grå) tas med i beräkningen.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster] [lista]	Inmatade rasterlager
Statistik	`STATISTIC`	[uppräkning] Standard: 0	Tillgänglig statistik. Alternativ: • 0 — Summa • 1 — Räkning • 2 — Medelvärde • 3 — Median • 4 — Standardavvikelse • 5 — Avvikelse • 6 — Minimum • 7 — Maximalt • 8 — Minoritet (minsta gemensamma värde) • 9 — Majoritet (vanligaste värdet) • 10 — Område (max - min) • 11 — Variety (antal unika värden)
Ignorera NoData-värden	`IGNORE_NODATA`	[boolean] Standard: Sann	Beräkna statistik även för alla cellstaplar, utan hänsyn till NoData-frekvensen.
Reference layer	`REF_LAYER`	[raster]	Referenslagret att skapa utdatalagret från (utsträckning, CRS, pixeldimensioner)
Utdatalager	`OUTPUT`	[samma som inmatning] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Utmatning av NoData-värde Valfritt	`OUTPUT_NO_DATA`	[numerisk: dubbel] Standard: -9999.0	Värde som ska användas för NoData i utdatalagret
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
CRS auktoritetsidentifier	`CRS_AUTHID`	[crs]	Koordinatreferenssystemet för det utgående rasterlagret
Extent	`EXTENT`	[sträng]	Den spatiala omfattningen av det utgående rasterlagret
Höjd i pixlar	`HEIGHT_IN_PIXE`	[numerisk: heltal]	Antalet rader i det utgående rasterlagret
Utgångsraster	`OUTPUT`	[raster]	Utgående rasterlager som innehåller resultatet
Totalt antal pixlar	`TOTAL_PIXEL_CO`	[numerisk: heltal]	Antalet pixlar i det utgående rasterlagret
Bredd i pixlar	`WIDTH_IN_PIXEL`	[numerisk: heltal]	Antalet kolumner i det utgående rasterlagret

Python-kod

Algoritm-ID: `native:cellstatistik`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Lika med frekvens

Utvärderar cell för cell frekvensen (antalet gånger) som värdena i en inmatningsstapel med raster är lika med värdet i ett värdelager. Utgångsrastrets utsträckning och upplösning definieras av ingångsrasterlagret och är alltid av typen `Int32`.

Om flerbandsraster används i datarasterstacken kommer algoritmen alltid att utföra analysen på det första bandet i rastren - använd GDAL för att använda andra band i analysen. NoData-värdet för utdata kan ställas in manuellt.

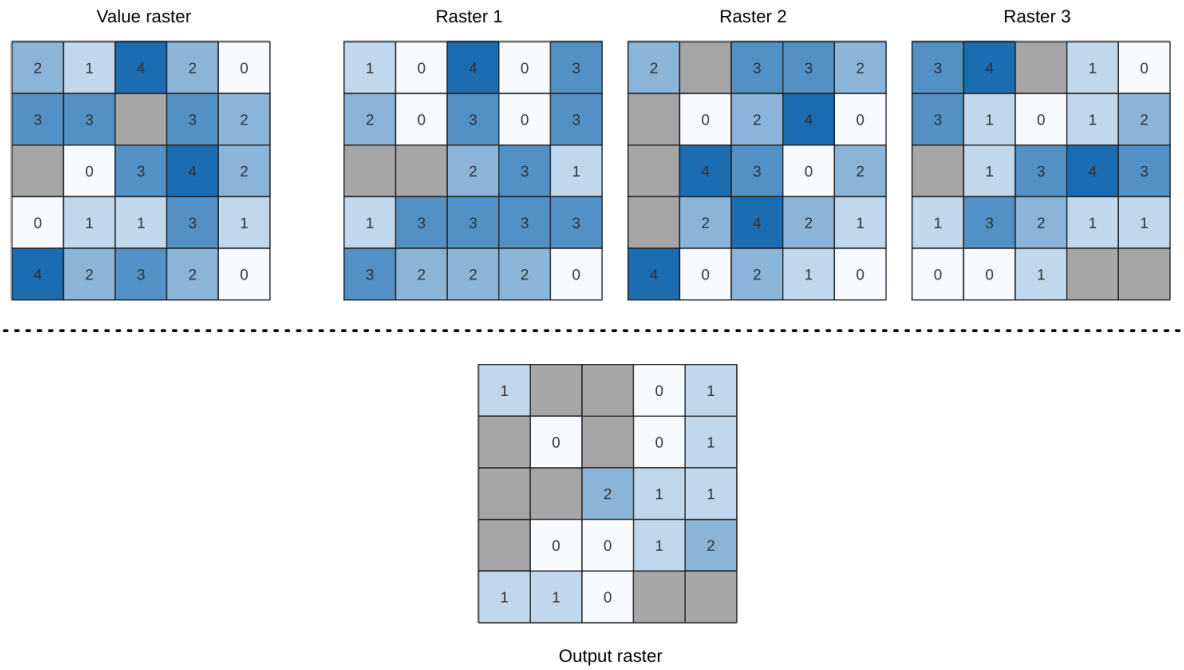


Fig. 24.22: För varje cell i utdatarastret representerar värdet det antal gånger som motsvarande celler i listan med raster är desamma som värderastret. `NoData`-celler (grå) tas med i beräkningen.

Se även

Större än frekvensen, *Mindre än frekvensen*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsvärde raster	`INPUT_VALUE_RA`	[raster]	Ingångsvärdeslagret fungerar som referenslager för provlagren
Värde rasterband	`INPUT_VALUE_RA`	[rasterband] Standard: Det första bandet i rasterlagret	Välj det band du vill använda som prov
Rasterlager för inmatning	`INPUT_RASTERS`	[raster] [lista]	Rasterlager som ska utvärderas. Om flerbandsraster används i datarasterstacken kommer algoritmen alltid att utföra analysen på det första bandet i rastren
Ignorera NoData-värden	`IGNORE_NODATA`	[boolean] Standard: Falsk	Om det inte är markerat kommer alla NoData-celler i värderastret eller datalagerstapeln att resultera i en NoData-cell i utdatarastret
Utdatalager	`OUTPUT`	[samma som inmatning] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Utmatning av NoData-värde Valfritt	OUTPUT_NO_DATA	[numerisk: dubbel] Standard: -9999.0	Värde som ska användas för NoData i utdatalagret
Alternativ för skapande Valfritt	CREATE_OPTIONS	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utdatalager	OUTPUT	[raster]	Utgående rasterlager som innehåller resultatet
CRS auktoritetsidentifier	CRS_AUTHID	[sträng]	Koordinatreferenssystemet för det utgående rasterlagret
Extent	EXTENT	[sträng]	Den spatiala omfattningen av det utgående rasterlagret
Antal celler med förekomster av lika värde	FOUND_LOCATION	[numerisk: heltal]	
Höjd i pixlar	HEIGHT_IN_PIXE	[numerisk: heltal]	Antalet rader i det utgående rasterlagret
Totalt antal pixlar	TOTAL_PIXEL_CO	[numerisk: heltal]	Antalet pixlar i det utgående rasterlagret
Medelfrekvens vid giltiga cellpositioner	MEAN_FREQUENCY	[numerisk: dubbel]	
Antal värdeförekomster	OCCURRENCE_COU	[numerisk: heltal]	
Bredd i pixlar	WIDTH_IN_PIXEL	[numerisk: heltal]	Antalet kolumner i det utgående rasterlagret

Python-kod

Algoritm-id: `native:equaltofrequency`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Fuzzifiera raster (gaussiskt medlemskap)

Omvandlar ett inmatningsraster till ett fuzzifierat raster genom att tilldela ett medlemsvärde till varje pixel med hjälp av en gaussisk medlemsfunktion. Medlemskapsvärdena sträcker sig från 0 till 1. I det fuzzifierade rastret innebär ett värde på 0 inget medlemskap i den definierade fuzzy-uppsättningen, medan ett värde på 1 innebär fullt medlemskap. Den gaussiska medlemskapsfunktionen definieras som $\mu(x) = e^{-f1*(x-f2)^2}$, där *f1* är spridningen och *f2* mittpunkten.

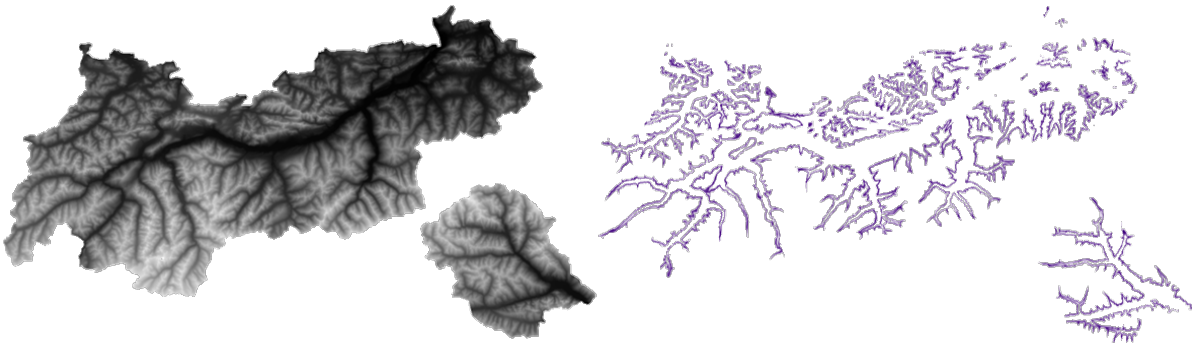


Fig. 24.23: Exempel på Fuzzify-raster. Ingångsrasterkälla: Land Tirol - data.tirol.gv.at.

Se även

Fuzzifiera raster (stort medlemskap), *Fuzzifiera raster (linjärt medlemskap)*, *Fuzzifiera raster (nära medlemskap)*, *Fuzzifiera raster (kraftmedlemskap)*, *Fuzzifiera raster (litet antal medlemmar)*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Input Raster	`INPUT`	[raster]	Inmatat rasterlager
Bandnummer	`BAND`	[rasterband] Standard: Det första bandet i rasterlagret	Om rastret är multibandigt väljer du det band som du vill fuzzifiera.
Funktionens mittpunkt	`FUZZYMIDPOINT`	[numerisk: dubbel] Standard: 10.0	Mittpunkt för den gaussiska funktionen
Funktionsfördelning	`FUZZYSPREAD`	[numerisk: dubbel] Standard: 0,01	Spridning av den gaussiska funktionen
Fuzzifierat raster	`OUTPUT`	[samma som inmatning] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Alternativ för skapande Valfritt	CREATE_OPTIONS	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).

Utgångar

Etikett	Namn	Typ	Beskrivning
Fuzzifierat raster	OUTPUT	[samma som inmatning]	Utgående rasterlager som innehåller resultatet
CRS auktoritetsidentifier	CRS_AUTHID	[crs]	Koordinatreferenssystemet för det utgående rasterlagret
Extent	EXTENT	[sträng]	Den spatiala omfattningen av det utgående rasterlagret
Bredd i pixlar	WIDTH_IN_PIXEL	[numerisk: heltal]	Antalet kolumner i det utgående rasterlagret
Höjd i pixlar	HEIGHT_IN_PIXE	[numerisk: heltal]	Antalet rader i det utgående rasterlagret
Totalt antal pixlar	TOTAL_PIXEL_CO	[numerisk: heltal]	Antalet pixlar i det utgående rasterlagret

Python-kod

Algoritm-ID: `native:fuzzifyrastergaussianmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Fuzzifiera raster (stort medlemskap)

Omvandlar ett inmatningsraster till ett fuzzifierat raster genom att tilldela ett medlemsvärde till varje pixel med hjälp av en stor medlemsfunktion. Medlemskapsvärdena sträcker sig från 0 till 1. I det fuzzifierade rastret innebär ett värde på 0 inget medlemskap i den definierade fuzzy-uppsättningen, medan ett värde på 1 innebär fullt medlemskap. Den stora medlemsfunktionen definieras som $\mu(x) = \frac{1}{1+\left(\frac{x}{f2}\right)^{-f1}}$, där *f1* är spridningen och *f2* mittpunkten.

Se även

Fuzzifiera raster (gaussiskt medlemskap), *Fuzzifiera raster (linjärt medlemskap)*, *Fuzzifiera raster (nära medlemskap)*, *Fuzzifiera raster (kraftmedlemskap)*, *Fuzzifiera raster (litet antal medlemmar)*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Input Raster	`INPUT`	[raster]	Inmatat rasterlager
Bandnummer	`BAND`	[rasterband] Standard: Det första bandet i rasterlagret	Om rastret är multibandigt väljer du det band som du vill fuzzifiera.
Funktionens mittpunkt	`FUZZYMIDPOINT`	[numerisk: dubbel] Standard: 50,0	Mittpunkt för den stora funktionen
Funktionsfördelning	`FUZZYSPREAD`	[numerisk: dubbel] Standard: 5,0	Utbredning av den stora funktionen
Fuzzifierat raster	`OUTPUT`	[samma som inmatning] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Fuzzifierat raster	`OUTPUT`	[samma som inmatning]	Utgående rasterlager som innehåller resultatet
CRS auktoritetsidentifier	`CRS_AUTHID`	[crs]	Koordinatreferenssystemet för det utgående rasterlagret
Extent	`EXTENT`	[sträng]	Den spatiala omfattningen av det utgående rasterlagret
Bredd i pixlar	`WIDTH_IN_PIXEL`	[numerisk: heltal]	Antalet kolumner i det utgående rasterlagret
Höjd i pixlar	`HEIGHT_IN_PIXE`	[numerisk: heltal]	Antalet rader i det utgående rasterlagret
Totalt antal pixlar	`TOTAL_PIXEL_CO`	[numerisk: heltal]	Antalet pixlar i det utgående rasterlagret

Python-kod

Algoritm-ID: `native:fuzzifyrasterlargemembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Fuzzifiera raster (linjärt medlemskap)

Omvandlar ett inmatningsraster till ett fuzzifierat raster genom att tilldela ett medlemsvärde till varje pixel med hjälp av en linjär medlemsfunktion. Medlemskapsvärdena sträcker sig från 0 till 1. I det fuzzifierade rastret innebär ett värde på 0 inget medlemskap i den definierade fuzzy-uppsättningen, medan ett värde på 1 innebär fullt medlemskap.

Den linjära funktionen definieras som $\mu(X)\begin{cases}0 & x \leq a \\ \frac{x-a}{b-a} & a < x < b \\ 1 & x \geq b\end{cases}$, där a är den låga gränsen och b den höga gränsen. Den här ekvationen tilldelar medlemsvärden med hjälp av en linjär transformation för pixelvärden mellan de låga och höga gränserna. Pixelvärden som är mindre än den låga gränsen får 0 medlemskap medan pixelvärden som är större än den höga gränsen får 1 medlemskap.

Se även

Fuzzifiera raster (gaussiskt medlemskap), *Fuzzifiera raster (stort medlemskap)*, *Fuzzifiera raster (nära medlemskap)*, *Fuzzifiera raster (kraftmedlemskap)*, *Fuzzifiera raster (litet antal medlemmar)*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Input Raster	`INPUT`	[raster]	Inmatat rasterlager
Bandnummer	`BAND`	[rasterband] Standard: Det första bandet i rasterlagret	Om rastret är multibandigt väljer du det band som du vill fuzzifiera.
Låg gräns för fuzzy-medlemskap	`FUZZYLOWBOUND`	[numerisk: dubbel] Standard: 0,0	Låg gräns för den linjära funktionen
Hög gräns för fuzzy-medlemskap	`FUZZYHIGHBOUND`	[numerisk: dubbel] Standard: 1,0	Hög gräns för den linjära funktionen
Fuzzifierat raster	`OUTPUT`	[samma som inmatning] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Alternativ för skapande Valfritt	CREATE_OPTIONS	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).

Utgångar

Etikett	Namn	Typ	Beskrivning
Fuzzifierat raster	OUTPUT	[samma som inmatning]	Utgående rasterlager som innehåller resultatet
CRS auktoritetsidentifier	CRS_AUTHID	[crs]	Koordinatreferenssystemet för det utgående rasterlagret
Extent	EXTENT	[sträng]	Den spatiala omfattningen av det utgående rasterlagret
Bredd i pixlar	WIDTH_IN_PIXEL	[numerisk: heltal]	Antalet kolumner i det utgående rasterlagret
Höjd i pixlar	HEIGHT_IN_PIXE	[numerisk: heltal]	Antalet rader i det utgående rasterlagret
Totalt antal pixlar	TOTAL_PIXEL_CO	[numerisk: heltal]	Antalet pixlar i det utgående rasterlagret

Python-kod

Algoritm-ID: `native:fuzzifyrasterlinearmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Fuzzifiera raster (nära medlemskap)

Omvandlar ett inmatningsraster till ett fuzzifierat raster genom att tilldela ett medlemsvärde till varje pixel med hjälp av en Near-medlemsfunktion. Medlemskapsvärdena sträcker sig från 0 till 1. I det fuzzifierade rastret innebär ett värde på 0 inget medlemskap i den definierade fuzzy-uppsättningen, medan ett värde på 1 innebär fullt medlemskap.

Near-medlemsfunktionen definieras som $\mu(x) = \frac{1}{1 + f1 * (x - f2)^2}$, där *f1* är spridningen och *f2* mittpunkten.

Se även

Fuzzifiera raster (gaussiskt medlemskap), *Fuzzifiera raster (stort medlemskap)*, *Fuzzifiera raster (linjärt medlemskap)*, *Fuzzifiera raster (kraftmedlemskap)*, *Fuzzifiera raster (litet antal medlemmar)*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Input Raster	`INPUT`	[raster]	Inmatat rasterlager
Bandnummer	`BAND`	[rasterband] Standard: Det första bandet i rasterlagret	Om rastret är multibandigt väljer du det band som du vill fuzzifiera.
Funktionens mittpunkt	`FUZZYMIDPOINT`	[numerisk: dubbel] Standard: 50,0	Mittpunkt för den nära funktionen
Funktionsfördelning	`FUZZYSPREAD`	[numerisk: dubbel] Standard: 0,01	Spridning av den närmaste funktionen
Fuzzifierat raster	`OUTPUT`	[samma som inmatning] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).

Utgångar

Etikett	Namn	Typ	Beskrivning
Fuzzifierat raster	`OUTPUT`	[samma som inmatning]	Utgående rasterlager som innehåller resultatet
CRS auktoritetsidentifier	`CRS_AUTHID`	[crs]	Koordinatreferenssystemet för det utgående rasterlagret
Extent	`EXTENT`	[sträng]	Den spatiala omfattningen av det utgående rasterlagret
Bredd i pixlar	`WIDTH_IN_PIXEL`	[numerisk: heltal]	Antalet kolumner i det utgående rasterlagret
Höjd i pixlar	`HEIGHT_IN_PIXE`	[numerisk: heltal]	Antalet rader i det utgående rasterlagret
Totalt antal pixlar	`TOTAL_PIXEL_CO`	[numerisk: heltal]	Antalet pixlar i det utgående rasterlagret

Python-kod

Algoritm-ID: `native:fuzzifyrasternearmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Fuzzifiera raster (kraftmedlemskap)

Omvandlar ett inmatningsraster till ett fuzzifierat raster genom att tilldela ett medlemsvärde till varje pixel med hjälp av en Power-medlemsfunktion. Medlemskapsvärdena sträcker sig från 0 till 1. I det fuzzifierade rastret innebär ett värde på 0 inget medlemskap i den definierade fuzzy-uppsättningen, medan ett värde på 1 innebär fullt medlemskap.

Power-funktionen definieras som $\mu(x)\begin{cases}0 & x \le a \\ \left(\frac{x-a}{b-a}\right)^{f1} & a < x < b \\ 1 & x \ge b\end{cases}$, där a är den låga gränsen, b är den höga gränsen och $f1$ är exponenten. Den här ekvationen tilldelar medlemsvärden med hjälp av power-transformationen för pixelvärden mellan den låga och den höga gränsen. Pixelvärden som är mindre än den låga gränsen får 0 medlemskap medan pixelvärden som är större än den höga gränsen får 1 medlemskap.

Se även

Fuzzifiera raster (gaussiskt medlemskap), *Fuzzifiera raster (stort medlemskap)*, *Fuzzifiera raster (linjärt medlemskap)*, *Fuzzifiera raster (nära medlemskap)*, *Fuzzifiera raster (litet antal medlemmar)*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Input Raster	`INPUT`	[raster]	Inmatat rasterlager
Bandnummer	`BAND`	[rasterband] Standard: Det första bandet i rasterlagret	Om rastret är multibandigt väljer du det band som du vill fuzzifiera.
Låg gräns för fuzzy-medlemskap	`FUZZYLOWBOUND`	[numerisk: dubbel] Standard: 0,0	Låg gräns för effektfunktionen
Hög gräns för fuzzy-medlemskap	`FUZZYHIGHBOUND`	[numerisk: dubbel] Standard: 1,0	Hög gräns för effektfunktionen
Hög gräns för fuzzy-medlemskap	`FUZZYEXPONENT`	[numerisk: dubbel] Standard: 2,0	Exponent för potensfunktionen
Fuzzifierat raster	`OUTPUT`	[samma som inmatning] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Alternativ för skapande Valfritt	CREATE_OPTIONS	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).

Utgångar

Etikett	Namn	Typ	Beskrivning
Fuzzifierat raster	OUTPUT	[samma som inmatning]	Utgående rasterlager som innehåller resultatet
CRS auktoritetsidentifier	CRS_AUTHID	[crs]	Koordinatreferenssystemet för det utgående rasterlagret
Extent	EXTENT	[sträng]	Den spatiala omfattningen av det utgående rasterlagret
Bredd i pixlar	WIDTH_IN_PIXEL	[numerisk: heltal]	Antalet kolumner i det utgående rasterlagret
Höjd i pixlar	HEIGHT_IN_PIXE	[numerisk: heltal]	Antalet rader i det utgående rasterlagret
Totalt antal pixlar	TOTAL_PIXEL_CO	[numerisk: heltal]	Antalet pixlar i det utgående rasterlagret

Python-kod

Algoritm-ID: `native:fuzzifyrasterpowermembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Fuzzifiera raster (litet antal medlemmar)

Omvandlar ett inmatningsraster till ett fuzzifierat raster genom att tilldela ett medlemsvärde till varje pixel med hjälp av en Small-medlemsfunktion. Medlemskapsvärdena sträcker sig från 0 till 1. I det fuzzifierade rastret innebär ett värde på 0 inget medlemskap i den definierade fuzzy-uppsättningen, medan ett värde på 1 innebär fullt medlemskap.

Den lilla medlemskapsfunktionen definieras som $\mu(x) = \frac{1}{1 + \left(\frac{x}{f2}\right)^{f1}}$, där *f1* är spridningen och *f2* mittpunkten.

Se även

Fuzzifiera raster (gaussiskt medlemskap), *Fuzzifiera raster (stort medlemskap)* *Fuzzifiera raster (linjärt medlemskap)*, *Fuzzifiera raster (nära medlemskap)*, *Fuzzifiera raster (kraftmedlemskap)*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Input Raster	`INPUT`	[raster]	Inmatat rasterlager
Bandnummer	`BAND`	[rasterband] Standard: Det första bandet i rasterlagret	Om rastret är multibandigt väljer du det band som du vill fuzzifiera.
Funktionens mittpunkt	`FUZZYMIDPOINT`	[numerisk: dubbel] Standard: 50,0	Mittpunkt för den lilla funktionen
Funktionsfördelning	`FUZZYSPREAD`	[numerisk: dubbel] Standard: 5,0	Utbredning av den lilla funktionen
Fuzzifierat raster	`OUTPUT`	[samma som inmatning] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Fuzzifierat raster	`OUTPUT`	[samma som inmatning]	Utgående rasterlager som innehåller resultatet
CRS auktoritetsidentifier	`CRS_AUTHID`	[crs]	Koordinatreferenssystemet för det utgående rasterlagret
Extent	`EXTENT`	[sträng]	Den spatiala omfattningen av det utgående rasterlagret
Bredd i pixlar	`WIDTH_IN_PIXEL`	[numerisk: heltal]	Antalet kolumner i det utgående rasterlagret
Höjd i pixlar	`HEIGHT_IN_PIXE`	[numerisk: heltal]	Antalet rader i det utgående rasterlagret
Totalt antal pixlar	`TOTAL_PIXEL_CO`	[numerisk: heltal]	Antalet pixlar i det utgående rasterlagret

Python-kod

Algoritm-ID: `native:fuzzifyrastersmallmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Större än frekvensen

Utvärderar cell för cell frekvensen (antalet gånger) som värdena i en inmatningsstapel av raster är lika med värdet i ett värderaster. Utgångsrastrets utsträckning och upplösning definieras av ingångsrasterlagret och är alltid av typen `Int32`.

Om flerbandsraster används i datarasterstacken kommer algoritmen alltid att utföra analysen på det första bandet i rastren - använd GDAL för att använda andra band i analysen. NoData-värdet för utdata kan ställas in manuellt.

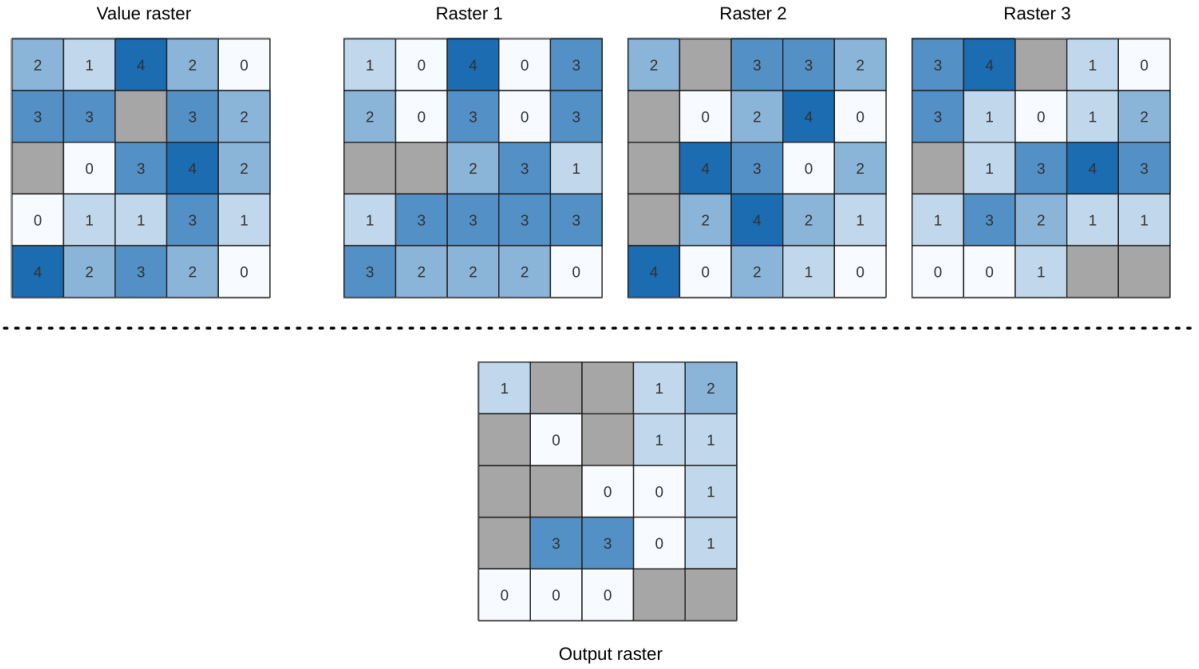


Fig. 24.24: För varje cell i utdatarastret representerar värdet det antal gånger som motsvarande celler i listan över raster är större än värdet raster. `NoData`-celler (grå) tas med i beräkningen.

Se även

Lika med frekvens, *Mindre än frekvensen*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsvärde raster	`INPUT_VALUE_RA`	[raster]	Ingångsvärdeslagret fungerar som referenslager för provlagren
Värde rasterband	`INPUT_VALUE_RA`	[rasterband] Standard: Det första bandet i rasterlagret	Välj det band du vill använda som prov
Rasterlager för inmatning	`INPUT_RASTERS`	[raster] [lista]	Rasterlager som ska utvärderas. Om flerbandsraster används i datarasterstacken kommer algoritmen alltid att utföra analysen på det första bandet i rastren
Ignorera NoData-värden	`IGNORE_NODATA`	[boolean] Standard: Falsk	Om det inte är markerat kommer alla NoData-celler i värderastret eller datalagerstapeln att resultera i en NoData-cell i utdatarastret
Utdatalager	`OUTPUT`	[samma som inmatning] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Utmatning av NoData-värde Valfritt	`OUTPUT_NO_DATA`	[numerisk: dubbel] Standard: -9999.0	Värde som ska användas för NoData i utdatalagret
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utdatalager	`OUTPUT`	[raster]	Utgående rasterlager som innehåller resultatet
CRS auktoritetsidentifier	`CRS_AUTHID`	[sträng]	Koordinatreferenssystemet för det utgående rasterlagret
Extent	`EXTENT`	[sträng]	Den spatiala omfattningen av det utgående rasterlagret
Antal celler med förekomster av lika värde	`FOUND_LOCATION`	[numerisk: heltal]	
Höjd i pixlar	`HEIGHT_IN_PIXE`	[numerisk: heltal]	Antalet rader i det utgående rasterlagret
Totalt antal pixlar	`TOTAL_PIXEL_CO`	[numerisk: heltal]	Antalet pixlar i det utgående rasterlagret
Medelfrekvens vid giltiga cellpositioner	`MEAN_FREQUENCY`	[numerisk: dubbel]	
Antal värdeförekomster	`OCCURRENCE_COU`	[numerisk: heltal]	
Bredd i pixlar	`WIDTH_IN_PIXEL`	[numerisk: heltal]	Antalet kolumner i det utgående rasterlagret

Python-kod

Algoritm-id: `native:greaterthanfrequency`

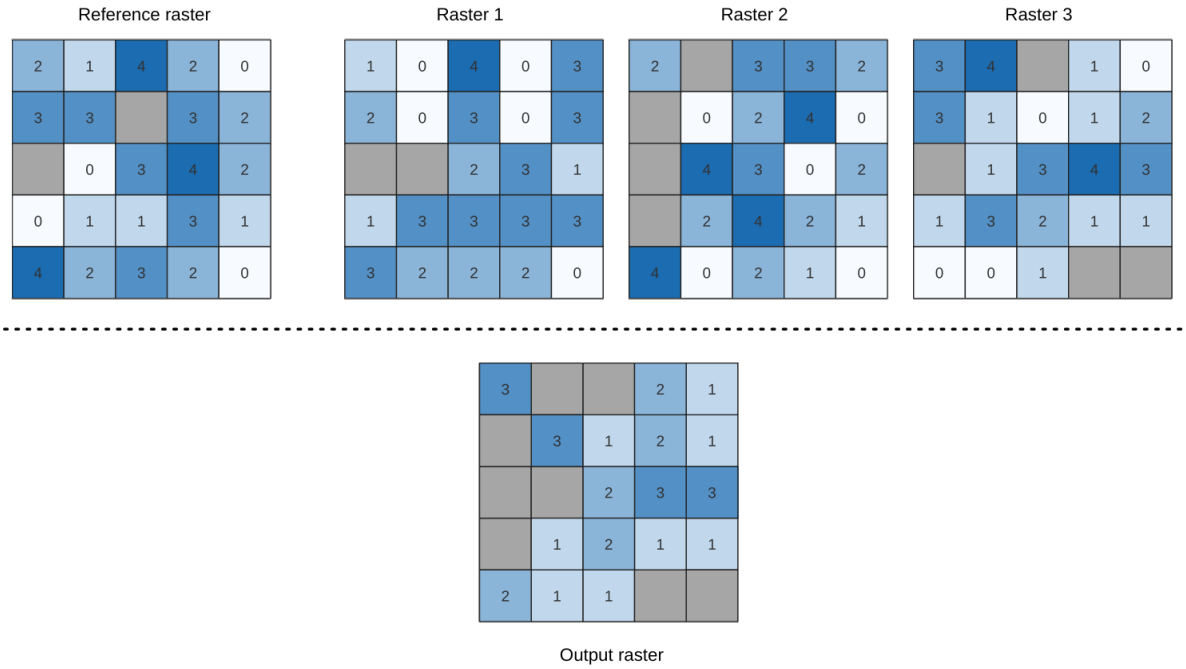
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Högsta positionen i rasterstacken

Utvärderar cell för cell positionen för det raster som har det högsta värdet i en stapel med raster. Positionsräkningarna börjar med 1 och sträcker sig till det totala antalet inmatningsraster. Ordningen på de ingående rastrerna är relevant för algoritmen. Om flera raster har det högsta värdet kommer det första rastret att användas för positionsvärdet.

Om flerbandsraster används i datarasterstacken kommer algoritmen alltid att utföra analysen på det första bandet i rastren - använd GDAL för att använda andra band i analysen. Alla NoData-celler i rasterlagerstacken kommer att resultera i en NoData-cell i utdatarastret om inte parametern ”ignore NoData” är markerad. NoData-värdet för utdata kan ställas in manuellt. Utdatarasterns utsträckning och upplösning definieras av ett referensrasterlager och är alltid av typen `Int32`.



Se även

Lägsta positionen i rasterstapeln

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Rasterlager för inmatning	`INPUT_RASTERS`	[raster] [lista]	Lista över rasterlager att jämföra med
Reference layer	`REFERENCE_LAYE`	[raster]	Referenslagret för skapandet av utdatalagret (utsträckning, CRS, pixeldimensioner)
Ignorera NoData-värden	`IGNORE_NODATA`	[boolean] Standard: Falsk	Om det inte är markerat kommer alla NoData-celler i datalagerstapeln att resultera i en NoData-cell i utdatarastern
Utdatalager	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utdatarastret som innehåller resultatet. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Utmatning av NoData-värde	`OUTPUT_NODATA_`	[numerisk: dubbel] Standard: -9999.0	Värde som ska användas för NoData i utdatalagret
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utdatalager	`OUTPUT`	[raster]	Utgående rasterlager som innehåller resultatet
CRS auktoritetsidentifier	`CRS_AUTHID`	[sträng]	Koordinatreferenssystemet för det utgående rasterlagret
Extent	`EXTENT`	[sträng]	Den spatiala omfattningen av det utgående rasterlagret
Bredd i pixlar	`WIDTH_IN_PIXEL`	[numerisk: heltal]	Antalet kolumner i det utgående rasterlagret
Höjd i pixlar	`HEIGHT_IN_PIXE`	[numerisk: heltal]	Antalet rader i det utgående rasterlagret
Totalt antal pixlar	`TOTAL_PIXEL_CO`	[numerisk: heltal]	Antalet pixlar i det utgående rasterlagret

Python-kod

Algoritm-id: `native:highestpositioninrasterstack`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Mindre än frekvensen

Utvärderar cell för cell frekvensen (antalet gånger) som värdena i en inmatningsstapel av raster är mindre än värdet i ett värderaster. Utgångsrastrets utsträckning och upplösning definieras av ingångsrasterlagret och är alltid av typen `Int32`.

Om flerbandsraster används i datarasterstacken kommer algoritmen alltid att utföra analysen på det första bandet i rastren - använd GDAL för att använda andra band i analysen. NoData-värdet för utdata kan ställas in manuellt.

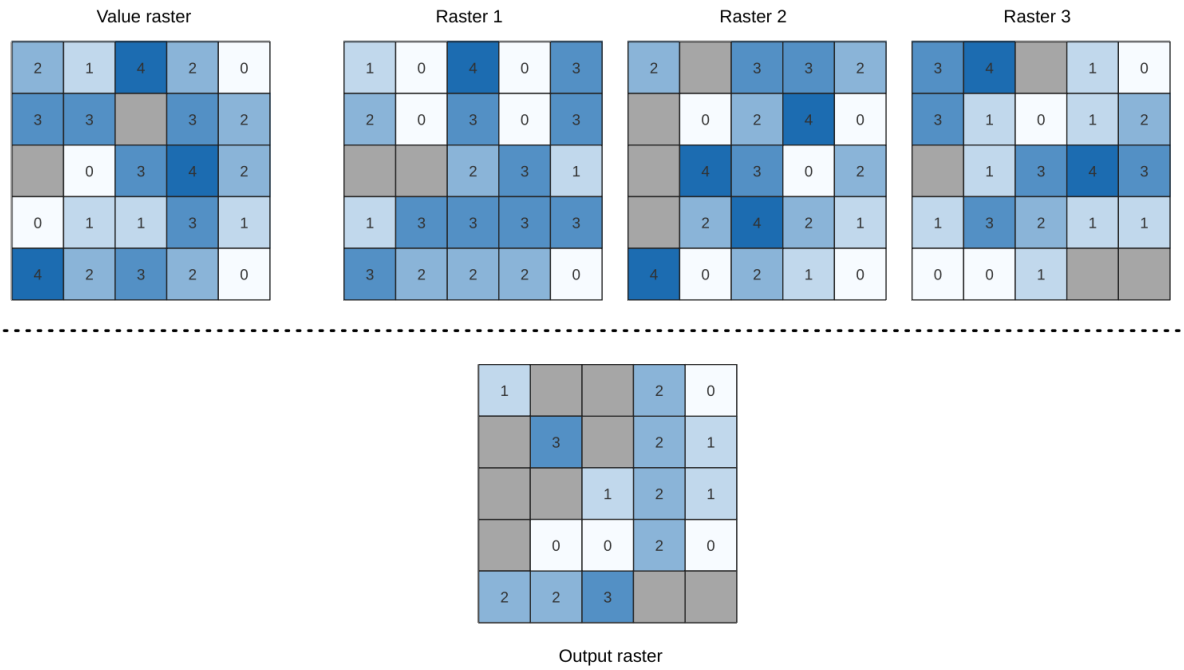


Fig. 24.25: För varje cell i utdatarastret representerar värdet det antal gånger som motsvarande celler i listan med raster är mindre än värdet raster. `NoData`-celler (grå) tas med i beräkningen.

Se även

Lika med frekvens, *Större än frekvensen*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsvärde raster	`INPUT_VALUE_RA`	[raster]	Ingångsvärdeslagret fungerar som referenslager för provlagren
Värde rasterband	`INPUT_VALUE_RA`	[rasterband] Standard: Det första bandet i rasterlagret	Välj det band du vill använda som prov
Rasterlager för inmatning	`INPUT_RASTERS`	[raster] [lista]	Rasterlager som ska utvärderas. Om flerbandsraster används i datarasterstacken kommer algoritmen alltid att utföra analysen på det första bandet i rastren
Ignorera NoData-värden	`IGNORE_NODATA`	[boolean] Standard: Falsk	Om det inte är markerat kommer alla NoData-celler i värderastret eller datalagerstapeln att resultera i en NoData-cell i utdatarastret
Utdatalager	`OUTPUT`	[samma som inmatning] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Utmatning av NoData-värde Valfritt	OUTPUT_NO_DATA	[numerisk: dubbel] Standard: -9999.0	Värde som ska användas för NoData i utdatalagret
Alternativ för skapande Valfritt	CREATE_OPTIONS	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utdatalager	OUTPUT	[raster]	Utgående rasterlager som innehåller resultatet
CRS auktoritetsidentifier	CRS_AUTHID	[sträng]	Koordinatreferenssystemet för det utgående rasterlagret
Extent	EXTENT	[sträng]	Den spatiala omfattningen av det utgående rasterlagret
Antal celler med förekomster av lika värde	FOUND_LOCATION	[numerisk: heltal]	
Höjd i pixlar	HEIGHT_IN_PIXE	[numerisk: heltal]	Antalet rader i det utgående rasterlagret
Totalt antal pixlar	TOTAL_PIXEL_CO	[numerisk: heltal]	Antalet pixlar i det utgående rasterlagret
Medelfrekvens vid giltiga cellpositioner	MEAN_FREQUENCY	[numerisk: dubbel]	
Antal värdeförekomster	OCCURRENCE_COU	[numerisk: heltal]	
Bredd i pixlar	WIDTH_IN_PIXEL	[numerisk: heltal]	Antalet kolumner i det utgående rasterlagret

Python-kod

Algoritm-id: `native:lessthanfrequency`

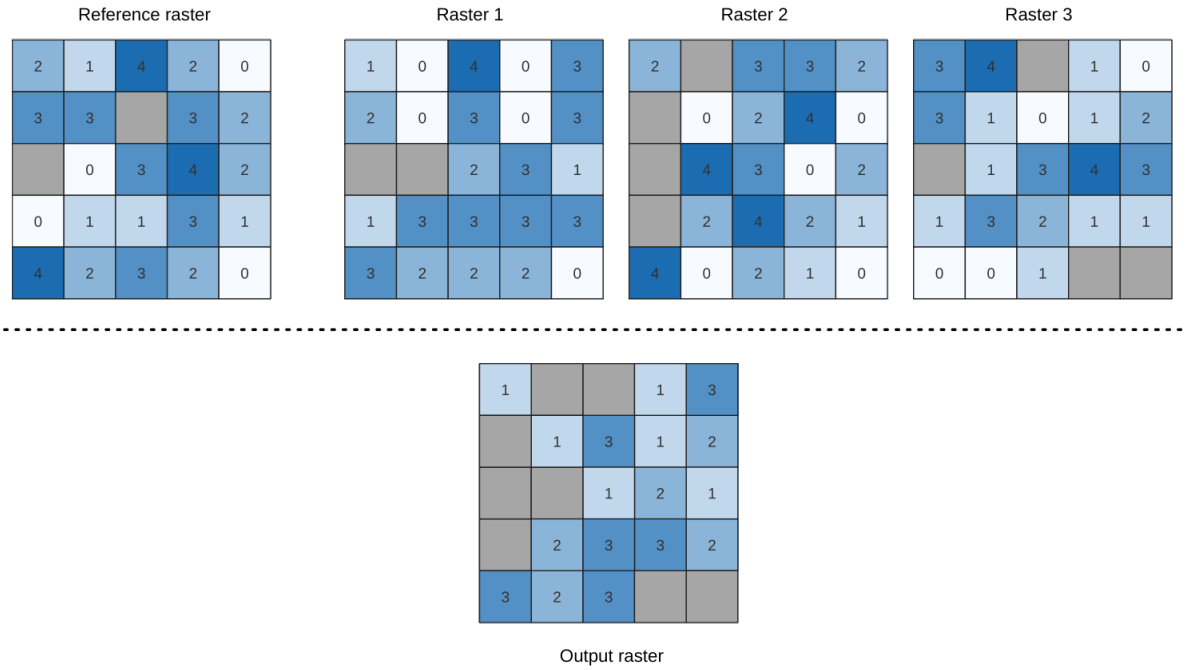
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Lägsta positionen i rasterstapeln

Utvärderar cell för cell positionen för det raster som har det lägsta värdet i en stapel med raster. Positionsräkningarna börjar med 1 och sträcker sig till det totala antalet inmatningsraster. Ordningen på de ingående rastrerna är relevant för algoritmen. Om flera raster har det lägsta värdet kommer det första rastret att användas för positionsvärdet.

Om flerbandsraster används i datarasterstacken kommer algoritmen alltid att utföra analysen på det första bandet i rastren - använd GDAL för att använda andra band i analysen. Alla NoData-celler i rasterlagerstacken kommer att resultera i en NoData-cell i utdatarastret om inte parametern ”ignore NoData” är markerad. NoData-värdet för utdata kan ställas in manuellt. Utdatarasterns utsträckning och upplösning definieras av ett referensrasterlager och är alltid av typen `Int32`.



Se även

Högsta positionen i rasterstacken

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Rasterlager för inmatning	`INPUT_RASTERS`	[raster] [lista]	Lista över rasterlager att jämföra med
Reference layer	`REFERENCE_LAYE`	[raster]	Referenslagret för skapandet av utdatalagret (utsträckning, CRS, pixeldimensioner)
Ignorera NoData-värden	`IGNORE_NODATA`	[boolean] Standard: Falsk	Om det inte är markerat kommer alla NoData-celler i datalagerstapeln att resultera i en NoData-cell i utdatarastern

continues on next page

Table 24.68 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Utdatalager	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utdatarastret som innehåller resultatet. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Utmatning av NoData-värde	`OUTPUT_NODATA_`	[numerisk: dubbel] Standard: -9999.0	Värde som ska användas för NoData i utdatalagret
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utdatalager	`OUTPUT`	[raster]	Utgående rasterlager som innehåller resultatet
CRS auktoritetsidentifier	`CRS_AUTHID`	[sträng]	Koordinatreferenssystemet för det utgående rasterlagret
Extent	`EXTENT`	[sträng]	Den spatiala omfattningen av det utgående rasterlagret
Bredd i pixlar	`WIDTH_IN_PIXEL`	[numerisk: heltal]	Antalet kolumner i det utgående rasterlagret
Höjd i pixlar	`HEIGHT_IN_PIXE`	[numerisk: heltal]	Antalet rader i det utgående rasterlagret
Totalt antal pixlar	`TOTAL_PIXEL_CO`	[numerisk: heltal]	Antalet pixlar i det utgående rasterlagret

Python-kod

Algoritm-id: `native:lowestpositioninrasterstack`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Raster boolean AND

Beräknar det booleska `AND` för en uppsättning inmatningsraster. Om alla indataraster har ett värde som inte är noll för en pixel, kommer den pixeln att sättas till `1` i utdatarastret. Om något av indatarastren har värden på `0` för pixeln kommer den att sättas till `0` i utdatarastret.

Parametern referenslager anger ett befintligt rasterlager som ska användas som referens när utdatarastern skapas. Utdatarastern kommer att ha samma utsträckning, CRS och pixeldimensioner som detta lager.

Som standard kommer en NoData-pixel i NÅGOT av inmatningslagren att resultera i en NoData-pixel i utdatarastret. Om alternativet *Behandla NoData-värden som falska* är markerat, behandlas NoData-ingångar på samma sätt som ett ingångsvärde på `0`.

Se även

Raster boolean OR

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster] [lista]	Lista över ingående rasterlager
Reference layer	`REF_LAYER`	[raster]	Referenslagret att skapa utdatalagret från (utsträckning, CRS, pixeldimensioner)
Behandla NoData-värden som falska	`NODATA_AS_FALS`	[boolean] Standard: Falsk	Behandla NoData-värden i indatafilerna som 0 när du utför operationen
Utdatalager	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utdatarastret som innehåller resultatet. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Utmatning av NoData-värde	`NO_DATA`	[numerisk: dubbel] Standard: -9999.0	Värde som ska användas för NoData i utdatalagret

continues on next page

Table 24.71 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Typ av utdata	DATA_TYPE	[uppräkning] Standard: 5	Typ av rasterdata som matas ut. Alternativ: • 0 — Byte (Åtta bitars osignerat heltal (quint8)) • 1 — Int16 (Sexton bitars signerat heltal (qint16)) • 2 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 3 — Int32 (Trettio tvåbitars signerade heltal (qint32)) • 4 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 5 — Float32 (trettio tvåbitars flyttal (float)) • 6 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 7 — CInt16 (Komplex Int16) • 8 — CInt32 (Komplex Int32) • 9 — CFloat32 (komplex float32) • 10 — CFloat64 (komplex float64) • 11 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)
Alternativ för skapande Valfritt	CREATE_OPTIONS	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).

Utgångar

Etikett	Namn	Typ	Beskrivning
Extent	EXTENT	[sträng]	Den spatiala omfattningen av det utgående rasterlagret
CRS auktoritetsidentifier	CRS_AUTHID	[crs]	Koordinatreferenssystemet för det utgående rasterlagret
Bredd i pixlar	WIDTH_IN_PIXEL	[numerisk: heltal]	Antalet kolumner i det utgående rasterlagret
Höjd i pixlar	HEIGHT_IN_PIXE	[numerisk: heltal]	Antalet rader i det utgående rasterlagret
Totalt antal pixlar	TOTAL_PIXEL_CO	[numerisk: heltal]	Antalet pixlar i det utgående rasterlagret
NoData pixelantal	NODATA_PIXEL_C	[numerisk: heltal]	Antalet NoData-pixlar i det utgående rasterlagret
Antal verkliga pixlar	TRUE_PIXEL_COU	[numerisk: heltal]	Antalet True-pixlar (värde = 1) i det utgående rasterlagret
Felaktigt pixelantal	FALSE_PIXEL_CO	[numerisk: heltal]	Antalet falska pixlar (värde = 0) i det utgående rasterlagret

continues on next page

Table 24.72 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Utdatalager	`OUTPUT`	[raster]	Utgående rasterlager som innehåller resultatet

Python-kod

Algoritm-ID: `native:rasterbooleanand`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Raster boolean OR

Beräknar det booleska `OR` för en uppsättning inmatningsraster. Om alla indataraster har ett nollvärde för en pixel, kommer den pixeln att sättas till `0` i utdatarastret. Om något av indatarastren har värden på `1` för pixeln kommer den att sättas till `1` i utdatarastret.

Parametern referenslager anger ett befintligt rasterlager som ska användas som referens när utdatarastern skapas. Utdatarastern kommer att ha samma utsträckning, CRS och pixeldimensioner som detta lager.

Som standard kommer en NoData-pixel i NÅGOT av inmatningslagren att resultera i en NoData-pixel i utdatarastret. Om alternativet *Behandla NoData-värden som falska* är markerat, behandlas NoData-ingångar på samma sätt som ett ingångsvärde på `0`.

Se även

Raster boolean AND

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster] [lista]	Lista över ingående rasterlager
Reference layer	`REF_LAYER`	[raster]	Referenslagret att skapa utdatalagret från (utsträckning, CRS, pixeldimensioner)
Behandla NoData-värden som falska	`NODATA_AS_FALS`	[boolean] Standard: Falsk	Behandla NoData-värden i indatafilerna som 0 när du utför operationen
Utdatalager	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utdatarastret som innehåller resultatet. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Utmatning av NoData-värde	NO_DATA	[numerisk: dubbel] Standard: -9999.0	Värde som ska användas för NoData i utdatalagret
Typ av utdata	DATA_TYPE	[uppräkning] Standard: 5	Typ av rasterdata som matas ut. Alternativ: • 0 — Byte (Åtta bitars osignerat heltal (quint8)) • 1 — Int16 (Sexton bitars signerat heltal (qint16)) • 2 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 3 — Int32 (Trettio tvåbitars signerade heltal (qint32)) • 4 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 5 — Float32 (trettio tvåbitars flyttal (float)) • 6 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 7 — CInt16 (Komplex Int16) • 8 — CInt32 (Komplex Int32) • 9 — CFloat32 (komplex float32) • 10 — CFloat64 (komplex float64) • 11 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)
Alternativ för skapande Valfritt	CREATE_OPTIONS	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).

Utgångar

Etikett	Namn	Typ	Beskrivning
Extent	EXTENT	[sträng]	Den spatiala omfattningen av det utgående rasterlagret
CRS auktoritetsidentifier	CRS_AUTHID	[crs]	Koordinatreferenssystemet för det utgående rasterlagret
Bredd i pixlar	WIDTH_IN_PIXEL	[numerisk: heltal]	Antalet kolumner i det utgående rasterlagret
Höjd i pixlar	HEIGHT_IN_PIXE	[numerisk: heltal]	Antalet rader i det utgående rasterlagret
Totalt antal pixlar	TOTAL_PIXEL_CO	[numerisk: heltal]	Antalet pixlar i det utgående rasterlagret
NoData pixelantal	NODATA_PIXEL_C	[numerisk: heltal]	Antalet NoData-pixlar i det utgående rasterlagret

continues on next page

Table 24.75 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Antal verkliga pixlar	`TRUE_PIXEL_COU`	[numerisk: heltal]	Antalet True-pixlar (värde = 1) i det utgående rasterlagret
Felaktigt pixelantal	`FALSE_PIXEL_CO`	[numerisk: heltal]	Antalet falska pixlar (värde = 0) i det utgående rasterlagret
Utdatalager	`OUTPUT`	[raster]	Utgående rasterlager som innehåller resultatet

Python-kod

Algoritm-id: `native:rasterbooleanor`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rasterkalkylator

Utför algebraiska operationer med hjälp av rasterlager.

Det resulterande lagret får sina värden beräknade enligt ett uttryck. Uttrycket kan innehålla numeriska värden, operatorer och referenser till något av lagren i det aktuella projektet.

Se även

Rasterkalkylator (virtuell), *Rasterkalkylator*, *Rasterkalkylator*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster] [lista]	Lista över ingående rasterlager
Uttryck	`EXPRESSION`	[uttryck]	*Rasterbaserat uttryck* som kommer att användas för att beräkna det utgående rasterlagret.

continues on next page

Table 24.76 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Utmatning omfattning Valfritt	`EXTENT`	[omfattning]	Ange den spatiala omfattningen för det utgående rasterlagret. Om omfattningen inte anges kommer den minsta omfattning som täcker alla valda referenslager att användas. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Cellstorlek för utmatning (lämna tomt för automatisk inställning) Valfritt	`CELLSIZE`	[numerisk: dubbel]	Cellstorlek för det utgående rasterlagret. Om cellstorleken inte anges kommer den minsta cellstorleken för det/de valda referenslagren att användas. Cellstorleken kommer att vara densamma för X- och Y-axlarna.
Utgång CRS Valfritt	`CRS`	[crs]	CRS för det utgående rasterlagret. Om CRS för utdata inte anges kommer CRS för det första referenslagret att användas.
Beräknad	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Beräknad	`OUTPUT`	[raster]	Utmatning av rasterfil med de beräknade värdena.

Python-kod

Algoritm-id: `native:rastercalc`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rasterkalkylator (virtuell)

Utför algebraiska operationer med hjälp av rasterlager och genererar resultat i minnet.

Det resulterande lagret får sina värden beräknade enligt ett uttryck. Uttrycket kan innehålla numeriska värden, operatorer och referenser till något av lagren i det aktuella projektet.

Ett virtuellt rasterlager är ett rasterlager som definieras av dess URI och vars pixlar beräknas i realtid. Det är inte en ny fil på disken; det virtuella lagret är fortfarande kopplat till de raster som används i beräkningen, vilket innebär att radering eller flyttning av dessa raster skulle förstöra det. Ett *Layernamn* kan anges, annars används beräkningsuttrycket som sådant. Om det virtuella lagret tas bort från projektet raderas det, och det kan göras beständigt i filen med hjälp av lagrets *Export* ► *Save as...* kontextuella meny.

Se även

Rasterkalkylator, *Rasterkalkylator*, *Rasterkalkylator*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster] [lista]	Lista över ingående rasterlager
Uttryck	`EXPRESSION`	[uttryck]	*Rasterbaserat uttryck* som kommer att användas för att beräkna det utgående rasterlagret.

continues on next page

Table 24.77 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Utmatning omfattning Valfritt	`EXTENT`	[omfattning]	Ange den spatiala omfattningen för det utgående rasterlagret. Om omfattningen inte anges kommer den minsta omfattning som täcker alla valda referenslager att användas. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Cellstorlek för utmatning (lämna tomt för automatisk inställning) Valfritt	`CELLSIZE`	[numerisk: dubbel]	Cellstorlek för det utgående rasterlagret. Om cellstorleken inte anges kommer den minsta cellstorleken för det/de valda referenslagren att användas. Cellstorleken kommer att vara densamma för X- och Y-axlarna.
Utgång CRS Valfritt	`CRS`	[crs]	CRS för det utgående rasterlagret. Om CRS för utdata inte anges kommer CRS för det första referenslagret att användas.
Namn på utmatningslager Valfritt	`LAYER_NAME`	[sträng]	Det namn som ska tilldelas det genererade lagret. Om det inte anges används texten i beräkningsuttrycket.

Utgångar

Etikett	Namn	Typ	Beskrivning
Beräknad	`OUTPUT`	[raster]	Utmatning av virtuellt rasterlager med de beräknade värdena.

Python-kod

Algoritm-id: `native:virtualrastercalc`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Egenskaper för rasterlager

Returnerar grundläggande egenskaper för det angivna rasterlagret, inklusive utsträckning, storlek i pixlar och pixeldimensioner (i kartenheter), antal band och NoData-värde.

Denna algoritm är avsedd att användas som ett sätt att extrahera dessa användbara egenskaper för att använda som ingångsvärden till andra algoritmer i en modell - t.ex. för att göra det möjligt att överföra ett befintligt rasters pixelstorlekar till en GDAL-rasteralgoritm.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatat rasterlager
Bandnummer Valfritt	`BAND`	[rasterband] Standard: Ej inställd	Om egenskaper för ett specifikt band också ska returneras. Om ett band anges returneras även noData-värdet för det valda bandet.

Utgångar

Etikett	Namn	Typ	Beskrivning
Antal band i rastret	`BAND_COUNT`	[numerisk: heltal]	Antalet band i rastret
CRS auktoritetsidentifier	`CRS_AUTHID`	[sträng]	Koordinatreferenssystemet för det utgående rasterlagret
Extent	`EXTENT`	[sträng]	Rasterlagrets utsträckning i CRS
Bandet har en NoData-värdeuppsättning	`HAS_NODATA_VAL`	[Boolean]	Anger om rasterlagret har ett värde inställt för NoData-pixlar i det valda bandet
Höjd i pixlar	`HEIGHT_IN_PIXE`	[numerisk: heltal]	Antalet kolumner i rasterlagret
Band NoData-värde	`NODATA_VALUE`	[numerisk: dubbel]	Värdet (om det är inställt) för NoData-pixlarna i det valda bandet
Pixelstorlek (höjd) i kartenheter	`PIXEL_HEIGHT`	[numerisk: heltal]	Vertikal storlek i kartenheter för pixeln
Pixelstorlek (bredd) i kartenheter	`PIXEL_WIDTH`	[numerisk: heltal]	Pixelns horisontella storlek i kartenheter
Bredd i pixlar	`WIDTH_IN_PIXEL`	[numerisk: heltal]	Antalet rader i rasterlagret
Maximal x-koordinat	`X_MAX`	[numerisk: dubbel]	
Minsta x-koordinat	`X_MIN`	[numerisk: dubbel]	
Maximal y-koordinat	`Y_MAX`	[numerisk: dubbel]	
Minsta y-koordinat	`Y_MIN`	[numerisk: dubbel]	

Python-kod

Algoritm-ID: `native:rasterlayerproperties`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Statistik för rasterlager

Beräknar grundläggande statistik från värdena i ett givet band i rasterlagret. Utdata laddas i menyn *Processing* ► *Results viewer*.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatat rasterlager
Bandnummer	`BAND`	[rasterband] Standard: Det första bandet i inmatningslagret	Om rastret är flerbandigt väljer du det band som du vill få statistik för.
Statistik	`OUTPUT_HTML_FI`	[html] Standard: `[Spara till temporär fil]`	Specifikation av utdatafilen: • Hoppa över utmatning • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Maximalt värde	`MAX`	[numerisk: dubbel]	
Medelvärde	`MEAN`	[numerisk: dubbel]	
Minsta värde	`MIN`	[numerisk: dubbel]	

continues on next page

Table 24.78 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Statistik	`OUTPUT_HTML_FI`	[html]	Utdatafilen innehåller följande information: • Analyserad fil: sökväg för rasterlagret • Minimivärde: rastrets minimivärde • Maximalt värde: rastrets maximala värde • Intervall: skillnaden mellan max- och minvärdena • Summa: total summa av värdena • Medelvärde: medelvärdet av värdena • Standardavvikelse: standardavvikelse för värdena • Kvadratsumma: summan av de kvadratiska skillnaderna för varje observation från det totala medelvärdet
Omfång	`RANGE`	[numerisk: dubbel]	
Standardavvikelse	`STD_DEV`	[numerisk: dubbel]	
Summa	`SUM`	[numerisk: dubbel]	
Summan av kvadraterna	`SUM_OF_SQUARES`	[numerisk: dubbel]	

Python-kod

Algoritm-ID: `native:rasterlayerstatistics`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rapport om unika värden för rasterlager

Returnerar antal och area för varje unikt värde i ett givet rasterlager. Beräkningen av arean görs i areaenheten för lagrets CRS.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatat rasterlager
Bandnummer	`BAND`	[rasterband] Standard: Det första bandet i inmatningslagret	Om rastret är flerbandigt väljer du det band som du vill få statistik för.

continues on next page

Table 24.79 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Rapport om unika värden	`OUTPUT_HTML_FI`	&Arkiv Standard: `[Spara till temporär fil]`	Specifikation av utdatafilen: • Hoppa över utmatning • Spara till en temporär fil • Spara till fil…
Tabell över unika värden	`OUTPUT_TABLE`	[vektor: tabell] Standard: `[Hoppa över utdata]`	Specifikation av tabellen för unika värden: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
CRS auktoritetsidentifier	`CRS_AUTHID`	[sträng]	Koordinatreferenssystemet för det utgående rasterlagret
Extent	`EXTENT`	[sträng]	Den spatiala omfattningen av det utgående rasterlagret
Höjd i pixlar	`HEIGHT_IN_PIXE`	[numerisk: heltal]	Antalet rader i det utgående rasterlagret
NoData pixelantal	`NODATA_PIXEL_C`	[numerisk: heltal]	Antalet NoData-pixlar i det utgående rasterlagret
Totalt antal pixlar	`TOTAL_PIXEL_CO`	[numerisk: heltal]	Antalet pixlar i det utgående rasterlagret
Rapport om unika värden	`OUTPUT_HTML_FI`	[html]	HTML-filen som skickas ut innehåller följande information: • Analyserad fil: sökvägen till rasterlagret • Utbredning: xmin, ymin, xmax, ymax koordinater för utbredningen • Projektion: projektion av lagret • Bredd i pixlar: antal kolumner och storlek på pixelbredd • Höjd i pixlar: antal rader och pixelbreddstorlek • Totalt antal pixlar: antal av alla pixlar • NoData pixel count: antal pixlar med NoData-värde
Tabell över unika värden	`OUTPUT_TABLE`	[vektor: tabell]	En tabell med tre kolumner: • *value*: pixelvärde • *count*: antal pixlar med detta värde • *m2* eller *deg2* eller *ft2* eller … : total yta av pixlar med detta värde. Kolumnnamnet beror på areaenheten i lagrets CRS och beräkningen görs i den enheten.
Bredd i pixlar	`WIDTH_IN_PIXEL`	[numerisk: heltal]	Antalet kolumner i det utgående rasterlagret

Python-kod

Algoritm-ID: `native:rasterlayeruniquevaluesreport`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Zonal statistik för rasterlager

Beräknar statistik för ett rasterlagers värden, kategoriserade efter zoner som definierats i ett annat rasterlager.

Se även

Zonvis statistik

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatat rasterlager
Bandnummer	`BAND`	[rasterband] Standard: Det första bandet i rasterlagret	Om rastret är flerbandigt väljer du det band som du vill beräkna statistiken för.
Zoner lager	`ZONES`	[raster]	Rasterlager som definierar zoner. Zonerna består av sammanhängande pixlar som har samma pixelvärde.
Zoner bandnummer	`ZONES_BAND`	[rasterband] Standard: Det första bandet i rasterlagret	Om rastret är flerbandigt väljer du det band som definierar zonerna
Statistik	`OUTPUT_TABLE`	[vektor: tabell] Standard: `[Skapa tillfälligt lager]`	Specifikation av utdatarapporten. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Reference layer Valfritt	REF_LAYER	[uppräkning] Standard: 0	Rasterlager som används för att beräkna de centroider som kommer att användas som referens när zonerna i utdatalagret bestäms. Ett av följande: • 0 — Inmatningslager: zonerna bestäms genom provtagning av zonrasterskiktets värde i centroiden för varje pixel från källrasterskiktet • 1 — Zonlager: det ingående rasterlagret kommer att samplas vid centroiden för varje pixel från zonrasterlagret

Utgångar

Etikett	Namn	Typ	Beskrivning
CRS auktoritetsidentifier	CRS_AUTHID	[sträng]	Koordinatreferenssystemet för det utgående rasterlagret
Extent	EXTENT	[sträng]	Den spatiala omfattningen av det utgående rasterlagret
Höjd i pixlar	HEIGHT_IN_PIXE	[numerisk: heltal]	Antalet rader i det utgående rasterlagret
NoData pixelantal	NODATA_PIXEL_C	[numerisk: heltal]	Antalet NoData-pixlar i det utgående rasterlagret
Statistik	OUTPUT_TABLE	[vektor: tabell]	Utmatningslagret innehåller följande information **för varje zon**: • Area (yta): ytan i kvadratiska rasterenheter i zonen; • Sum: den totala summan av pixelvärdena i zonen; • Count: antalet pixlar i zonen; • Min: det lägsta pixelvärdet i zonen; • Max: det maximala pixelvärdet i zonen; • Mean: medelvärdet av pixelvärdena i zonen;
Totalt antal pixlar	TOTAL_PIXEL_CO	[numerisk: heltal]	Antalet pixlar i det utgående rasterlagret
Bredd i pixlar	WIDTH_IN_PIXEL	[numerisk: heltal]	Antalet kolumner i det utgående rasterlagret

Python-kod

Algoritm-ID: `native:rasterlayerzonalstats`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rasterytans volym

Beräknar volymen under en rasteryta i förhållande till en given basnivå. Detta är främst användbart för digitala höjdmodeller (DEM).

Parametrar

Etikett	Namn	Typ	Beskrivning
INPUT-lager	`INPUT`	[raster]	Ingångsraster som representerar en yta
Bandnummer	`BAND`	[rasterband] Standard: Det första bandet i rasterlagret	Om rastret är flerbandigt väljer du det band som ska definiera ytan.
Basisnivå	`LEVEL`	[numerisk: dubbel] Standard: 0,0	Definiera en bas eller ett referensvärde. Denna bas används vid volymberäkningen enligt parametern `Metod` (se nedan).
Metod	`METHOD`	[uppräkning] Standard: 0	Definiera metoden för volymberäkningen som ges av skillnaden mellan rasterpixelvärdet och `Base level`. Alternativ för: • 0 — Räkna endast över basnivån: endast pixlar över basnivån kommer att läggas till volymen. • 1 — Räkna endast under basnivån: endast pixlar under basnivån kommer att läggas till volymen. • 2 — Subtrahera volymer under basnivån: pixlar över basnivån läggs till volymen, pixlar under basnivån subtraheras från volymen. • 3 — Lägg till volymer under basnivån: Lägg till volymen oavsett om pixeln ligger över eller under basnivån. Detta motsvarar summan av de absoluta värdena av skillnaden mellan pixelvärdet och basnivån.

continues on next page

Table 24.84 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Rapport om ytvolym	`OUTPUT_HTML_FI`	[html] Standard: `[Spara till temporär fil]`	Specifikation av HTML-utdatarapporten. *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Tabell över ytvolym	`OUTPUT_TABLE`	[vektor: tabell] Standard: `[Hoppa över utdata]`	Specifikation av utmatningstabellen. *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Volym	`VOLUME`	[numerisk: dubbel]	Den beräknade volymen
Område	`AREA`	[numerisk: dubbel]	Området i kvadratiska kartenheter
Pixel_antal	`PIXEL_COUNT`	[numerisk: heltal]	Det totala antalet pixlar som har analyserats
Rapport om ytvolym	`OUTPUT_HTML_FI`	[html]	Utdatarapport (med volym, yta och pixelantal) i HTML-format
Tabell över ytvolym	`OUTPUT_TABLE`	[vektor: tabell]	Utmatningstabellen (som innehåller volym, yta och pixelantal)

Python-kod

Algoritm-ID: `native:rastersurfacevolym`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Omklassificering per lager

Omklassificerar ett rasterband genom att tilldela nya klassvärden baserat på de intervall som anges i en vektortabell.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Ett rasterlager	`INPUT_RASTER`	[raster]	Rasterlager att omklassificera
Bandnummer	`RASTER_BAND`	[rasterband] Standard: Det första bandet i rasterlagret	Om rastret är flerbandigt väljer du det band som du vill omklassificera.
Lager som innehåller klassresor	`INPUT_TABLE`	[vektor: vilken som helst]	Vektorlager som innehåller de värden som ska användas för klassificering.
Minsta fält för klassvärde	`MIN_FIELD`	[tabellfält: numerisk]	Fält med det lägsta värdet i intervallet för klassen. Använd `-inf` för att fånga det lägsta möjliga värdet.
Maximalt klassvärdesfält	`MAX_FIELD`	[tabellfält: numerisk]	Fält med det högsta värdet i intervallet för klassen. Använd `inf` för att fånga det högsta möjliga värdet.
Fält för utdatavärde	`VALUE_FIELD`	[tabellfält: numerisk]	Fält med det värde som kommer att tilldelas de pixlar som faller inom klassen (mellan motsvarande min- och maxvärden). Använd `nan` för att ställa in värdet för intervallet till NoData.
Omklassificerat raster	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Utmatning av NoData-värde	`NO_DATA`	[numerisk: dubbel] Standard: -9999.0	Värde som ska tillämpas på NoData-värden.
Range-gränser	`RANGE_BOUNDARI`	[uppräkning] Standard: 0	Definierar jämförelseregler för klassificeringen. Alternativ: • 0 — min < värde <= max • 1 — min <= värde < max • 2 — min <= värde <= max • 3 — min < värde < max
Använd NoData när inget intervall matchar värdet	`NODATA_FOR_MIS`	[boolean] Standard: Falsk	Tillämpar NoData-värdet på bandvärden som inte tillhör någon klass. Om False, behålls det ursprungliga värdet.

continues on next page

Table 24.86 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Typ av utdata	DATA_TYPE	[uppräkning] Standard: 5	Definierar formatet för den utgående rasterfilen. Alternativ för: • 0 — Byte (Åtta bitars osignerat heltal (quint8)) • 1 — Int16 (Sexton bitars signerat heltal (qint16)) • 2 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 3 — Int32 (Trettio tvåbitars signerade heltal (qint32)) • 4 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 5 — Float32 (trettio tvåbitars flyttal (float)) • 6 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 7 — CInt16 (Komplex Int16) • 8 — CInt32 (Komplex Int32) • 9 — CFloat32 (komplex float32) • 10 — CFloat64 (komplex float64) • 11 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)
Alternativ för skapande Valfritt	CREATE_OPTIONS	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).

Utgångar

Etikett	Namn	Typ	Beskrivning
Omklassificerat raster	OUTPUT	[raster]	Utmatat rasterlager med omklassificerade bandvärden

Python-kod

Algoritm-ID: `native:reclassifybylayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Omklassificering per tabell

Omklassificerar ett rasterband genom att tilldela nya klassvärden baserat på de intervall som anges i en fast tabell.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Ett rasterlager	`INPUT_RASTER`	[raster]	Rasterlager att omklassificera
Bandnummer	`RASTER_BAND`	[rasterband] Standard: 1	Rasterband för vilket du vill räkna om värden.
Omklassificeringsta	`TABLE`	[vektor: tabell]	En tabell med 3 kolumner som ska fyllas med de värden som anger gränserna för varje klass (`Minimum` och `Maximum`) och det nya `Value` som ska tilldelas de bandvärden som faller inom klassen. Värdet `-inf` kan användas som minimum, `inf` som maximum och `nan` kan användas för att ställa in utdatavärdet till NoData.
Omklassificerat raster	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av rasterlagret för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Utmatning av NoData-värde	`NO_DATA`	[numerisk: dubbel] Standard: -9999.0	Värde som ska tillämpas på NoData-värden.
Range-gränser	`RANGE_BOUNDARI`	[uppräkning] Standard: 0	Definierar jämförelseregler för klassificeringen. Alternativ: • 0 — min < värde <= max • 1 — min <= värde < max • 2 — min <= värde <= max • 3 — min < värde < max

continues on next page

Table 24.88 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Använd NoData när inget intervall matchar värdet	NODATA_FOR_MIS	[boolean] Standard: Falsk	Tillämpar NoData-värdet på bandvärden som inte tillhör någon klass. Om False, behålls det ursprungliga värdet.
Typ av utdata	DATA_TYPE	[uppräkning] Standard: 5	Definierar formatet för den utgående rasterfilen. Alternativ för: • 0 — Byte (Åtta bitars osignerat heltal (quint8)) • 1 — Int16 (Sexton bitars signerat heltal (qint16)) • 2 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 3 — Int32 (Trettio tvåbitars signerade heltal (qint32)) • 4 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 5 — Float32 (trettio tvåbitars flyttal (float)) • 6 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 7 — CInt16 (Komplex Int16) • 8 — CInt32 (Komplex Int32) • 9 — CFloat32 (komplex float32) • 10 — CFloat64 (komplex float64) • 11 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)
Alternativ för skapande Valfritt	CREATE_OPTIONS	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).

Utgångar

Etikett	Namn	Typ	Beskrivning
Omklassificerat raster	OUTPUT	[raster]	Utmatat rasterlager med omklassificerade bandvärden

Python-kod

Algoritm-ID: `native:reclassifybytable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Omskalning av raster

Omskalerar rasterlagret till ett nytt värdeintervall, samtidigt som formen (fördelningen) på rastrets histogram (pixelvärden) bevaras. Ingångsvärden mappas med hjälp av en linjär interpolering från källrastrets lägsta och högsta pixelvärden till målrastrets lägsta och högsta pixelintervall.

Som standard bevarar algoritmen det ursprungliga NoData-värdet, men det finns en möjlighet att åsidosätta det.

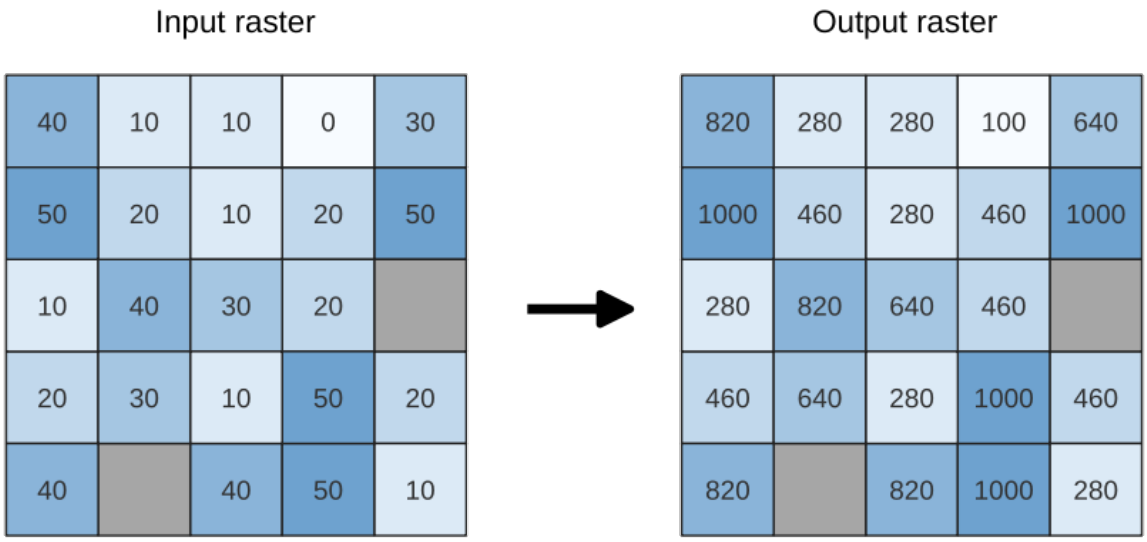


Fig. 24.26: Omskalning av värden för ett rasterlager från [0 - 50] till [100 - 1000]

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Input Raster	`INPUT`	[raster]	Rasterlager att använda för omskalning
Bandnummer	`BAND`	[rasterband] Standard: Det första bandet i inmatningslagret	Om rastret är multibandigt väljer du ett band.
Nytt minimivärde	`MINIMUM`	[numerisk: dubbel] Standardvärde: 0,0	Lägsta pixelvärde som ska användas i det skalade lagret
Nytt maximalt värde	`MAXIMUM`	[numerisk: dubbel] Standardvärde: 255.0	Maximalt pixelvärde att använda i det omskalade lagret
Nytt NoData-värde Valfritt	`NODATA`	[numerisk: dubbel] Standardvärde: Ej inställt	Värde som ska tilldelas NoData-pixlarna. Om den inte är inställd bevaras de ursprungliga NoData-värdena.
Rescaled	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av rasterlagret för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Rescaled	`OUTPUT`	[raster]	Utmatning av rasterlager med omskalade bandvärden

Python-kod

Algoritm-ID: `native:rescaleraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rund raster

Avrundar cellvärdena i en rasterdatauppsättning enligt det angivna antalet decimaler.

Alternativt kan ett negativt antal decimaler användas för att avrunda värden till potenser av basen n. Till exempel, med ett basvärde n på 10 och decimaler på -1, avrundar algoritmen cellvärden till multiplar av 10, -2 avrundar till multiplar av 100, och så vidare. Man kan välja godtyckliga basvärden, men algoritmen tillämpar samma multiplikativa princip. Avrundning av cellvärden till multiplar av en bas n kan användas för att generalisera rasterlager.

Algoritmen bevarar datatypen för inmatningsrastret. Därför kan byte/integer-raster endast avrundas till multiplar av basen n, annars utfärdas en varning och rastret kopieras som byte/integer-raster.

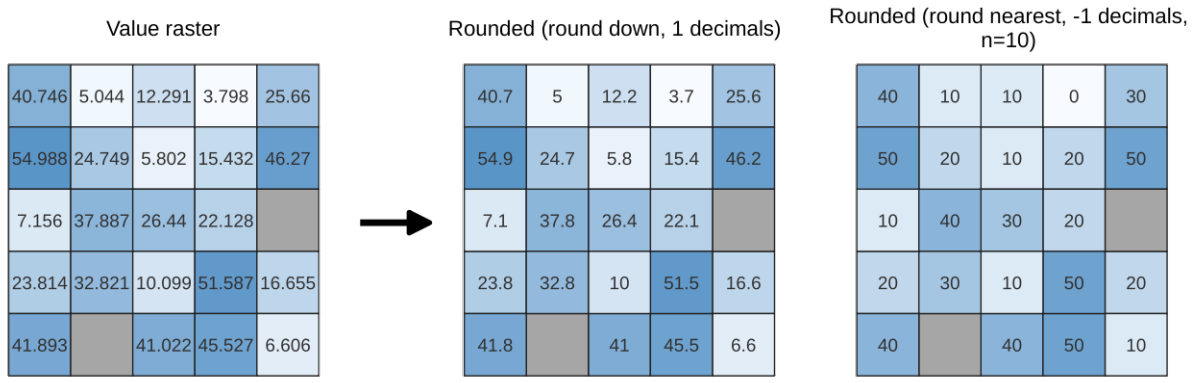


Fig. 24.27: Avrundning av värden i ett raster

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Ingångsraster	`INPUT`	[raster]	Rastret till processen.
Bandnummer	`BAND`	[rasterband] Standard: 1	Rasterns band
Rundgångsriktning	`ROUNDING_DIREC`	Konfigurationens id %1 finns redan i listan Standard: 1	Hur man väljer målet avrundat värde. Alternativen är: • 0 — Runda upp • 1 — Avrundat till närmaste • 2 — Runda ner

continues on next page

Table 24.90 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Antal decimaler	`DECIMAL_PLACES`	[numerisk: heltal] Standard: 2	Antal decimaler att avrunda till. Använd negativa värden för att avrunda cellvärden till en multipel av en bas n
Utgångsraster	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utdatafilen. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Bas n för avrundning till multiplar av n	`BASE_N`	[numerisk: heltal] Standard: 10	När parametern `DECIMAL_PLACES` är negativ avrundas rastervärdena till multiplar av basvärdet n
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utgångsraster	`OUTPUT`	[raster]	Det utgående rasterlagret med avrundade värden för det valda bandet.

Python-kod

Algoritm-ID: `native:roundrastervalues`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exempel på rastervärden

Extraherar rastervärden på punktplatserna. Om rasterlagret är flerbandigt samplas varje band.

Attributtabellen för det resulterande lagret kommer att ha lika många nya kolumner som rasterlagrets bandantal.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: punkt]	Punktvektorlager att använda för provtagning
Ett rasterlager	`RASTERCOPY`	[raster]	Rasterlager som ska provtas vid de angivna punktlägena.
Prefix för utdatakolumn	`COLUMN_PREFIX`	[sträng] Standard: 'SAMPLE_'	Prefix för namnen på de kolumner som läggs till.
Samplad Valfritt	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret som innehåller de samplade värdena. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Samplad	`OUTPUT`	[vektor: punkt]	Utmatningslagret som innehåller de samplade värdena.

Python-kod

Algoritm-id: `native:rastersampling`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Zonalt histogram

Lägger till fält som representerar räkningar av varje unikt värde från ett rasterlager som ingår i polygonfunktioner.

Attributtabellen för utdatalagret kommer att ha lika många fält som de unika värdena för det rasterlager som skär polygonen/polygonerna.

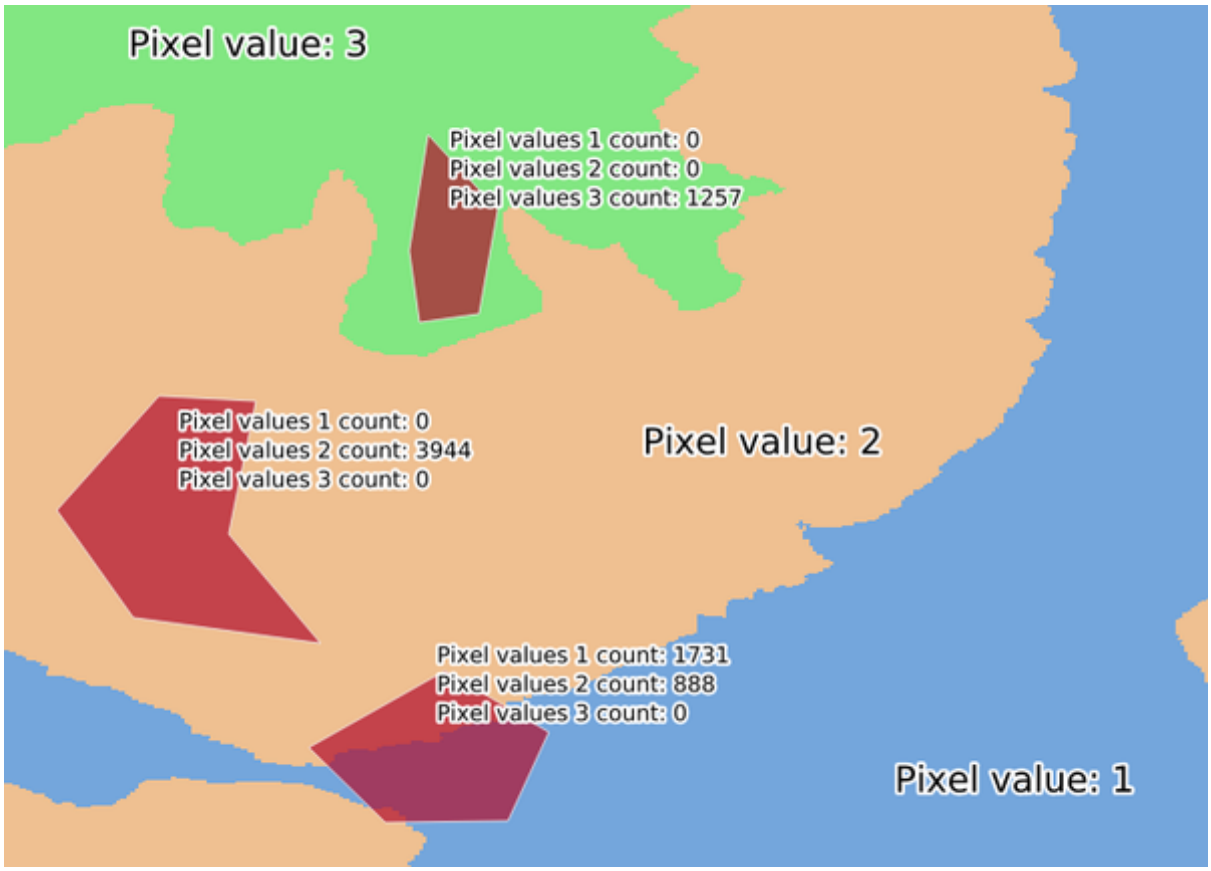


Fig. 24.28: Exempel på histogram för rasterlager

Parametrar

Etikett	Namn	Typ	Beskrivning
Ett rasterlager	`INPUT_RASTER`	[raster]	Inmatat rasterlager.
Bandnummer	`RASTER_BAND`	[rasterband] Standard: Det första bandet i inmatningslagret	Om rastret är multibandigt väljer du ett band.
Vektorlager som innehåller zoner	`INPUT_VECTOR`	[vektor: polygon]	Vektorpolygonlager som definierar zonerna.
Prefix för utdatakolumn	`COLUMN_PREFIX` Valfritt	[sträng] Standard: ”HISTO_	Prefix för utdatakolumnernas namn.
Utgångszoner	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange polygonlagret för utdatavektorn. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Utgångszoner	`OUTPUT`	[vektor: polygon]	Polygonlager för utmatningsvektor.

Python-kod

Algoritm-ID: `native:zonalhistogram`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Zonvis statistik

Beräknar statistik för ett rasterlager för varje egenskap i ett överlappande polygonvektorlager.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: polygon]	Vektorpolygonlager som innehåller zonerna.
Ett rasterlager	`INPUT_RASTER`	[raster]	Inmatat rasterlager.
Rasterband	`RASTER_BAND`	[rasterband] Standard: Det första bandet i inmatningslagret	Om rastret är flerbandigt väljer du ett band för statistiken.
Prefix för utdatakolumn	`COLUMN_PREFIX`	[sträng] Standard: '_'	Prefix för utdatakolumnernas namn.
Statistik att beräkna	`STATISTICS`	[uppräkning] [lista] Standard: [0,1,2]	Lista över statistiska operatorer för utdata. Alternativ: • 0 — Räkna • 1 — Summa • 2 — Medelvärde • 3 — Median • 4 — St. dev. • 5 — Minimum • 6 — Maximalt • 7 — Område • 8 — Minoritet • 9 — Majoritet • 10 — Variationsrikedom • 11 — Avvikelse
Zonal statistik	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange polygonlagret för utdatavektorn. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Zonal statistik	`OUTPUT`	[vektor: polygon]	Zonens vektorlager med extra statistik.

Python-kod

Algoritm-ID: `native:zonalstatisticsfb`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.16 Skapande av raster

Skapa konstant rasterlager

Skapar ett rasterlager för en given utsträckning och cellstorlek som fylls med det angivna värdet.

Dessutom kan en datatyp för utdata specificeras. Algoritmen avbryts om ett värde har angetts som inte kan representeras av den valda rasterdatatypen för utdata.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Önskad omfattning	`EXTENT`	[omfattning]	Ange den spatiala omfattningen av det utgående rasterlagret. Det kommer internt att utökas till en multipel av kakelstorleken. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Mål CRS	`TARGET_CRS`	[crs] Standard: Projekt CRS	CRS för det utgående rasterlagret
Pixelstorlek	`PIXEL_SIZE`	[numerisk: dubbel] Standard: 1,0	Pixelstorlek (X=Y) i kartenheter.
Konstant värde	`NUMBER`	[numerisk: dubbel] Standard: 1,0	Konstant pixelvärde för det utgående rasterlagret.
konstant	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Typ av rasterdata vid utmatning	`OUTPUT_TYPE` Standard: 5	[uppräkning]	Definierar datatypen för den utgående rasterfilen. Alternativ för: • 0 — Byte • 1 — heltal16 • 2 — Osignerat heltal16 • 3 — Integer32 • 4 — Obetecknad heltal32 • 5 — Float32 • 6 — Float64
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).

Utgångar

Etikett	Namn	Typ	Beskrivning
konstant	`OUTPUT`	[raster]	Raster som täcker önskad utsträckning med angiven pixelstorlek och värde.

Python-kod

Algoritm-ID: `native:skapakonstantrasterlager`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa slumpmässigt rasterlager (binomial fördelning)

Genererar ett rasterlager för given utsträckning och cellstorlek fyllt med binomiskt fördelade slumpmässiga värden.

Som standard kommer värdena att väljas med ett N på 10 och en sannolikhet på 0,5. Detta kan åsidosättas genom att använda den avancerade parametern för N och sannolikhet. Rasterdatatypen är inställd på Integer-typer (Integer16 som standard). Binomialfördelningens slumpmässiga värden definieras som positiva heltal. Ett raster med flyttal representerar en omräkning av heltalsvärden till flyttal.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Önskad omfattning	`EXTENT`	[omfattning]	Ange den spatiala omfattningen av det utgående rasterlagret. Det kommer internt att utökas till en multipel av kakelstorleken. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Mål CRS	`TARGET_CRS`	[crs] Standard: Projekt CRS	CRS för det utgående rasterlagret
Pixelstorlek	`PIXEL_SIZE`	[numerisk: dubbel] Standard: 1,0	Pixelstorlek (X=Y) i kartenheter.
Utgångsraster	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Typ av rasterdata vid utmatning	`OUTPUT_TYPE` Standard: 0	[uppräkning]	Definierar datatypen för den utgående rasterfilen. Alternativ för: • 0 — heltal16 • 1 — Osignerat heltal16 • 2 — Integer32 • 3 — Osignerat heltal32 • 4 — Float32 • 5 — Float64
N	`N`	[numerisk: heltal] Standard: 10	
Sannolikhet	`PROBABILITY`	[numerisk: dubbel] Standard: 0,5	
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utgångsraster	`OUTPUT`	[raster]	Raster som täcker önskad utsträckning med cellstorleken fylld med slumpmässiga värden

Python-kod

Algoritm-ID: `native:skapareochombinomialrasterlager`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa slumpmässigt rasterlager (exponentiell fördelning)

Genererar ett rasterlager för given utsträckning och cellstorlek fyllt med exponentiellt fördelade slumpmässiga värden.

Som standard kommer värdena att väljas med ett lambda på 1,0. Detta kan åsidosättas genom att använda den avancerade parametern för lambda. Rasterdatatypen är inställd på Float32 som standard eftersom de slumpmässiga värdena för exponentiell distribution är flyttal.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Önskad omfattning	`EXTENT`	[omfattning]	Ange den spatiala omfattningen av det utgående rasterlagret. Det kommer internt att utökas till en multipel av kakelstorleken. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Mål CRS	`TARGET_CRS`	[crs] Standard: Projekt CRS	CRS för det utgående rasterlagret
Pixelstorlek	`PIXEL_SIZE`	[numerisk: dubbel] Standard: 1,0	Pixelstorlek (X=Y) i kartenheter.
Utgångsraster	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Typ av rasterdata vid utmatning	`OUTPUT_TYPE` Standard: 0	[uppräkning]	Definierar datatypen för den utgående rasterfilen. Alternativ för: • 0 — Float32 • 1 — Float64
Lambda	`LAMBDA`	[numerisk: dubbel] Standard: 1,0	
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utgångsraster	`OUTPUT`	[raster]	Raster som täcker önskad utsträckning med cellstorleken fylld med slumpmässiga värden

Python-kod

Algoritm-ID: `` `native:skapareochomexponentielltrasterlager ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa slumpmässigt rasterlager (gammafördelning)

Genererar ett rasterlager för given utsträckning och cellstorlek fyllt med gammafördelade slumpmässiga värden.

Som standard kommer värdena att väljas med ett alfa- och betavärde på 1,0. Detta kan åsidosättas genom att använda den avancerade parametern för alfa och beta. Rasterdatatypen är inställd på Float32 som standard eftersom de slumpmässiga värdena för gammadistributionen är flyttal.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Önskad omfattning	`EXTENT`	[omfattning]	Ange den spatiala omfattningen av det utgående rasterlagret. Det kommer internt att utökas till en multipel av kakelstorleken. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Mål CRS	`TARGET_CRS`	[crs] Standard: Projekt CRS	CRS för det utgående rasterlagret
Pixelstorlek	`PIXEL_SIZE`	[numerisk: dubbel] Standard: 1,0	Pixelstorlek (X=Y) i kartenheter.
Utgångsraster	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Typ av rasterdata vid utmatning	`OUTPUT_TYPE` Standard: 0	[uppräkning]	Definierar datatypen för den utgående rasterfilen. Alternativ för: • 0 — Float32 • 1 — Float64
Alpha	`ALPHA`	[numerisk: dubbel] Standard: 1,0	
Beta	`BETA`	[numerisk: dubbel] Standard: 1,0	
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utgångsraster	`OUTPUT`	[raster]	Raster som täcker önskad utsträckning med cellstorleken fylld med slumpmässigt fördelade värden

Python-kod

Algoritm-ID: `` `native:createrandomgammarasterlayer ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa slumpmässigt rasterlager (geometrisk fördelning)

Genererar ett rasterlager för given utsträckning och cellstorlek fyllt med geometriskt fördelade slumpmässiga värden.

Som standard kommer värdena att väljas med en sannolikhet på 0,5. Detta kan åsidosättas genom att använda den avancerade parametern för medelvärde. Rasterdatatypen är inställd på Integer-typer (Integer16 som standard). De slumpmässiga värdena för geometrisk distribution definieras som positiva heltal. Ett raster med flyttal representerar en omräkning av heltalsvärden till flyttal.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Önskad omfattning	`EXTENT`	[omfattning]	Ange den spatiala omfattningen av det utgående rasterlagret. Det kommer internt att utökas till en multipel av kakelstorleken. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Mål CRS	`TARGET_CRS`	[crs] Standard: Projekt CRS	CRS för det utgående rasterlagret
Pixelstorlek	`PIXEL_SIZE`	[numerisk: dubbel] Standard: 1,0	Pixelstorlek (X=Y) i kartenheter.
Utgångsraster	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Typ av rasterdata vid utmatning	`OUTPUT_TYPE` Standard: 0	[uppräkning]	Definierar datatypen för den utgående rasterfilen. Alternativ för: • 0 — heltal16 • 1 — Osignerat heltal16 • 2 — Integer32 • 3 — Osignerat heltal32 • 4 — Float32 • 5 — Float64
Sannolikhet	`PROBABILITY`	[numerisk: dubbel] Standard: 0,5	
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utgångsraster	`OUTPUT`	[raster]	Raster som täcker önskad utsträckning med cellstorleken fylld med slumpmässigt fördelade värden

Python-kod

Algoritm-ID: `native:skapande av ett slumpmässigt geometriskt rasterlager`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa slumpmässigt rasterlager (negativ binomialfördelning)

Genererar ett rasterlager för given utsträckning och cellstorlek fyllt med negativa binomialt fördelade slumpmässiga värden.

Som standard kommer värdena att väljas med en fördelningsparameter k på 10,0 och en sannolikhet på 0,5. Detta kan åsidosättas genom att använda de avancerade parametrarna för k och sannolikhet. Rasterdatatypen är inställd på Integer-typer (Integer16 som standard). Slumpvärdena för den negativa binomialfördelningen definieras som positiva heltal. Ett raster med flyttal representerar en omvandling av heltalsvärden till flyttal.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Önskad omfattning	`EXTENT`	[omfattning]	Ange den spatiala omfattningen av det utgående rasterlagret. Det kommer internt att utökas till en multipel av kakelstorleken. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Mål CRS	`TARGET_CRS`	[crs] Standard: Projekt CRS	CRS för det utgående rasterlagret
Pixelstorlek	`PIXEL_SIZE`	[numerisk: dubbel] Standard: 1,0	Pixelstorlek (X=Y) i kartenheter.
Utgångsraster	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Typ av rasterdata vid utmatning	`OUTPUT_TYPE` Standard: 0	[uppräkning]	Definierar datatypen för den utgående rasterfilen. Alternativ för: • 0 — heltal16 • 1 — Osignerat heltal16 • 2 — Integer32 • 3 — Osignerat heltal32 • 4 — Float32 • 5 — Float64
Distributionsparam k	`K_PARAMETER`	[numerisk: heltal] Standard: 10	
Sannolikhet	`PROBABILITY`	[numerisk: dubbel] Standard: 0,5	
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utgångsraster	`OUTPUT`	[raster]	Raster som täcker önskad utsträckning med cellstorleken fylld med slumpmässigt fördelade värden

Python-kod

Algoritm-id: `native:createrandomnegativebinomialrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa slumpmässigt rasterlager (normalfördelning)

Genererar ett rasterlager för given utsträckning och cellstorlek fyllt med normalfördelade slumpmässiga värden.

Som standard kommer värdena att väljas med ett medelvärde på 0,0 och en standardavvikelse på 1,0. Detta kan åsidosättas genom att använda de avancerade parametrarna för medelvärde och standardavvikelse. Rasterdatatypen är som standard inställd på Float32 eftersom normalfördelningens slumpmässiga värden är flyttal.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Önskad omfattning	`EXTENT`	[omfattning]	Ange den spatiala omfattningen av det utgående rasterlagret. Det kommer internt att utökas till en multipel av kakelstorleken. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Mål CRS	`TARGET_CRS`	[crs] Standard: Projekt CRS	CRS för det utgående rasterlagret
Pixelstorlek	`PIXEL_SIZE`	[numerisk: dubbel] Standard: 1,0	Pixelstorlek (X=Y) i kartenheter.
Utgångsraster	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Typ av rasterdata vid utmatning	`OUTPUT_TYPE` Standard: 0	[uppräkning]	Definierar datatypen för den utgående rasterfilen. Alternativ för: • 0 — Float32 • 1 — Float64
Medelvärdet av normalfördelningen	`MEAN`	[numerisk: dubbel] Standard: 0,0	
Standardavvikelse för normalfördelning	`STDDEV`	[numerisk: dubbel] Standard: 1,0	
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utgångsraster	`OUTPUT`	[raster]	Raster som täcker önskad utsträckning med cellstorleken fylld med slumpmässigt fördelade värden

Python-kod

Algoritm-id: `native:createrandomnormalrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa slumpmässigt rasterlager (poissonfördelning)

Genererar ett rasterlager för given utsträckning och cellstorlek fyllt med poissonfördelade slumpmässiga värden.

Som standard kommer värdena att väljas med ett medelvärde på 1,0. Detta kan åsidosättas genom att använda den avancerade parametern för medelvärde. Rasterdatatypen är inställd på Integer-typer (Integer16 som standard). Poisson-distributionens slumpmässiga värden är positiva heltal. Ett raster med flyttal representerar en omvandling av heltalsvärden till flyttal.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Önskad omfattning	`EXTENT`	[omfattning]	Ange den spatiala omfattningen av det utgående rasterlagret. Det kommer internt att utökas till en multipel av kakelstorleken. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Mål CRS	`TARGET_CRS`	[crs] Standard: Projekt CRS	CRS för det utgående rasterlagret
Pixelstorlek	`PIXEL_SIZE`	[numerisk: dubbel] Standard: 1,0	Pixelstorlek (X=Y) i kartenheter.
Utgångsraster	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Typ av rasterdata vid utmatning	`OUTPUT_TYPE` Standard: 0	[uppräkning]	Definierar datatypen för den utgående rasterfilen. Alternativ för: • 0 — heltal16 • 1 — Osignerat heltal16 • 2 — Integer32 • 3 — Osignerat heltal32 • 4 — Float32 • 5 — Float64
Måttlig	`MEAN`	[numerisk: dubbel] Standard: 1,0	
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utgångsraster	`OUTPUT`	[raster]	Raster som täcker önskad utsträckning med cellstorleken fylld med slumpmässigt fördelade värden

Python-kod

Algoritm-ID: `` `native:skapaandandompoissonrasterlager ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa slumpmässigt rasterlager (enhetlig fördelning)

Genererar ett rasterlager för given utsträckning och cellstorlek fyllt med slumpmässiga värden.

Som standard kommer värdena att ligga mellan det lägsta och högsta värdet för den angivna rastertypen för utdata. Detta kan åsidosättas genom att använda de avancerade parametrarna för nedre och övre gränsvärde. Om gränserna har samma värde eller båda är noll (standard) kommer algoritmen att skapa slumpmässiga värden inom hela värdeintervallet för den valda rasterdatatypen. Om du väljer gränser utanför det acceptabla intervallet för den utgående rastertypen avbryts algoritmen.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Önskad omfattning	`EXTENT`	[omfattning]	Ange den spatiala omfattningen av det utgående rasterlagret. Det kommer internt att utökas till en multipel av kakelstorleken. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Mål CRS	`TARGET_CRS`	[crs] Standard: Projekt CRS	CRS för det utgående rasterlagret
Pixelstorlek	`PIXEL_SIZE`	[numerisk: dubbel] Standard: 1,0	Pixelstorlek (X=Y) i kartenheter.
Utgångsraster	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Typ av rasterdata vid utmatning	`OUTPUT_TYPE` Standard: 5	[uppräkning]	Definierar datatypen för den utgående rasterfilen. Alternativ för: • 0 — Byte • 1 — heltal16 • 2 — Osignerat heltal16 • 3 — Integer32 • 4 — Obetecknad heltal32 • 5 — Float32 • 6 — Float64
Nedre gräns för slumptalsområde	`LOWER_BOUND`	[numerisk: dubbel] Standard: 0,0	
Övre gräns för slumptalsintervall	`UPPER_BOUND`	[numerisk: dubbel] Standard: 0,0	
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utgångsraster	`OUTPUT`	[raster]	Raster som täcker önskad utsträckning med cellstorleken fylld med slumpmässigt fördelade värden

Python-kod

Algoritm-ID: `native:skapaettandomuniformtrasterlager`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.17 Rasteranalys av terräng

Bildförhållande

Beräknar aspekten för den digitala terrängmodellen i indata. Det slutliga aspektrasterskiktet innehåller värden från 0 till 360 som uttrycker lutningsriktningen, med början från norr (0°) och fortsätter medurs.

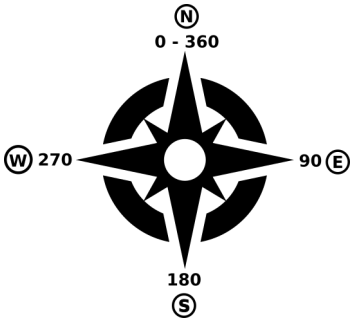


Fig. 24.29: Aspect-värden

Följande bild visar aspektlagret omklassificerat med en färgramp:

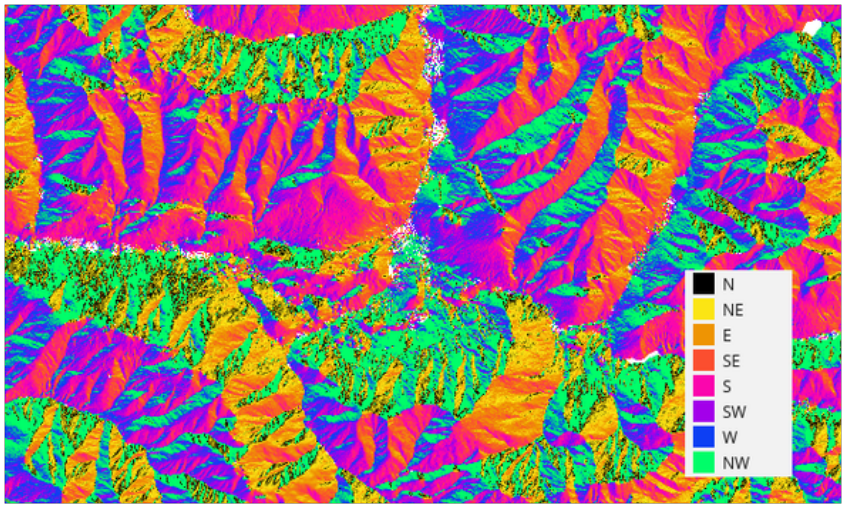


Fig. 24.30: Omklassificering av aspektlager

Parametrar

Etikett	Namn	Typ	Beskrivning
Elevationsskikt	`INPUT`	[raster]	Rasterlager för digital terrängmodell
Z-faktor	`Z_FACTOR`	[numerisk: dubbel] Standard: 1,0	Vertikal överdrift. Denna parameter är användbar när Z-enheterna skiljer sig från X- och Y-enheterna, t.ex. fot och meter. Du kan använda den här parametern för att justera för detta. Standardvärdet är 1 (ingen överdrift).
Aspekt	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange rasterlagret för utdataaspekten. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Aspekt	`OUTPUT`	[raster]	Det utgående aspektrasterskiktet

Python-kod

Algoritm-ID: `qgis:aspekt`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

DTM-filter (lutningsbaserat)

Kan användas för att filtrera en digital höjdmodell för att klassificera dess celler i mark- och objektceller (ej mark).

Verktyget använder begrepp som beskrivs av Vosselman (2000) och baseras på antagandet att en stor höjdskillnad mellan två närliggande celler sannolikt inte orsakas av en brant sluttning i terrängen. Sannolikheten för att den högre cellen inte är mark ökar när avståndet mellan de två cellerna minskar. I filtret definieras därför en maximal höjdskillnad (`dz_max`) mellan två celler som en funktion av avståndet (`d`) mellan cellerna (`dz_max( d ) = d`). En cell klassificeras som terräng om det inte finns någon cell inom kernelradien till vilken höjdskillnaden är större än den tillåtna maximala höjdskillnaden på avståndet mellan dessa två celler.

Parametern för ungefärlig terränglutning (`s`) används för att modifiera filterfunktionen så att den motsvarar den totala lutningen i studieområdet (`dz_max( d ) = d * s`). Ett 5 % confidensintervall (`ci = 1,65 * sqrt( 2 * stddev )`) kan användas för att modifiera filterfunktionen ytterligare genom att antingen lätta på (`dz_max( d ) = d * s + ci`) eller förstärka (`dz_max( d ) = d * s - ci`) filterkriteriet.

Referenser:: Vosselman, G. (2000): Lutningsbaserad filtrering av laseraltimetridata. IAPRS, Vol. XXXIII, Part B3, Amsterdam, Nederländerna, 935-942

Se även

Detta verktyg är en portning av SAGA DTM Filter (sluttningsbaserat)

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Rasterlager för digital terrängmodell
Bandnummer	`BAND`	[rasterband]	DEM:s band att ta hänsyn till
Kernel-radie (pixlar)	`RADIUS`	[numerisk: heltal] Standard: 5	Filterkärnans radie (i pixlar). Måste vara tillräckligt stor för att nå markceller bredvid icke-markobjekt.

continues on next page

Table 24.93 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Terrängens lutning (%, pixelstorlek/vertikal enheter)	`TERRAIN_SLOPE`	[numerisk: dubbel] Standard: 30,0	Den ungefärliga terränglutningen i `%`. Terrängens lutning måste justeras för att ta hänsyn till förhållandet mellan höjdenheter och rasterpixeldimensioner. Används för att lätta på filterkriteriet i brantare terräng.
Ändring av filter	`FILTER_MODIFIC`	Konfigurationens id %1 finns redan i listan Standard: 0	Välj om du vill använda filterkärnan utan modifiering eller om du vill använda ett konfidensintervall för att lätta på eller förstärka höjdkriteriet. • 0 - Ingen • 1 - Filter för avslappning • 2 - Förstärka
Standardavvikelse	`STANDARD_DEVIA`	[numerisk: dubbel] Standard: 0,1	Den standardavvikelse som används för att beräkna ett 5% confidence-intervall som tillämpas på tröskelvärdet för höjd.
Utmatningslager (mark) Valfritt	`OUTPUT_GROUND`	[raster] Standard: `[Spara till temporär fil]`	Ange den filtrerade DEM som endast innehåller celler som klassificerats som mark. *En av*: • Hoppa över utmatning • Spara till en temporär fil • Spara till fil…
Utmatningslager (objekt som inte är markobjekt) Valfritt	`OUTPUT_NONGROU`	[raster] Standard: `[Hoppa över utdata]`	Ange de icke-markobjekt som filtret tar bort. *En av*: • Hoppa över utmatning • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utmatningslager (mark)	`OUTPUT_GROUND`	[raster]	Den filtrerade DEM som endast innehåller celler som klassificerats som mark.
Utmatningslager (objekt som inte är markobjekt)	`OUTPUT_NONGROU`	[raster]	De icke-jordiska objekt som filtret tar bort.

Python-kod

Algoritm ID: `native:dtmslopebaseratfilter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Terrängskuggning

Beräknar rasterlagret för terrängskuggning med en digital terrängmodell som indata.

Skuggningen av lagret beräknas utifrån solens position: du har möjlighet att ändra både den horisontella vinkeln (azimut) och den vertikala vinkeln (solhöjd) för solen.

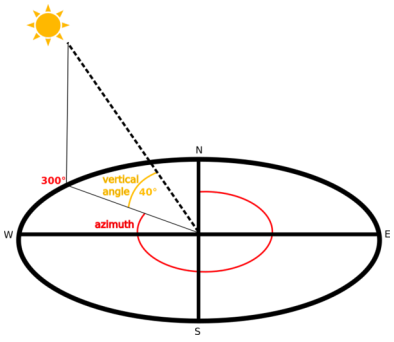


Fig. 24.31: Azimut och vertikal vinkel

Terrängskuggningslagret innehåller värden från 0 (fullständig skugga) till 255 (fullständig sol). Terrängskuggning används vanligtvis för att bättre förstå områdets relief.

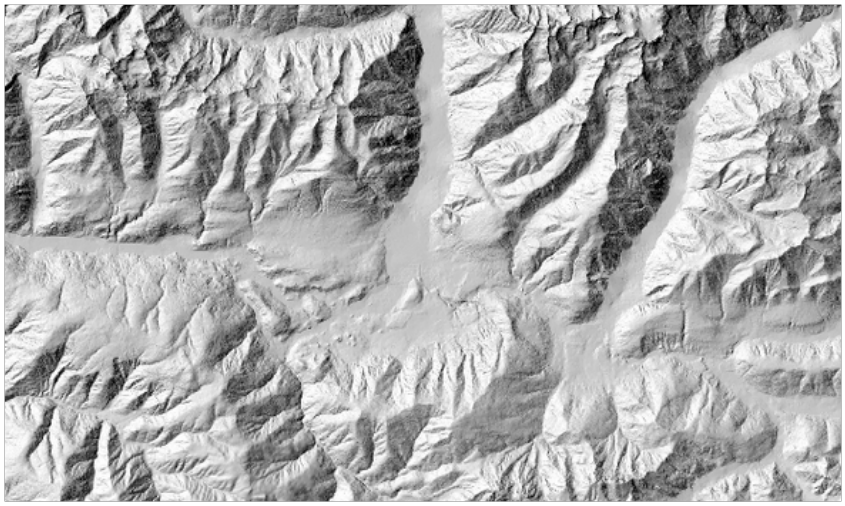


Fig. 24.32: Terrängskuggningslager med azimut 300 och vertikal vinkel 45

Särskilt intressant är det att ge terrängskuggningslagret ett transparensvärde och överlappa det med höjdrastret:

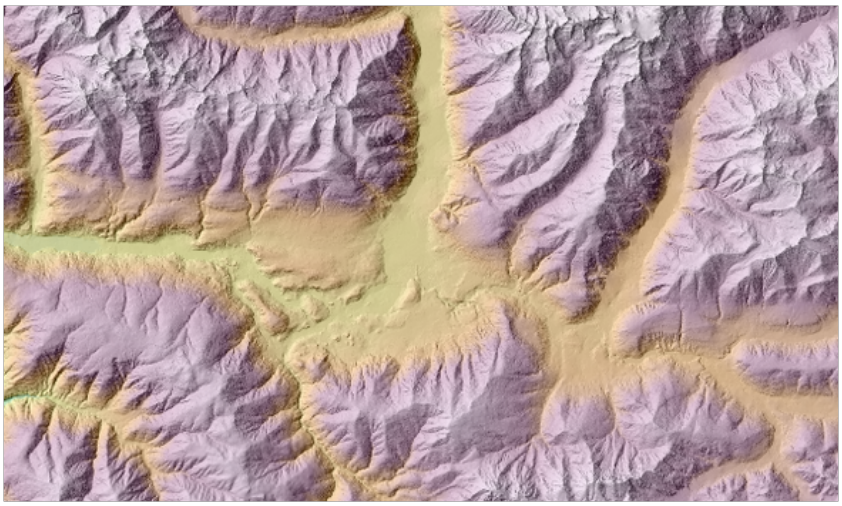


Fig. 24.33: Överlappning av terrängskuggning med höjdlagret

Parametrar

Etikett	Namn	Typ	Beskrivning
Elevationsskikt	`INPUT`	[raster]	Rasterlager för digital terrängmodell
Z-faktor	`Z_FACTOR`	[numerisk: dubbel] Standard: 1,0	Vertikal överdrift. Denna parameter är användbar när Z-enheterna skiljer sig från X- och Y-enheterna, t.ex. fot och meter. Du kan använda den här parametern för att justera för detta. Om du ökar värdet på den här parametern överdrivs slutresultatet (det ser mer ”kuperat” ut). Standardvärdet är 1 (ingen överdrift).

continues on next page

Table 24.95 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Azimut (horisontell vinkel)	`AZIMUTH`	[numerisk: dubbel] Standard: 300,0	Ställ in solens horisontella vinkel (i grader) (medurs riktning). Intervall: 0 till 360. 0 är norr.
Vertikal vinkel	`V_ANGLE`	[numerisk: dubbel] Standard: 40,0	Ställ in solens vertikala vinkel (i grader), det vill säga solens höjd. Värdena kan gå från 0 (lägsta höjd) till 90 (högsta höjd).
Terrängskuggning	`OUTPUT`	[raster] Standard: `Spara till temporär fil`	Ange det utgående rasterlagret för terrängskuggning. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Terrängskuggning	`OUTPUT`	[raster]	Det utmatade rasterlagret för terrängskuggning

Python-kod

Algoritm-id: `qgis:hillshade`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Hypsometriska kurvor

Beräknar hypsometriska kurvor för en inmatad digital höjdmodell. Kurvorna produceras som CSV-filer i en utdatamapp som anges av användaren.

En hypsometrisk kurva är ett kumulativt histogram över höjdvärden i ett geografiskt område.

Du kan använda hypsometriska kurvor för att upptäcka skillnader i landskapet som beror på områdets geomorfologi.

Parametrar

Etikett	Namn	Typ	Beskrivning
DEM för att analysera	`INPUT_DEM`	[raster]	Rasterlager för digital terrängmodell att använda för beräkning av höjder
Begränsande lager	`BOUNDARY_LAYER`	[vektor: polygon]	Polygonvektorlager med gränser för områden som används för att beräkna hypsometriska kurvor
Steg	`STEP`	[numerisk: dubbel] Standard: 100.0	Vertikalt avstånd mellan kurvorna

continues on next page

Table 24.96 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Använd % of area istället för absolut värde	`USE_PERCENTAGE`	[boolean] Standard: Falsk	Skriv procentuell area till fältet ”Area” i CSV-filen i stället för den absoluta arean
Hypsometriska kurvor	`OUTPUT_DIRECTO`	mapp	Ange utdatamappen för de hypsometriska kurvorna. *En av*: • Spara till en temporär katalog • Spara till katalog

Utgångar

Etikett	Namn	Typ	Beskrivning
Hypsometriska kurvor	`OUTPUT_DIRECTO`	mapp	Katalog som innehåller filerna med de hypsometriska kurvorna. För varje objekt från indatavektorskiktet skapas en CSV-fil med värden för area och höjd. Filnamnen börjar med `histogram_`, följt av lagernamn och 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

Python-kod

Algoritm-ID: `qgis:hypsometriska kurvor`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Avlastning

Skapar ett skuggat reliefskikt från digitala höjddata. Du kan ange relieffärgen manuellt eller låta algoritmen automatiskt välja alla reliefklasser.

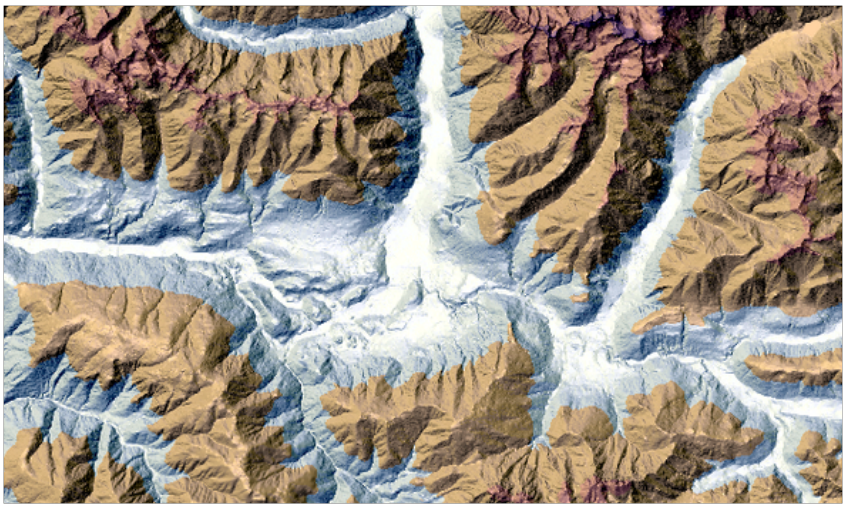


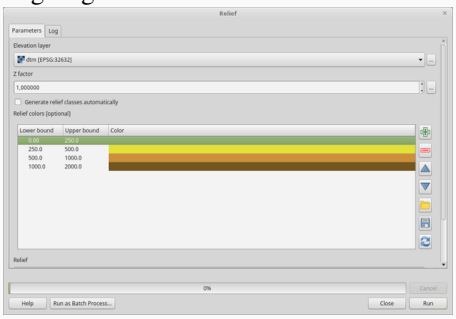
Fig. 24.34: Avlastande lager

Parametrar

Etikett	Namn	Typ	Beskrivning
Elevationsskikt	`INPUT`	[raster]	Rasterlager för digital terrängmodell
Z-faktor	`Z_FACTOR`	[numerisk: dubbel] Standard: 1,0	Vertikal överdrift. Denna parameter är användbar när Z-enheterna skiljer sig från X- och Y-enheterna, t.ex. fot och meter. Du kan använda den här parametern för att justera för detta. Om du ökar värdet på den här parametern överdrivs slutresultatet (det ser mer ”kuperat” ut). Standardvärdet är 1 (ingen överdrift).
Genererar hjälpklasser automatiskt	`AUTO_COLORS`	[boolean] Standard: Falsk	Om du markerar det här alternativet kommer algoritmen att skapa alla färgklasser för relief automatiskt

continues on next page

Table 24.97 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Relief färger Valfritt	`COLORS`	[tabellwidget]	Använd tabellwidgeten om du vill välja relieffärgerna manuellt. Du kan lägga till så många färgklasser som du vill: för varje klass kan du välja den nedre och övre gränsen och slutligen genom att klicka på färgraden kan du välja färg tack vare färgwidgeten.  Fig. 24.35: Manuell inställning av färgklasser för relief Knapparna i högerpanelen ger dig möjlighet att: lägga till eller ta bort färgklasser, ändra ordningen på de färgklasser som redan är definierade, öppna en befintlig fil med färgklasser och spara de aktuella klasserna som fil.
Relief	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange det utgående reliefrasterskiktet. *En av*: • Spara till en temporär fil • Spara till fil…
Frekvensfördelning Valfritt	`FREQUENCY_DIST`	[vektor: tabell] Standard: `[Hoppa över utdata]`	Ange CSV-tabellen för frekvensfördelningen för utdata. *En av*: • Hoppa över utmatning • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Relief	`OUTPUT`	[raster]	Det utgående reliefrasterskiktet
Frekvensfördelning	`OUTPUT`	[vektor: tabell]	Frekvensfördelningen för utdata

Python-kod

Algoritm ID: `qgis:lättnad`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Index för robusthet

Beräknar det kvantitativa mått på terrängens heterogenitet som beskrivs av Riley et al (1999). Det beräknas för varje plats genom att sammanfatta förändringen i höjd inom 3x3-pixelnätet.

Varje pixel innehåller höjdskillnaden mellan en cell i mitten och de 8 celler som omger den.

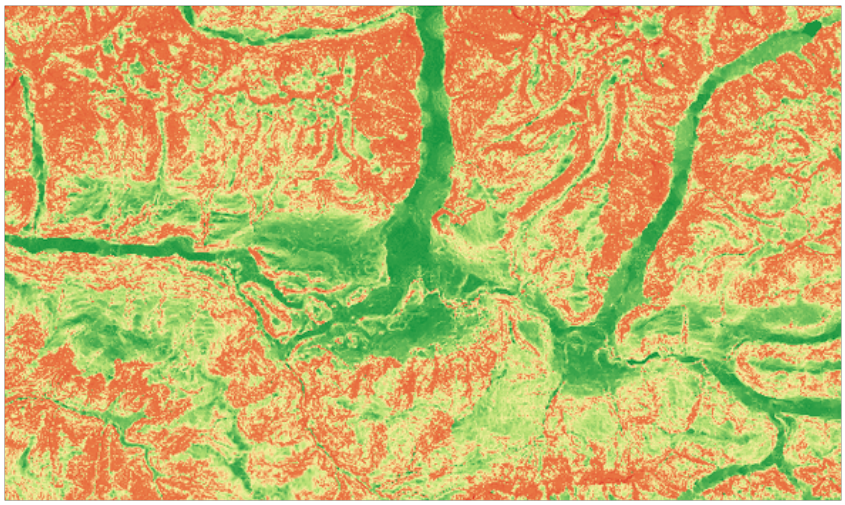


Fig. 24.36: Robusthetslager från låga (rött) till höga värden (grönt)

Parametrar

Etikett	Namn	Typ	Beskrivning
Elevationsskikt	`INPUT`	[raster]	Rasterlager för digital terrängmodell
Z-faktor	`Z_FACTOR`	[numerisk: dubbel] Standard: 1,0	Vertikal överdrift. Denna parameter är användbar när Z-enheterna skiljer sig från X- och Y-enheterna, t.ex. fot och meter. Du kan använda den här parametern för att justera för detta. Om du ökar värdet på denna parameter överdrivs slutresultatet (så att det ser mer robust ut). Standardvärdet är 1 (ingen överdrift).
Ruggedness	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange rasterlagret för utdatans robusthet. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Ruggedness	`OUTPUT`	[raster]	Rasterlager för utmatningens robusthet

Python-kod

Algoritm ID: `qgis:ruggednessindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Lutning

Beräknar lutningen från ett inmatat rasterlager. Lutningen är terrängens lutningsvinkel och uttrycks i **grader**.

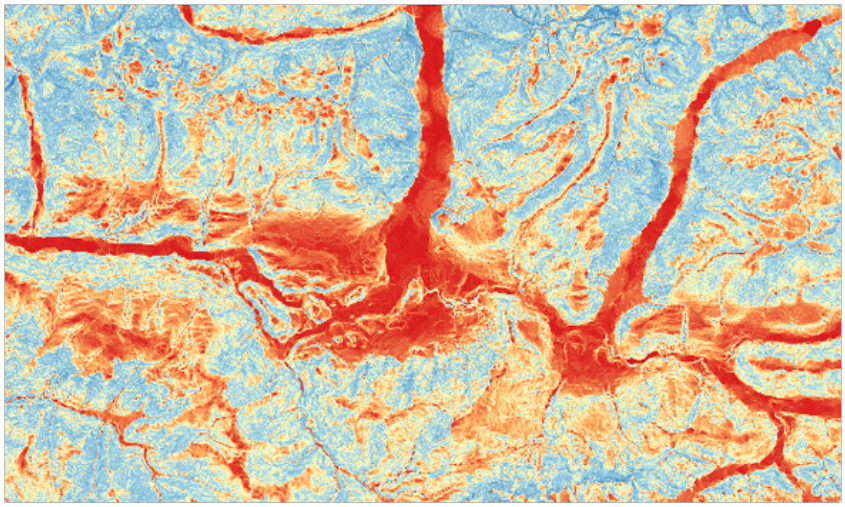


Fig. 24.37: Flacka områden i rött, branta områden i blått

Parametrar

Etikett	Namn	Typ	Beskrivning
Elevationsskikt	`INPUT`	[raster]	Rasterlager för digital terrängmodell
Z-faktor	`Z_FACTOR`	[numerisk: dubbel] Standard: 1,0	Vertikal överdrift. Denna parameter är användbar när Z-enheterna skiljer sig från X- och Y-enheterna, t.ex. fot och meter. Du kan använda den här parametern för att justera för detta. Om värdet på denna parameter ökas överdrivs slutresultatet (det blir brantare). Standardvärdet är 1 (ingen överdrift).
Slope	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange rasterlagret för utdatans lutning. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Slope	`OUTPUT`	[raster]	Det utgående rasterlagret för lutning

Python-kod

Algoritm-ID: `qgis:lutning`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.18 Rasterverktyg

Rikta in raster

Justerar rastret genom att omsampla det till samma cellstorlek och återprojicera till samma CRS som ett referensraster.

> **Varning**
>
> **Den här algoritmen är ENDAST tillgänglig i sammanhanget** *Model Designer*. För andra kontexter, använd istället *Rikta in raster*.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatat rasterlager att justera
Metod för omprovtagning	`RESAMPLING_MET`	[uppräkning] Standard: 0	Method to use for input layer resampling. `Available options` are: • 0 — Närmaste granne • 1 — Bilineär (2x2 kernel) • 2 — Kubisk (4x4-kärna) • 3 — Kubisk B-Spline (4x4 kernel) • 4 — Lanczos (6x6 kärnor) • 5 — Genomsnitt • 6 — Läge • 7 — Maximalt • 8 — Minimum • 9 — Median • 10 — Första kvartilen (Q1) • 11 — Tredje kvartilen (Q3)
Skala om värden enligt cellstorlek	`RESCALE`	[boolean] Standard: Nej	
Reference layer	`REFERENCE_LAYE`	[raster]	Ett rasterlager som kommer att användas för att hämta utsträckning, cellstorlek och CRS som kommer att tillämpas på inmatningslager.
Överträffa referens CRS Valfritt	`CRS`	[crs]	CRS som ska användas istället för referenslagrets
Åsidosätta referenscellstorlek X Valfritt	`CELL_SIZE_X`	[numerisk: dubbel]	Cellstorlek i X-riktning som ska användas i stället för referenslagrets
Åsidosätta referenscellstorlek Y Valfritt	`CELL_SIZE_Y`	[numerisk: dubbel]	Cellstorlek i Y-riktning som ska användas i stället för referenslagrets
Åsidosätta offset för referensnät X Valfritt	`GRID_OFFSET_X`	[numerisk: dubbel]	Offset i X-riktning för att applicera på cellerna i rutnätet
Åsidosätta referensnätets offset Y Valfritt	`GRID_OFFSET_Y`	[numerisk: dubbel]	Offset i Y-riktning för att applicera på cellernas rutnät

continues on next page

Table 24.98 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Klipp till omfattning Valfritt	`EXTENT`	[omfattning]	Ange omfattningen av det utgående rasterlagret. Det kommer internt att utökas till en multipel av kakelstorleken. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Avgränsad raster	`OUTPUT`	[raster]	utmatningsraster med omsamplade pixlar

Utgångar

Etikett	Namn	Typ	Beskrivning
Avgränsad raster	`OUTPUT`	[raster]	utmatningsraster med omsamplade pixlar

Python-kod

Algoritm-id: `native:alignsingleraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rikta in raster

Justerar raster genom att omsampla dem till samma cellstorlek och omprojicera till samma CRS som ett referensraster.

Varning

Den här algoritmen är INTE tillgänglig i sammanhanget *Model Designer*. Använd istället *Rikta in raster*.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster] [lista]	Lista över ingående rasterlager med tillhörande omsamplingsalternativ (fylls i som ett `QgsProcessingParameterAlignRasterLayers`-objekt — görs i GUI genom att trycka på *Configure Raster…*-knappen för varje valt lager): **Inmatningslager [sträng] (`inputFile`)** Fullständig sökväg för inmatningslagret som ska justeras **Utdatafil [sträng] (`outputFile`)** Fullständig sökväg för motsvarande justerade utgångslager **Avprovningsmetod [uppräkning] (`resampleMethod`)** Method to use for input layer resampling. `Available options` are: • 0 — Närmaste granne • 1 — Bilineär (2x2 kernel) • 2 — Kubisk (4x4-kärna) • 3 — Kubisk B-Spline (4x4 kernel) • 4 — Lanczos (6x6 kärnor) • 5 — Genomsnitt • 6 — Läge • 7 — Maximalt • 8 — Minimum • 9 — Median • 10 — Första kvartilen (Q1) • 11 — Tredje kvartilen (Q3) **Skala om värden enligt cellstorleken** [boolean] (`rescale`)
Reference layer	`REFERENCE_LAYE`	[raster]	Ett rasterlager som kommer att användas för att hämta utsträckning, cellstorlek och CRS som kommer att tillämpas på inmatningslager.
Överträffa referens CRS Valfritt	`CRS`	[crs]	CRS som ska användas istället för referenslagrets
Åsidosätta referenscellstorlek X Valfritt	`CELL_SIZE_X`	[numerisk: dubbel]	Cellstorlek i X-riktning som ska användas i stället för referenslagrets
Åsidosätta referenscellstorlek Y Valfritt	`CELL_SIZE_Y`	[numerisk: dubbel]	Cellstorlek i Y-riktning som ska användas i stället för referenslagrets
Åsidosätta offset för referensnät X Valfritt	`GRID_OFFSET_X`	[numerisk: dubbel]	Offset i X-riktning för att applicera på cellerna i rutnätet

continues on next page

Table 24.99 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Åsidosätta referensnätets offset Y Valfritt	`GRID_OFFSET_Y`	[numerisk: dubbel]	Offset i Y-riktning för att applicera på cellernas rutnät
Klipp till omfattning Valfritt	`EXTENT`	[omfattning]	Ange omfattningen av det utgående rasterlagret. Det kommer internt att utökas till en multipel av kakelstorleken. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`

Utgångar

Etikett	Namn	Typ	Beskrivning
Justerade raster	`OUTPUT_LAYERS`	[raster] [lista]	utmatning av raster med omsamplade pixlar

Python-kod

Algoritm-ID: `` `native:alignrasters` ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Konvertera karta till raster

Skapar en rasterbild av kartans innehåll.

Ett *maptema* kan väljas för att återge en förutbestämd uppsättning lager med en definierad stil för varje lager.

Alternativt kan ett enda lager väljas om inget karttema har angetts.

Om varken karttema eller lager har angetts kommer det aktuella kartinnehållet att återges. Den minsta utsträckning som anges kommer internt att utökas till att vara en multipel av kakelstorleken.

Parametrar

Etikett	Namn	Typ	Beskrivning
Minsta omfattning som ska återges (xmin, xmax, ymin, ymax)	`EXTENT`	[omfattning]	Ange omfattningen av det utgående rasterlagret. Det kommer internt att utökas till en multipel av kakelstorleken. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Storlek på kakelplatta	`TILE_SIZE`	[numerisk: heltal] Standard: 1024	Storlek på kakelplattan i det utgående rasterlagret. Minsta värde: 64.
Kartläggningsenhet per pixel	`MAP_UNITS_PER_`	[numerisk: dubbel] Standard: 100.0	Pixelstorlek (i kartenheter). Minsta värde: 0,0
Göra bakgrunden transparent	`MAKE_BACKGROUN`	[boolean] Standard: Falsk	Gör det möjligt att exportera kartan med en transparent bakgrund. Utmatar en RGBA-bild (istället för RGB) om den är inställd på `True`.
Mappa tema till rendering Valfritt	`MAP_THEME`	[uppräkning]	Använd ett befintligt *mapptema* för renderingen.
Ett enda lager att rendera Valfritt	`LAYER`	[uppräkning]	Välj ett enda lager för renderingen
Utdatalager	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Utdatalager	`OUTPUT`	[raster]	Utgående rasterlager

Python-kod

Algoritm-ID: `native:rasterize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Fylla NoData-celler

Återställer NoData-värdena i indatarastret till ett valt värde, vilket resulterar i ett rasterdataset utan NoData-pixlar.

Algoritmen respekterar den ingående rasterdatatypen, t.ex. kommer ett fyllnadsvärde med flyttal att trunkeras när det tillämpas på ett heltalsraster.

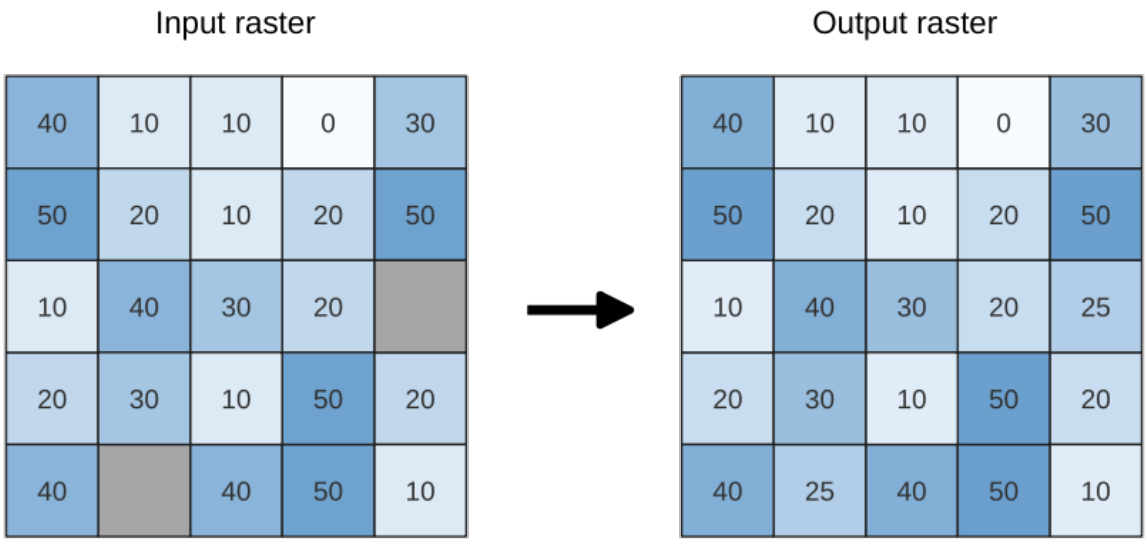


Fig. 24.38: Fylla i NoData-värden (i grått) i ett raster

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Ingångsraster	`INPUT`	[raster]	Rastret till processen.
Bandnummer	`BAND`	[rasterband] Standard: 1	Rasterns band
Fyllnadsvärde	`FILL_VALUE`	[numerisk: dubbel] Standard: 1,0	Ange det värde som ska användas för NoData-pixlarna
Utgångsraster	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Alternativ för skapande Valfritt	`CREATE_OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Utgångsraster	`OUTPUT`	[raster]	Det utgående rasterlagret med fyllda dataceller.

Python-kod

Algoritm-ID: `native:fillnodata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Generera XYZ-plattor (katalog)

Genererar ”XYZ”-rasterplattor med hjälp av det aktuella QGIS-projektet som enskilda bilder till en katalogstruktur. Alternativt kan en HTML-utdatafil för Leaflet skapas med de genererade plattorna som ett kartlager.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Extent (xmin, xmax, ymin, ymax)	`EXTENT`	[omfattning]	Ange omfattningen av plattorna. Den kommer internt att utökas till en multipel av brickans storlek. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Minsta zoom	`ZOOM_MIN`	[numerisk: heltal] Standard: 12	Minimum 0, maximum 25.
Maximal zoom	`ZOOM_MAX`	[numerisk: heltal] Standard: 12	Minimum 0, maximum 25.
DPI	`DPI`	[numerisk: heltal] Standard: 96	Minst 48, högst 600.
Bakgrundsfärg Valfritt	`BACKGROUND_COL`	[färg] Standard: QColor(0, 0, 0, 0)	Välj bakgrundsfärg för plattorna
Aktivera antialiasing	`ANTIALIAS`	[boolean] Standard: Sann	Avgör om antialiasing ska vara aktiverat
Kakelformat	`TILE_FORMAT`	[uppräkning] Standard: 0	En av: • 0 — PNG • 1 — JPG
Kvalitet (endast JPG) Valfritt	`QUALITY`	[numerisk: heltal] Standard: 75	Minst 1, högst 100.
Flyttbar storlek Valfritt	`METATILESIZE`	[numerisk: heltal] Standard: 4	Ange en anpassad metatil storlek när du genererar XYZ-rutor. Större värden kan påskynda renderingen av plattor och ge bättre märkning (färre luckor utan märkning) på bekostnad av att mer minne används. Minimum 1, maximum 20.

continues on next page

Table 24.102 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Kakelplattans bredd Valfritt	`TILE_WIDTH`	[numerisk: heltal] Standard: 256	Minst 1, högst 4096.
Kakelplattans höjd Valfritt	`TILE_HEIGHT`	[numerisk: heltal] Standard: 256	Minst 1, högst 4096.
Använd Y-axel med inverterad kakelplatta (TMS-konventioner)	`TMS_CONVENTION`	[boolean] Standard: Falsk	
Utmatningskatalog	`OUTPUT_DIRECTO`	mapp Standard: `[Spara till tillfällig mapp]`	Specifikation av utdatakatalogen (för brickorna). *En av*: • Hoppa över utmatning • Spara till en temporär katalog • Spara till katalog
html-utskrift (broschyr) Valfritt	`OUTPUT_HTML`	[html] Standard: `[Spara till temporär fil]`	Specifikation av HTML-filen för utdata. *En av*: • Hoppa över utmatning • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Leaflets HTML-titel för utdata Valfritt	`HTML_TITLE`	[sträng] Standard: Ej inställd	HTML <title>-tagg som används för Leaflets HTML-utdatafil.
Tilldelning av HTML-utdata från Leaflet Valfritt	`HTML_ATTRIBUTI`	[sträng] Standard: Ej inställd	Anpassad kartattribuering som används för Leaflets HTML-utdatafil. HTML-länkar är möjliga.
Inkludera OpenStreetMap-baskarta i Leaflets HTML-utdata	`HTML_OSM`	[boolean] Standard: Falsk	Ett OpenStreetMap-baskartlager (källa: https://tile.openstreetmap.org) ingår i Leaflets HTML-utdatafil. Korrekt kartattribuering läggs till automatiskt.

Utgångar

Etikett	Namn	Typ	Beskrivning
Utmatningskatalog	`OUTPUT_DIRECTO`	mapp	Katalog för utdata (för plattorna)
html-utskrift (broschyr)	`OUTPUT_HTML`	[html]	Den utgående HTML-filen (Leaflet)

Python-kod

Algoritm ID: ”original:tilesxyzdirectory

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Generera XYZ-plattor (MBTiles)

Genererar ”XYZ”-plattor i rasterformat med hjälp av det aktuella QGIS-projektet som en enda fil i formatet ”MBTiles”.

Parametrar

Etikett	Namn	Typ	Beskrivning
Extent (xmin, xmax, ymin, ymax)	`EXTENT`	[omfattning]	Ange omfattningen av plattorna. Den kommer internt att utökas till en multipel av brickans storlek. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Minsta zoom	`ZOOM_MIN`	[numerisk: heltal] Standard: 12	Minimum 0, maximum 25.
Maximal zoom	`ZOOM_MAX`	[numerisk: heltal] Standard: 12	Minimum 0, maximum 25.
DPI	`DPI`	[numerisk: heltal] Standard: 96	Minst 48, högst 600.
Bakgrundsfärg Valfritt	`BACKGROUND_COL`	[färg] Standard: QColor(0, 0, 0, 0)	Välj bakgrundsfärg för plattorna
Aktivera antialiasing	`ANTIALIAS`	[boolean] Standard: Sann	Avgör om antialiasing ska vara aktiverat
Kakelformat	`TILE_FORMAT`	[uppräkning] Standard: 0	En av: • 0 — PNG • 1 — JPG

continues on next page

Table 24.104 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Kvalitet (endast JPG) Valfritt	`QUALITY`	[numerisk: heltal] Standard: 75	Minst 1, högst 100.
Flyttbar storlek Valfritt	`METATILESIZE`	[numerisk: heltal] Standard: 4	Ange en anpassad metatil storlek när du genererar XYZ-rutor. Större värden kan påskynda renderingen av plattor och ge bättre märkning (färre luckor utan märkning) på bekostnad av att mer minne används. Minimum 1, maximum 20.
Utdatafil (för MBTiles)	`OUTPUT_FILE`	&Arkiv Standard: `[Spara till temporär fil]`	Specifikation av utdatafilen. *En av*: • Hoppa över utmatning • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Utdatafil (för MBTiles)	`OUTPUT_FILE`	&Arkiv	Utdatafilen.

Python-kod

Algoritm-ID: `original:tilesxyzmbtiles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.19 Vektoranalys

Grundläggande statistik för fält

Genererar grundläggande statistik för ett fält i attributtabellen för ett vektorlager. Numeriska fält, datum-, tids- och strängfält stöds. Den statistik som returneras beror på fälttypen.

Statistik kan genereras som en tabell eller en HTML-fil och finns tillgänglig i *Processing* ► *Results viewer*.

Standardmeny: *Vektor* ► *Analysverktyg*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsvektor	`INPUT_LAYER`	[vektor: vilken som helst]	Vektorlager för att beräkna statistiken på
Fält att beräkna statistik på	`FIELD_NAME`	[tabellfält: alla]	Alla tabellfält som stöds för att beräkna statistiken
Statistik Valfritt	`OUTPUT`	[vektor: tabell] Standard: `[Skapa tillfälligt lager]`	Ange utmatningstabellen för den genererade statistiken. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Statistikrapport Valfritt	`OUTPUT_HTML_FI`	[html] Standard: `[Spara till temporär fil]`	Specifikation av filen för den beräknade statistiken. *En av*: • Hoppa över utmatning • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Statistik	`OUTPUT`	[vektor: tabell]	Tabell som innehåller den beräknade statistiken
Statistikrapport	`OUTPUT_HTML_FI`	[html]	HTML-fil med den beräknade statistiken
Räkna	`COUNT`	[numerisk: heltal]	
Antal unika värden	`UNIQUE`	[numerisk: heltal]	
Antal tomma (null) värden	`EMPTY`	[numerisk: heltal]	
Antal icke-tomma värden	`FILLED`	[numerisk: heltal]	
Minsta värde	`MIN`	[samma som inmatning]	
Maximalt värde	`MAX`	[samma som inmatning]	
Minsta längd	`MIN_LENGTH`	[numerisk: heltal]	
Maximala längden	`MAX_LENGTH`	[numerisk: heltal]	
Genomsnittlig längd	`MEAN_LENGTH`	[numerisk: dubbel]	
Koefficient för variation	`CV`	[numerisk: dubbel]	
Summa	`SUM`	[numerisk: dubbel]	
Medelvärde	`MEAN`	[numerisk: dubbel]	
Standardavvikelse	`STD_DEV`	[numerisk: dubbel]	
Omfång	`RANGE`	[numerisk: dubbel]	
Median	`MEDIAN`	[numerisk: dubbel]	
Minority (sällsynt förekommande värde)	`MINORITY`	[samma som inmatning]	
Majoritet (mest frekvent förekommande värde)	`MAJORITY`	[samma som inmatning]	
Första kvartilen	`FIRSTQUARTILE`	[numerisk: dubbel]	
Tredje kvartilen	`THIRDQUARTILE`	[numerisk: dubbel]	

continues on next page

Table 24.105 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Interkvartilintervall (IQR)	`IQR`	[numerisk: dubbel]	

Python-kod

Algoritm ID: `qgis:grundläggande statistik för fält`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Klättra längs linjen

Beräknar den totala stigningen och nedstigningen längs linjegeometrier. Inmatningslagret måste ha Z-värden. Om Z-värden inte finns tillgängliga kan algoritmen *Drape (ställ in Z-värde från raster)* användas för att lägga till Z-värden från ett DEM-lager.

Utmatningslagret är en kopia av inmatningslagret med ytterligare fält som innehåller den totala stigningen (`climb`), den totala nedstigningen (`descent`), den lägsta höjden (`minelev`) och den högsta höjden (`maxelev`) för varje linjegeometri. Om inmatningslagret innehåller fält med samma namn som de tillagda fälten, byter de namn (fältnamnen ändras till ”name_2”, ”name_3”, etc., varvid det första icke-duplicerade namnet hittas).

Parametrar

Etikett	Namn	Typ	Beskrivning
Linjelager	`INPUT`	[vektor: linje]	Linjelager att beräkna stigningen för. Måste ha Z-värden
Klättringslager	`OUTPUT`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Specifikation av utdatalagret (linje). *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Klättringslager	`OUTPUT`	[vektor: linje]	Linjelager som innehåller nya attribut med resultat från klättringsberäkningar.
Total klättring	`TOTALCLIMB`	[numerisk: dubbel]	Summan av stigningen för alla linjegeometrier i inmatningslagret
Total nedstigning	`TOTALDESCENT`	[numerisk: dubbel]	Summan av nedstigningen för alla linjegeometrier i inmatningslagret
Minsta höjd	`MINELEVATION`	[numerisk: dubbel]	Den lägsta höjden för geometrierna i lagret
Maximal höjd över havet	`MAXELEVATION`	[numerisk: dubbel]	Den maximala höjden för geometrierna i lagret

Python-kod

Algoritm-ID: `qgis:climbalongline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Räkna punkter i polygonen

Tar ett punkt- och ett polygonlager och räknar antalet punkter från punktlagret i var och en av polygonerna i polygonlagret.

Ett nytt polygonlager genereras, med exakt samma innehåll som det inmatade polygonlagret, men med ett extra fält med poängantalet för varje polygon.

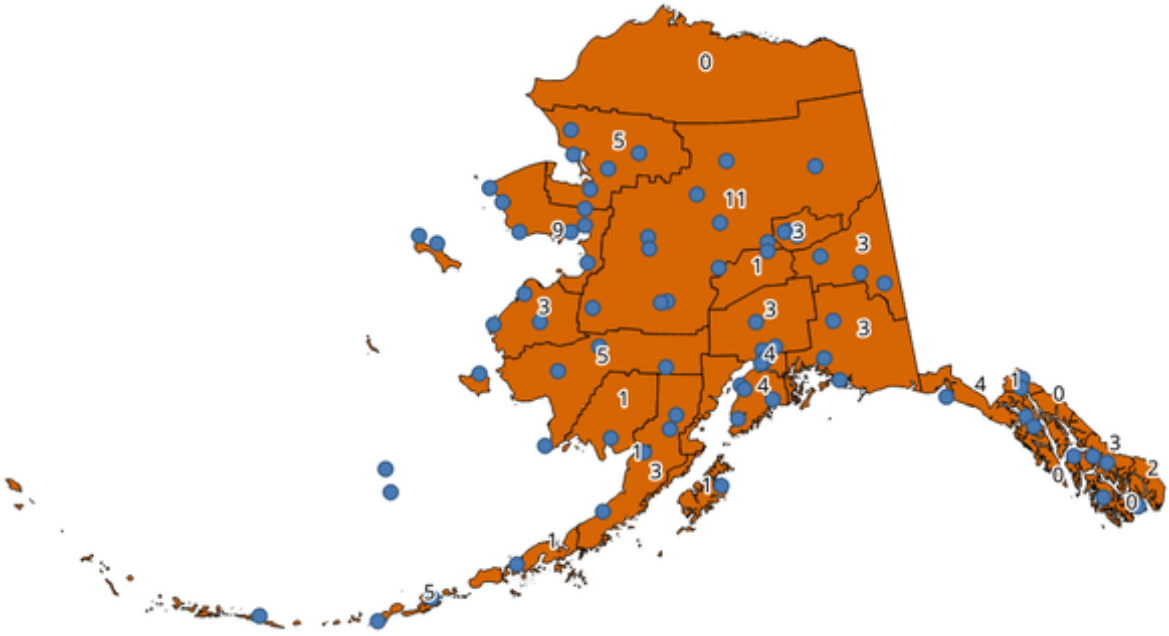


Fig. 24.39: Etiketterna i polygonerna visar antalet punkter

Ett valfritt viktfält kan användas för att tilldela vikter till varje punkt. Alternativt kan ett unikt klassfält anges. Om båda alternativen används kommer viktfältet att ha företräde och det unika klassfältet kommer att ignoreras.

☑ Tillåter *features in-place modification* av polygonfeatures

`Förvalda menyn`: *Vektor* ► *Analysverktyg*

Parametrar

Etikett	Namn	Typ	Beskrivning
Polygoner	`POLYGONS`	[vektor: polygon]	Polygonlager vars egenskaper är associerade med antalet punkter de innehåller
Punkter	`POINTS`	[vektor: punkt]	Punktlager med funktioner att räkna
Viktfält Valfritt	`WEIGHT`	[tabellfält: alla]	Ett fält från punktlagret. Det antal som genereras blir summan av viktfältet för de punkter som ingår i polygonen. Om viktfältet inte är numeriskt kommer antalet att vara `0`.
Klassfält Valfritt	`CLASSFIELD`	[tabellfält: alla]	Punkterna klassificeras utifrån det valda attributet och om flera punkter med samma attributvärde finns inom polygonen räknas bara en av dem. Det slutliga antalet punkter i en polygon är därför antalet olika klasser som finns i den.
*namn på fältet *Count**	`FIELD`	[sträng] Standard: ”NUMPOINTS	Namnet på det fält som ska lagra antalet poäng
Räkna	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Specifikation av utdatalagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Räkna	`OUTPUT`	[vektor: polygon]	Resulterande lager med attributtabellen som innehåller den nya kolumnen med poängantalet

Python-kod

Algoritm-id: `native:countpointsinpolygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

DBSCAN-klustring

Klustrar punktfunktioner baserat på en 2D-implementering av DBSCAN-algoritmen (Density-based spatial clustering of applications with noise).

Algoritmen kräver två parametrar, en minsta klusterstorlek och det största tillåtna avståndet mellan klusterpunkterna.

Se även

ST-DBSCAN klustring, *K-means klustring*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: punkt]	Lager att analysera
Minsta klusterstorlek	`MIN_SIZE`	[numerisk: heltal] Standard: 5	Minsta antal funktioner för att generera ett kluster
Maximalt avstånd mellan klustrade punkter	`EPS`	[numerisk: dubbel] Standard: 1,0	Avstånd bortom vilket två funktioner inte kan tillhöra samma kluster (eps)
Clusters	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange vektorlagret för resultatet av klustringen. *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Behandla gränspunkter som brus (DBSCAN*)	`DBSCAN*`	[boolean] Standard: Falsk	Om den är markerad behandlas punkter på gränsen till ett kluster som punkter utan kluster, och endast punkter i det inre av ett kluster taggas som klustrade.
Klustrets fältnamn	`FIELD_NAME`	[sträng] Standard: ”CLUSTER_ID	Namn på det fält där det associerade klusternumret ska lagras
Namn på fält för klusterstorlek	`SIZE_FIELD_NAM`	[sträng] Standard: ”CLUSTER_SIZE	Namnet på fältet med antalet funktioner i samma kluster

Utgångar

Etikett	Namn	Typ	Beskrivning
Clusters	`OUTPUT`	[vektor: punkt]	Vektorlager som innehåller de ursprungliga objekten med ett fält som anger vilket kluster de tillhör
Antal kluster	`NUM_CLUSTERS`	[numerisk: heltal]	Antalet kluster som upptäckts

Python-kod

Algoritm-ID: `native:dbscanclustering`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Avståndsmatris

Beräknar för punktobjekt avstånd till närmaste objekt i samma lager eller i ett annat lager.

Standardmeny: *Vektor* ► *Analysverktyg*

Se även

Sammanfoga attribut efter närmaste

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningspunkt lager	`INPUT`	[vektor: punkt]	Punktlager för vilket avståndsmatrisen beräknas (**från** punkter)
Ange unikt ID-fält	`INPUT_FIELD`	[tabellfält: alla]	Fält som ska användas för att unikt identifiera egenskaper i inmatningslagret. Används i tabellen för utdataattribut.
Målpunkt lager	`TARGET`	[vektor: punkt]	Punktlager som innehåller den eller de punkter som ligger närmast sökningen (**till** punkter)
Målets unika ID-fält	`TARGET_FIELD`	[tabellfält: alla]	Fält som ska användas för att unikt identifiera egenskaper i mållagret. Används i tabellen för utdataattribut.

continues on next page

Table 24.109 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Typ av utmatningsmatris	`MATRIX_TYPE`	[uppräkning] Standard: 0	Olika typer av beräkningar finns tillgängliga: • 0 — Linjär (N * *k* x 3) avståndsmatris: för varje inmatningspunkt rapporteras avståndet till var och en av de *k* närmaste målpunkterna. Utmatningsmatrisen består av upp till *k* rader per inmatningspunkt, och varje rad har tre kolumner: *InputID*, *TargetID* och *Distance*. • 1 — Standard (N x T) avståndsmatris • 2 — Sammanfattande avståndsmatris (medelvärde, std. dev., min, max): för varje inmatningspunkt rapporteras statistik över avstånden till dess målpunkter.
Använd endast de närmaste (k) målpunkterna	`NEAREST_POINTS`	[numerisk: heltal] Standard: 0	Du kan välja att beräkna avståndet till alla punkter i mållagret (*0*) eller begränsa till ett antal (*k*) av de närmaste objekten.
Avståndsmatris	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Specifikation av utmatningsvektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Avståndsmatris	`OUTPUT`	[vektor: punkt]	Punkt (eller MultiPoint för fallet ”Linear (N * *k* x 3)”) vektorlager som innehåller avståndsberäkningen för varje indatafunktion. Dess funktioner och attributtabell beror på den valda typen av utmatningsmatris.

Python-kod

Algoritm-ID: `qgis:distancematrix`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Avstånd till närmaste hubb (linje till hubb)

Skapar linjer som förbinder varje funktion i en inmatningsvektor med den närmaste funktionen i ett destinationslager. Avstånden beräknas baserat på *center* för varje objekt.

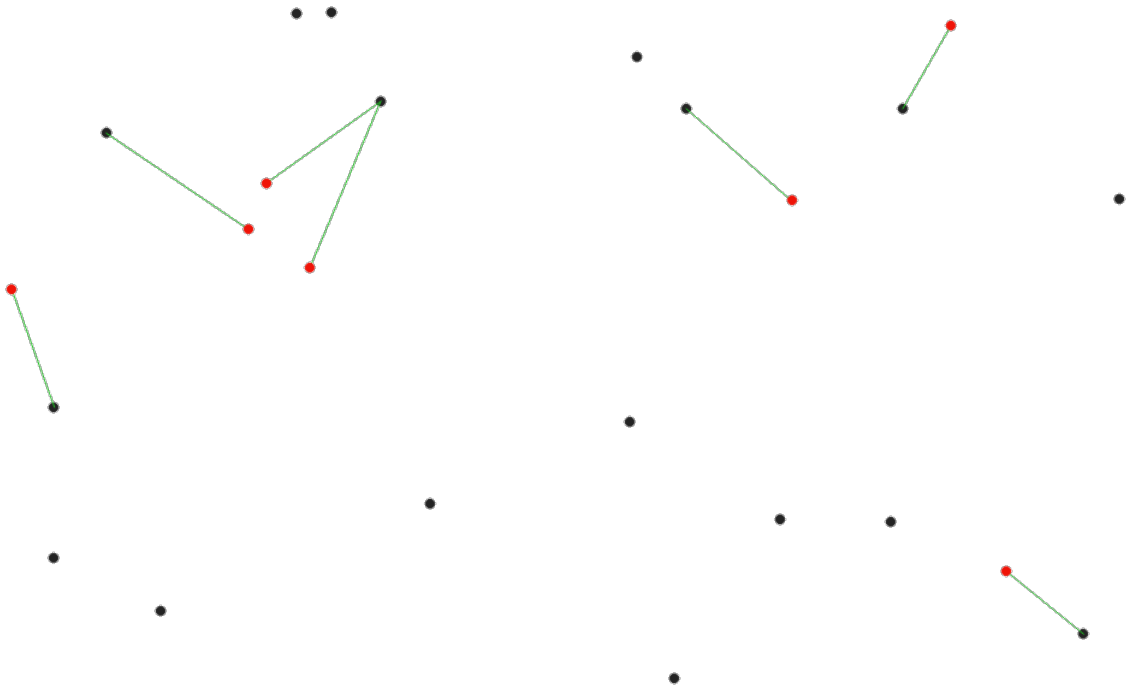


Fig. 24.40: Visa närmaste hubb för de röda inmatningsfunktionerna

Se även

Avstånd till närmaste hubb (punkter), *Sammanfoga attribut efter närmaste*

Parametrar

Etikett	Namn	Typ	Beskrivning
Källpunkter lager	`INPUT`	[vektor: geometri]	Vektorlager för vilket den närmaste funktionen söks
Destination hubbar lager	`HUBS`	[vektor: geometri]	Vektorlager som innehåller de funktioner som ska sökas efter
Navlagrets namnattribut	`FIELD`	[tabellfält: alla]	Fält som ska användas för att unikt identifiera egenskaper i destinationsskiktet. Används i tabellen för utdataattribut
Mätenhet	`UNIT`	[uppräkning] Standard: 0	Enheter i vilka avståndet till närmaste objekt ska rapporteras: • 0 — Meter • 1 — Fötter • 2 — mil • 3 — Kilometrar • 4 — Lagerenheter
Hubavstånd	`OUTPUT`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange det utgående linjevektorskiktet som förbinder de matchande punkterna. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Hubavstånd	`OUTPUT`	[vektor: linje]	Linjevektorlager med attributen för de ingående objekten, identifieraren för deras närmaste objekt och det beräknade avståndet.

Python-kod

Algoritm-ID: `qgis:distancetonearesthublinetohub`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Avstånd till närmaste hubb (punkter)

Skapar ett punktlager som representerar *center* för de ingående funktionerna med tillägg av två fält som innehåller identifieraren för den närmaste funktionen (baserat på dess mittpunkt) och avståndet mellan punkterna.

Se även

Avstånd till närmaste hubb (linje till hubb), *Sammanfoga attribut efter närmaste*

Parametrar

Etikett	Namn	Typ	Beskrivning
Källpunkter lager	`INPUT`	[vektor: vilken som helst]	Vektorlager för vilket den närmaste funktionen söks
Destination hubbar lager	`HUBS`	[vektor: vilken som helst]	Vektorlager som innehåller de funktioner som ska sökas efter
Navlagrets namnattribut	`FIELD`	[tabellfält: alla]	Fält som ska användas för att unikt identifiera egenskaper i destinationsskiktet. Används i tabellen för utdataattribut
Mätenhet	`UNIT`	[uppräkning] Standard: 0	Enheter i vilka avståndet till närmaste objekt ska rapporteras: • 0 — Meter • 1 — Fötter • 2 — mil • 3 — Kilometrar • 4 — Lagerenheter
Hubavstånd	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange utmatningspunktens vektorlager med närmaste nav. *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Hubavstånd	`OUTPUT`	[vektor: punkt]	Punktvektorlager som representerar mitten av källobjekten med deras attribut, identifieraren för deras närmaste objekt och det beräknade avståndet.

Python-kod

Algoritm-ID: `qgis:avståndtillnärmstahubpunkter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Förbindelse med linjer (navlinjer)

Skapar nav- och ekerdiagram genom att ansluta linjer från punkter i lagret Spoke till matchande punkter i lagret Hub.

Avgörandet av vilken hubb som hör till varje punkt baseras på en matchning mellan Hub ID-fältet på hubbpunkterna och Spoke ID-fältet på ekerpunkterna.

Om inmatningslagren inte är punktlager kommer en punkt på geometriernas yta att användas som anslutningsplats.

Som tillval kan geodetiska linjer skapas, som representerar den kortaste vägen på ytan av en ellipsoid. När geodetiskt läge används är det möjligt att dela de skapade linjerna vid antimeridianen (±180 graders longitud), vilket kan förbättra återgivningen av linjerna. Dessutom kan avståndet mellan hörnen specificeras. Ett mindre avstånd resulterar i en tätare och mer exakt linje.

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

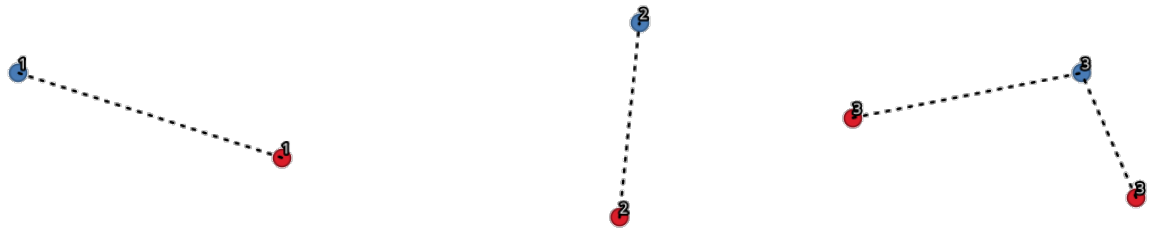


Fig. 24.41: Sammanfoga punkter baserat på ett gemensamt fält/attribut

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Hublager	`HUBS`	[vektor: geometri]	Inmatningslager
Hub ID-fält	`HUB_FIELD`	[tabellfält: alla]	Fält för navlagret med ID för att gå med

continues on next page

Table 24.111 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Hublagerfält att kopiera (lämna tomt för att kopiera alla fält) Valfritt	`HUB_FIELDS`	[tabellfält: valfritt] [lista]	Det eller de fält i navlagret som ska kopieras. Om inget eller inga fält väljs tas alla fält med.
Spoke lager	`SPOKES`	[vektor: geometri]	Ytterligare lager av ekerpunkter
Spoke ID-fält	`SPOKE_FIELD`	[tabellfält: alla]	Fält för det talade lagret med ID för att gå med
Spoke layer-fält som ska kopieras (lämna tomt för att kopiera alla fält) Valfritt	`SPOKE_FIELDS`	[tabellfält: valfritt] [lista]	Fält(en) i det talskikt som ska kopieras. Om inga fält väljs tas alla fält med.
Skapa geodetiska linjer	`GEODESIC`	[boolean] Standard: Falsk	Skapa geodetiska linjer (den kortaste vägen på ytan av en ellipsoid)
Hublinjer	`OUTPUT`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange utdatahubbens linjevektorlager. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Avstånd mellan hörnpunkter (endast geodetiska linjer)	`GEODESIC_DISTA`	[numerisk: dubbel] Standard: 1000,0 (kilometer)	Avstånd mellan på varandra följande hörn (i kilometer). Ett mindre avstånd resulterar i en tätare och mer exakt linje
Delade linjer vid antimeridianen (±180 graders longitud)	`ANTIMERIDIAN_S`	[boolean] Standard: Falsk	Delade linjer på ±180 graders longitud (för att förbättra återgivningen av linjerna)

Utgångar

Etikett	Namn	Typ	Beskrivning
Hublinjer	`OUTPUT`	[vektor: linje]	Det resulterande linjelagret som förbinder matchande punkter i ingående lager

Python-kod

Algoritm-ID: `native:hublines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

K-means klustring

Beräknar det 2D-avståndsbaserade k-means-klusternumret för varje inmatad funktion.

K-means-klustring syftar till att dela upp funktionerna i k kluster där varje funktion tillhör det kluster som har det närmaste medelvärdet. Medelpunkten representeras av barycentrum för de klustrade funktionerna.

Om indatageometrierna är linjer eller polygoner baseras klusterbildningen på objektets centroid.

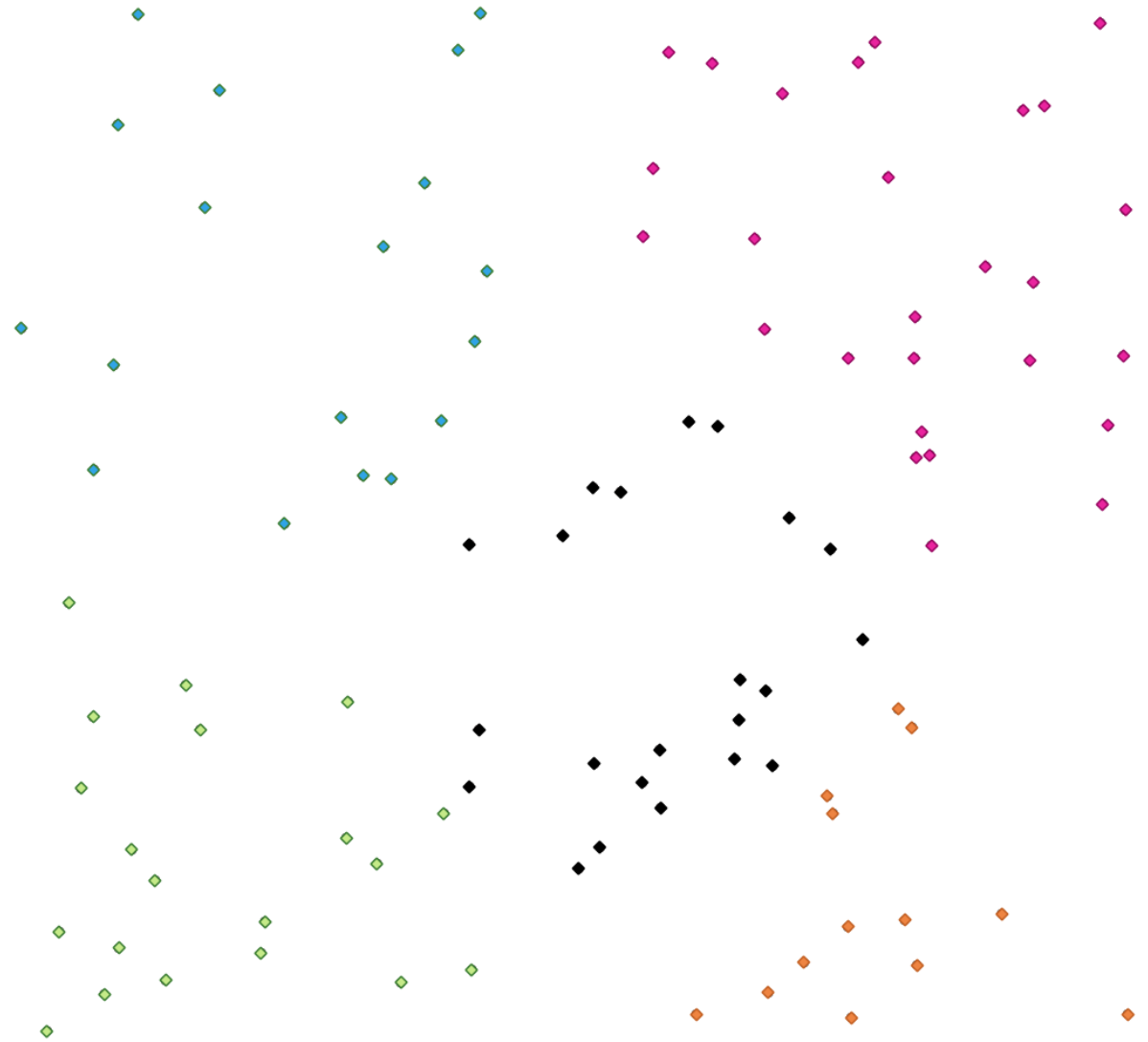


Fig. 24.42: Ett kluster med fem klasspoäng

Se även

DBSCAN-klustring, *ST-DBSCAN klustring*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Lager att analysera
Antal kluster	`CLUSTERS`	[numerisk: heltal] Standard: 5	Antal kluster som ska skapas med funktionerna
Clusters	`OUTPUT`	[vektor: samma som indata] Standard: `[Skapa tillfälligt lager]`	Ange utdatavektorlagret för de genererade klustren. *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Klustrets fältnamn	`FIELD_NAME`	[sträng] Standard: ”CLUSTER_ID	Namn på det fält där det associerade klusternumret ska lagras
Namn på fält för klusterstorlek	`SIZE_FIELD_NAM`	[sträng] Standard: ”CLUSTER_SIZE	Namnet på fältet med antalet funktioner i samma kluster

Utgångar

Etikett	Namn	Typ	Beskrivning
Clusters	`OUTPUT`	[vektor: samma som indata]	Vektorlager som innehåller de ursprungliga egenskaperna med fält som anger vilket kluster de tillhör och deras nummer i det

Python-kod

Algoritm-ID: `native:kmeansclustering`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Lista unika värden

Listar unika värden för ett attributtabellfält och räknar deras antal.

Standardmeny: *Vektor* ► *Analysverktyg*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Lager att analysera
Målområde(n)	`FIELDS`	[tabellfält: valfritt] [lista]	Område(n) som ska analyseras
Unika värden Valfritt	`OUTPUT`	[vektor: tabell] Standard: `[Skapa tillfälligt lager]`	Ange lagret för sammanfattningstabellen med unika värden. *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
HTML-rapport Valfritt	`OUTPUT_HTML_FI`	[html] Standard: `[Spara till temporär fil]`	HTML-rapport av unika värden i *Bearbetning* ► *Resultatvisning*. *En av*: • Hoppa över utmatning • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Unika värden	`OUTPUT`	[vektor: tabell]	Sammanfattande tabellager med unika värden
HTML-rapport	`OUTPUT_HTML_FI`	[html]	HTML-rapport över unika värden. Kan öppnas från *Bearbetning* ► *Resultatvisare*
Totala unika värden	`TOTAL_VALUES`	[numerisk: heltal]	Antalet unika värden i inmatningsfältet
Unika värden sammanlänkade	`UNIQUE_VALUES`	[sträng]	En sträng med en kommaseparerad lista över unika värden som finns i inmatningsfältet

Python-kod

Algoritm-ID: `qgis:listuniquevalues`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Medelkoordinat(er)

Beräknar ett punktlager med masscentrum för geometrierna i ett inmatningslager.

Ett attribut kan specificeras som innehållande vikter som ska tillämpas på varje egenskap vid beräkning av masscentrum.

Om ett attribut har valts i parametern grupperas funktionerna enligt värdena i detta fält. Istället för en enda punkt med masscentrum för hela lagret, kommer utdatalagret att innehålla ett masscentrum för funktionerna i varje kategori.

Standardmeny: *Vektor* ► *Analysverktyg*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Viktfält Valfritt	`WEIGHT`	[tabellfält: numerisk]	Fält som ska användas om du vill göra ett viktat medelvärde
Unikt ID-fält	`UID`	[tabellfält: alla]	Unikt fält på vilket beräkningen av medelvärdet kommer att göras
Medelkoordinater	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange (punktvektor)-lagret för resultatet. *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Medelkoordinater	`OUTPUT`	[vektor: punkt]	Resulterande punkt(er) lager

Python-kod

Algoritm-ID: `native:medelvärde koordinater`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Analys av närmaste granne

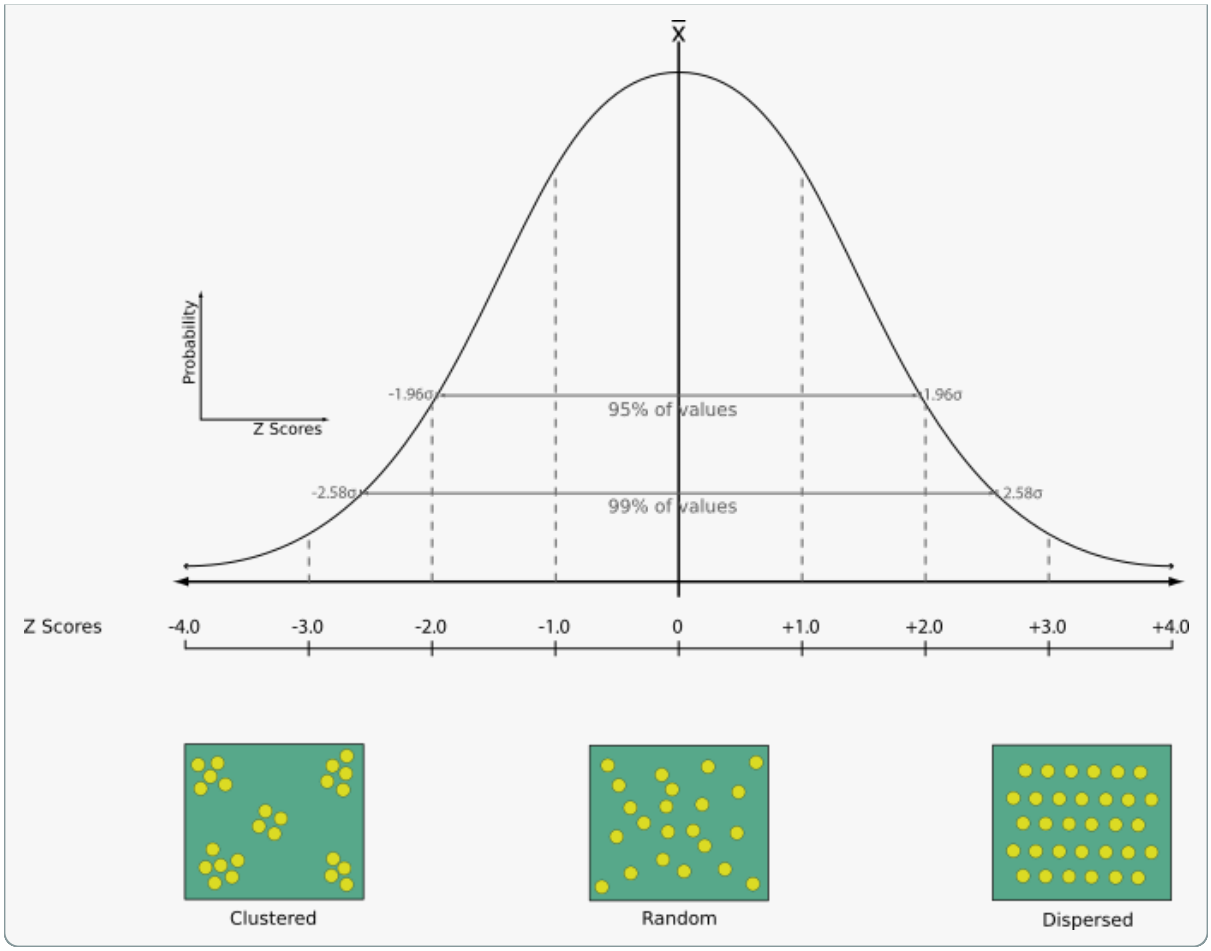
Utför närmaste granne-analys för ett punktlager. Utdata visar hur dina data är fördelade (klustrade, slumpmässigt eller fördelade).

Utdata genereras som en HTML-fil med de beräknade statistiska värdena:

- Observerat medelavstånd
- Förväntat medelavstånd
- Index för närmaste granne
- Antal poäng
- Z-poäng: Genom att jämföra Z-Score med normalfördelningen får du reda på hur dina data är fördelade. En låg Z-score innebär att det är osannolikt att data är resultatet av en spatialt slumpmässig process, medan en hög Z-score innebär att det är sannolikt att dina data är resultatet av en spatialt slumpmässig process.

Observera

Den här algoritmen använder ellipsoidbaserade mätningar och respekterar de aktuella *ellipsoidinställningarna*.



Standardmeny: *Vektor ► Analysverktyg*

Se även

Sammanfoga attribut efter närmaste

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: punkt]	Punktvektorlager för att beräkna statistiken över
Närmaste granne Valfritt	`OUTPUT_HTML_FI`	[html] Standard: `[Spara till temporär fil]`	Specifikation av HTML-filen för den beräknade statistiken. *En av*: • Hoppa över utmatning • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Närmaste granne	`OUTPUT_HTML_FI`	[html]	HTML-fil med den beräknade statistiken
Observerat medelavstånd	`OBSERVED_MD`	[numerisk: dubbel]	Observerat medelavstånd
** Förväntat medelavstånd**	`EXPECTED_MD`	[numerisk: dubbel]	Förväntat medelavstånd
Index för närmaste granne	`NN_INDEX`	[numerisk: heltal]	Index för närmaste granne
Antal poäng	`POINT_COUNT`	[numerisk: heltal]	Antal poäng
Z-poäng	`Z_SCORE`	[numerisk: dubbel]	Z-poäng

Python-kod

Algoritm-ID: `original:analys av närmaste granne`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Analys av överlappning

Beräknar den yta och procentuella täckning med vilken objekt från ett inmatningslager överlappas av objekt från ett urval av överlagringslager.

Nya attribut läggs till i utdatalagret som rapporterar den totala överlappningsytan och procentandelen av indatafunktionen som överlappas av vart och ett av de valda överlagringslagren.

Observera

Den här algoritmen använder ellipsoidbaserade mätningar och respekterar de aktuella *ellipsoidinställningarna*.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: polygon]	Det ingående lagret.
Overlay-lager	`LAYERS`	[vektor: polygon] [lista]	De överliggande lagren.
Överlappning	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Nätstorlek Valfritt	`GRID_SIZE`	[numerisk: dubbel] Standard: Ej inställd	Om det anges, snappas indatageometrierna till ett rutnät med den angivna storleken och resultatets hörn beräknas på samma rutnät. Kräver GEOS 3.9.0 eller senare.

Utgångar

Etikett	Namn	Typ	Beskrivning
Överlappning	`OUTPUT`	[samma som inmatning]	Utmatningslagret med ytterligare fält som rapporterar överlappningen (i kartenheter och procent) av den ingående funktionen som överlappas av vart och ett av de valda lagren.

Python-kod

Algoritm-ID: `native:beräkna vektoröverlappningar`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Kortaste linjen mellan funktionerna

Skapar ett linjelager som den kortaste linjen mellan käll- och destinationsskiktet. Som standard tas endast hänsyn till den första närmaste funktionen i destinationslagret. Antalet n-närmaste grannfunktioner kan specificeras. Om ett maximalt avstånd anges kommer endast objekt som ligger närmare än detta avstånd att beaktas.

De utgående funktionerna innehåller alla källagrets attribut, alla attribut från den n-närmaste funktionen och det extra fältet för avståndet.

Viktigt

Denna algoritm använder rent kartesiska beräkningar för avstånd och tar inte hänsyn till geodetiska eller ellipsoidiska egenskaper när den bestämmer objektets närhet. Koordinatsystemet för mätning och utdata baseras på koordinatsystemet för källagret.

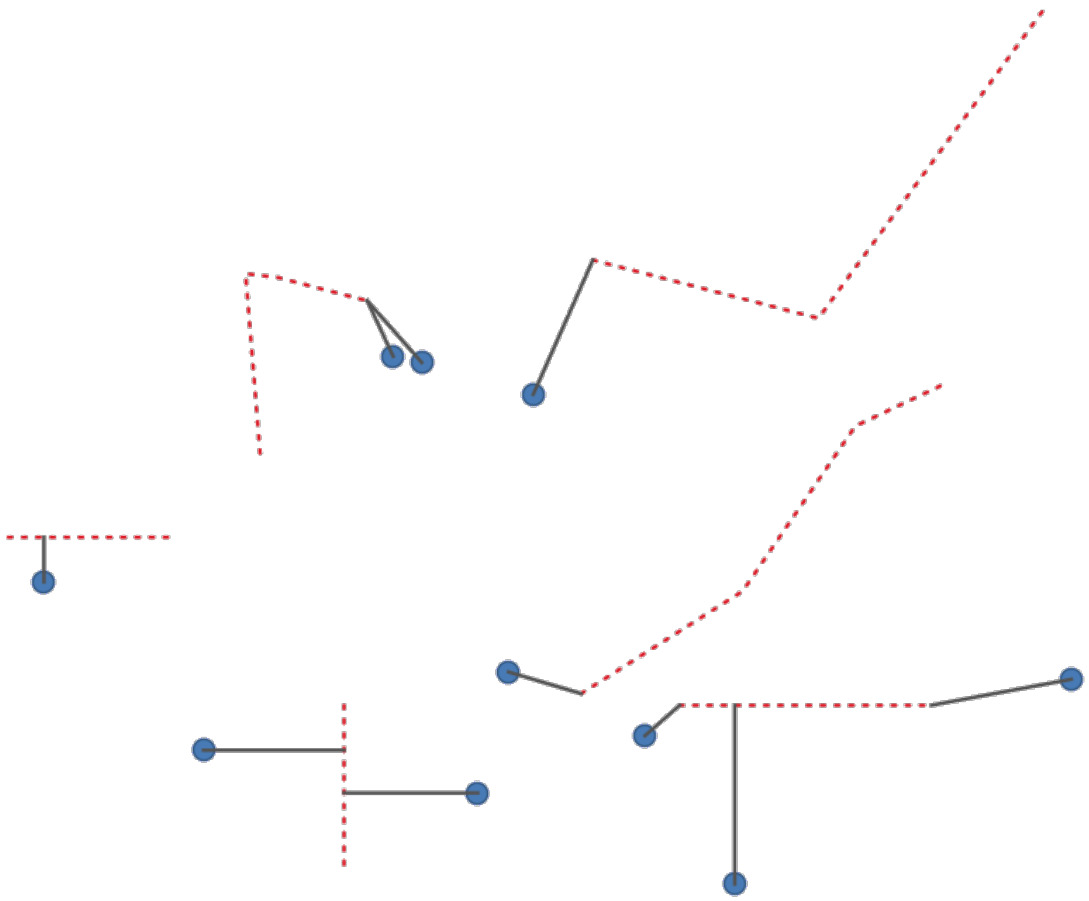


Fig. 24.43: Kortaste linjen från punkter till linjer

Parametrar

Etikett	Namn	Typ	Beskrivning
Källager	`SOURCE`	[vektor: geometri]	Ursprungslager för vilket man söker efter närmaste grannar
Destinationslager	`DESTINATION`	[vektor: geometri]	Mållager i vilket man söker efter närmaste grannar
Metod	`METHOD`	[uppräkning] Standard: 0	Beräkningsmetod för kortaste avstånd Möjliga värden är: • 0 — Avstånd till närmaste punkt på objektet • 1 — Avstånd till objektets tyngdpunkt
Maximalt antal grannar	`NEIGHBORS`	[numerisk: heltal] Standard: 1	Maximalt antal grannar att leta efter
Maximalt avstånd Valfritt	`DISTANCE`	[numerisk: dubbel]	Endast destinationsfunktioner som ligger närmare än detta avstånd kommer att beaktas.
Kortaste linjerna	`OUTPUT`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Utdatalager	`OUTPUT`	[vektor: linje]	Linjevektorlager som sammanfogar källfunktioner med deras närmaste granne(r) i destinationslagret. Innehåller alla attribut för både käll- och destinationsobjekten samt det beräknade avståndet.

Python-kod

Algoritm-id: `native:shortestline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

ST-DBSCAN klustring

Klustrar punktfunktioner baserat på en 2D-implementering av spatiotemporal Density-based clustering of applications with noise (ST-DBSCAN)-algoritm.

Se även

DBSCAN-klustring, *K-means klustring*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: punkt]	Lager att analysera
Datum/tid-fält	`DATETIME_FIELD`	[tabellfält: datum]	Fält som innehåller den temporala informationen
Minsta klusterstorlek	`MIN_SIZE`	[numerisk: heltal] Standard: 5	Minsta antal funktioner för att generera ett kluster
Maximalt avstånd mellan klustrade punkter	`EPS`	[numerisk: dubbel] Standard: 1,0	Avstånd bortom vilket två funktioner inte kan tillhöra samma kluster (eps)
Maximal tidslängd mellan klustrade punkter	`EPS2`	[numerisk: dubbel] Standard: 0,0 (dagar)	Tidsperiod efter vilken två funktioner inte kan tillhöra samma kluster (eps2). Tillgängliga tidsenheter är millisekunder, sekunder, minuter, timmar, dagar och veckor.
Clusters	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange vektorlagret för resultatet av klustringen. *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Behandla gränspunkter som brus (DBSCAN*)	`DBSCAN*`	[boolean] Standard: Falsk	Om den är markerad behandlas punkter på gränsen till ett kluster som punkter utan kluster, och endast punkter i det inre av ett kluster taggas som klustrade.
Klustrets fältnamn	`FIELD_NAME`	[sträng] Standard: ”CLUSTER_ID	Namn på det fält där det associerade klusternumret ska lagras
Namn på fält för klusterstorlek	`SIZE_FIELD_NAM`	[sträng] Standard: ”CLUSTER_SIZE	Namnet på fältet med antalet funktioner i samma kluster

Utgångar

Etikett	Namn	Typ	Beskrivning
Clusters	`OUTPUT`	[vektor: punkt]	Vektorlager som innehåller de ursprungliga objekten med ett fält som anger vilket kluster de tillhör
Antal kluster	`NUM_CLUSTERS`	[numerisk: heltal]	Antalet kluster som upptäckts

Python-kod

Algoritm-id: `native:stdbscanclustering`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Statistik per kategori

Beräknar statistik för ett fält beroende på en överordnad klass. Den överordnade klassen är en kombination av värden från andra fält.

Parametrar

Etikett	Namn	Typ	Beskrivning
Input vektorlager	`INPUT`	[vektor: vilken som helst]	Vektorlager för indata med unika klasser och värden
Fält att beräkna statistik på (om tomt beräknas endast antal) Valfritt	`VALUES_FIELD_N`	[tabellfält: alla]	Om den är tom kommer endast antalet att beräknas
Fält med kategorier	`CATEGORIES_FIE`	[tabellfält: valfritt] [lista]	De fält som (tillsammans) definierar kategorierna
Statistik per kategori	`OUTPUT`	[vektor: tabell] Standard: `[Skapa tillfälligt lager]`	Ange utmatningstabellen för den genererade statistiken. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Statistik per kategori	`OUTPUT`	[vektor: tabell]	Tabell som innehåller statistiken

Beroende på vilken typ av fält som analyseras returneras följande statistik för varje grupperat värde:

Statistik	Text	Numeric	Datum
Räkna (`COUNT`)	✓	✓	✓
Unika värden (`UNIQUE`)	✓		✓
Tomma (null) värden (`EMPTY`)	✓		✓
Icke-tomma värden (`FILLED`)	✓		✓
Minimivärde (`MIN`)	✓	✓	✓
Maximalt värde (`MAX`)	✓	✓	✓
Område (`RANGE`)		✓	
Summa (`SUM`)		✓	
Medelvärde (`MEAN`)		✓	
Medianvärde (`MEDIAN`)		✓	
Standardavvikelse (`STD_DEV`)		✓	
Variationskoefficient (`CV`)		✓	
Minoritet (sällsynt förekommande värde - `MINORITET`)		✓	
Majoritet (mest frekvent förekommande värde - `MAJORITY`)		✓	
Första kvartilen (`FIRSTQUARTILE`)		✓	
Tredje kvartilen (`THIRDQUARTILE`)		✓	
Interkvartilt intervall (`IQR`)		✓	
Minsta längd (`MIN_LENGTH`)	✓		
Medellängd (`MEAN_LENGTH`)	✓		
Maximal längd (`MAX_LENGTH`)	✓		

Python-kod

Algoritm-ID: `qgis:statisticsbycategories`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Summera linjelängder

Tar ett polygonlager och ett linjelager och mäter den totala längden på linjerna och det totala antalet linjer som korsar varje polygon.

Det resulterande lagret har samma egenskaper som det ingående polygonlagret, men med två ytterligare attribut som innehåller längden och antalet linjer över varje polygon.

Observera

Den här algoritmen använder ellipsoidbaserade mätningar och respekterar de aktuella *ellipsoidinställningarna*.

Tillåter *features in-place modification* av polygonfeatures

Standardmeny: *Vektor* ► *Analysverktyg*

Parametrar

Etikett	Namn	Typ	Beskrivning
Linjer	`LINES`	[vektor: linje]	Ingångsvektor linjelager
Polygoner	`POLYGONS`	[vektor: polygon]	Polygon vektorlager
Linjelängd fältnamn	`LEN_FIELD`	[sträng] Standard: ”LENGTH” (längd)	Namn på fältet för linjelängd
Fältnamn för antal rader	`COUNT_FIELD`	[sträng] Standard: ”COUNT	Namn på fältet för antal rader
Linjens längd	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående polygonlagret med genererad statistik. *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Linjens längd	`OUTPUT`	[vektor: polygon]	Polygonutgångsskikt med fält för linjelängd och linjeantal

Python-kod

Algoritm-ID: `native:sumlinelengths`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.20 Vektortäckning

Lös upp täckning

Opererar på en täckning (representerad som en uppsättning polygonfunktioner med exakt matchande kantgeometri) för att lösa upp (sammanfoga) geometrierna. Det ger en kraftigt optimerad metod för unionering av dessa funktioner jämfört med standardverktygen för Dissolve.

Se även

Lös upp

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: polygon]	Vektorlager för inmatad polygon
Upplöst	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Upplöst	`OUTPUT`	[vektor: polygon]	Det utgående polygonvektorlagret med upplösta geometrier.

Python-kod

Algoritm-ID: `native:coverageunion`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Förenkla täckningen

Opererar på en täckning (representerad som en uppsättning polygonfunktioner med exakt matchande kantgeometri) för att tillämpa en Visvalingam-Whyatt-förenkling på kanterna, vilket minskar komplexiteten i proportion till den angivna toleransen, samtidigt som en giltig täckning bibehålls (dvs. inga kanter kommer att korsa eller beröra varandra efter förenklingen). Geometrier kommer aldrig att tas bort, men de kan förenklas ner till bara en triangel. Dessutom kommer vissa geometrier (t.ex. polygoner som har för få icke upprepade punkter) att returneras oförändrade.

Om indatasetet inte är en giltig täckning på grund av överlappningar kommer det fortfarande att förenklas, men ogiltig topologi som korsande kanter kommer fortfarande att vara ogiltig.

Kräver version av GEOS >= 3.12

Se även

Förenkla, *Smidig*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: polygon]	Vektorlager för inmatad polygon
Tolerans	`TOLERANCE`	[numerisk: dubbel] Standard: 0,0	Det maximala avståndet (i valfri enhet) mellan två på varandra följande toppar som ska slås samman.
Bevara gränsdragningen	`PRESERVE_BOUND`	[boolean] Standard: Falsk	När den är aktiverad bevaras täckningens ytterkanter utan förenkling
Förenklat	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Förenklat	`OUTPUT`	[vektor: polygon]	Det utgående polygonvektorlagret, med ett lägre antal hörn.

Python-kod

Algoritm-ID: `` `native:coveragesimplify ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Validera täckning

Analyserar en täckning (representerad som en uppsättning polygonelement med exakt matchande kantgeometri) för att hitta platser där antagandet om exakt matchande kanter inte uppfylls. Ogiltighet inkluderar polygoner som överlappar varandra eller som har luckor som är mindre än den angivna luckbredden.

Kräver version av GEOS >= 3.12

Se även

Kontrollera giltighet

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: polygon]	Vektorlager för inmatad polygon
Gapets bredd	`GAP_WIDTH`	[numerisk: dubbel] Standard: 0,0	Den maximala bredden på gapet som ska detekteras
Ogiltiga kanter	`INVALID_EDGES`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Ogiltiga kanter	`OUTPUT`	[vektor: linje]	Vektorlagret för utdatalinjen visar felkanter för varje ogiltig polygon.
Validitetsresultat	`IS_VALID`	[boolean]	Returnerar om täckningen är giltig eller inte.

Python-kod

Algoritm-ID: `native:coveragevalidate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.21 Skapande av vektor

Array av förskjutna (parallella) linjer

Skapar kopior av linjefunktioner i ett lager genom att skapa flera förskjutna versioner av varje funktion. Varje ny version förskjuts stegvis med ett angivet avstånd.

Positiva avstånd förskjuter linjerna åt vänster och negativa avstånd förskjuter dem åt höger.

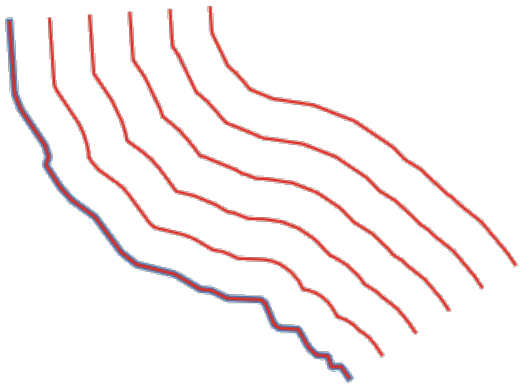


Fig. 24.44: I blått källagret, i rött det förskjutna lagret

Tillåter *funktioner på plats modifiering* av linjefunktioner

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Offset-linjer, *En rad översatta funktioner*

Parametrar

Grundläggande parametrar

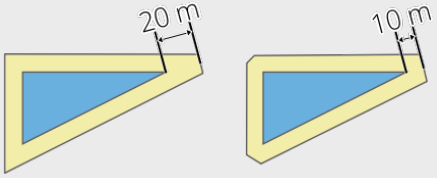
Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje]	Vektorlager för inmatningslinje som ska användas för offsets.
Antal funktioner att skapa	`COUNT`	[numerisk: heltal] Standard: 10	Antal offsetkopior som ska genereras för varje funktion
Offset stegavstånd	`OFFSET`	[numerisk: dubbel] Standard: 1,0	Avstånd mellan två på varandra följande offsetkopior
Offset-linjer	`OUTPUT`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange det utgående linjelagret med förskjutna funktioner. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Segment	`SEGMENTS`	[numerisk: dubbel] Standard: 8	Antal linjesegment som ska användas för att approximera en kvartscirkel när man skapar avrundade offsets
Join style	`JOIN_STYLE`	[uppräkning] Standard: 0	Ange om runda, geringade eller fasade skarvar ska användas vid förskjutning av hörn i en linje. Ett av följande: • 0 — Omgång • 1 — Miter • 2 — Avfasning Fig. 24.45: Runda, geringade och fasade fogningsstilar

continues on next page

Table 24.120 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Mitergräns	`MITER_LIMIT`	[numerisk: dubbel] Standard: 2,0	Ställer in det maximala avståndet från offsetgeometrin som ska användas när man skapar en geringad fog som en faktor av offsetavståndet (gäller endast för geringade fogstilar). Minsta värde: 1.0  Fig. 24.46: En 10 m buffert med en gräns på 2 och en 10 m buffert med en gräns på 1

Utgångar

Etikett	Namn	Typ	Beskrivning
Offset-linjer	`OUTPUT`	[vektor: linje]	Utgående linjelager med original- och offsetfunktioner. Attributen kopieras till motsvarande utgångar. Följande attribut relaterade till inparametrarna läggs till: `instance` indexerar `COUNT` offsetlinjerna och `offset` betecknar offsetavståndet från den ursprungliga linjen.

Python-kod

Algoritm-ID: `native:arrayoffsetlines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

En rad översatta funktioner

Skapar kopior av features i ett lager genom att skapa flera översatta versioner av varje feature. Varje kopia förskjuts stegvis med en förinställd mängd i X-, Y- och/eller Z-axeln.

M-värden som finns i geometrin kan också översättas.

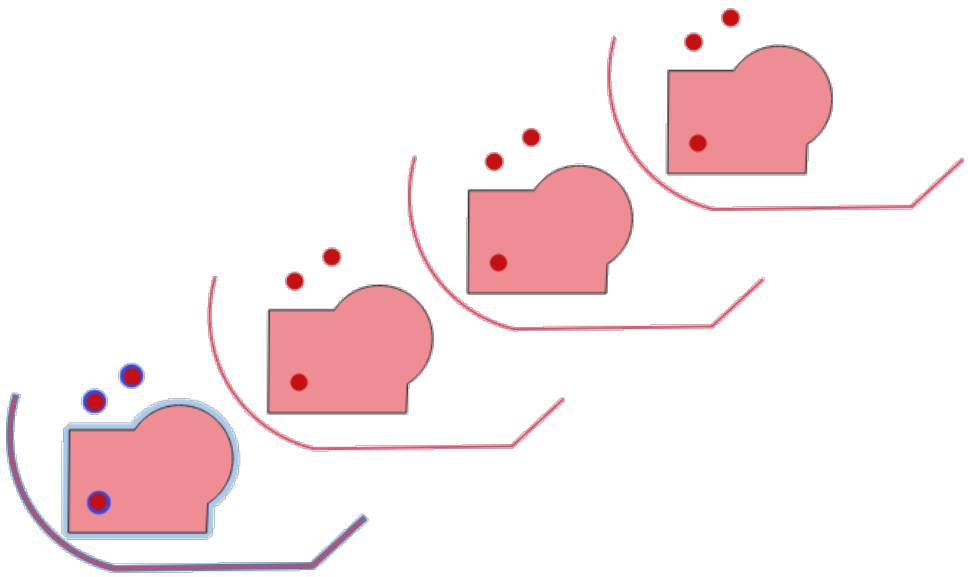


Fig. 24.47: Inmatningslager i blå toner, utmatningslager med översatta funktioner i röda toner

Tillåter *features in-place modification* av punkt-, linje- och polygonfeatures

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Översätt, *Array av förskjutna (parallella) linjer*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager för att översätta
Antal funktioner att skapa	`COUNT`	[numerisk: heltal] Standard: 10	Antal kopior som ska genereras för varje funktion
Stegavstånd (x-axel)	`DELTA_X`	[numerisk: dubbel] Standard: 0,0	Förskjutning som ska tillämpas på X-axeln
Stegavstånd (y-axeln)	`DELTA_Y`	[numerisk: dubbel] Standard: 0,0	Förskjutning som ska tillämpas på Y-axeln
Stegavstånd (z-axel)	`DELTA_Z`	[numerisk: dubbel] Standard: 0,0	Förskjutning som ska tillämpas på Z-axeln

continues on next page

Table 24.121 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Stegavstånd (m-värden)	`DELTA_M`	[numerisk: dubbel] Standard: 0,0	Förskjutning att applicera på M
Översatt	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Utgående vektorlager med översatta (flyttade) kopior av funktionerna. De ursprungliga funktionerna kopieras också. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Översatt	`OUTPUT`	[samma som inmatning]	Utgående vektorlager med översatta (flyttade) kopior av funktionerna. De ursprungliga funktionerna kopieras också.

Python-kod

Algoritm-ID: `native:arrayöversatta funktioner`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa rutnät

Skapar ett vektorlager med ett rutnät som täcker en given utsträckning. Rutnätscellerna kan ha olika former:

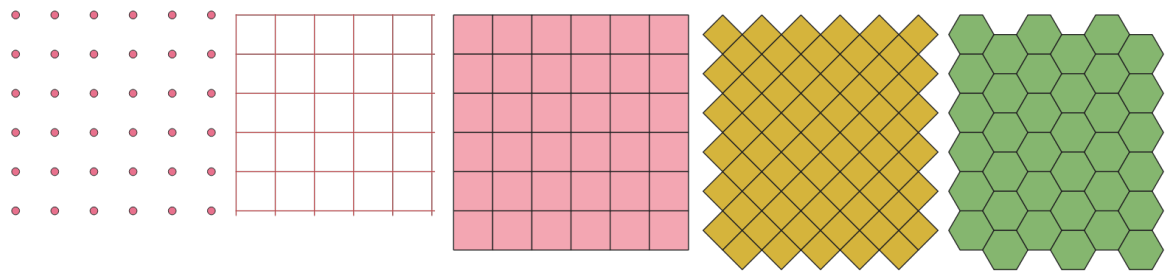


Fig. 24.48: Olika former av rutnätsceller appliceras i samma utsträckning, utan överlappningar

Storleken och/eller placeringen av varje element i rutnätet definieras med hjälp av ett horisontellt och vertikalt avstånd. CRS för utmatningslagret måste definieras. Rutnätets utbredning och avståndsvärdena måste uttryckas i koordinaterna och enheterna för detta CRS.

Standardmeny: *Vektor ► Forskningsverktyg*

Parametrar

Etikett	Namn	Typ	Beskrivning
Typ av rutnät	`TYPE`	[uppräkning] Standard: 0	Formen på rutnätet. En av: • 0 — Punkt • 1 — Linje • 2 — Rektangel (polygon) • 3 — Diamant (polygon) • 4 — Hexagon (polygon)
Grid utsträckning	`EXTENT`	[omfattning]	Rutnätets utbredning Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Horisontellt avstånd	`HSPACING`	[numerisk: dubbel] Standard: 1,0	Storlek på en rutnätscell på X-axeln
Vertikalt avstånd	`VSPACING`	[numerisk: dubbel] Standard: 1,0	Storlek på en rutnätscell på Y-axeln
Horisontell överlagring	`HOVERLAY`	[numerisk: dubbel] Standard: 0,0	Överlagringsavstånd mellan två på varandra följande rutnätsceller på X-axeln
Vertikal överlagring	`VOVERLAY`	[numerisk: dubbel] Standard: 0,0	Överlagra avståndet mellan två på varandra följande rutnätsceller på Y-axeln
Grid CRS	`CRS`	[crs] Standard: *Projekt CRS*	Referenssystem för koordinater som ska tillämpas på nätet

continues on next page

Table 24.122 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Denna algoritm skapar ett vektorlager med ett rutnät som täcker en angiven utsträckning. Elementen i rutnätet kan vara punkter, linjer eller polygoner. Storleken och/eller placeringen av varje elementen i rutnätet bestäms av de horisontella och vertikala avstånden. Koordinatsystemet i utdata måste anges. Rutnätets utsträckning och avståndsvärden måste anges i koordinaterna och enheterna för koordinatsystemet. Den övre vänstra punkten (minX, maxY) används som referenspunkt. Det betyder att, vid den punkten, är ett element garanterat att placeras. Om inte bredden och höjden av den valda utsträckningen är en multipel av de valda avstånden, är detta inte sant för de andra punkterna som definierar den utsträckningen.	`OUTPUT`	[vektor: geometri] Standard: `[Skapa tillfälligt lager]`	Resulterande vektorrasterlager. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Denna algoritm skapar ett vektorlager med ett rutnät som täcker en angiven utsträckning. Elementen i rutnätet kan vara punkter, linjer eller polygoner. Storleken och/eller placeringen av varje elementen i rutnätet bestäms av de horisontella och vertikala avstånden. Koordinatsystemet i utdata måste anges. Rutnätets utsträckning och avståndsvärden måste anges i koordinaterna och enheterna för koordinatsystemet. Den övre vänstra punkten (minX, maxY) används som referenspunkt. Det betyder att, vid den punkten, är ett element garanterat att placeras. Om inte bredden och höjden av den valda utsträckningen är en multipel av de valda avstånden, är detta inte sant för de andra punkterna som definierar den utsträckningen.	`OUTPUT`	[vektor: geometri]	Resulterande vektornätskikt. Geometritypen för utdata (punkt, linje eller polygon) beror på *Grid type*. Funktioner skapas från topp till botten, från vänster till höger. Attributtabellen är fylld med: • Raderade objektet ID permanent • koordinater på sidorna `vänster`, `höger`, `överst` och `` nederst`` • och deras placering i rutnätet: `row_index` och `column_index` (tillgängligt för rutnätstyperna punkt, rektangel och hexagon)

Python-kod

Algoritm-ID: `native:creategrid`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa punktlager från tabell

Skapar ett punktlager från en tabell med kolumner som innehåller koordinatfält.

Förutom X- och Y-koordinater kan du också ange Z- och M-fält.

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Vektorlager för indata eller en tabell.
X-fält	`XFIELD`	[tabellfält: alla]	Fält som innehåller X-koordinaten
Y-fält	`YFIELD`	[tabellfält: alla]	Fält som innehåller Y-koordinaten
Z-fält Valfritt	`ZFIELD`	[tabellfält: alla]	Fält som innehåller Z-koordinaten
M fält Valfritt	`MFIELD`	[tabellfält: alla]	Fält som innehåller M-värdet
Mål CRS	`TARGET_CRS`	[crs] Standard: `EPSG:4326`	Koordinatreferenssystem att använda för lagret. De angivna koordinaterna antas vara kompatibla.
Pekar från bordet	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange det resulterande punktlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Pekar från bordet	`OUTPUT`	[vektor: punkt]	Det resulterande punktlagret

Python-kod

Algoritm-id: `native:createpointslayerfromtable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Generera punkter (pixelcentrum) längs linjen

Genererar ett punktvektorlager från ett raster- och linjelager.

Punkterna motsvarar de pixelcentroider som skär linjelagret.

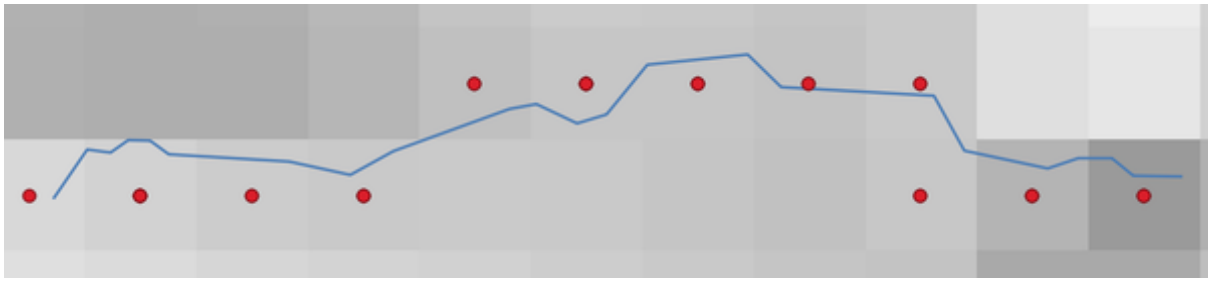


Fig. 24.49: Punkter för pixel-centroiderna

Parametrar

Etikett	Namn	Typ	Beskrivning
Ett rasterlager	`INPUT_RASTER`	[raster]	Inmatat rasterlager
Ett vektorlager	`INPUT_VECTOR`	[vektor: linje]	Ingångslinje vektorlager
Pekar ut längs linjen	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Resulterande punktlager med pixelcentroider. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Pekar ut längs linjen	`OUTPUT`	[vektor: punkt]	Resulterande punktlager med pixel-centroider

Python-kod

Algoritm-ID: `qgis:generatepointspixelcentroidsalongline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Generera punkter (pixelcentrum) inuti polygonen

Genererar ett punktvektorskikt från ett raster- och polygonlager.

Punkterna motsvarar de pixelcentroider som skär polygonlagret.

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

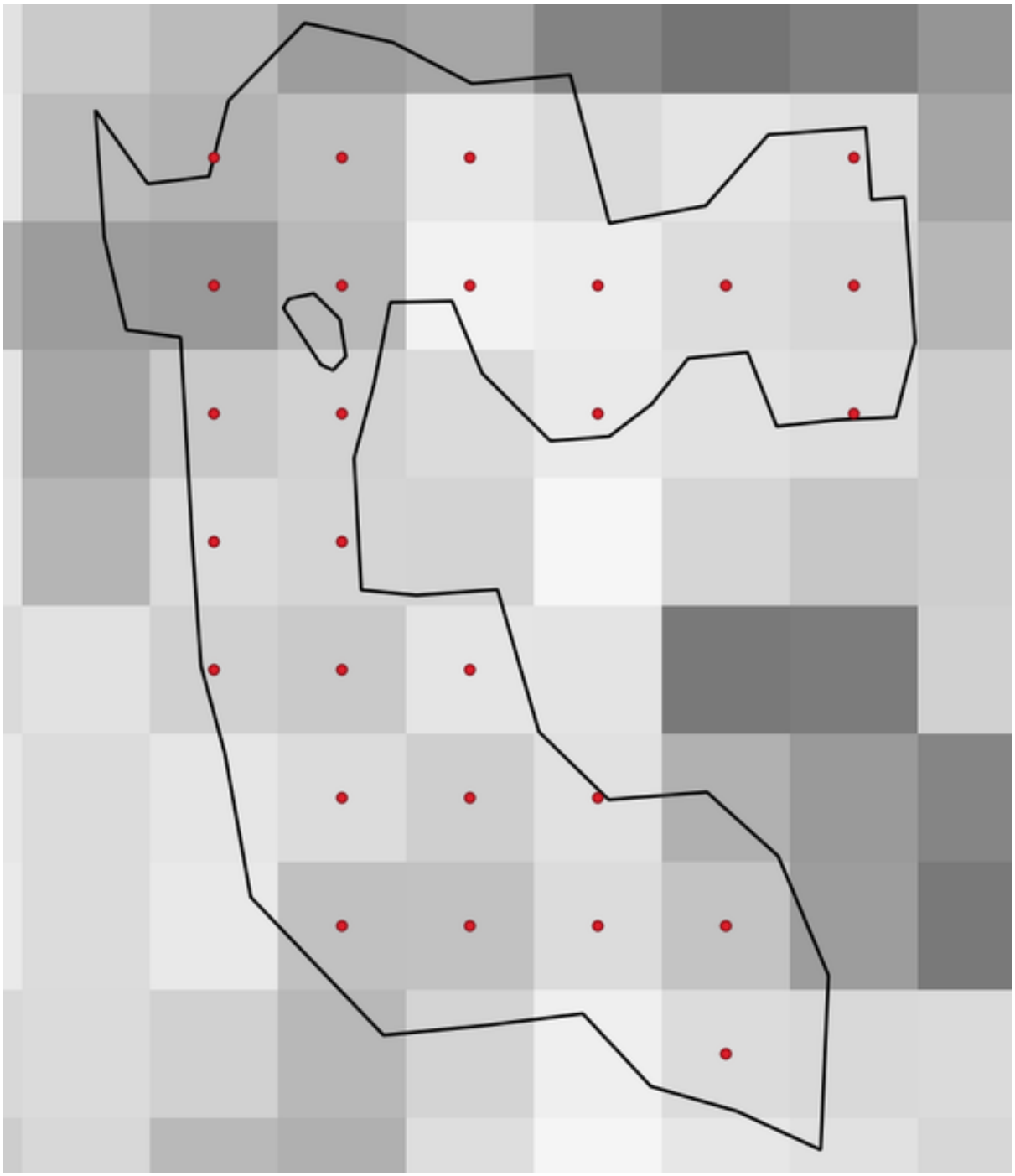


Fig. 24.50: Punkter för pixel-centroiderna

Parametrar

Etikett	Namn	Typ	Beskrivning
Ett rasterlager	`INPUT_RASTER`	[raster]	Inmatat rasterlager
Ett vektorlager	`INPUT_VECTOR`	[vektor: polygon]	Vektorlager för inmatad polygon
Punkter inuti polygoner	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Resulterande punktlager av pixel-centroider. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Punkter inuti polygoner	`OUTPUT`	[vektor: punkt]	Resulterande punktlager av pixel-centroider

Python-kod

Algoritm-ID: `` `native:genererapunktspixelcentroiderinsidepolygoner ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Importera geotaggade foton

Skapar ett punktlager som motsvarar de geotaggade platserna från JPEG- eller HEIC/HEIF-bilder från en källmapp.

Punktlagret kommer att innehålla en enda PointZ-funktion per indatafil från vilken geotaggarna kan läsas. Eventuell höjdinformation från geotaggarna kommer att användas för att ställa in punktens Z-värde.

Förutom longitud och latitud läggs även information om höjd, riktning och tidsstämpel, om sådan finns på fotot, till punkten som attribut.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsmapp	`FOLDER`	mapp	Sökväg till källmappen som innehåller de geotaggade bilderna
Scanning rekursivt	`RECURSIVE`	[boolean] Standard: Falsk	Om den är markerad kommer mappen och dess undermappar att skannas

continues on next page

Table 24.124 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Foton Valfritt	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange punktvektorlagret för de geotaggade bilderna. *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Invalid photos table (ogiltig bildtabell) Valfritt	`INVALID`	[vektor: tabell] Standard: `[Hoppa över utdata]`	Ange tabellen över olästa eller ej geotaggade foton. *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Foton	`OUTPUT`	[vektor: punkt]	Punktvektorlager med geotaggade foton. Lagrets form fylls automatiskt med banor och inställningar för förhandsgranskning av foton.
Invalid photos table (ogiltig bildtabell) Valfritt	`INVALID`	[vektor: tabell]	Tabell över olästa eller icke geotaggade foton kan också skapas.

Python-kod

Algoritm-ID: `native:importphotos`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Pekar ut vägen

Konverterar ett punktlager till ett linjelager genom att sammanfoga punkter i en ordning som definieras av ett uttryck eller ett fält i det ingående punktlagret.

Punkterna kan grupperas efter ett fält eller ett uttryck för att särskilja linjefunktioner.

Förutom linjevektorlagret matas en textfil ut som beskriver den resulterande linjen som en startpunkt och en sekvens av bäringar / riktningar (relativt azimut) och avstånd.

Observera

Den här algoritmen använder ellipsoidbaserade mätningar och respekterar de aktuella *ellipsoidinställningarna*.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningspunkt lager	`INPUT`	[vektor: punkt]	Vektorlager för inmatningspunkt
Skapa stängda vägar	`CLOSE_PATH`	[boolean] Standard: Falsk	Om den är markerad kommer linjens första och sista punkt att anslutas och stänga den genererade sökvägen
Orderuttryck Valfritt	`ORDER_EXPRESSI`	[uttryck]	Fält eller uttryck som anger i vilken ordning punkterna i sökvägen ska kopplas samman. Om det inte anges används funktionens ID (`$id`).
Sortera text som innehåller siffror på ett naturligt sätt	`NATURAL_SORT`	[boolean] Standard: Falsk	Om markerad sorteras funktionerna naturligt utifrån det angivna uttrycket (t.ex. 'a9' < 'a10').
Stiggruppsuttryck Valfritt	`GROUP_EXPRESSI`	[uttryck]	Punktfunktioner med samma värde i fältet eller uttrycket kommer att grupperas i samma linje. Om den inte är inställd ritas en enda bana med alla inmatade punkter.
Stigar	`OUTPUT`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange linjevektorlagret för sökvägen. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Directory för textutmatning Valfritt	`OUTPUT_TEXT_DI`	mapp Standard: `[Hoppa över utdata]`	Ange den katalog som ska innehålla beskrivningsfilerna för punkter och banor. *En av*: • Hoppa över utmatning • Spara till en temporär katalog • Spara till katalog

Utgångar

Etikett	Namn	Typ	Beskrivning
Stigar	`OUTPUT`	[vektor: linje]	Linjevektorlager för banan
Directory för textutmatning	`OUTPUT_TEXT_DI`	mapp	Katalog som innehåller beskrivningsfiler för punkter och banor

Python-kod

Algoritm-ID: `native:pointstopath`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Slumpmässiga punkter längs linjen

Skapar ett nytt punktlager med punkter som placeras på linjerna i ett annat lager.

För varje rad i inmatningslagret läggs ett givet antal punkter till i det resulterande lagret. Proceduren för att lägga till en punkt är att:

1. slumpmässigt välja en linjefunktion från inmatningslagret
2. om objektet består av flera delar, slumpmässigt välja en del av det
3. slumpmässigt välja ett segment av den linjen
4. slumpmässigt välja en position på det segmentet.

Förfarandet innebär att kurviga delar av linjerna (med relativt korta segment) får fler punkter än raka delar (med relativt långa segment), vilket framgår av illustrationen nedan, där resultatet av algoritmen *Random points along lines* kan jämföras med resultatet av algoritmen *Random points on lines* (som ger punkter med en i genomsnitt jämn fördelning längs linjerna).

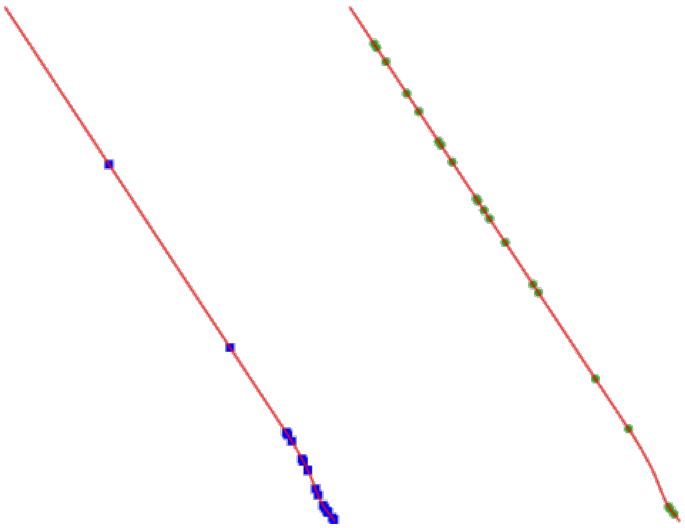


Fig. 24.51: Exempel på algoritmutdata. Till vänster: *Slumpmässiga punkter längs linje*, höger: *Slumpmässiga punkter på linjer*

Ett minimiavstånd kan anges för att undvika att punkterna ligger för nära varandra.

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Slumpmässiga punkter på linjer

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningspunkt lager	`INPUT`	[vektor: linje]	Ingångslinje vektorlager
Antal poäng	`POINTS_NUMBER`	[numerisk: heltal] Standard: 1	Antal poäng att skapa
Minsta avstånd mellan punkterna	`MIN_DISTANCE`	[numerisk: dubbel] Standard: 0,0	Det minsta avståndet mellan punkter
Slumpmässiga poäng	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Slumpmässiga punkter för utdata. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Slumpmässiga poäng	`OUTPUT`	[vektor: punkt]	Utmatningslagret för slumpmässiga punkter.

Python-kod

Algoritm-ID: `qgis:qgisrandompointsalongline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Slumpmässiga punkter i utbredning

Skapar ett nytt punktlager med ett givet antal slumpmässiga punkter, alla inom en given utsträckning.

En avståndsfaktor kan anges för att undvika att punkter ligger för nära varandra. Om det minsta avståndet mellan punkterna gör det omöjligt att skapa nya punkter, kan antingen avståndet minskas eller det maximala antalet försök ökas.

Standardmeny: *Vektor* ► *Forskningsverktyg*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Input omfattning	`EXTENT`	[omfattning]	Kartutbredning för slumpmässiga punkter Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Antal poäng	`POINTS_NUMBER`	[numerisk: heltal] Standard: 1	Antal punkter som ska skapas
Minsta avstånd mellan punkterna	`MIN_DISTANCE`	[numerisk: dubbel] Standard: 0,0	Det minsta avståndet mellan punkter
Mål CRS	`TARGET_CRS`	[crs] Standard: *Projekt CRS*	CRS för lagret med slumpmässiga punkter
Slumpmässiga poäng	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Slumpmässiga punkter för utdata. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Maximalt antal sökförsök givet minsta möjliga avstånd	`MAX_ATTEMPTS`	[numerisk: heltal] Standard: 200	Maximalt antal försök att placera ut poängen

Utgångar

Etikett	Namn	Typ	Beskrivning
Slumpmässiga poäng	`OUTPUT`	[vektor: punkt]	Utmatningslagret för slumpmässiga punkter.

Python-kod

Algoritm-ID: `native:randompointsinextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Slumpmässiga punkter i lagergränser

Skapar ett nytt punktlager med ett givet antal slumpmässiga punkter, som alla ligger inom ett givet lagers utsträckning.

Ett minimiavstånd kan anges för att undvika att punkterna ligger för nära varandra.

Standardmeny: *Vektor* ► *Forskningsverktyg*

> **Varning**
>
> Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: polygon]	Polygonlager för inmatning som definierar området
Antal poäng	`POINTS_NUMBER`	[numerisk: heltal] Standard: 1	Antal poäng att skapa
Minsta avstånd mellan punkterna	`MIN_DISTANCE`	[numerisk: dubbel] Standard: 0,0	Det minsta avståndet mellan punkter
Slumpmässiga poäng	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Slumpmässiga punkter för utdata. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Slumpmässiga poäng	`OUTPUT`	[vektor: punkt]	Utmatningslagret för slumpmässiga punkter.

Python-kod

Algoritm-ID: `qgis:randompointsinlayerbounds`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Slumpmässiga punkter i polygoner

Skapar ett punktlager med punkter som placeras inuti polygonerna i ett annat lager.

För varje geometri (polygon/multipolygon) i inmatningslagret läggs det angivna antalet punkter till i resultatlagret.

Minimiavstånd per funktion och globala minimiavstånd kan anges för att undvika att punkterna ligger för nära varandra i det utgående punktlagret. Om ett minimiavstånd anges kanske det inte är möjligt att generera det angivna antalet punkter för varje funktion. Det totala antalet genererade punkter och missade punkter finns tillgängliga som utdata från algoritmen.

Illustrationen nedan visar effekten av minimiavstånd per funktion och globala minimiavstånd samt minimiavstånd noll/icke-noll (genererade med samma frö, så åtminstone den första genererade punkten blir densamma).

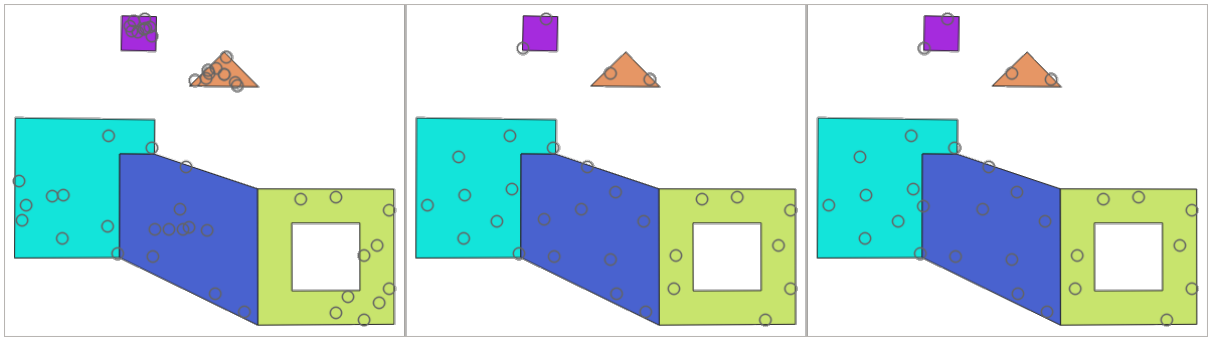


Fig. 24.52: Tio punkter per polygonfunktion, *vänster*: min. avstånd = 0, *mitt*: min. avstånd = 1, *höger*: min. avstånd = 1, globalt min. avstånd = 0

Det maximala antalet försök per punkt kan anges. Detta är endast relevant för minsta avstånd som inte är noll.

Ett frö för slumptalsgeneratorn kan tillhandahållas, vilket gör det möjligt att få identiska slumptalssekvenser för olika körningar av algoritmen.

Attributen för den polygon på vilken en punkt genererades kan inkluderas (*Include polygon attributes*).

Om du vill ha ungefär samma punkttäthet för alla funktioner kan du datadefiniera antalet punkter med hjälp av ytan för polygonfunktionens geometri.

Se även

Slumpmässiga punkter inuti polygoner

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Polygonlager för inmatning	`INPUT`	[vektor: linje]	Vektorlager för inmatad polygon
Antal poäng för varje funktion	`POINTS_NUMBER`	[numerisk: heltal] Standard: 1	Antal poäng att skapa
Minsta avstånd mellan punkterna Valfritt	`MIN_DISTANCE`	[numerisk: dubbel] Standard: 0,0	Det minsta avståndet mellan punkter inom en polygonfunktion
Slumpmässiga punkter i polygoner	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Slumpmässiga punkter för utdata. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Globalt minsta avstånd mellan punkter Valfritt	MIN_DISTANCE_G	[numerisk: dubbel] Standard: 0,0	Det globala minsta avståndet mellan punkter. Bör vara mindre än *Minsta avstånd mellan punkter (per funktion)* för att den parametern ska ha någon effekt.
Maximalt antal sökförsök (för Min. dist. > 0) Valfritt	MAX_TRIES_PER_	[numerisk: heltal] Standard: 10	Det maximala antalet försök per punkt. Endast relevant om det minsta avståndet mellan punkterna är inställt (och större än 0).
Slumpmässigt frö Valfritt	SEED	[numerisk: heltal] Standard: Ej inställd	Det frö som ska användas för slumptalsgeneratorn.
Inkludera polygonattribut	INCLUDE_POLYGO	[boolean] Standard: Sann	Om den är inställd kommer en punkt att få attributen från den linje som den är placerad på.

Utgångar

Etikett	Namn	Typ	Beskrivning
Slumpmässiga punkter i polygoner	OUTPUT	[vektor: punkt]	Utmatningslagret för slumpmässiga punkter.
Antal funktioner med tom eller ingen geometri	FEATURES_WITH_	[numerisk: heltal]	
Totalt antal genererade poäng	OUTPUT_POINTS	[numerisk: heltal]	
Antal missade poäng	POINTS_MISSED	[numerisk: heltal]	Antalet punkter som inte kunde genereras på grund av minimiavståndsbegränsningen.
Antal funktioner med missade poäng	POLYGONS_WITH_	[numerisk: heltal]	Inkluderar inte funktioner med tom eller ingen geometri

Python-kod

Algoritm-id: `native:randompointsinpolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Slumpmässiga punkter inuti polygoner

Skapar ett nytt punktlager med ett givet antal slumpmässiga punkter i varje polygon i det inmatade polygonlagret.

Två provtagningsstrategier finns tillgängliga:

- Antal poäng: antal poäng för varje funktion
- Poängtäthet: täthet av poäng för varje funktion

Ett minimiavstånd kan anges för att undvika att punkterna ligger för nära varandra.

Standardmeny: *Vektor* ► *Forskningsverktyg*

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Slumpmässiga punkter i polygoner

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: polygon]	Vektorlager för inmatad polygon
Strategi för urval	`STRATEGY`	[uppräkning] Standard: 0	Samplingsstrategi att använda. En av: • 0 — Antal poäng: antal poäng för varje funktion • 1 — Poängtäthet: antal poäng för varje funktion
Punktantal eller täthet	`VALUE`	[numerisk: dubbel] Standard: 1,0	Antalet punkter eller deras täthet, beroende på vald *Samplingsstrategi*.
Minsta avstånd mellan punkterna	`MIN_DISTANCE`	[numerisk: dubbel] Standard: 0,0	Det minsta avståndet mellan punkter
Slumpmässiga poäng	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Slumpmässiga punkter för utdata. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Slumpmässiga poäng	`OUTPUT`	[vektor: punkt]	Utmatningslagret för slumpmässiga punkter.

Python-kod

Algoritm-ID: `qgis:randompointsinsidepolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Slumpmässiga punkter på linjer

Skapar ett punktlager med punkter som placeras på linjerna i ett annat lager.

För varje geometrisk funktion (linje/flertal linjer) i inmatningslagret läggs det angivna antalet punkter till i resultatlagret.

Minimiavstånd per funktion och globala minimiavstånd kan anges för att undvika att punkterna ligger för nära varandra i det utgående punktlagret. Om ett minimiavstånd anges kanske det inte är möjligt att generera det angivna antalet punkter för varje funktion. Det totala antalet genererade punkter och missade punkter finns tillgängliga som utdata från algoritmen.

Illustrationen nedan visar effekten av minimiavstånd per funktion och globala minimiavstånd samt minimiavstånd noll/icke-noll (genererade med samma frö, så åtminstone den första genererade punkten blir densamma).

Fig. 24.53: Funktion med fem punkter per linje, *vänster*: min. avstånd = 0, *mitt*: min.avstånd != 0, *höger*: min. avstånd != 0, globalt min. avstånd = 0

Det maximala antalet försök per punkt kan anges. Detta är endast relevant för minsta avstånd som inte är noll.

Ett frö för slumptalsgeneratorn kan tillhandahållas, vilket gör det möjligt att få identiska slumptalssekvenser för olika körningar av algoritmen.

Attributen för den linjefunktion på vilken en punkt genererades kan inkluderas (*Include line attributes*).

Om du vill ha ungefär samma punkttäthet för alla linjefunktioner kan du datadefiniera antalet punkter med hjälp av längden på linjefunktionens geometri.

Se även

Slumpmässiga punkter längs linjen

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsledning lager	`INPUT`	[vektor: linje]	Ingångslinje vektorlager
Antal poäng för varje funktion	`POINTS_NUMBER`	[numerisk: heltal] Standard: 1	Antal poäng att skapa

continues on next page

Table 24.131 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Minsta avstånd mellan punkterna (per funktion) Valfritt	MIN_DISTANCE	[numerisk: dubbel] Standard: 0,0	Det minsta avståndet mellan punkter inom en linjefunktion
Slumpmässiga punkter på linjer	OUTPUT	[vektor: punkt] Standard: [Skapa tillfälligt lager]	Slumpmässiga punkter för utdata. *En av*: • Skapa ett tillfälligt lager (TEMPORARY_OUTPUT) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Globalt minsta avstånd mellan punkter Valfritt	MIN_DISTANCE_G	[numerisk: dubbel] Standard: 0,0	Det globala minsta avståndet mellan punkter. Bör vara mindre än *Minsta avstånd mellan punkter (per funktion)* för att den parametern ska ha någon effekt.
Maximalt antal sökförsök (för Min. dist. > 0) Valfritt	MAX_TRIES_PER_	[numerisk: heltal] Standard: 10	Det maximala antalet försök per punkt. Endast relevant om det minsta avståndet mellan punkterna är inställt (och större än 0).
Slumpmässigt frö Valfritt	SEED	[numerisk: heltal] Standard: Ej inställd	Det frö som ska användas för slumptalsgeneratorn.
Inkludera linjeattribut	INCLUDE_LINE_A	[boolean] Standard: Sann	Om den är inställd kommer en punkt att få attributen från den linje som den är placerad på.

Utgångar

Etikett	Namn	Typ	Beskrivning
Slumpmässiga punkter på linjer	OUTPUT	[vektor: punkt]	Utmatningslagret för slumpmässiga punkter.
Antal funktioner med tom eller ingen geometri	FEATURES_WITH_	[numerisk: heltal]	
Antal funktioner med missade poäng	LINES_WITH_MIS	[numerisk: heltal]	Inkluderar inte funktioner med tom eller ingen geometri
Totalt antal genererade poäng	POINTS_GENERAT	[numerisk: heltal]	
Antal missade poäng	POINTS_MISSED	[numerisk: heltal]	Antalet punkter som inte kunde genereras på grund av minimiavståndsbegränsningen.

Python-kod

Algoritm-ID: `native:randompointsonlines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rasterpixlar till punkter

Skapar ett vektorlager med punkter som motsvarar varje pixel i ett rasterlager.

Konverterar ett rasterlager till ett vektorlager genom att skapa punktfunktioner för varje enskild pixels centrum i rasterlagret. Alla NoData-pixlar hoppas över i utdata.

Parametrar

Etikett	Namn	Typ	Beskrivning
Ett rasterlager	`INPUT_RASTER`	[raster]	Inmatat rasterlager
Bandnummer	`RASTER_BAND`	[rasterband]	Rasterband att extrahera data från
Fältets namn	`FIELD_NAME`	[sträng] Standard: 'VÄRDE'	Namn på det fält där värdet för rasterbandet lagras
Vektorpunkter	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange det resulterande punktlagret med pixlarnas centroider. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Vektorpunkter	`OUTPUT`	[vektor: punkt]	Resulterande punktlager med pixlarnas centroider

Python-kod

Algoritm-ID: `native:pixelstopoints`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rasterpixlar till polygoner

Skapar ett vektorlager med polygoner som motsvarar varje pixel i ett rasterlager.

Konverterar ett rasterlager till ett vektorlager genom att skapa polygonfunktioner för varje enskild pixels utbredning i rasterlagret. Eventuella NoData-pixlar hoppas över i utdata.

Parametrar

Etikett	Namn	Typ	Beskrivning
Ett rasterlager	`INPUT_RASTER`	[raster]	Inmatat rasterlager
Bandnummer	`RASTER_BAND`	[rasterband]	Rasterband att extrahera data från
Fältets namn	`FIELD_NAME`	[sträng] Standard: 'VÄRDE'	Namn på det fält där värdet för rasterbandet lagras
Vektorpolygoner	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det resulterande polygonlagret med pixelutsträckningar. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Vektorpolygoner	`OUTPUT`	[vektor: polygon]	Resulterande polygonlager med pixelexterienter

Python-kod

Algoritm-id: `native:pixelstopolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Regelbundna poäng

Skapar ett nytt punktlager med punkterna placerade i ett regelbundet rutnät inom en given utsträckning.

Rutnätet specificeras antingen genom avståndet mellan punkterna (samma avstånd för alla dimensioner) eller genom antalet punkter som ska genereras. I det senare fallet kommer avståndet att bestämmas utifrån omfattningen. För att generera ett helt rektangulärt rutnät genereras i det senare fallet minst det antal punkter som användaren har angett.

Slumpmässiga förskjutningar av punktavståndet kan tillämpas, vilket resulterar i ett oregelbundet punktmönster.

Standardmeny: *Vektor ► Forskningsverktyg*

Parametrar

Etikett	Namn	Typ	Beskrivning
Input omfattning (xmin, xmax, ymin, ymax)	`EXTENT`	[omfattning]	Kartutbredning för slumpmässiga punkter Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Punktavstånd/antal	`SPACING`	[numerisk: heltal] Standard: 100	Avståndet mellan punkterna, eller antalet punkter, beroende på om `Use point spacing` är markerat eller inte.
Initialinsats från hörn (vänster sida)	`INSET`	[numerisk: dubbel] Standard: 0,0	Förskjuter punkterna i förhållande till det övre vänstra hörnet. Värdet används för både X- och Y-axeln.
Tillämpa slumpmässig förskjutning av punktavståndet	`RANDOMIZE`	[boolean] Standard: Falsk	Om kryssrutan är markerad kommer punkterna att ha ett slumpmässigt avstånd
Använd punktavstånd	`IS_SPACING`	[boolean] Standard: Sann	Om den inte är markerad tas ingen hänsyn till punktavståndet
Utgående lager CRS	`CRS`	[crs] Standard: *Projekt CRS*	CRS för lagret med slumpmässiga punkter
Regelbundna poäng	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange utmatningen för det reguljära punktlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Regelbundna poäng	`OUTPUT`	[vektor: punkt]	Det utgående vanliga punktlagret.

Python-kod

Algoritm-ID: `qgis:regularpoints`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.22 Vektor allmänt

Tilldela projektion

Tilldelar en ny projektion till ett vektorlager.

Det skapar ett nytt lager med exakt samma funktioner och geometrier som det ingående lagret, men tilldelat ett nytt CRS. Geometrierna är **inte** omprojicerade, de är bara tilldelade till ett annat CRS.

Denna algoritm kan användas för att reparera lager som har tilldelats en felaktig projektion.

Attributen ändras inte av denna algoritm.

Se även

Definiera Shapefile-projektion, *Hitta projektion*, *Återprojicera lagret*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Vektorlager med felaktig eller saknad CRS
Tilldelad CRS	`CRS`	[crs] Standard: `EPSG:4326 – WGS84`	Välj den nya CRS som ska tilldelas till vektorlagret
Tilldelad CRS Valfritt	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Tilldelad CRS	`OUTPUT`	[samma som inmatning]	Vektorlager med tilldelad projektion

Python-kod

Algoritm-id: `native:assignprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Batch Nominatim geokodare

Utför batchgeokodning med hjälp av Nominatim-tjänsten mot ett strängfält i ett inmatningslager. Utmatningslagret kommer att ha en punktgeometri som återspeglar den geokodade platsen samt ett antal attribut som är associerade med den geokodade platsen.

Tillåter *features in-place modification* av punktfunktioner

Observera

Denna algoritm är förenlig med användningspolicyn för geokodningstjänsten Nominatim som tillhandahålls av OpenStreetMap Foundation.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Vektorlager för att geokoda funktionerna
Adressfält	`FIELD`	[tabellfält: sträng]	Fält som innehåller de adresser som ska geokodas
Geokodad	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret som endast innehåller de geokodade adresserna. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Geokodad	`OUTPUT`	[vektor: punkt]	Vektorlager med punktfunktioner som motsvarar de geokodade adresserna

Python-kod

Algoritm-id: `native:batchnominatimgeocoder`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Konvertera lager till spatiala bokmärken

Skapar spatiala bokmärken som motsvarar omfattningen av funktioner som finns i ett lager.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje, polygon]	Ingångsvektorlagret
Bokmärk destination	`DESTINATION`	[uppräkning] Standard: 0	Välj destination för bokmärkena. En av följande: • 0 — Bokmärken för projekt • 1 — Användarens bokmärken
Namnfält	`NAME_EXPRESSIO`	[uttryck]	Fält eller uttryck som ger namn åt de genererade bokmärkena
Gruppfält	`GROUP_EXPRESSI`	[uttryck]	Fält eller uttryck som ger grupper för de genererade bokmärkena

Utmatningar

Etikett	Namn	Typ	Beskrivning
Antal bokmärken som lagts till	`COUNT`	[numerisk: heltal]	

Python-kod

Algoritm-ID: `native:layertobookmarks`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Konvertera spatiala bokmärken till lager

Skapar ett nytt lager som innehåller polygonfunktioner för lagrade spatiala bokmärken. Exporten kan filtreras till endast bokmärken som hör till det aktuella projektet, till alla användares bokmärken eller en kombination av båda.

Parametrar

Etikett	Namn	Typ	Beskrivning
Källa till bokmärke	`SOURCE`	[uppräkning] [lista] Standard: [0,1]	Välj källor för bokmärkena. En eller flera av: • 0 — Bokmärken för projekt • 1 — Användarens bokmärken
Utmatning CRS	`CRS`	[crs] Standard: `EPSG:4326 - WGS 84`	CRS för det utgående lagret
Utmatning	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Utmatning	`OUTPUT`	[vektor: polygon]	Vektorlagret för utdata (bokmärken)

Python-kod

Algoritm-ID: `native:bookmarkstolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa attributindex

Skapar ett index mot ett fält i attributtabellen för att snabba upp sökningar. Stödet för indexskapande beror både på lagrets dataleverantör och på fälttypen.

Inga utdata skapas: indexet lagras på själva lagret.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Välj det vektorlager som du vill skapa ett attributindex för
Attribut till index	`FIELD`	[tabellfält: alla]	Fält för vektorlagret

Utmatningar

Etikett	Namn	Typ	Beskrivning
Indexerat lager	`OUTPUT`	[samma som inmatning]	En kopia av inmatningsvektorlagret med ett index för det angivna fältet

Python-kod

Algoritm-ID: `native:createattributeindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa spatialt index

Skapar ett index för att påskynda åtkomsten till funktionerna i ett lager baserat på deras spatiala placering. Stöd för skapande av spatiala index är beroende av lagrets dataleverantör.

Inga nya utgående lager skapas.

Standardmeny: *Vektor* ► *Verktyg för datahantering*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager

Utmatningar

Etikett	Namn	Typ	Beskrivning
Indexerat lager	`OUTPUT`	[samma som inmatning]	En kopia av inmatningsvektorlagret med ett spatialt index

Python-kod

Algoritm ID: `native:createspatialindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Definiera Shapefile-projektion

Ställer in CRS (projektion) för en befintlig Shapefile-dataset till den angivna CRS. Det är mycket användbart när ett dataset i Shapefile-format saknar filen `prj` och du vet den korrekta projektionen.

I motsats till algoritmen *Tilldela projektion* modifierar den det aktuella lagret och ger inte ut ett nytt lager.

Observera

För Shapefile-dataset kommer filerna `.prj` och `.qpj` att skrivas över - eller skapas om de saknas - för att matcha det tillhandahållna CRS.

Standardmeny: *Vektor* ► *Verktyg för datahantering*

Se även

Tilldela projektion, *Hitta projektion*, *Återprojicera lagret*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Vektorlager med saknad projektionsinformation
CRS	`CRS`	[crs]	Välj den CRS som ska tilldelas till vektorlagret

Utmatningar

Etikett	Namn	Typ	Beskrivning
	`INPUT`	[samma som inmatning]	Ingångsvektorlagret med den definierade projektionen

Python-kod

Algoritm-ID: `qgis:definecurrentprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Ta bort duplicerade geometrier

Hittar och tar bort duplicerade geometrier.

Attributen kontrolleras inte, så om två funktioner har identiska geometrier men olika attribut kommer endast en av dem att läggas till i resultatlagret.

Observera

Denna algoritm kräver inte giltiga geometrier som indata.

Se även

Geometrier för droppar, *Ta bort nollgeometrier*, *Ta bort dubbletter per attribut*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Lagret med duplicerade geometrier som du vill rengöra
Rensad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Antal kasserade duplicerade poster	`DUPLICATE_COUN`	[numerisk: heltal]	Antal kasserade duplicerade poster
Rensad	`OUTPUT`	[samma som inmatning]	Utmatningslagret utan några duplicerade geometrier
Antal bevarade poster	`RETAINED_COUNT`	[numerisk: heltal]	Antal unika poster

Python-kod

Algoritm-ID: `native:deleteduplicategeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Ta bort dubbletter per attribut

Raderar duplicerade rader genom att endast beakta det angivna fältet/fälten. Den första matchande raden behålls och duplikat kasseras.

Alternativt kan dessa duplicerade poster sparas i en separat utdata för analys.

Se även

Ta bort duplicerade geometrier

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Det ingående lagret
Fält att matcha dubbletter med	`FIELDS`	[tabellfält: valfritt] [lista]	Fält som definierar duplikat. Funktioner med identiska värden för alla dessa fält betraktas som duplikat.
Filtrerad (inga dubbletter)	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret som innehåller de unika egenskaperna. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Filtrerad (dubbletter) Valfritt	`DUPLICATES`	[samma som inmatning] Standard: `[Skip output]`	Ange utdatalagret som endast innehåller duplikaten. *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Filtrerad (dubbletter) Valfritt	`DUPLICATES`	[samma som inmatning] Standard: `[Skip output]`	Vektorlager som innehåller de borttagna funktionerna. Produceras inte om det inte anges (lämnas som `[Skip output]`).
Antal kasserade duplicerade poster	`DUPLICATE_COUN`	[numerisk: heltal]	Antal kasserade duplicerade poster
Filtrerad (inga dubbletter)	`OUTPUT`	[samma som inmatning]	Vektorlager som innehåller de unika egenskaperna.
Antal bevarade poster	`RETAINED_COUNT`	[numerisk: heltal]	Antal unika poster

Python-kod

Algoritm-ID: `native:removeduplicatesbyattribute`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Upptäck förändringar i datasetet

Jämför två vektorlager och fastställer vilka funktioner som är oförändrade, tillagda eller borttagna mellan de två. Den är utformad för att jämföra två olika versioner av samma dataset.

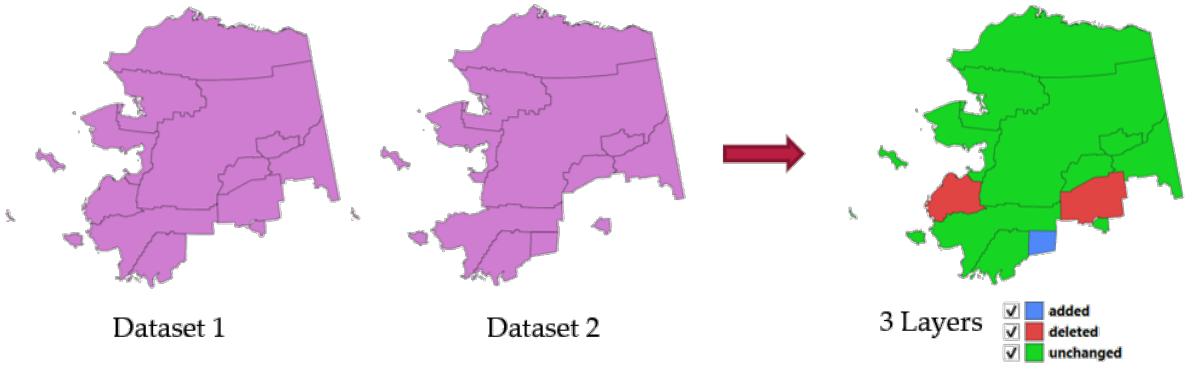


Fig. 24.54: Detektera ändring av dataset exempel

Parametrar

Etikett	Namn	Typ	Beskrivning
Originallager	`ORIGINAL`	[vektor: geometri]	Vektorlagret betraktas som originalversionen
Reviderat lager	`REVISED`	[vektor: geometri]	Det reviderade eller modifierade vektorlagret
Attribut att ta hänsyn till vid matchning Valfritt	`COMPARE_ATTRIB`	[tabellfält: valfritt] [lista]	Attribut som ska beaktas för matchning. Som standard jämförs alla attribut.
Beteende vid jämförelse av geometri Valfritt	`MATCH_TYPE`	[uppräkning] Standard: 1	Definierar kriterierna för jämförelse. Alternativ: • 0 — Exakt matchning: inkluderar geometriernas ordning och antal toppar • 1 — Tolerant matchning (topologisk jämlikhet): geometrierna anses vara lika
Oförändrade funktioner Valfritt	`UNCHANGED`	[vektor: samma som Originallagret]	Ange utmatningsvektorlagret som innehåller de oförändrade funktionerna. *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

continues on next page

Table 24.137 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Tillagda funktioner Valfritt	ADDED	[vektor: samma som Originallagret]	Ange utmatningsvektorlagret som innehåller de extra funktionerna. *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (TEMPORARY_OUTPUT) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Borttagna funktioner Valfritt	DELETED	[vektor: samma som Originallagret]	Ange det utdatavektorlager som innehåller de borttagna funktionerna. *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (TEMPORARY_OUTPUT) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Oförändrade funktioner	UNCHANGED	[vektor: samma som Originallagret]	Vektorlager som innehåller de oförändrade funktionerna.
Tillagda funktioner	ADDED	[vektor: samma som Originallagret]	Vektorlager som innehåller de tillagda funktionerna.
Borttagna funktioner	DELETED	[vektor: samma som Originallagret]	Vektorlager som innehåller de borttagna objekten.
Antal oförändrade funktioner	UNCHANGED_COUN	[numerisk: heltal]	Antal oförändrade funktioner.
Antal funktioner som lagts till i det reviderade lagret	ADDED_COUNT	[numerisk: heltal]	Antal funktioner som lagts till i det reviderade lagret.
Antal funktioner som tagits bort från det ursprungliga lagret	DELETED_COUNT	[numerisk: heltal]	Antal funktioner som tagits bort från det ursprungliga lagret.

Python-kod

Algoritm-ID: `native:upptäcka vektorförändringar`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Geometrier för droppar

Skapar en enkel *geometrilös* kopia av attributtabellen för inmatningslagret. Den behåller attributtabellen för källagret.

Om filen sparas i en lokal mapp kan du välja mellan många olika filformat.

Tillåter *features in-place modification* av punkt-, linje- och polygonfeatures

Se även

Ta bort duplicerade geometrier, *Ta bort nollgeometrier*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlagret
Droppade geometrier	`OUTPUT`	[vektor: tabell]	Ange det geometrilösa utdatalagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Droppade geometrier	`OUTPUT`	[vektor: tabell]	Det geometrilösa utdatalagret. En kopia av den ursprungliga attributtabellen.

Python-kod

Algoritm-ID: `native:dropgeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exekvera SQL

Kör en enkel eller komplex fråga som endast baseras på SELECT med `SQL`-syntax på källagret.

Indatakällor identifieras med `input1`, `input2` ... `inputN` och en enkel fråga kommer att se ut som `SELECT * FROM input1`.

Förutom en enkel fråga kan du lägga till uttryck eller variabler i själva parametern `SQL-fråga`. Detta är särskilt användbart om denna algoritm körs i en bearbetningsmodell och du vill använda en modellinmatning som en parameter i frågan. Ett exempel på en fråga kan då vara `SELECT * FROM [% @table %]` där `@table` är den variabel som identifierar modellinmatningen.

Resultatet av sökningen läggs till som ett nytt lager.

Se även

SpatiaLite kör SQL, *PostgreSQL kör SQL*

Parametrar

Etikett	Namn	Typ	Beskrivning
Ytterligare datakällor för indata (kallas input1, …, inputN i frågan)	`INPUT_DATASOUR`	[vektor: valfri] [lista]	Lista över lager att fråga. I SQL-editorn kan du referera till dessa lager med deras **verkliga** namn eller också med **input1**, **input2**, **inputN** beroende på hur många lager som har valts.
SQL-fråga	`INPUT_QUERY`	[sträng]	Skriv in strängen för din SQL-fråga, t.ex. `SELECT * FROM input1`.
fält för unik identifierare Valfritt	`INPUT_UID_FIEL`	[sträng]	Ange kolumnen med unikt ID
Geometriområde Valfritt	`INPUT_GEOMETRY`	[sträng]	Ange geometrifältet
Typ av geometri Valfritt	`INPUT_GEOMETRY`	[uppräkning] Standard: 0	Välj geometri för resultatet. Som standard kommer algoritmen att autodetektera den. Ett av följande: • 0 — Automatisk detektering • 1 — Ingen geometri • 2 — Punkt • 3 — LinjeString • 4 — Polygon • 5 — MultiPoint • 6 — MultiLineString • 7 — MultiPolygon
CRS Valfritt	`INPUT_GEOMETRY`	[crs]	Den CRS som ska tilldelas det utgående lagret
SQL-utdata	`OUTPUT`	[vektor: geometri] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret som skapas av frågan. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
SQL-utdata	`OUTPUT`	[vektor: geometri]	Vektorlager som skapats av frågan

Python-kod

Algoritm ID: `qgis:executesql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exportera lager till DXF

Exporterar lager till DXF-fil. För varje lager kan du välja ett fält vars värden används för att dela upp funktioner i genererade destinationslager i DXF-utdata.

Se även

Skapa nya DXF-filer

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	LAYERS	[vektor: geometri] [lista]	Lista över inmatningsvektorlager med tillhörande alternativ (fylls i som ett QgsProcessingParameterDxfLayers-objekt — görs i GUI genom att trycka på *Configure Layer…*-knappen för varje valt lager): **Lager [sträng] (lager)** Fullständig sökväg till inmatningslagret som ska exporteras **Attribut för utdatalager [tablefield: any] (attributeIndex)** Attributindex för att dela upp inmatningslagret med unika värden **Namn på utmatningslager [sträng] (overriddenLayerName)** Åsidosatt lagernamn som ska användas i den exporterade DXF-filen (det ursprungliga projektlagret förblir orört) **Tillåt datadefinierade symbolblock** [boolean] (buildDataDefinedBlocks) **Maximalt antal symbolblock [numeric: integer] (dataDefinedBlocksMaximumNumberOfClasses)** -1 betyder ingen begränsning.
Symbologiläge	SYMBOLOGY_MODE	[uppräkning] Standard: 0	Typ av symbologi som ska tillämpas på utdatalagren. Du kan välja mellan: • 0 — Ingen symbologi • 1 — Symbologi för funktioner • 2 — Symbollager för symbologi
Symbologisk skala	SYMBOLOGY_SCAL	[skala] Standard: 1:1 000 000	Standardskala för dataexport.
Karttema Valfritt	MAP_THEME	[karttema]	Anpassa lagrets styling till det tillhandahållna karttemat.
Kodning	ENCODING	[uppräkning]	Kodning för att applicera på lager.
CRS	CRS	[crs]	Välj CRS för det utgående lagret.
Omfattning Valfritt	EXTENT	[omfattning]	Begränsa exporterade funktioner till dem med geometrier som skär den angivna utsträckningen.
Använd lagertitel som namn	USE_LAYER_TITL	[boolean] Standard: Falsk	Namnge utdatalagret med lagertiteln (som anges i lagrets metadata eller QGIS Server-egenskaper) istället för lagernamnet.
Force 2D	FORCE_2D	[boolean] Standard: Falsk	
Exportera etiketter som MTEXT-element	MTEXT	[boolean] Standard: Sann	Exporterar etiketter som MTEXT- eller TEXT-element
Använd endast utvalda funktioner	SELECTED_FEATU	[boolean] Standard: Falsk	Exporterar endast de valda funktionerna.
DXF	OUTPUT	[file] Standard: [Spara till temporär fil]	Specifikation av DXF-filen för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Utmatningar

Etikett	Namn	Typ	Beskrivning
DXF	`OUTPUT`	[file]	`.DXF`-fil som innehåller de ingående lagren

Python-kod

Algoritm-ID: `native:dxfexport`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Extrahera utvalda funktioner

Sparar de markerade objekten som ett nytt lager.

Observera

Om det valda lagret inte har några valda funktioner kommer det nyskapade lagret att vara tomt.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Lager för att spara valet från
Utvalda egenskaper	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange vektorlagret för de valda funktionerna. *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Utvalda egenskaper	`OUTPUT`	[samma som inmatning]	Vektorlager med endast de valda funktionerna, eller ingen funktion om ingen har valts.

Python-kod

Algoritm-ID: `native:spara utvalda funktioner`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Extrahera Shapefile-kodning

Extraherar information om attributkodning som är inbäddad i en Shapefile. Både den kodning som anges av en valfri `.cpg`-fil och eventuella kodningsdetaljer som finns i `.dbf` LDID-huvudblocket beaktas.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	ESRI Shapefile (`.SHP`) Layer för att extrahera kodningsinformationen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Shapefile-kodning	`ENCODING`	[sträng]	Kodningsinformation som anges i indatafilen
CPG-kodning	`CPG_ENCODING`	[sträng]	Kodningsinformation som anges i en valfri `.CPG`-fil
LDID-kodning	`LDID_ENCODING`	[sträng]	Kodningsinformation som anges i `.dbf` LDID-huvudblock

Python-kod

Algoritm-ID: `` `native:shpencodinginfo` ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Hitta projektion

Skapar en kortlista med kandidatkoordinatreferenssystem, t.ex. för ett lager med okänd projektion.

Det område som lagret förväntas täcka måste anges via parametern target area. Koordinatreferenssystemet för detta målområde måste vara känt för QGIS.

Algoritmen fungerar genom att testa lagrets utsträckning i alla kända referenssystem och sedan lista alla för vilka gränserna skulle vara nära målområdet om lagret var i denna projektion.

Se även

Tilldela projektion, *Definiera Shapefile-projektion*, *Återprojicera lagret*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Lager med okänd projektion
Målområde för lagret (xmin, xmax, ymin, ymax)	`TARGET_AREA`	[omfattning]	Det område som lagret täcker. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
CRS-kandidater	`OUTPUT`	[vektor: tabell] Standard: `[Skapa tillfälligt lager]`	Ange tabellen (geometrilöst lager) för CRS-förslagen (EPSG-koder). *One of*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
CRS-kandidater	`OUTPUT`	[vektor: tabell]	En tabell med alla CRS-koder (EPSG-koder) för de matchande kriterierna.

Python-kod

Algoritm-ID: `qgis:findprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Platt relation

Plattar ut ett *relationship* för ett vektorlager och exporterar ett enda lager som innehåller en överordnad funktion per relaterad underordnad funktion. Denna huvudfunktion innehåller alla attribut för de relaterade funktionerna. Detta gör det möjligt att ha relationen som en vanlig tabell som t.ex. kan exporteras till CSV.

> **Varning**
>
> Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

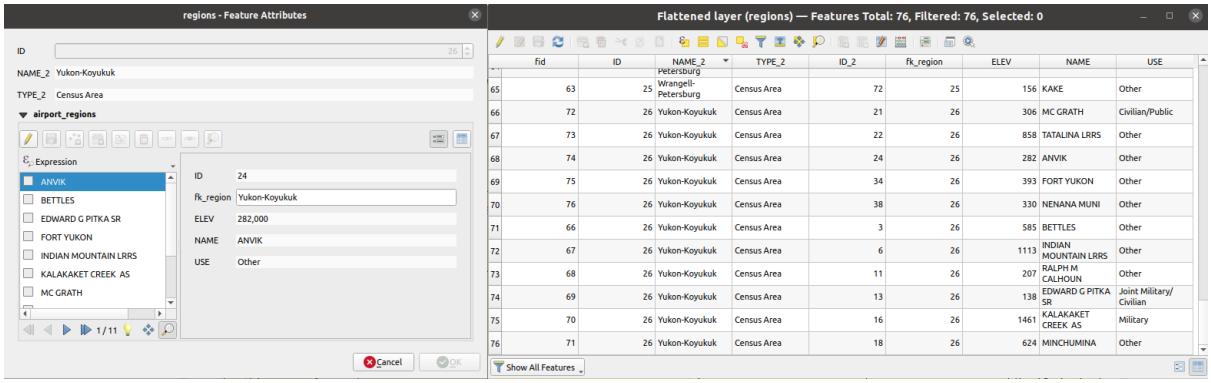


Fig. 24.55: Form av en region med relaterade barn (vänster) - En duplicerad regionfunktion för varje relaterat barn, med sammanfogade attribut (höger)

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Lager med relation som ska avnormaliseras
Utplattat lager Valfritt	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående (tillplattade) lagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Utplattat lager	`OUTPUT`	[samma som inmatning]	Ett lager som innehåller huvudfunktioner med alla attribut för de relaterade funktionerna

Python-kod

Algoritm-ID: `native:flattenrelationships`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Sammanfoga attribut efter fältvärde

Tar ett inmatat vektorlager och skapar ett nytt vektorlager som är en utökad version av det inmatade lagret, med ytterligare attribut i dess attributtabell.

De ytterligare attributen och deras värden hämtas från ett andra vektorlager. Ett attribut väljs i var och en av dem för att definiera anslutningskriterierna.

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Sammanfoga attribut efter närmaste, *Anslut attribut efter plats*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlager. Utmatningslagret kommer att bestå av funktionerna i detta lager med attribut från matchande funktioner i det andra lagret.
Tabellfält	`FIELD`	[tabellfält: alla]	Fält för det källager som ska användas för sammanfogningen
Inmatningslager 2	`INPUT_2`	[vektor: vilken som helst]	Lager med attributtabell för att ansluta
Tabellfält 2	`FIELD_2`	[tabellfält: alla]	Fält i det andra (join) lagret som ska användas för join Fältets typ måste vara lika med (eller kompatibel med) indatatabellens fälttyp.
Lager 2 fält att kopiera Valfritt	`FIELDS_TO_COPY`	[tabellfält: valfritt] [lista]	Välj de specifika fält som du vill lägga till. Som standard läggs alla fält till.
Join typ	`METHOD`	[uppräkning] Standard: 1	Typen av det sista sammanfogade lagret. Ett av följande: • 0 — Skapa en separat funktion för varje matchande funktion (en-till-många) • 1 — Ta endast attribut för den första matchande funktionen (en-till-en)
Avvisa poster som inte kunde kopplas samman	`DISCARD_NONMAT`	[boolean] Standard: Sann	Kontrollera om du inte vill behålla de funktioner som inte kunde anslutas
Prefix för sammanfogat fält Valfritt	`PREFIX`	[sträng]	Lägg till ett prefix för gemensamma fält för att enkelt kunna identifiera dem och undvika kollisioner mellan fältnamn
Sammanlagt lager Valfritt	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange utmatningsvektorlagret för sammanfogningen. *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Oöverträffade funktioner från första lagret Valfritt	`NON_MATCHING`	[samma som inmatning] Standard: `[Skip output]`	Ange utdatavektorlagret för funktioner som inte kan sammanföras från det första lagret. *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Antal sammanfogade funktioner från indatatabellen	`JOINED_COUNT`	[numerisk: heltal]	
Oöverträffade funktioner från första lagret Valfritt	`NON_MATCHING`	[samma som inmatning]	Vektorlager med de icke-matchade funktionerna
Sammanlagt lager Valfritt	`OUTPUT`	[samma som inmatning]	Utgående vektorlager med tillagda attribut från sammanfogningen
Antal funktioner som inte kan sammanföras från indatatabellen Valfritt	`UNJOINABLE_COU`	[numerisk: heltal]	

Python-kod

Algoritm-ID: `` `native:joinattributestable ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Anslut attribut efter plats

Tar ett inmatat vektorlager och skapar ett nytt vektorlager som är en utökad version av det inmatade lagret, med ytterligare attribut i dess attributtabell.

De ytterligare attributen och deras värden hämtas från ett andra vektorlager. Ett spatialt kriterium tillämpas för att välja de värden från det andra lagret som läggs till varje egenskap från det första lagret.

Standardmeny: *Vektor* ► *Verktyg för datahantering*

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Sammanfoga attribut efter närmaste, *Sammanfoga attribut efter fältvärde*, *Join-attribut per plats (sammanfattning)*

Utforska spatiala relationer

Geometriska predikat är booleska funktioner som används för att bestämma den spatiala relationen mellan en egenskap och en annan genom att jämföra om och hur deras geometrier delar en del av utrymmet.

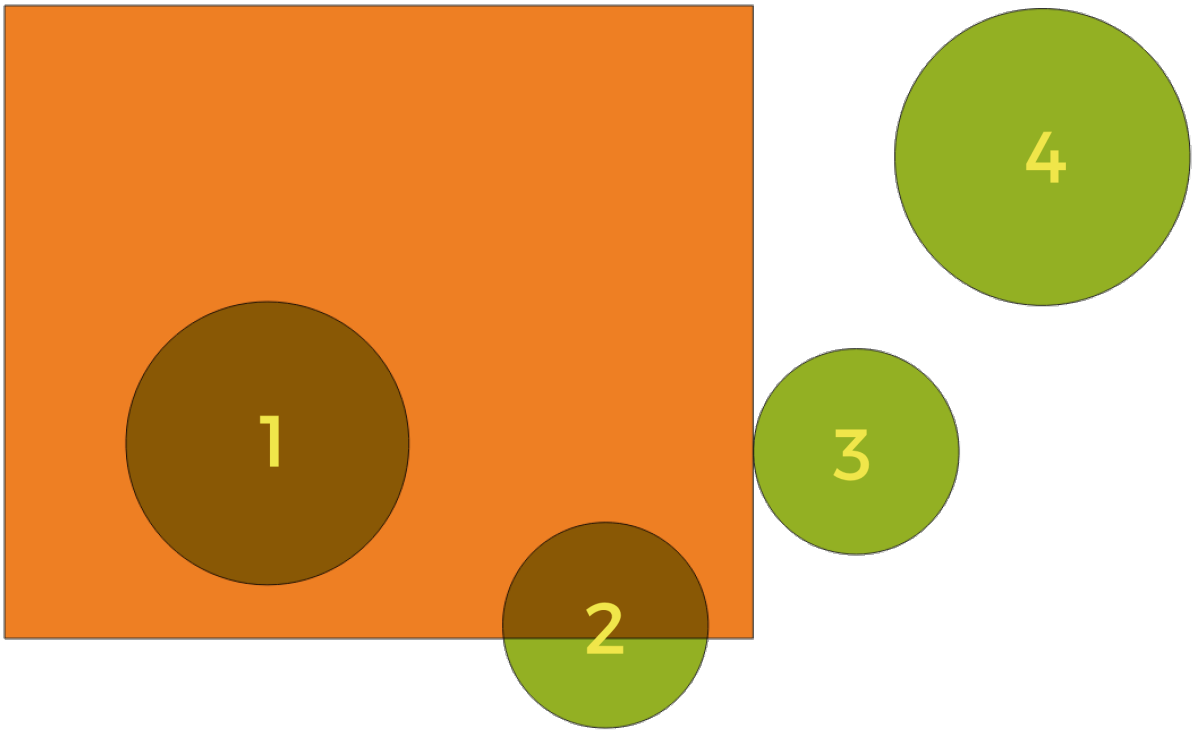


Fig. 24.56: Letar efter spatiala relationer mellan lager

I figuren ovan letar vi efter de gröna cirklarna genom att spatialt jämföra dem med den orangefärgade rektangeln. Tillgängliga geometriska predikat är:

Korsningspunkt
: Testar om en geometri korsar en annan. Returnerar 1 (sant) om geometrierna korsar varandra spatialt (delar någon del av utrymmet - överlappar eller rör vid varandra) och 0 om de inte gör det. I bilden ovan kommer detta att returnera cirklarna 1, 2 och 3.

Innehåller
: Returnerar 1 (sant) om och endast om inga punkter i b ligger i utsidan av a, och minst en punkt i insidan av b ligger i insidan av a. I bilden returneras ingen cirkel, men rektangeln skulle göra det om man letade efter den tvärtom, eftersom den innehåller cirkel 1 helt och hållet. Detta är motsatsen till *är inom*.

Disjunkt
: Returnerar 1 (sant) om geometrierna inte delar någon del av utrymmet (ingen överlappning, ingen beröring). Endast cirkel 4 returneras.

Equal
: Returnerar 1 (true) om och endast om geometrierna är exakt likadana. Inga cirklar kommer att returneras.

Touch
: Testar om en geometri berör en annan. Returnerar 1 (true) om geometrierna har minst en gemensam punkt, men deras inre delar inte korsar varandra. Endast cirkel 3 returneras.

Överlappning
: Testar om en geometri överlappar en annan. Returnerar 1 (true) om geometrierna delar utrymme, har samma dimension, men inte är helt inneslutna av varandra. Endast cirkel 2 returneras.

Är inom

Testar om en geometri är inuti en annan. Returnerar 1 (true) om geometri a är helt inuti geometri b. Endast cirkel 1 returneras.

Kors

Returnerar 1 (true) om de angivna geometrierna har vissa, men inte alla, inre punkter gemensamt och den faktiska korsningen har en lägre dimension än den högsta angivna geometrin. Till exempel kommer en linje som korsar en polygon att korsas som en linje (true). Två linjer som korsar varandra kommer att korsas som en punkt (sant). Två polygoner korsar varandra som en polygon (falsk). I bilden kommer inga cirklar att returneras.

Parametrar

Etikett	Namn	Typ	Beskrivning
Join to features in	`INPUT`	[vektor: geometri]	Ingångsvektorlager. Utmatningslagret kommer att bestå av funktionerna i detta lager med attribut från matchande funktioner i det andra lagret.
Var är funktionerna	`PREDICATE`	[uppräkning] [lista] Standard: [0]	Typ av spatial relation som källobjektet bör ha med målobjektet för att de ska kunna sammanfogas. En eller flera av: • 0 — korsa • 1 — innehåller • 2 — lika • 3 — tryck på • 4 — överlappning • 5 — är inom • 6 — kors Om mer än ett villkor väljs måste minst ett av dem (OR-operationen) uppfyllas för att en funktion ska kunna extraheras.
Genom att jämföra med	`JOIN`	[vektor: geometri]	Det förenade lagret. Funktioner i detta vektorlager kommer att **lägga till** sina attribut i källagrets attributtabell om de uppfyller det spatiala förhållandet.
Fält att lägga till (lämna tomt för att använda alla fält) Valfritt	`JOIN_FIELDS`	[tabellfält: valfritt] [lista]	Välj de specifika fält som du vill lägga till från länklagret. Som standard läggs alla fält till.
Join typ	`METHOD`	[uppräkning]	Typen av det sista sammanfogade lagret. Ett av följande: • 0 — Skapa en separat funktion för varje matchande funktion (en-till-många) • 1 — Ta endast attribut för den första matchande funktionen (en-till-en) • 2 — Ta endast attribut för funktionen med störst överlappning (en-till-en)
Avvisa poster som inte kunde kopplas samman	`DISCARD_NONMAT`	[boolean] Standard: Falsk	Ta bort de funktioner från inmatningslagret som inte kunde kopplas samman från utdata
Prefix för sammanfogat fält Valfritt	`PREFIX`	[sträng]	Lägg till ett prefix för gemensamma fält för att enkelt kunna identifiera dem och undvika kollisioner mellan fältnamn

continues on next page

Table 24.144 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Sammanlagt lager Valfritt	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange utmatningsvektorlagret för sammanfogningen. *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Oöverträffade funktioner från första lagret Valfritt	`NON_MATCHING`	[samma som inmatning] Standard: `[Skip output]`	Ange utdatavektorlagret för funktioner som inte kan sammanföras från det första lagret. *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Antal sammanfogade funktioner från indatatabellen	`JOINED_COUNT`	[numerisk: heltal]	
Oöverträffade funktioner från första lagret Valfritt	`NON_MATCHING`	[samma som inmatning]	Vektorlager av de icke-matchade funktionerna
Sammanlagt lager	`OUTPUT`	[samma som inmatning]	Utgående vektorlager med tillagda attribut från sammanfogningen

Python-kod

Algoritm-ID: `` `native:joinattributesbylocation ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Join-attribut per plats (sammanfattning)

Tar ett inmatat vektorlager och skapar ett nytt vektorlager som är en utökad version av det inmatade lagret, med ytterligare attribut i dess attributtabell.

De ytterligare attributen och deras värden hämtas från ett andra vektorlager. Ett spatialt kriterium tillämpas för att välja de värden från det andra lagret som läggs till varje egenskap från det första lagret.

Algoritmen beräknar en statistisk sammanfattning för värdena från matchande funktioner i det andra lagret (t.ex. maxvärde, medelvärde etc.).

Se även

Anslut attribut efter plats

Utforska spatiala relationer

Geometriska predikat är booleska funktioner som används för att bestämma den spatiala relationen mellan en egenskap och en annan genom att jämföra om och hur deras geometrier delar en del av utrymmet.

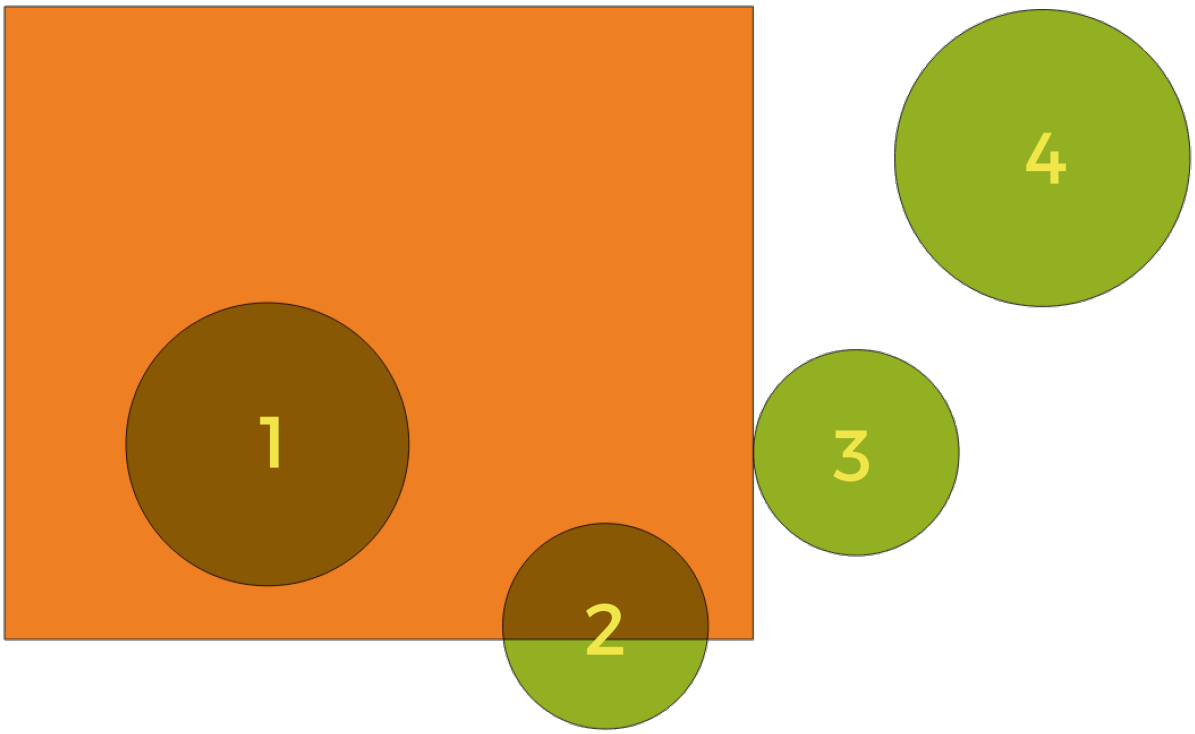


Fig. 24.57: Letar efter spatiala relationer mellan lager

I figuren ovan letar vi efter de gröna cirklarna genom att spatialt jämföra dem med den orangefärgade rektangeln. Tillgängliga geometriska predikat är:

Korsningspunkt
: Testar om en geometri korsar en annan. Returnerar 1 (sant) om geometrierna korsar varandra spatialt (delar någon del av utrymmet - överlappar eller rör vid varandra) och 0 om de inte gör det. I bilden ovan kommer detta att returnera cirklarna 1, 2 och 3.

Innehåller
: Returnerar 1 (sant) om och endast om inga punkter i b ligger i utsidan av a, och minst en punkt i insidan av b

ligger i insidan av a. I bilden returneras ingen cirkel, men rektangeln skulle göra det om man letade efter den tvärtom, eftersom den innehåller cirkel 1 helt och hållet. Detta är motsatsen till *är inom*.

Disjunkt
Returnerar 1 (sant) om geometrierna inte delar någon del av utrymmet (ingen överlappning, ingen beröring). Endast cirkel 4 returneras.

Equal
Returnerar 1 (true) om och endast om geometrierna är exakt likadana. Inga cirklar kommer att returneras.

Touch
Testar om en geometri berör en annan. Returnerar 1 (true) om geometrierna har minst en gemensam punkt, men deras inre delar inte korsar varandra. Endast cirkel 3 returneras.

Överlappning
Testar om en geometri överlappar en annan. Returnerar 1 (true) om geometrierna delar utrymme, har samma dimension, men inte är helt inneslutna av varandra. Endast cirkel 2 returneras.

Är inom
Testar om en geometri är inuti en annan. Returnerar 1 (true) om geometri a är helt inuti geometri b. Endast cirkel 1 returneras.

Kors
Returnerar 1 (true) om de angivna geometrierna har vissa, men inte alla, inre punkter gemensamt och den faktiska korsningen har en lägre dimension än den högsta angivna geometrin. Till exempel kommer en linje som korsar en polygon att korsas som en linje (true). Två linjer som korsar varandra kommer att korsas som en punkt (sant). Två polygoner korsar varandra som en polygon (falsk). I bilden kommer inga cirklar att returneras.

Parametrar

Etikett	Namn	Typ	Beskrivning
Join to features in	`INPUT`	[vektor: geometri]	Ingångsvektorlager. Utmatningslagret kommer att bestå av funktionerna i detta lager med attribut från matchande funktioner i det andra lagret.
Var är funktionerna	`PREDICATE`	[uppräkning] [lista] Standard: [0]	Typ av spatial relation som källobjektet bör ha med målobjektet för att de ska kunna sammanfogas. En eller flera av: • 0 — korsa • 1 — innehåller • 2 — lika • 3 — tryck på • 4 — överlappning • 5 — är inom • 6 — kors Om mer än ett villkor väljs måste minst ett av dem (OR-operationen) uppfyllas för att en funktion ska kunna extraheras.
Genom att jämföra med	`JOIN`	[vektor: geometri]	Det förenade lagret. Funktioner i detta vektorlager kommer att **lägga till** sammanfattningar av sina attribut till källagrets attributtabell om de uppfyller det spatiala förhållandet.

continues on next page

Table 24.145 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Fält att sammanfatta (lämna tomt för att använda alla fält) Valfritt	JOIN_FIELDS	[tabellfält: valfritt] [lista]	Välj de specifika fält som du vill lägga till från länklagret. Som standard läggs alla fält till.
Summarier att beräkna (lämna tomt för att använda alla fält) Valfritt	SUMMARIES	[uppräkning] [lista] Standard: []	För varje indatafunktion beräknas statistik på sammanfogade fält i deras matchande funktioner. En eller flera av: • 0 — antal • 1 — unik • 2 — min • 3 — max • 4 — intervall • 5 — summa • 6 — medelvärde • 7 — median • 8 — stddev • 9 — minoritet • 10 — majoritet • 11 — q1 • 12 — q3 • 13 — iqr • 14 — tom • 15 — fylld • 16 — min_längd • 17 — max_längd • 18 — medelvärdeslängd
Avvisa poster som inte kunde kopplas samman	DISCARD_NONMAT	[boolean] Standard: Falsk	Ta bort de funktioner från inmatningslagret som inte kunde kopplas samman från utdata
Sammanlagt lager	OUTPUT	[samma som inmatning] Standard: [Skapa tillfälligt lager]	Ange utmatningsvektorlagret för sammanfogningen. *En av*: • Skapa ett tillfälligt lager (TEMPORARY_OUTPUT) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Sammanlagt lager	OUTPUT	[samma som inmatning]	Vektorlager för utdata med sammanfattade attribut från sammanfogningen

Python-kod

Algoritm ID: `qgis:joinbylocationsummary`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Sammanfoga attribut efter närmaste

Tar ett inmatat vektorlager och skapar ett nytt vektorlager med ytterligare fält i dess attributtabell. De ytterligare attributen och deras värden hämtas från ett andra vektorlager. Funktioner sammanfogas genom att hitta de närmaste funktionerna från varje lager.

Som standard ansluts endast den närmaste funktionen, men anslutningen kan också ansluta till de k närmaste grannfunktionerna.

Om ett maximalt avstånd anges kommer endast objekt som är närmare än detta avstånd att matchas.

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Analys av närmaste granne, *Sammanfoga attribut efter fältvärde*, *Anslut attribut efter plats*, *Avståndsmatris*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Det ingående lagret.
Inmatningslager 2	`INPUT_2`	[vektor: geometri]	Det sammanfogande lagret.
Lager 2-fält att kopiera (lämna tomt för att kopiera alla fält)	`FIELDS_TO_COPY`	[fält]	Fält i Join-lagret som ska kopieras (om de är tomma kopieras alla fält).
Avvisa poster som inte kunde kopplas samman	`DISCARD_NONMAT`	[boolean] Standard: Falsk	Ta bort de poster i inmatningslagret som inte kunde sammanfogas från utdata
Prefix för sammanfogat fält	`PREFIX`	[sträng]	Prefix för sammanfogat fält
Maximalt antal närmaste grannar	`NEIGHBORS`	[numerisk: heltal] Standard: 1	Maximalt antal närmaste grannar
Maximalt avstånd	`MAX_DISTANCE`	[numerisk: dubbel]	Maximalt sökavstånd

continues on next page

Table 24.146 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Sammanlagt lager Valfritt	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det vektorlager som innehåller de sammanfogade funktionerna. *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Oöverträffade funktioner från första lagret	`NON_MATCHING`	[samma som inmatning] Standard: `[Skip output]`	Ange det vektorlager som innehåller de funktioner som inte kunde sammanfogas. *One of*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Sammanlagt lager	`OUTPUT`	[samma som inmatning]	Det utgående sammanfogade lagret.
Oöverträffade funktioner från första lagret	`NON_MATCHING`	[samma som inmatning]	Lager som innehåller de funktioner från det första lagret som inte kunde kopplas till några funktioner i det kopplade lagret.
Antal sammanfogade funktioner från indatatabellen	`JOINED_COUNT`	[numerisk: heltal]	Antal funktioner från inmatningstabellen som har kopplats samman.
Antal funktioner som inte kan sammanföras från indatatabellen	`UNJOINABLE_COU`	[numerisk: heltal]	Antal funktioner från indatatabellen som inte kunde länkas.

Python-kod

Algoritm-ID: `native:joinbynearest`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

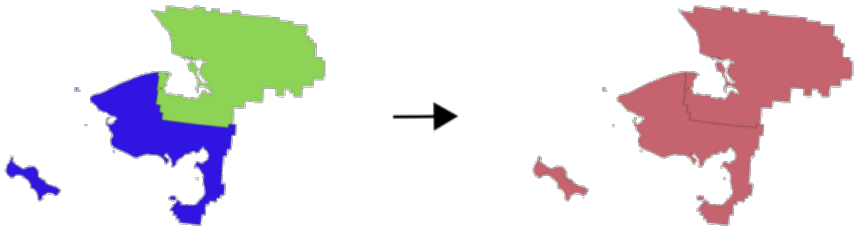
Sammanfoga vektorlager

Kombinerar flera vektorlager av **samme geometrityp** till ett enda.

Attributtabellen för det resulterande lagret kommer att innehålla fälten från alla ingående lager. Om fält med samma namn men olika typer hittas konverteras det exporterade fältet automatiskt till ett fält av strängtyp. Nya fält som lagrar det ursprungliga lagrets namn och källa läggs också till.

Om något av inmatningslagren innehåller Z- eller M-värden, kommer även utmatningslagret att innehålla dessa värden. På samma sätt, om något av inmatningslagren är flerdelat, kommer utmatningslagret också att vara ett flerdelat lager.

Alternativt kan det koordinatreferenssystem (CRS) som ska användas för det sammanslagna lagret ställas in. Om det inte anges kommer CRS att hämtas från det första inmatningslagret. Alla lager kommer att reprojiceras för att matcha detta CRS.



Standardmeny: *Vektor ► Verktyg för datahantering*

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Delat vektorlager

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`LAYERS`	[vektor: valfri] [lista]	De lager som ska sammanfogas till ett enda lager. Lagren bör vara av samma geometrityp.
Destination CRS Valfritt	`CRS`	[crs]	Välj CRS för utmatningslagret. Om inget anges används CRS för det första inmatningslagret.
Sammanslagna	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Sammanslagna	`OUTPUT`	[samma som inmatning]	Utgående vektorlager som innehåller alla funktioner och attribut från de ingående lagren.

Python-kod

Algoritm-ID: `native:mergevectorlayers`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Ordning efter uttryck

Sorterar ett vektorlager enligt ett uttryck: ändrar funktionsindexet enligt ett uttryck.

Var försiktig, det kanske inte fungerar som förväntat med vissa leverantörer, beställningen kanske inte hålls varje gång.

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlager att sortera
Uttryck	`EXPRESSION`	[uttryck]	Uttryck som ska användas för sorteringen
Sortera i stigande ordning	`ASCENDING`	[boolean] Standard: Sann	Om markerad kommer vektorlagret att sorteras från små till stora värden.
Sortera nollor först	`NULLS_FIRST`	[boolean] Standard: Falsk	Om markerad, placeras nollvärden först
Orderad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Orderad	`OUTPUT`	[samma som inmatning]	Utgång (sorterad) vektorlager

Python-kod

Algoritm-ID: `native:orderbyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Reparera Shapefile

Reparerar en trasig ESRI Shapefile-datauppsättning genom att (åter)skapa SHX-filen.

Parametrar

Etikett	Namn	Typ	Beskrivning
Input Shapefile (Shapefil)	`INPUT`	[file]	Fullständig sökväg till ESRI Shapefile-datasetet med en saknad eller trasig SHX-fil

Utmatningar

Etikett	Namn	Typ	Beskrivning
Reparerat lager	`OUTPUT`	[vektor: geometri]	Ingångsvektorlagret med SHX-filen reparerad

Python-kod

Algoritm-id: `native:repairshapefile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Återprojicera lagret

Återprojicerar ett vektorlager i ett annat CRS. Det återprojicerade lagret har samma funktioner och attribut som inmatningslagret.

Tillåter *features in-place modification* av punkt-, linje- och polygonfeatures

Se även

Tilldela projektion, *Definiera Shapefile-projektion*, *Hitta projektion*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager för att återprojicera
Mål CRS	`TARGET_CRS`	[crs] Standard: `EPSG:4326 - WGS 84`	Referenssystem för destinationskoordinater
Konvertera krökta geometrier till raka segment	`CONVERT_CURVED`	[boolean] Standard: Falsk	Om detta är markerat kommer krökta geometrier att konverteras till raka segment i processen, vilket undviker potentiella distorsionsproblem.
Omprojekterad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Koordinatdrift Valfritt	`OPERATION`	[sträng]	Specifik operation som ska användas för en viss reprojiceringsuppgift, istället för att alltid tvinga fram användning av det aktuella projektets transformationsinställningar. Användbart när ett visst lager ska reprojiceras och kontroll över den exakta transformationspipelinen krävs. Kräver proj-version >= 6. Läs mer på *Datumtransformationer*.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Omprojekterad	`OUTPUT`	[samma som inmatning]	Utgående (reprojicerat) vektorlager

Python-kod

Algoritm-ID: `native:reprojectlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Spara vektorfunktioner till fil

Sparar vektorfunktioner till en angiven fildataset.

För datasetformat som stöder lager kan en valfri parameter för lagernamn användas för att ange en anpassad sträng. Valfria GDAL-definierade alternativ för dataset och lager kan anges. Mer information om detta finns i online-dokumentationen för GDAL <https://gdal.org/en/latest/drivers/vector/index.html>`_ om formatet.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Vektorfunktioner	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlager.
Sparade funktioner	`OUTPUT`	[samma som inmatning] Standard: `[Spara till temporär fil]`	Ange den fil som funktionerna ska sparas i. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Lagernamn Valfritt	LAYER_NAME	[sträng]	Namn som ska användas för utdatalagret
Alternativ för GDAL-dataset Valfritt	DATASOURCE_OPT	[sträng]	Alternativ för skapande av GDAL-dataset för utdataformatet. Separera enskilda alternativ med semikolon.
GDAL-lageralternativ Valfritt	LAYER_OPTIONS	[sträng]	Alternativ för skapande av GDAL-lager i utdataformatet. Separera enskilda alternativ med semikolon.
Åtgärd att vidta i redan befintlig fil	ACTION_ON_EXIS	[uppräkning] Standard: 0	Hur man hanterar befintliga funktioner. Giltiga metoder är: • 0 — Skapa eller skriva över fil • 1 — Skapa eller skriva över lager • 2 — Lägg till funktioner i ett befintligt lager, men skapa inte nya fält • 3 — Lägg till funktioner i befintliga lager och skapa nya fält om det behövs

Utmatningar

Etikett	Namn	Typ	Beskrivning
Sparade funktioner	OUTPUT	[samma som inmatning]	Vektorlager med de sparade funktionerna.
Filnamn och sökväg	FILE_PATH	[sträng]	Namn och sökväg för utdatafilen.
Lagernamn	LAYER_NAME	[sträng]	Namn på lagret, om det finns något.

Python-kod

Algoritm-ID: `native:spara funktioner`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Ställ in lagerkodning

Ställer in den kodning som används för att läsa ett lagers attribut. Inga permanenta ändringar görs i lagret, utan det påverkar endast hur lagret läses under den aktuella sessionen.

Observera

Ändring av kodning stöds endast för vissa datakällor för vektorlager.

Parametrar

Etikett	Namn	Typ	Beskrivning
Sparade funktioner	`INPUT`	[vektor: geometri]	Vektorlager för att ställa in kodningen.
Kodning	`ENCODING`	[sträng]	Textkodning som ska tilldelas lagret i den aktuella QGIS-sessionen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Denna algoritm beräknar den totala stigningen och nedstigningen längs linjegeometrier.	`OUTPUT`	[samma som inmatning]	Ingångsvektorlager med den inställda kodningen.

Python-kod

Algoritm-ID: `` `native:setlayerencoding ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Dela upp funktioner efter tecken

Funktioner delas upp i flera utdatafunktioner genom att dela upp ett fältvärde vid ett angivet tecken. Om ett lager t.ex. innehåller funktioner med flera kommaseparerade värden i ett enda fält kan denna algoritm användas för att dela upp dessa värden i flera utdatafunktioner. Geometrier och andra attribut förblir oförändrade i utdata. Eventuellt kan separatorsträngen vara ett reguljärt uttryck för ökad flexibilitet.

Tillåter *features in-place modification* av punkt-, linje- och polygonfeatures

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlager
Split med hjälp av värden i fältet	`FIELD`	[tabellfält: alla]	Fält att använda för delning
Dela värde med tecken	`CHAR`	[sträng]	Karaktär att använda för klyvning
Använd separator för reguljära uttryck	`REGEX`	[boolean] Standard: Falsk	
Split	`OUTPUT`	[samma som inmatning] Standard: `Skapa tillfälligt lager`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Split	`OUTPUT`	[samma som inmatning]	Utgångsvektorlagret.

Python-kod

Algoritm-ID: `native:splitfeaturesbycharacter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Delat vektorlager

Skapar en uppsättning vektorer i en utdatamapp baserat på ett inmatningslager och ett attribut. Utdatamappen kommer att innehålla lika många lager som de unika värden som finns i önskat fält.

Antalet filer som genereras är lika med antalet olika värden som hittas för det angivna attributet.

Det är den motsatta funktionen till *merging*.

Standardmeny: *Vektor* ► *Verktyg för datahantering*

Se även

Sammanfoga vektorlager

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlager
Unikt ID-fält	`FIELD`	[tabellfält: alla]	Fält att använda för delning
Utmatningskatalog	`OUTPUT`	[mapp] Standard: `[Spara till tillfällig mapp]`	Ange katalogen för utdatalagren. *En av*: • Spara till en temporär katalog • Spara till katalog

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Typ av utdatafil Valfritt	`FILE_TYPE`	[uppräkning] Standard: `gpkg` i dialogfönstret	Välj filnamnstillägg för utdatafilerna. Om det inte anges eller om det är ogiltigt kommer utdatafilernas format att vara det som anges i bearbetningsinställningen ”Default output vector layer extension”.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Utmatningskatalog	`OUTPUT`	[mapp]	Katalogen för utdatalagren
Utgående lager	`OUTPUT_LAYERS`	[samma som inmatning] [lista]	De utgående vektorlagren som blir resultatet av delningen.

Python-kod

Algoritm-ID: `native:splitvectorlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Avkorta tabellen

Trunkerar ett lager genom att ta bort alla funktioner inom lagret.

> **Varning**
>
> Denna algoritm modifierar lagret på plats, och borttagna funktioner kan inte återställas!

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlager

Utmatningar

Etikett	Namn	Typ	Beskrivning
Trunkerat lager	`OUTPUT`	[mapp]	Inmatningslagret, alla funktioner raderade

Python-kod

Algoritm ID: `native:truncatetable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.23 Vektorgeometri

Lägg till geometriattribut

Beräknar geometriska egenskaper för objekten i ett vektorlager och inkluderar dem i utdatalagret.

Det genererar ett nytt vektorlager med samma innehåll som det ingående lagret, men med ytterligare attribut, som innehåller geometriska mätningar baserade på ett valt CRS.

De attribut som läggs till i tabellen beror på geometritypen och dimensionen för det ingående lagret:

- för **punkt** lager: X (`xcoord`), Y (`ycoord`), Z (`zcoord`) koordinater och/eller M-värde (`mvalue`)
- för **line**-lager: `längd` och, för geometrityperna LineString och CompoundCurve, egenskapen `sinuosity` och raksträcka (`straightdis`)
- för **polygon**-lager: ”omkrets” och ”yta

Observera

Den här algoritmen använder ellipsoidbaserade mätningar och respekterar de aktuella *ellipsoidinställningarna*.

Standardmeny: *Vektor* ► *Geometriverktyg*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Beräkna med hjälp av	`CALC_METHOD`	[uppräkning] Standard: 0	Beräkningsparametrar som ska användas för de geometriska egenskaperna. En av: • 0 — Lager CRS • 1 — Projekt CRS • 2 — Ellipsoidformad
Lagt till geom-information	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret (inmatningskopia med geometri). *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Lagt till geom-information	`OUTPUT`	[samma som inmatning]	Kopia av inmatningsvektorlagret med tillägg av geometriska fält

Python-kod

Algoritm-ID: `qgis:exportaddgeometrycolumns`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Affin transform

Tillämpar en affin transformation på lagergeometrierna. Affina transformationer kan inkludera translation, skalning och rotation. Åtgärderna utförs i följande ordning: skala, rotation och translation.

Z- och M-värden (om sådana finns) kan översättas och skalas.

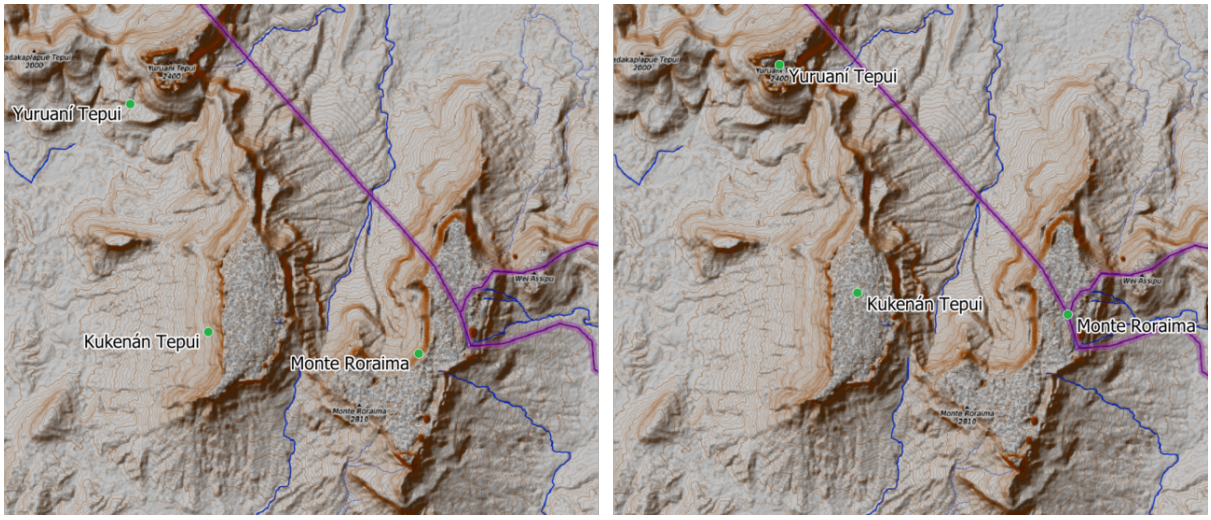


Fig. 24.58: Vektorpunktlager (gröna prickar) före (vänster) och efter (höger) en affin transformation (översättning).

Tillåter *features in-place modification* av punkt-, linje- och polygonfeatures

Se även

Översätt

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Översättning (x-axeln)	`DELTA_X`	[numerisk: dubbel] Standard: 0,0	Förskjutning som ska tillämpas på X-axeln.
Översättning (y-axeln)	`DELTA_Y`	[numerisk: dubbel] Standard: 0,0	Förskjutning som ska tillämpas på Y-axeln.
Translatering (z-axel)	`DELTA_Z`	[numerisk: dubbel] Standard: 0,0	Förskjutning som ska tillämpas på Z-axeln.
Översättning (m-värden)	`DELTA_M`	[numerisk: dubbel] Standard: 0,0	Offset som ska tillämpas på m-värden.
Skalfaktor (x-axel)	`SCALE_X`	[numerisk: dubbel] Standard: 1,0	Skalningsvärde (expansion eller kontraktion) som ska tillämpas på X-axeln.

continues on next page

Table 24.151 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Skalfaktor (y-axel)	`SCALE_Y`	[numerisk: dubbel] Standard: 1,0	Skalningsvärde (expansion eller kontraktion) som ska tillämpas på Y-axeln.
Skalfaktor (z-axel)	`SCALE_Z`	[numerisk: dubbel] Standard: 1,0	Skalningsvärde (expansion eller kontraktion) som ska tillämpas på Z-axeln.
Skalfaktor (m-värden)	`SCALE_M`	[numerisk: dubbel] Standard: 1,0	Skalningsvärde (expansion eller kontraktion) som ska tillämpas på m-värden.
Rotation runt z-axeln (grader moturs)	`ROTATION_Z`	[numerisk: dubbel] Standard: 0,0	Rotationens vinkel i grader.
Transformerad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Transformerad	`OUTPUT`	[samma som inmatning]	Utgående (transformerat) vektorlager.

Python-kod

Algoritm-ID: `native:affinetransform`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Sammanslagna

Tar ett vektor- eller tabellager och skapar ett nytt lager genom att aggregera funktioner baserat på ett `group by`-uttryck.

Funktioner för vilka `group by`-uttrycket returnerar samma värde grupperas tillsammans.

Det är möjligt att gruppera alla källfunktioner tillsammans med hjälp av ett konstant värde i parametern `group by`, t.ex: NULL.

Det är också möjligt att gruppera funktioner efter flera fält med hjälp av Array-funktionen, t.ex: Array(”Fält1”, ”Fält2”).

Geometrier (om sådana finns) kombineras till en flerdelad geometri för varje grupp. Utdataattribut beräknas beroende på varje given aggregatdefinition.

Den här algoritmen gör det möjligt att använda standard *aggregatsfunktioner* i QGIS Expression-motorn.

Observera

Den här algoritmen använder ellipsoidbaserade mätningar och respekterar de aktuella *ellipsoidinställningarna*.

Se även

Samla in geometrier, *Lös upp*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlager
Gruppera efter uttryck	`GROUP_BY`	[tabellfält: alla] Standard: 'NULL'	Välj grupperingsfält. Om *NULL* kommer alla funktioner att grupperas.

continues on next page

Table 24.153 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Aggregat	AGGREGATES	Konfigurationens id %1 finns redan i listan	Lista över fältdefinitioner för utdatalagret. Exempel på en fältdefinition: *{'aggregate': 'sum', 'delimiter': ',', 'input': ' $area', 'length': 10, 'name': 'totarea', 'precision': 0, 'typ': 6}* Som standard innehåller listan alla fält i inmatningslagret. I GUI kan du redigera dessa fält och deras definitioner, och du kan också: • Klicka på knappen för att lägga till ett nytt fält. • Klicka på för att ta bort det markerade fältet. • Använd och för att ändra ordningen på fälten. • Klicka på för att återställa till standardinställningen (fälten i inmatningslagret). För vart och ett av de fält som du vill hämta information från måste du definiera följande: ***Input uttryck* (`input`) [uttryck]** Fält eller uttryck från inmatningslagret. ***Aggregeringsfunktion* (`aggregate`) [uppräkning]** Funktion som ska användas på indatauttrycket för att returnera det aggregerade värdet. Dessa är huvudsakligen *Aggregeringsfunktioner*, plus dedikerade ”first_value”- och ”last_value”-funktioner för ordnade listobjekt. Standardinställning: *concatenate* (för strängdatatyp), *sum* (för numerisk datatyp) ***Delimiter* (`avgränsare`) [sträng]** Textsträng för att separera aggregerade värden, t.ex. vid konkatenering. Standard: , ***Namn på utdatafältet* (`namn`) [sträng]** Namnet på det aggregerade fältet i utdatalagret. Som standard behålls namnet på inmatningsfältet. ***Type* (`typ`) [uppräkning]** Datatyp för utdatafältet. Tillgängliga typer kanske inte är kompatibla med leverantören av utdatalagret. En av följande: **Uppmärksamma** För vissa fälttyper, t.ex. listor, hjälper en extra parameter `sub_type` till att förfina den specifika typen av data. Den ställs in automatiskt i det grafiska gränssnittet men kan behövas om du kör algoritmen i Python eller

Table 24.153 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Ladda fält från lagret	Endast GUI	[vektor: vilken som helst]	Du kan ladda fält från ett annat lager och använda dem för aggregeringen
Aggregerad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående (aggregerade) lagret *One of*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Aggregerad	`OUTPUT`	[samma som inmatning]	Multigeometriskt vektorlager med de aggregerade värdena

Python-kod

Algoritm-ID: `` `native:aggregat` ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Begränsning

Returnerar stängningen av den kombinatoriska gränsen för indatageometrierna (dvs. geometrins topologiska gräns).

Endast för polygon- och linjelager.

För **polygongeometrier** består gränsen av alla de linjer som utgör ringarna i polygonen.

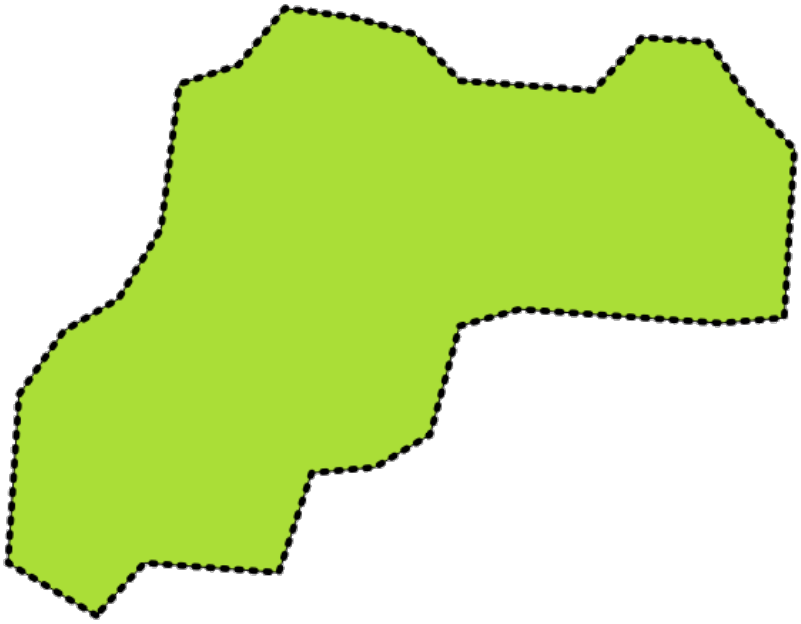


Fig. 24.59: Gränser (svart streckad linje) för källans polygonlager

För **linjegeometrier** är gränserna deras ändpunkter.

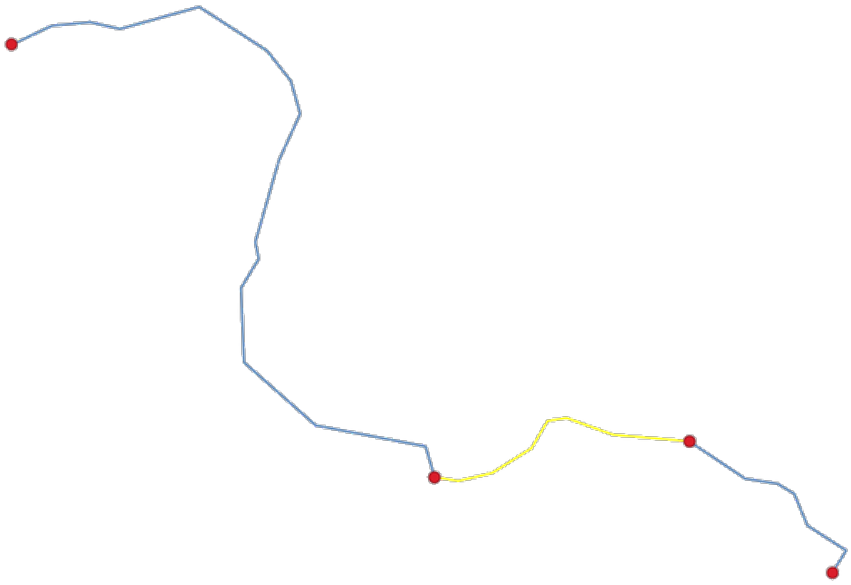


Fig. 24.60: Gränsskikt (röda punkter) för linjer. I gult en vald funktion.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje, polygon]	Vektorlager för linje- eller polygoninmatning
Begränsning	`OUTPUT`	[vektor: punkt, linje] Standard: `[Skapa tillfälligt lager]`	Ange det utgående (gräns)lagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Begränsning	`OUTPUT`	[vektor: punkt, linje]	Gränser från inmatningslagret (punkt för linje och linje för polygon)

Python-kod

Algoritm-ID: `native:boundary`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Begränsande boxar

Beräknar begränsningsrutan (envelope) för varje objekt i ett inmatningslager. Polygon- och linjegeometrier stöds.

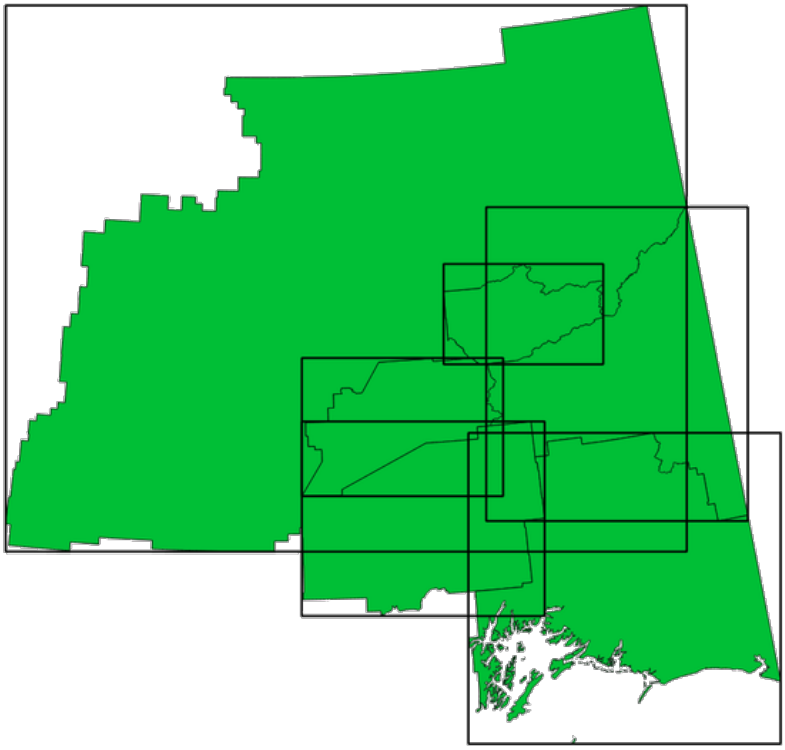


Fig. 24.61: Svarta linjer representerar de avgränsande boxarna för varje polygonfunktion

Tillåter *features in-place modification* av polygonfeatures

Se även

Minsta begränsande geometri

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje, polygon]	Vektorlager för linje- eller polygoninmatning
Begränsningar	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret (bounding box). *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Begränsningar	`OUTPUT`	[vektor: polygon]	Inmatningslagrets avgränsande rutor. Förutom input-attributen innehåller output-lagret också följande fält: `bredd`, `höjd`, `area` och `perimeter` för den genererade polygonen.

Python-kod

Algoritm-ID: `` `native:boundingboxes ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Buffra

Beräknar ett buffertområde för alla funktioner i ett inmatningslager med hjälp av ett fast eller datadefinierat avstånd.

Det är möjligt att använda ett negativt avstånd för polygoninmatningslager. I detta fall kommer bufferten att resultera i en mindre polygon (setback).

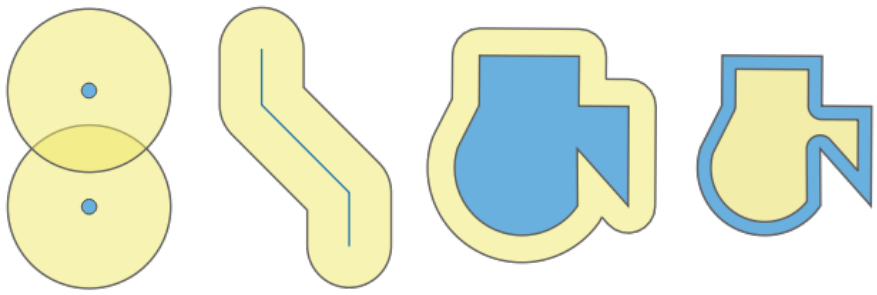


Fig. 24.62: Buffert (i gult) för punkter, linjer, polygoner med positiv buffert och polygoner med negativ buffert

Tillåter *features in-place modification* av polygonfeatures

Standardmeny: *Vector* ► *Geoprocessing Tools*

Varning

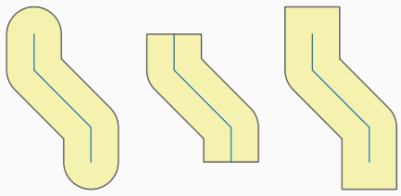
Den här algoritmen kan ta bort befintliga primärnycklar eller FID-värden och återskapa dem i utdatalagren, beroende på indataparametrarna.

Se även

Buffert med variabelt avstånd, *Buffert med flera ringar (konstant avstånd)*, *Buffert med variabel bredd (av M-värde)*

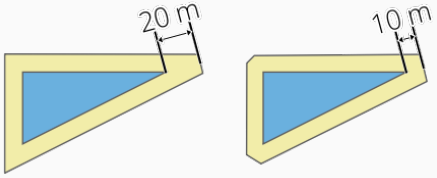
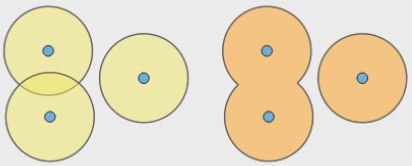
Parametrar

Grundläggande parametrar

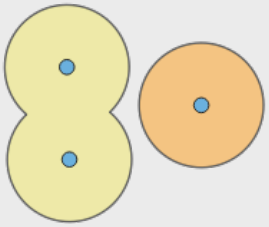
Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Avstånd inom vilket ett objekt kommer att snäppa till dina placerade stödlinjer. Efter att du har ställt in detta måste du starta om Scribus för att ställa in denna inställning.	`DISTANCE`	[numerisk: dubbel] Standard: 10.0	Buffertavstånd (från gränsen för varje funktion). Du kan använda knappen Data Defined (Datadefinierad) till höger för att välja ett fält från vilket radien ska beräknas. På så sätt kan du ha olika radier för varje funktion.
Segment	`SEGMENTS`	[numerisk: heltal] Standard: 5	Styr antalet linjesegment som ska användas för att approximera en kvartscirkel när man skapar avrundade offsets.
Typ av slutlock	`END_CAP_STYLE`	[uppräkning] Standard: 0	Styr hur radavslut ska hanteras i bufferten. Ett av följande: • 0 — Omgång • 1 — Platta • 2 — Torg  Fig. 24.63: Runda, platta och fyrkantiga lock
Join style	`JOIN_STYLE`	[uppräkning] Standard: 0	Anger om runda, geringade eller avfasade skarvar ska användas vid förskjutning av hörn i en linje. Alternativen är: • 0 — Omgång • 1 — Miter • 2 — Avfasning Fig. 24.64: Runda, geringade och fasade fogningsstilar

continues on next page

Table 24.154 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Mitergräns	`MITER_LIMIT`	[numerisk: dubbel] Standard: 2,0	Ställer in det maximala avståndet från offsetgeometrin som ska användas när man skapar en geringad fog som en faktor av offsetavståndet (gäller endast för geringade fogstilar). Minsta värde: 1.0  Fig. 24.65: En 10 m buffert med en gräns på 2 och en 10 m buffert med en gräns på 1
Lös upp resultatet	`DISSOLVE`	[boolean] Standard: Falsk	Lös upp den slutliga bufferten. Om `True` (markerat) kommer överlappande buffertar att lösas upp (kombineras) till en enda flerdelad funktion.  Fig. 24.66: Standard (tre endelade inslag - vänster), upplöst (1 flerdelat inslag med 2 delar - höger)
Buffrad	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret (buffert). *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Håll isär olika funktioner	`SEPARATE_DISJO`	[boolean] Standard: Falsk	Om `True` (markerad) och dissolved är markerad, kommer objekt som inte överlappar eller berör varandra att exporteras som separata objekt (istället för delar av ett flerdelat objekt).  Fig. 24.67: Resultat i 2 enstaka delar

Utgångar

Etikett	Namn	Typ	Beskrivning
Buffrad	`OUTPUT`	[vektor: polygon]	Utgående (buffert) polygonlager

Python-kod

Algoritm-ID: `native:buffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Centroider

Skapar ett nytt punktlager med punkter som representerar mittpunkterna i geometrierna i inmatningslagret.

Centroiden är en enda punkt som representerar barycentrum (av alla delar) av objektet, så den kan ligga utanför objektets gränser. Men det kan också vara en punkt på varje del av objektet.

Attributen för punkterna i det utgående lagret är desamma som för de ursprungliga funktionerna.

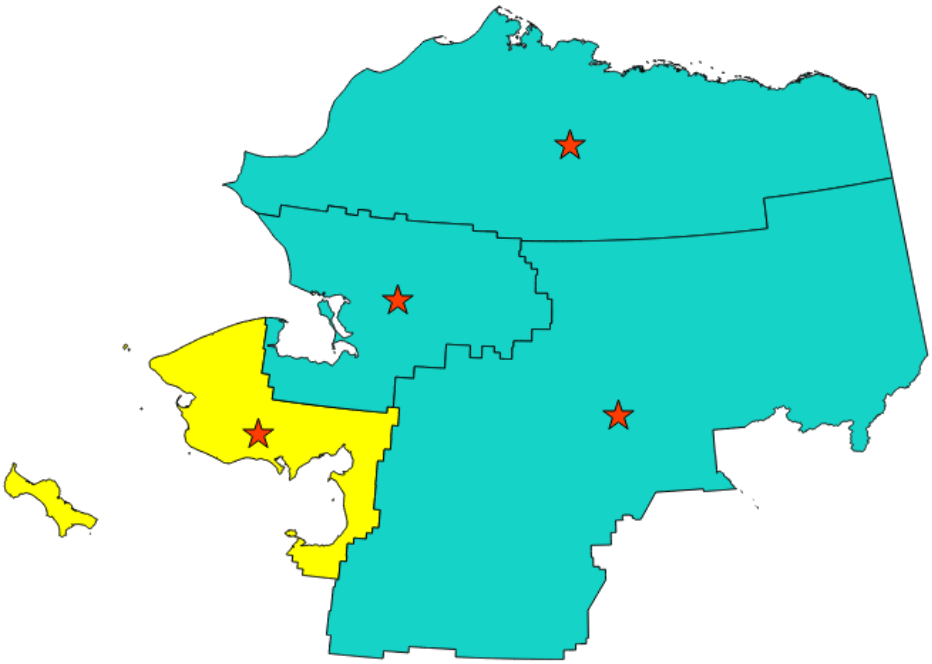


Fig. 24.68: De röda stjärnorna representerar centroiderna för funktionerna i inmatningslagret.

Tillåter *features in-place modification* av punktfeatures

Standardmeny: *Vektor* ► *Geometriverktyg*

Varning

Den här algoritmen kan ta bort befintliga primärnycklar eller FID-värden och återskapa dem i utdatalagren, beroende på indataparametrarna.

Se även

Punkt på ytan

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Skapa centroid för varje del	`ALL_PARTS`	[boolean] Standard: Falsk	Om True (markerat) skapas en centroid för varje del av geometrin
Centroider	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange det utgående lagret (centroid). *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Centroider	`OUTPUT`	[vektor: punkt]	Utgångspunkt vektorlager (centroider)

Python-kod

Algoritm-ID: `native:centroids`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Kontrollera giltighet

Utför en validitetskontroll av geometrierna i ett vektorlager.

Geometrierna klassificeras i tre grupper (giltig, ogiltig och fel) och för varje grupp genereras ett vektorlager med dess egenskaper:

- Lagret **Valid output** innehåller endast de giltiga funktionerna (utan topologiska fel).
- Lagret **Invalid output** innehåller alla ogiltiga funktioner som algoritmen har hittat.
- Lagret **Error output** är ett punktlager som pekar på var de ogiltiga funktionerna hittades.

Attributtabellerna för de genererade lagren kommer att innehålla viss ytterligare information (”message” för **error**-lagret, ”FID” och ”_errors” för **invalid**-lagret och endast ”FID” för **valid**-lagret):

Attributtabellen för varje genererat vektorlager kommer att innehålla ytterligare information (antal fel som hittats och typ av fel):

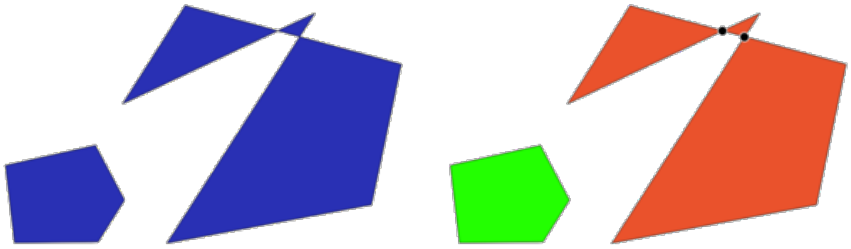


Fig. 24.69: Vänster: inmatningslagret. Höger: det giltiga lagret (grönt), det ogiltiga lagret (orange)

Standardmeny: *Vektor* ► *Geometriverktyg*

Se även

Fixa geometrier och kärntillägget *Tillägget Geometrikontroll*, *Validera täckning*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT_LAYER`	[vektor: geometri]	Ingångsvektorlager
Metod	`METHOD`	[uppräkning] Standard: 2	Metod som ska användas för att kontrollera giltigheten. Alternativ: • 0: Den som valts i digitaliseringsinställningarna • 1: QGIS • 2: GEOS
Ignore ring self intersection	`IGNORE_RING_SE`	[boolean] Standard: Falsk	Ignorera självskärande ringar när du kontrollerar giltigheten.
Giltig utdata	`VALID_OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det vektorlager som ska innehålla en kopia av de giltiga egenskaperna i källlagret. *One of*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Ogiltig utdata	`INVALID_OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Vektorlager som innehåller en kopia av de ogiltiga funktionerna i källlagret med fältet `_errors` som innehåller en sammanfattning av de fel som hittades. *One of*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Felutmatning	`ERROR_OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Punktlager av den exakta positionen för de validitetsproblem som upptäckts med fältet `message` som beskriver det eller de fel som hittats. *One of*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Antal fel	`ERROR_COUNT`	[numerisk: heltal]	Antalet geometrier som orsakade fel.
Felutmatning	`ERROR_OUTPUT`	[vektor: punkt]	Punktlager med den exakta positionen för de validitetsproblem som upptäckts med fältet `message` som beskriver det eller de fel som upptäckts.
Antal ogiltiga funktioner	`INVALID_COUNT`	[numerisk: heltal]	Antalet ogiltiga geometrier.
Ogiltig utdata	`INVALID_OUTPUT`	[samma som inmatning]	Vektorlager som innehåller en kopia av de ogiltiga funktionerna i källlagret med fältet `_errors` som innehåller en sammanfattning av de fel som hittats.
Antal giltiga funktioner	`VALID_COUNT`	[numerisk: heltal]	Antalet giltiga geometrier.
Giltig utdata	`VALID_OUTPUT`	[samma som inmatning]	Vektorlager som innehåller en kopia av de giltiga egenskaperna i källlagret.

Python-kod

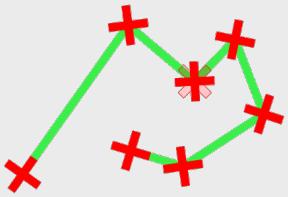
Algoritm-ID: `qgis:checkvalidity`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

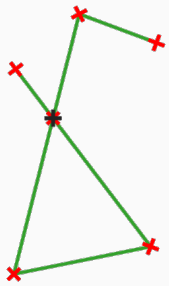
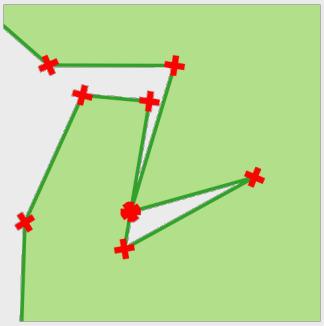
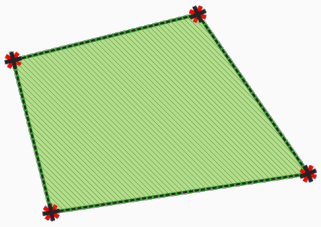
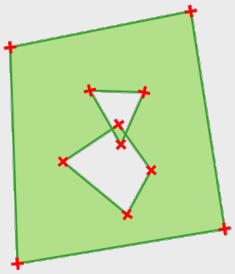
Olika typer av felmeddelanden och deras innebörd

Table 24.157: Några exempel på felaktiga geometrikontroller i QGIS

Geometrisammanhang	GEOS-validering och felmeddelande	QGIS-validering och felmeddelande
På varandra följande punkter på en linje har samma koordinater 	✗ *Upprepad punkt*	✗ *Linje a innehåller x duplikatnod(er) vid b*

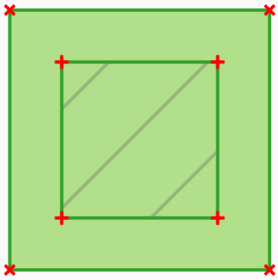
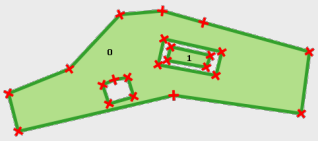
continues on next page

Table 24.157 – fortsättning från föregående sida

Geometrisammanhang	GEOS-validering och felmeddelande	QGIS-validering och felmeddelande
Delar av en linje skär varandra 		*Segmenten a och b av linje c skär varandra vid d*
Polygongeometri rör vid sig själv och genererar en ring 	*Ringens egen korsning*	*Ringens egen korsning*
Två ringar (yttre eller inre) i en polygongeometri är identiska 	*Duplicera ringar*	
Geometrin rör vid sig själv 	*Självskärande*	

continues on next page

Table 24.157 – fortsättning från föregående sida

Geometrisammanhang	GEOS-validering och felmeddelande	QGIS-validering och felmeddelande
En polygongeometri ligger ovanpå en annan polygongeometri 	✗ *Inkapslat skal*	
En del av en multipolygongeometri ligger inom ett hål i en multipolygongeometri 		✗ *Polygon a ligger inuti polygon b*
Punktgeometrin har inte ett korrekt koordinatpar. Koordinatparet innehåller inte ett latitudvärde och ett longitudvärde i nämnd ordning.	✗ * Ogiltig koordinat*	

Samla in geometrier

Tar ett vektorlager och samlar dess geometrier till nya flerdelade geometrier.

Ett eller flera attribut kan anges för att endast samla in geometrier som tillhör samma klass (som har samma värde för de angivna attributen), alternativt kan alla geometrier samlas in.

Alla utdatageometrier konverteras till multigeometrier, även de som bara har en enda del. Den här algoritmen löser inte upp överlappande geometrier - de samlas ihop utan att formen på varje geometridel ändras.

Se algoritmerna ”Promote to multipart” eller ”Aggregate” för alternativa möjligheter.

Standardmeny: *Vektor* ► *Geometriverktyg*

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Sammanslagna, *Befordra till multipart*, *Lös upp*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Unika ID-fält	`FIELD`	[tabellfält: valfritt] [lista]	Välj ett eller flera attribut för att samla geometrierna
Uppsamlad	`OUTPUT`	[samma som inmatning]	Vektorlager med samlade geometrier

Utgångar

Etikett	Namn	Typ	Beskrivning
Uppsamlad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange utdatavektorlagret för de insamlade geometrierna. *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Python-kod

Algoritm-ID: `native:samla`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Konkavt skrov

Beräknar det konkava skrovet av funktionerna från ett inmatat punktlager.

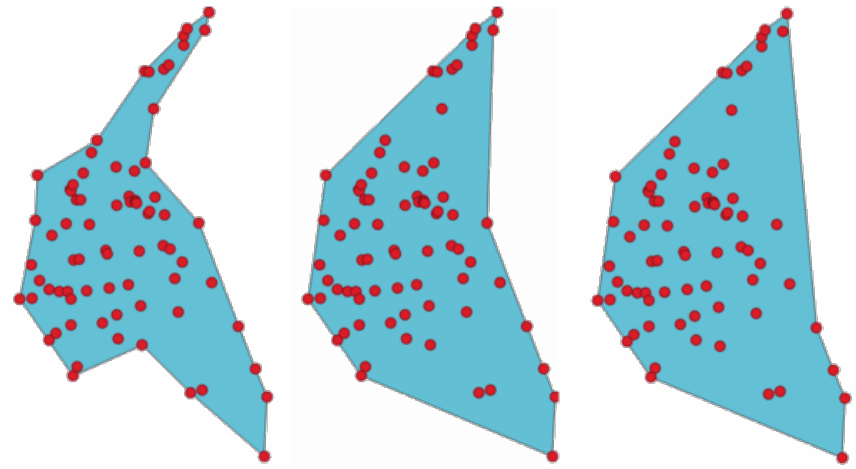


Fig. 24.70: Konkava skrov med olika tröskelvärden (0,3, 0,6, 0,9)

Se även

Konvex skrovform

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningspunktlag	`INPUT`	[vektor: punkt]	Vektorlager för inmatningspunkt
Tröskelvärde	`ALPHA`	[numerisk: dubbel] Standard: 0,3	Tal från 0 (maximalt konkavt skrov) till 1 (konvext skrov).
Tillåtna hål	`HOLES`	[boolean] Standard: Sann	Välj om du vill tillåta hål i det slutliga konkava skrovet
Dela upp geometrier med flera delar i geometrier med en del	`NO_MULTIGEOMET`	[boolean] Standard: Falsk	Kontrollera om du vill ha endelade geometrier i stället för flerdelade.
Konkavt skrov	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Konkavt skrov	`OUTPUT`	[vektor: polygon]	Utgångsvektorlagret

Python-kod

Algoritm-ID: `` `native:concavehull` ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Konvertera geometrityp

Skapar ett nytt lager baserat på ett befintligt lager, med en annan typ av geometri.

Attributtabellen för det utgående lagret är densamma som för det ingående lagret.

Alla konverteringar är inte möjliga. Ett linjelager kan t.ex. konverteras till ett punktlager, men ett punktlager kan inte konverteras till ett linjelager.

Se även

Polygonisera, *Denna algoritm skapar ett vektorlager med ett rutnät som täcker en angiven utsträckning. Elementen i rutnätet kan vara punkter, linjer eller polygoner. Storleken och/eller placeringen av varje elementen i rutnätet bestäms av de horisontella och vertikala avstånden. Koordinatsystemet i utdata måste anges. Rutnätets utsträckning och avståndsvärden måste anges i koordinaterna och enheterna för koordinatsystemet. Den övre vänstra punkten (minX, maxY) används som referenspunkt. Det betyder att, vid den punkten, är ett element garanterat att placeras. Om inte bredden och höjden av den valda utsträckningen är en multipel av de valda avstånden, är detta inte sant för de andra punkterna som definierar den utsträckningen.*, *Polygoner till linjer*, *Pekar ut vägen*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Ny geometrityp	`TYPE`	[uppräkning] Standard: 0	Geometrityp som ska tillämpas på utdataobjekten. En av: • 0 — Centroider • 1 — Noder • 2 — Linjestringar • 3 — Multilinjestrings • 4 — Polygoner
Varningen verifierar att en PDF-fil har lästs in i en bildram, där den rastreras eller konverteras till en bitmapp. Dess upplösning kan vara mindre än idealisk. Se PDF Export för att lära dig hur du minimerar detta problem.	`OUTPUT`	[vektor: geometri] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Varningen verifierar att en PDF-fil har lästs in i en bildram, där den rastreras eller konverteras till en bitmapp. Dess upplösning kan vara mindre än idealisk. Se PDF Export för att lära dig hur du minimerar detta problem.	`OUTPUT`	[vektor: geometri]	Utgående vektorlager - typen beror på parametrarna

Python-kod

Algoritm-ID: `qgis:convertgeometrytype`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Konvertera till krökta geometrier

Konverterar en geometri till dess krökta geometriekvivalent.

Redan krökta geometrier kommer att bibehållas utan ändringar.

Tillåter *features in-place modification* av linje- och polygonfeatures

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje eller polygon]	Ingångsvektorlager
Maximal tolerans för avstånd	`DISTANCE`	[numerisk: dubbel] Standard: 0,000001	Det maximala tillåtna avståndet mellan hörnens ursprungliga placering och var de skulle hamna i den konverterade krökta geometrin
Maximal vinkeltolerans	`ANGLE`	[numerisk: dubbel] Standard: 0,000001	Segment anses vara lämpliga att ersätta med en båge om alla punkter är regelbundet placerade på kandidatbågen. Denna parameter anger den maximala vinkelavvikelse (i grader) som tillåts vid test av regelbundet punktavstånd. Mellan 0 och 45°.
Kurvor	`OUTPUT`	[vektor: sammansatt kurva eller kurvapolygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Kurvor	`OUTPUT`	[vektor: sammansatt kurva eller kurvapolygon]	Utgående vektorlager med krökta geometrier

Python-kod

Algoritm ID: `native:konverteringskurvor`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Konvex skrovform

Beräknar det konvexa skrovet för varje funktion i ett inmatningslager.

Se algoritmen ”Minimum bounding geometry” för en beräkning av ett konvext skrov som täcker hela lagret eller grupperade undergrupper av funktioner.

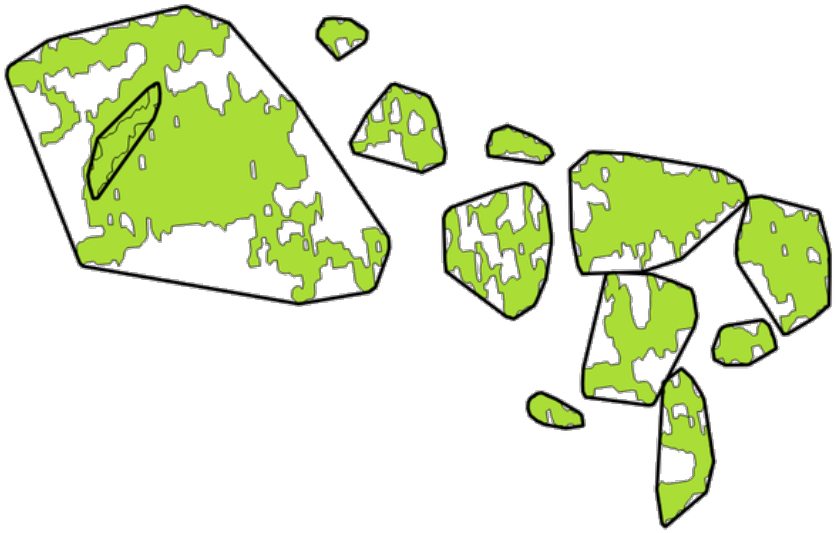


Fig. 24.71: Svarta linjer identifierar det konvexa skrovet för varje lagerfunktion

Tillåter *features in-place modification* av polygonfeatures

Standardmeny: *Vector* ► *Geoprocessing Tools*

Se även

Minsta begränsande geometri, *Konkavt skrov*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Konvexa skrov	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Konvexa skrov	`OUTPUT`	[vektor: polygon]	Vektorlagret för utdata (konvex skrov). Förutom inmatningsattributen innehåller utdatalagret också följande fält: `area` och `perimeter` för den genererade polygonen.

Python-kod

Algoritm-ID: `native:convexhull`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa lager från utsträckning

Skapar ett nytt vektorlager som innehåller en enda funktion med geometri som matchar omfattningen av inmatningslagret.

Den kan användas i modeller för att konvertera en bokstavlig utsträckning (formatet `xmin`, `xmax`, `ymin`, `ymax`) till ett lager som kan användas för andra algoritmer som kräver en lagerbaserad indata.

Se även

Skapa lager från punkt

Parametrar

Etikett	Namn	Typ	Beskrivning
Extent (xmin, xmax, ymin, ymax)	`INPUT`	[omfattning]	Inmatningens omfattning Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Extent	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Extent	`OUTPUT`	[vektor: polygon]	Vektorlagret för utdata (omfattning)

Python-kod

Algoritm-ID: `native:extenttolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa lager från punkt

Skapar ett nytt vektorlager som innehåller en enda funktion med geometri som matchar en punktparameter. Det kan användas i modeller för att konvertera en punkt till ett punktlager för algoritmer som kräver en lagerbaserad inmatning.

Se även

Skapa lager från utsträckning

Parametrar

Etikett	Namn	Typ	Beskrivning
Punkt	`INPUT`	[koordinater]	Inmatningspunkt, inklusive CRS-information (exempel: `397254, 6214446 [EPSG:32632]`). Om CRS inte tillhandahålls kommer projektets CRS att användas. Punkten kan anges genom att klicka på kartbilden.
Punkt	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Punkt	`OUTPUT`	[vektor: punkt]	Vektorlagret för utdatapunkten som innehåller inmatningspunkten.

Python-kod

Algoritm-ID: `native:pointtolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa kilbuffertar

Skapar kilformade buffertar från inmatningspunkter.

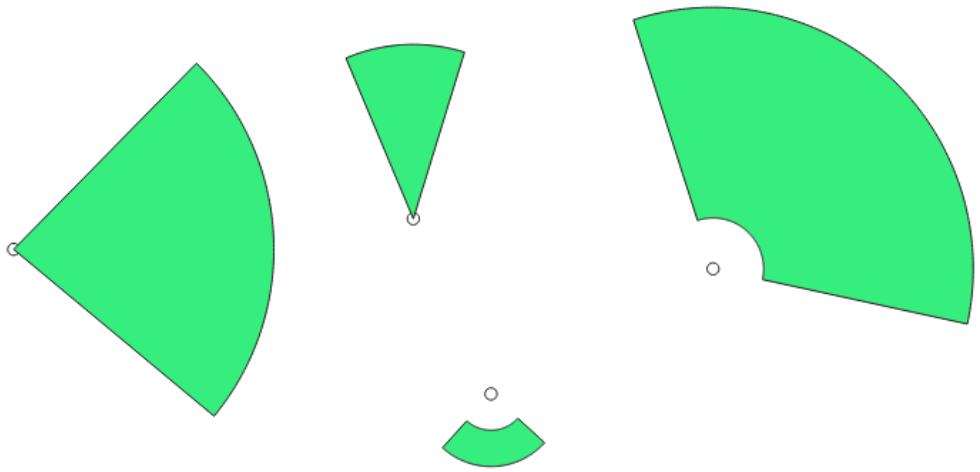


Fig. 24.72: Buffertar för kilar

Den ursprungliga utdata från denna algoritm är CurvePolygon-geometrier, men dessa kan automatiskt segmentiseras till polygoner beroende på utdataformatet.

Se även

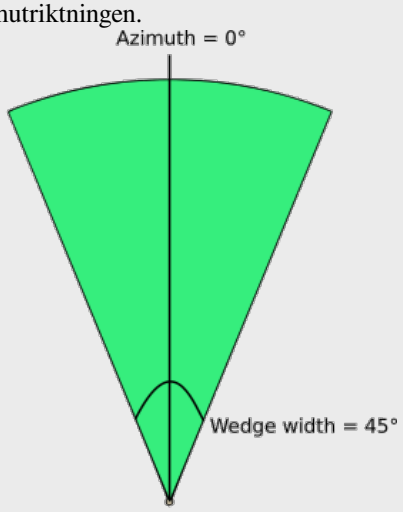
Buffra, *Buffert med variabel bredd (av M-värde)*, *Avsmalnande buffertar*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: punkt]	Vektorlager för inmatningspunkt
Azimut (grader från norr)	`AZIMUTH`	[numerisk: dubbel] Standard: 0,0	Vinkel (i grader) som kilens mittvärde

continues on next page

Table 24.160 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Kantbredd (i grader)	`WIDTH`	[numerisk: dubbel] Standard: 45,0	Bredd (i grader) på bufferten. Kilen kommer att sträcka sig till halva vinkelbredden på vardera sidan om azimutriktningen.  Fig. 24.73: Azimut- och breddvärden för kilbufferten
Yttre radie	`OUTER_RADIUS`	[numerisk: dubbel] Standard: 1,0	Kilens yttre *storlek* (längd): storleken avser från källpunkten till kilformens kant.
Innersta radien Valfritt	`INNER_RADIUS`	[numerisk: dubbel] Standard: 0,0	Värde för inre radie. Om 0 kommer kilen att börja från källpunkten.
Buffertar	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Buffertar	`OUTPUT`	[vektor: polygon]	Det utgående (kilbufferten) vektorlagret

Python-kod

Algoritm-ID: `native:wedgebuffers`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Delaunay-triangulering

Skapar ett polygonlager med Delaunay-triangulering som motsvarar det inmatade punktlagret.

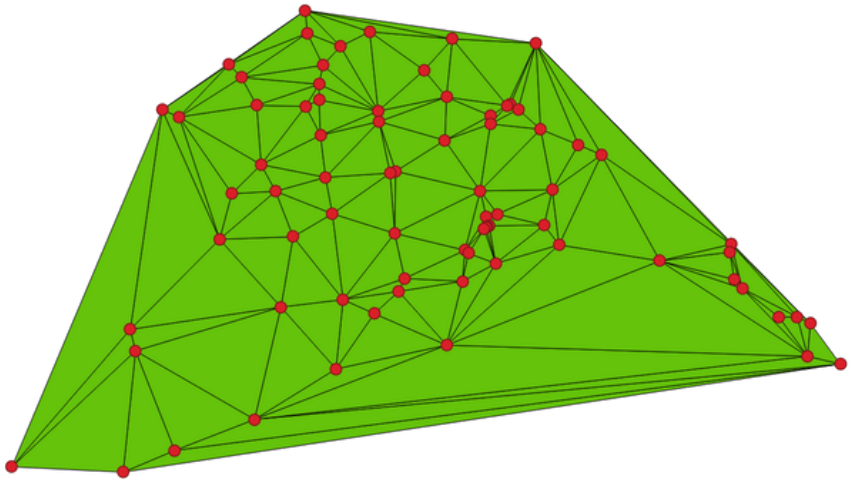


Fig. 24.74: Delaunay-triangulering på punkter

Standardmeny: *Vektor* ► *Geometriverktyg*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: punkt]	Vektorlager för inmatningspunkt
Tolerans Valfritt	`TOLERANCE`	[numerisk: dubbel] Standard: 0,0	Anger en valfri snäppningstolerans som kan användas för att förbättra trianguleringens robusthet.
Lägg till punkt-ID:n i utdata	`ADD_ATTRIBUTES`	[boolean] Standard: Sann	Anger om fält som lagrar ID för involverade punktfunktioner ska läggas till i utdata. Om False används ett ”id”-fält för att identifiera polygonerna.
Delaunay-triangulering	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Delaunay-triangulering	`OUTPUT`	[vektor: polygon]	Vektorlager för utdata (Delaunay-triangulering)

Python-kod

Algoritm-ID: `native:delaunaytriangulering`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Ta bort hål

Tar ett polygonlager och tar bort hål i polygonerna. Det skapar ett nytt vektorlager där polygoner med hål har ersatts av polygoner med endast sin yttre ring. Attributen ändras inte.

En valfri parameter för minsta yta gör att endast hål som är mindre än ett angivet tröskelvärde för ytan tas bort. Om denna parameter lämnas på `0,0` tas alla hål bort.

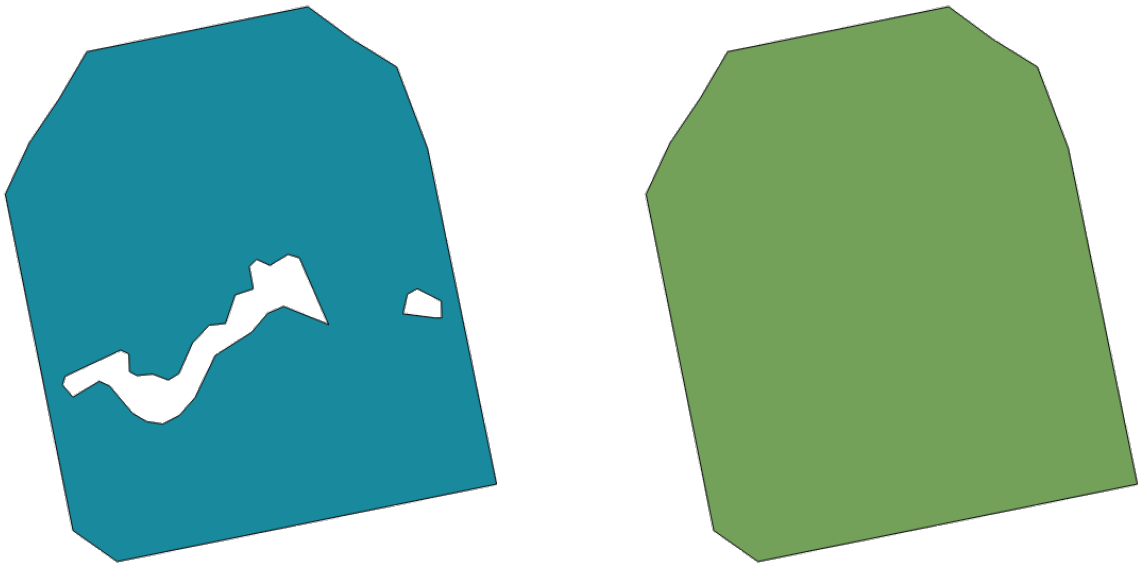


Fig. 24.75: Före och efter rengöringen

Tillåter *features in-place modification* av polygonfeatures

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: polygon]	Vektorlager för inmatad polygon
Ta bort hål med area mindre än Valfritt	`MIN_AREA`	[numerisk: dubbel] Standard: 0,0	Endast hål med en yta som är mindre än detta tröskelvärde kommer att raderas. Med ett värde på `0.0` kommer **alla** hål att raderas.
Renad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Renad	`OUTPUT`	[samma som inmatning]	Det utgående (rensade) vektorlagret

Python-kod

Algoritm-ID: `native:radera hål`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Förtätning efter antal

Tar ett polygon- eller linjelager och genererar ett nytt lager där geometrierna har ett större antal hörn än det ursprungliga lagret.

Om geometrierna har Z- eller M-värden kommer dessa att interpoleras linjärt vid de extra hörnen.

Antalet nya hörn som ska läggas till i varje segment anges som en inmatningsparameter.

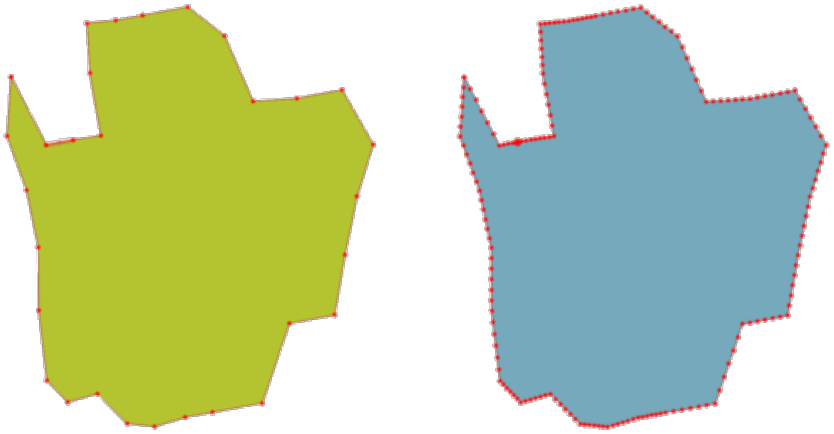


Fig. 24.76: Röda punkter visar topparna före och efter förtätningen

Tillåter *features in-place modification* av linje- och polygonfeatures

Standardmeny: *Vektor ► Geometriverktyg*

Se även

Förtätning med intervall

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje, polygon]	Vektorlager för linje- eller polygoninmatning
Tillägg att lägga till	`VERTICES`	[numerisk: heltal] Standard: 1	Antal toppar som ska läggas till i varje segment
Densifierad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Densifierad	`OUTPUT`	[samma som inmatning]	Det utgående (förtätade) vektorlagret

Python-kod

Algoritm-ID: `native:densifygeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Förtätning med intervall

Tar ett polygon- eller linjelager och genererar ett nytt lager där geometrierna har ett större antal hörn än det ursprungliga lagret.

Geometrierna förtätas genom att lägga till regelbundet placerade extra hörn inuti varje segment så att det maximala avståndet mellan två hörn inte överstiger det angivna avståndet.

Om geometrierna har Z- eller M-värden kommer dessa att interpoleras linjärt vid de extra hörnen.

Denna algoritm exporterar ett vektorlager till en PostgreSQL-databas, där en ny tabell skapas.

Om du anger ett avstånd på 3 konverteras segmentet `[0 0] -> [10 0]` till `[0 0] -> [2,5 0] -> [5 0] -> [7,5 0] -> [10 0]`, eftersom det krävs 3 extra hörn på segmentet och ett avstånd på 2,5 gör att de fördelas jämnt över segmentet.

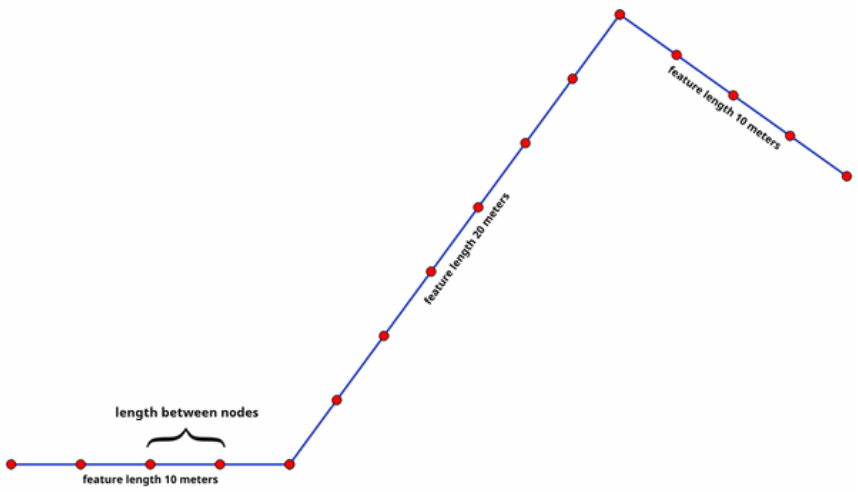


Fig. 24.77: Förtäta geometrin vid ett givet intervall

Tillåter *features in-place modification* av linje- och polygonfeatures

Se även

Förtätning efter antal

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje, polygon]	Vektorlager för linje- eller polygoninmatning
Intervall mellan toppar som ska läggas till	`INTERVAL`	[numerisk: dubbel] Standard: 1,0	Maximalt avstånd mellan två på varandra följande toppar
Densifierad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Densifierad	`OUTPUT`	[samma som inmatning]	Det utgående (förtätade) vektorlagret

Python-kod

Algoritm-ID: `` `native:densifygeometriesgivenaninterval ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Lös upp

Tar ett vektorlager och kombinerar dess funktioner till nya funktioner. Ett eller flera attribut kan anges för att upplösa funktioner som tillhör samma klass (med samma värde för de angivna attributen), alternativt kan alla funktioner upplösas till en enda funktion.

Alla utdatageometrier konverteras till multigeometrier. Om indata är ett polygonlager kommer gemensamma gränser för intilliggande polygoner som löses upp att raderas. Om den valfria inställningen ”Keep disjoint features separate” är aktiverad kommer funktioner och delar som inte överlappar eller rör vid varandra att exporteras som separata funktioner (i stället för delar av en enda flerdelad funktion).

Den resulterande attributtabellen kommer att ha samma fält som inmatningslagret. Värdena i utdataskiktets fält är värdena för den första indatafunktionen som bearbetas.

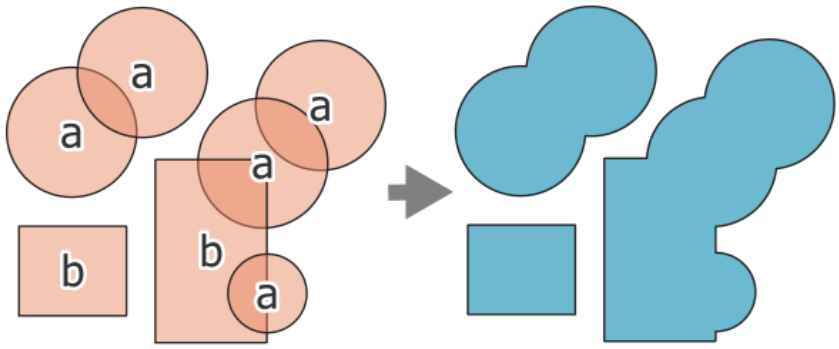


Fig. 24.78: Upplösning av ett lager till en enda flerdelad funktion

Standardmeny: *Vector* ► *Geoprocessing Tools*

Varning

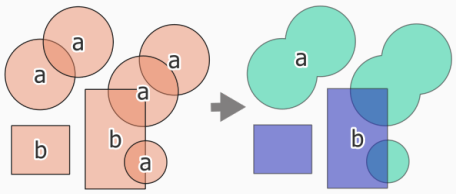
Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

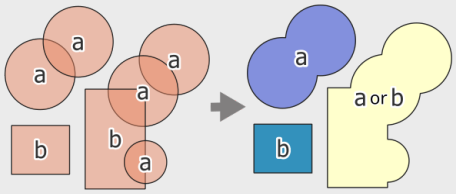
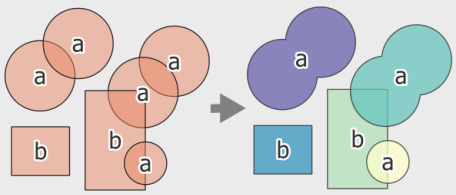
Lös upp täckning, *Sammanslagna*, *Samla in geometrier*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Lös upp fältet/fälten Valfritt	`FIELD`	[tabellfält: valfritt] [lista] Standard: []	Objekt som har samma värde för det/de valda fälten ersätts med ett enda objekt och deras geometrier slås samman. Om inget fält anges upplöses alla funktioner, vilket resulterar i en enda (flerdelad) funktion.  Fig. 24.79: Lös upp polygonlagret på ett gemensamt attribut (2 flerdelade funktioner)
Upplöst	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Håll isär olika funktioner	`SEPARATE_DISJO`	[boolean] Standard: Falsk	Delar av upplösta features exporteras som separata features (istället för delar av en multipart feature).  Fig. 24.80: source (vänster), dissolve all (3 distinkta funktioner - höger)  Fig. 24.81: källa (vänster), upplöses på fältet (5 distinkta funktioner - höger)

Utgångar

Etikett	Namn	Typ	Beskrivning
Upplöst	`OUTPUT`	[samma som inmatning]	Utgångsvektorlagret med upplösta geometrier

Python-kod

Algoritm ID: `native:upplösning`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Drape (ställ in Z-värde från raster)

Använder värden som samplas från ett band i ett rasterlager för att ställa in Z-värdet för varje överlappande vertex i objektets geometri. Rastervärdena kan eventuellt skalas med en förinställd mängd.

Om det redan finns Z-värden i lagret skrivs de över med det nya värdet. Om inga Z-värden finns kommer geometrin att uppgraderas så att den inkluderar Z-dimensionen.

Tillåter *features in-place modification* av punkt-, linje- och polygonfeatures med Z aktiverat

Se även

Ställ in M-värde från raster, *Ställ in Z-värde*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Ett rasterlager	`RASTER`	[raster]	Rasterlager med Z-värden
Bandnummer	`BAND`	[rasterband] Standard: 1	Det rasterband som Z-värdena ska hämtas från
Värde för NoData eller icke korsande hörn	`NODATA`	[numerisk: dubbel] Standard: 0,0	Värde att använda om vertexen inte korsar (en giltig pixel i) rastret
Skalfaktor	`SCALE`	[numerisk: dubbel] Standard: 1,0	Skalningsvärde: bandvärdena multipliceras med detta värde.
Offset	`OFFSET`	[numerisk: dubbel] Standard: 0,0	Offset-värde: det läggs algebraiskt till bandvärdena efter tillämpning av ”Scale factor”.

continues on next page

Table 24.161 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Uppdaterad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret (med Z-värden från rasterlagret). *One of*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Uppdaterad	`OUTPUT`	[samma som inmatning]	Det utgående vektorlagret med Z-värden från rasterlagret

Python-kod

Algoritm-ID: `native:setzfromraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Släpp M/Z-värden

Tar bort M- (mått) eller Z- (höjd) värden från indatageometrier.

Se även

Ställ in M-värde, *Ställ in Z-värde*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager med M- eller Z-värden
Drop M Values	`DROP_M_VALUES`	[boolean] Standard: Falsk	Tar bort M-värdena från geometrierna
Drop Z Values	`DROP_Z_VALUES`	[boolean] Standard: Falsk	Tar bort Z-värdena från geometrierna
Z/M droppade	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Z/M droppade	`OUTPUT`	[samma som inmatning]	Det utgående vektorlagret (identiskt med det ingående lagret, förutom att M- och/eller Z-dimensionerna har tagits bort från geometrierna).

Python-kod

Algoritm-ID: `native:dropmzvalues`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Eliminera utvalda polygoner

Kombinerar valda polygoner i inmatningslagret med vissa intilliggande polygoner genom att radera deras gemensamma gräns. Den angränsande polygonen kan antingen vara den med störst eller minst yta eller den som delar den största gemensamma gränsen med den polygon som ska elimineras.

Eliminate används normalt för att ta bort sliver-polygoner, dvs. små polygoner som är ett resultat av polygonskärningsprocesser där gränserna för indata är liknande men inte identiska.

Standardmeny: *Vector* ► *Geoprocessing Tools*

Se även

Fixa geometrier

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: polygon]	Vektorlager för inmatad polygon
Sammanfoga urvalet med den närliggande polygonen med	`MODE`	[uppräkning] Standard: Ej inställd	Välj den parameter som ska användas för att ta bort de markerade polygonerna: • 0 — Största område • 1 — Minsta yta • 2 — Största gemensamma gräns
Eliminerad	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Eliminerad	`OUTPUT`	[vektor: polygon]	Det utgående polygonvektorlagret.

Python-kod

Algoritm-ID: `qgis:eliminerar utvalda polygoner`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Explodera linjer

Tar ett linjelager och skapar ett nytt där varje linjelager ersätts av en uppsättning linjer som representerar segmenten i den ursprungliga linjen.

Varje linje i det resulterande lagret innehåller bara en start- och en slutpunkt, utan några mellanliggande hörnpunkter mellan dem.

Om inmatningslagret består av CircularStrings eller CompoundCurves kommer utdatalagret att vara av samma typ och endast innehålla enstaka kurvsegment.

Observera

- Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.
- Denna algoritm kräver inte giltiga geometrier som indata.

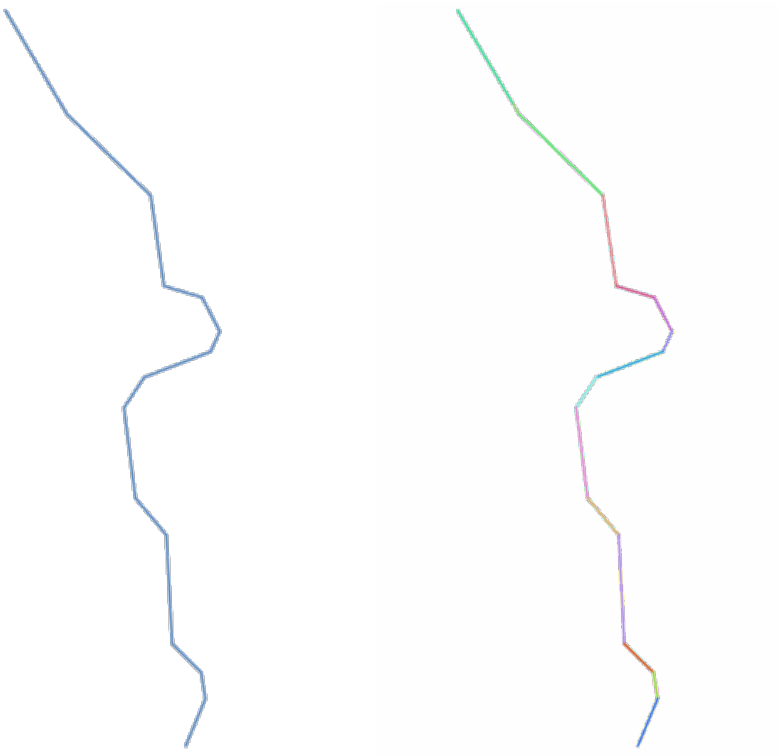


Fig. 24.82: Det ursprungliga linjelagret och det sprängda

Tillåter *funktioner på plats modifiering* av linjefunktioner

Se även

Dela upp, *Linje delsträng*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje]	Ingångslinje vektorlager
Exploded	`OUTPUT`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Exploded	`OUTPUT`	[vektor: linje]	Det utgående linjevektorlagret med funktioner som representerar varje segment i det ingående lagret.

Python-kod

Algoritm-ID: `native:explodelines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Förlänga linjer

Förlänger linjens geometri med en angiven mängd i början och slutet av linjen.

Linjer förlängs med hjälp av bäringen för det första och sista segmentet i linjen.

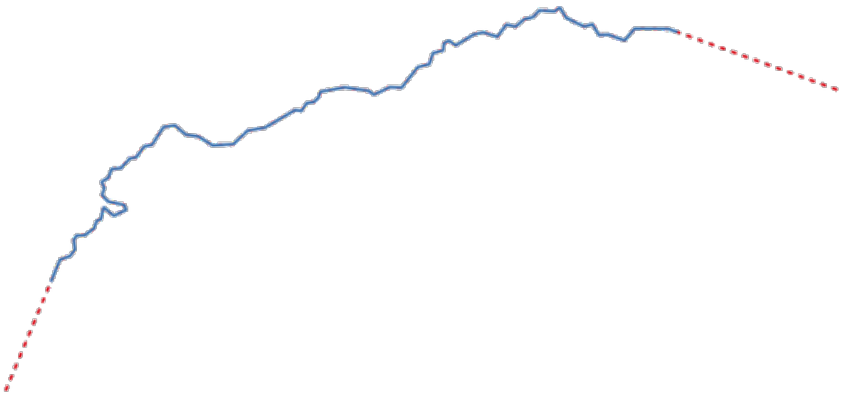


Fig. 24.83: De röda strecken representerar den initiala och slutliga förlängningen av det ursprungliga lagret

Tillåter *funktioner på plats modifiering* av linjefunktioner

Se även

Linje delsträng

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje]	Ingångslinje vektorlager
Startavstånd	`START_DISTANCE`	[numerisk: dubbel] Standard: 0,0	Avstånd med vilket linjens första segment ska förlängas (startpunkt)
Slutavstånd	`END_DISTANCE`	[numerisk: dubbel] Standard: 0,0	Avstånd med vilket linjens sista segment ska förlängas (slutpunkt)
Utökad	`OUTPUT`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Utökad	`OUTPUT`	[vektor: linje]	Det utgående (utökade) linjevektorskiktet.

Python-kod

Algoritm-ID: `native:extendlines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Extrahera M-värden

Extraherar M-värden från geometrier till funktionsattribut.

Som standard extraheras endast M-värdet från det första vertexet i varje funktion, men algoritmen kan som tillval beräkna statistik för alla geometrins M-värden, inklusive summa, medelvärde, minimum och maximum.

Se även

Extrahera Z-värden, *Ställ in M-värde*, *Släpp M/Z-värden*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Summa att beräkna	`SUMMARIES`	[uppräkning] Standard: [0]	Statistik över M-värdena i en geometri. En eller flera av: • 0 — Första • 1 — Sista • 2 — Räkning • 3 — Summa • 4 — Medelvärde • 5 — Median • 6 — St.dev (pop) • 7 — Minimum • 8 — Maximalt • 9 — Område • 10 — Minoritet • 11 — Majoritet • 12 — Variationsrikedom • 13 — Q1 • 14 — Q3 • 15 — IQR
Prefix för utdatakolumn	`COLUMN_PREFIX`	[sträng] Standard: ”m_	Prefixet för kolumnen för utdata (M)
Extraherad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Extraherad	`OUTPUT`	[samma som inmatning]	Utgångsvektorlagret (med M värden)

Python-kod

Algoritm-ID: `native:extraheravärden`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Extrahera specifika toppar

Tar ett vektorlager och genererar ett punktlager med punkter som representerar specifika hörn i indatageometrierna.

Den här algoritmen kan t.ex. användas för att extrahera den första eller sista punkten i geometrin. De attribut som är associerade med varje punkt är desamma som är associerade med den funktion som vertexen tillhör.

Parametern vertex indices accepterar en kommaseparerad sträng som anger indexen för de toppar som ska extraheras. Det första vertexet motsvarar index 0, det andra vertexet har index 1 osv. Negativa index kan användas för att hitta toppar i slutet av geometrin, t.ex. motsvarar ett index på -1 den sista toppunkten, -2 motsvarar den näst sista toppunkten osv.

Ytterligare fält läggs till i vertikalerna som anger den specifika vertexpositionen (t.ex. 0, -1, etc.), det ursprungliga vertexindexet, vertexens del och dess index inom delen (samt dess ring för polygoner), avstånd längs den ursprungliga geometrin och bisectorvinkel för vertex för den ursprungliga geometrin.

Tillåter *features in-place modification* av punktfeatures

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Extrahera toppar, *Filtrera toppar efter M-värde*, *Filtrera toppar efter Z-värde*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Vertexindex	`VERTICES`	[sträng] Standard: '0'	Kommaseparerad sträng med index för de hörnpunkter som ska extraheras.
Vertices	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Vertices	`OUTPUT`	[vektor: punkt]	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.

Python-kod

Algoritm-ID: `native:extrahera specifika vertikaler`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Extrahera toppar

Tar ett vektorlager och genererar ett punktlager med punkter som representerar hörnen i indatageometrierna.

De attribut som är kopplade till varje punkt är desamma som är kopplade till den funktion som vertexen tillhör.

Ytterligare fält läggs till i vertikalerna och anger vertexindex (med början vid 0), objektets del och dess index inom delen (samt dess ring för polygoner), avstånd längs originalgeometrin och vertexets bisektionsvinkel för originalgeometrin.

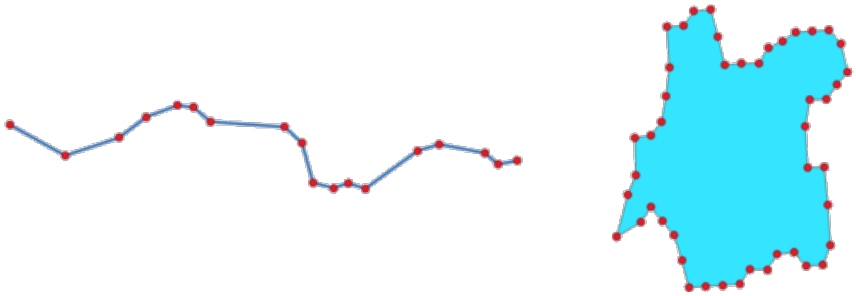


Fig. 24.84: Extraherade hörnpunkter för linje- och polygonlager

Tillåter *features in-place modification* av punktfeatures

Standardmeny: *Vektor* ► *Geometriverktyg*

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Extrahera specifika toppar, *Filtrera toppar efter M-värde*, *Filtrera toppar efter Z-värde*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Vertices	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Vertices	`OUTPUT`	[vektor: punkt]	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 its part. 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`: is the index of a vertex in a specific ring. • `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.

Python-kod

Algoritm-ID: `` `native:extrahera vertikaler ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Extrahera Z-värden

Extraherar Z-värden från geometrier till funktionsattribut.

Som standard extraheras endast Z-värdet från det första toppunktet i varje funktion, men algoritmen kan som tillval beräkna statistik för alla geometrins Z-värden, inklusive summa, medelvärde, minimum och maximum.

Se även

Extrahera M-värden, *Ställ in Z-värde*, *Släpp M/Z-värden*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Summa att beräkna	`SUMMARIES`	[uppräkning] Standard: [0]	Statistik över Z-värdena i en geometri. En eller flera av: • 0 — Första • 1 — Sista • 2 — Räkning • 3 — Summa • 4 — Medelvärde • 5 — Median • 6 — St.dev (pop) • 7 — Minimum • 8 — Maximalt • 9 — Område • 10 — Minoritet • 11 — Majoritet • 12 — Variationsrikedom • 13 — Q1 • 14 — Q3 • 15 — IQR
Prefix för utdatakolumn	`COLUMN_PREFIX`	[sträng] Standard: ”z_	Prefixet för den utgående (Z) kolumnen
Extraherad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Extraherad	`OUTPUT`	[samma som inmatning]	Utgångsvektorlagret (med Z-värden)

Python-kod

Algoritm-ID: `native:extraherazvärden`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Filtrera toppar efter M-värde

Filtrerar bort toppunkter baserat på deras M-värde och returnerar geometrier med endast toppunkter som har ett M-värde som är större än eller lika med det angivna minimivärdet och/eller mindre än eller lika med maximivärdet.

Om minimivärdet inte anges testas endast maximivärdet, och på samma sätt om maximivärdet inte anges testas endast minimivärdet.

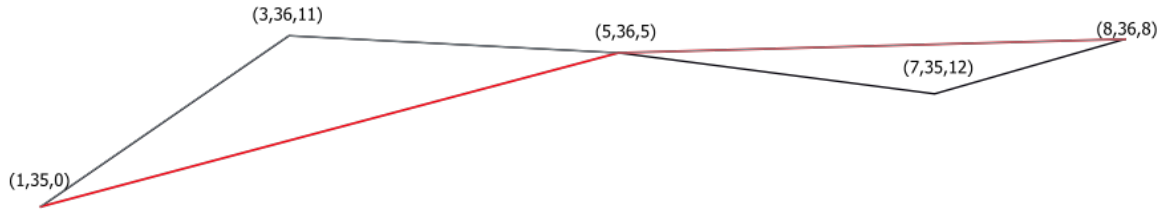


Fig. 24.85: Den röda linjen representerar den svarta linjen med endast toppar vars M-värde är <=10.

Tillåter *features in-place modification* av punkt-, linje- och polygonfunktioner med M aktiverat

Varning

Beroende på indatageometriattributen och de filter som används kan det hända att de geometrier som skapas med den här algoritmen inte längre är giltiga.

Se även

Filtrera toppar efter Z-värde, *Extrahera toppar*, *Extrahera specifika toppar*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager för att ta bort toppar från
Minimum Valfritt	`MIN`	[numerisk: dubbel] Standard: Ej inställd	Minsta tillåtna M-värden
Maximalt Valfritt	`MAX`	[numerisk: dubbel] Standard: Ej inställd	Maximalt antal M värden tillåtna
Filtrerad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Filtrerad	`OUTPUT`	[samma som inmatning]	Utgångsvektorlagret av funktioner med endast de filtrerade topparna.

Python-kod

Algoritm-ID: `native:filterverticesbym`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Filtrera toppar efter Z-värde

Filtrerar bort toppunkter baserat på deras Z-värde och returnerar geometrier med endast toppunkter som har ett Z-värde som är större än eller lika med det angivna minimivärdet och/eller mindre än eller lika med maximivärdet.

Om minimivärdet inte anges testas endast maximivärdet, och på samma sätt om maximivärdet inte anges testas endast minimivärdet.

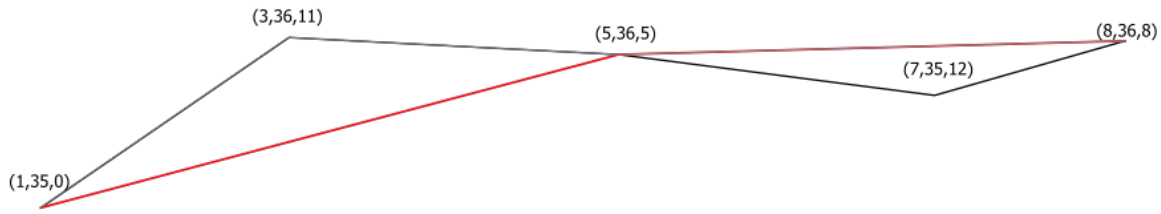


Fig. 24.86: Den röda linjen representerar den svarta linjen med endast toppar vars Z-värde är <=10.

Tillåter *features in-place modification* av punkt-, linje- och polygonfeatures med Z aktiverat

Varning

Beroende på indatageometriattributen och de filter som används kan det hända att de geometrier som skapas med den här algoritmen inte längre är giltiga. Du kan behöva köra algoritmen *Fixa geometrier* för att säkerställa att de är giltiga.

Se även

Filtrera toppar efter M-värde, *Extrahera toppar*, *Extrahera specifika toppar*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager för att ta bort toppar från
Minimum Valfritt	`MIN`	[numerisk: dubbel] Standard: Ej inställd	Minsta tillåtna Z-värden
Maximalt Valfritt	`MAX`	[numerisk: dubbel] Standard: Ej inställd	Maximalt antal Z-värden tillåtna
Filtrerad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Filtrerad	`OUTPUT`	[samma som inmatning]	Utgångsvektorlagret av funktioner med endast de filtrerade topparna.

Python-kod

Algoritm-ID: `native:filterverticesbyz`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Fixa geometrier

Försöker skapa en giltig representation av en given ogiltig geometri utan att förlora någon av de ingående topparna. Redan giltiga geometrier returneras utan ytterligare åtgärder. Skickar alltid ut lager med flera geometrier.

Tillåter *features in-place modification* av punkt-, linje- och polygonfunktioner utan M aktiverat

Varning

M-värden kommer att tas bort från utdata.

Se även

Kontrollera giltighet

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Reparationsmetod	`METHOD`	[uppräkning] Standard: 1	Metod som används för att reparera geometrierna. En av: • 0 — `Linework`: kombinerar alla ringar till en uppsättning nodade linjer och extraherar sedan giltiga polygoner från detta linework • 1 — `Structure`: gör först alla ringar giltiga och slår sedan ihop skal och subtraherar hål från skal för att generera ett giltigt resultat. Antar att hål och skal är korrekt kategoriserade. Kräver QGIS version byggd med GEOS 3.10 eller senare (kontrollera *Hjälp* ► *Om* menyn).
Fasta geometrier	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Fasta geometrier	`OUTPUT`	[samma som inmatning]	Utgångsvektorlagret med fasta geometrier.

Python-kod

Algoritm-ID: `native:fixgeometrier`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Kraft högerhands-regel

Tvingar polygongeometrier att följa högerhandsregeln, enligt vilken det område som avgränsas av en polygon ligger till höger om gränsen. I synnerhet är den yttre ringen orienterad i medurs riktning och alla inre ringar i moturs riktning.

Tillåter *features in-place modification* av polygonfeatures

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: polygon]	Ingångsvektorlager
Omorienterad	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Omorienterad	`OUTPUT`	[vektor: polygon]	Utgångsvektorlagret med omorienterade geometrier.

Python-kod

Algoritm-ID: `native:forcerhr`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Geodetisk linje delad vid antimeridianen

Delar upp en linje i flera geodetiska segment när linjen korsar antimeridianen (±180 graders longitud).

Delning vid antimeridianen underlättar den visuella framställningen av linjerna i vissa projektioner. Den returnerade geometrin kommer alltid att vara en geometri med flera delar.

När linjesegment i indatageometrin korsar antimeridianen delas de upp i två segment, varvid brytpunktens latitud bestäms med hjälp av en geodetisk linje som förbinder punkterna på vardera sidan om detta segment. Projektets aktuella ellipsoidinställning kommer att användas vid beräkning av denna brytpunkt.

Om indatageometrin innehåller M- eller Z-värden kommer dessa att interpoleras linjärt för de nya hörn som skapas vid antimeridianen.

Observera

Den här algoritmen använder ellipsoidbaserade mätningar och respekterar de aktuella *ellipsoidinställningarna*.

Tillåter *funktioner på plats modifiering* av linjefunktioner

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje]	Ingångslinje vektorlager
Split	`OUTPUT`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange utdatans linjevektorlager. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Split	`OUTPUT`	[vektor: linje]	Utgångslinjens vektorlager delas vid antimeridianen.

Python-kod

Algoritm-ID: `native:antimeridiansplit`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Geometri genom uttryck

Uppdaterar befintliga geometrier (eller skapar nya geometrier) för indatafunktioner med hjälp av ett QGIS-uttryck.

Detta möjliggör komplexa geometrimodifieringar som kan utnyttja all flexibilitet i QGIS uttrycksmotor för att manipulera och skapa geometrier för utdatafunktioner.

För hjälp med QGIS uttrycksfunktioner, se den inbyggda hjälp som finns i *expression builder*.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlager
Typ av geometri för utdata	`OUTPUT_GEOMETR`	[uppräkning] Standard: 0	Geometrin för utdata är starkt beroende av uttrycket: om du t.ex. skapar en buffert måste geometritypen vara polygon. En av följande: • 0 — Polygon • 1 — Linje • 2 — Punkt
Utgångsgeometrin har z-värden	`WITH_Z`	[boolean] Standard: Falsk	Välj om utdatageometrin ska inkludera Z-dimensionen
Utgångsgeometrin har m värden	`WITH_M`	[boolean] Standard: Falsk	Välj om utdatageometrin ska inkludera M-dimensionen
Geometriskt uttryck	`EXPRESSION`	[uttryck] Standard: '$geometry'	Lägg till det geometriska uttryck som du vill använda. Du kan använda knappen för att öppna dialogrutan Expression. I dialogen listas alla relevanta uttryck, tillsammans med deras hjälp och guide.
Modifierad geometri	`OUTPUT`	[vektor: geometri] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Modifierad geometri	`OUTPUT`	[vektor: geometri]	Utgångsvektorlagret

Python-kod

Algoritm-ID: `native:geometrybyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Interpolera punkt på linje

Skapar en punktgeometri som interpoleras på ett angivet avstånd längs linje- eller kurvgeometrier.

Z- och M-värdena interpoleras linjärt från befintliga värden.

Om en geometri med flera delar påträffas beaktas endast den första delen vid beräkningen av delsträngen.

Om det angivna avståndet är större än den inmatade funktionens längd, kommer den resulterande funktionen att ha en nollgeometri.

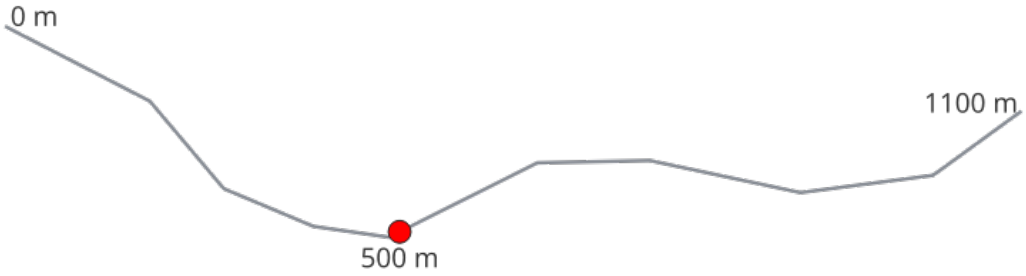


Fig. 24.87: Interpolerad punkt 500 meter från linjens början

Se även

Punkter längs geometrin

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje, polygon]	Vektorlager för linje- eller polygoninmatning
Avstånd inom vilket ett objekt kommer att snäppa till dina placerade stödlinjer. Efter att du har ställt in detta måste du starta om Scribus för att ställa in denna inställning.	`DISTANCE`	[numerisk: dubbel] Standard: 0,0	Avstånd från början av linjen
Interpolerade punkter	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Interpolerade punkter	`OUTPUT`	[vektor: punkt]	Utgångspunktens vektorlager med objekt på ett visst avstånd längs linjen eller polygongränsen

Python-kod

Algoritm-ID: `native:interpolatepoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Behåll de största delarna

Tar ett lager med polygoner eller multipolygoner och returnerar ett nytt lager där endast de *n* största polygonerna i varje multipolygonfunktion sparas. Om en funktion har *n* eller färre delar kommer funktionen bara att kopieras.

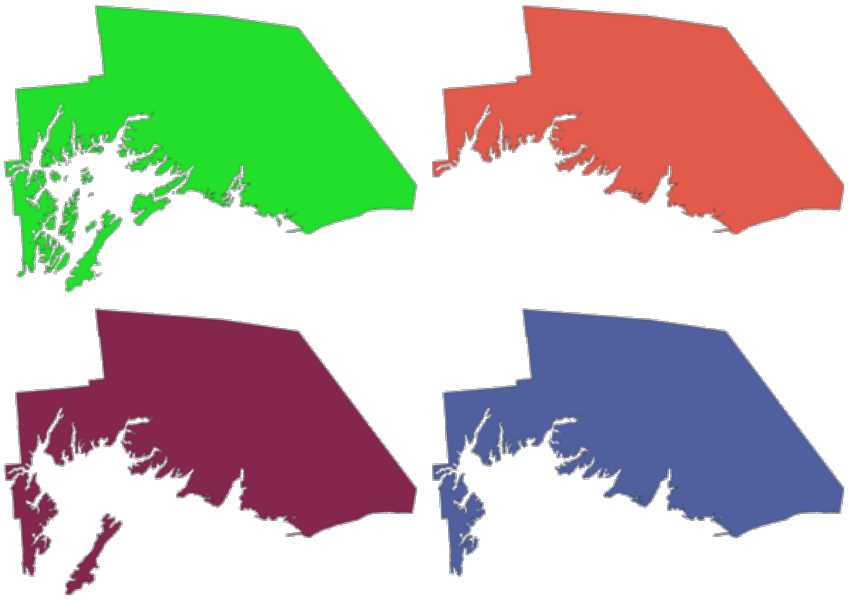


Fig. 24.88: Medurs från övre vänstra hörnet: originalfilm i flera delar, en, två och tre största delar hålls

Parametrar

Etikett	Namn	Typ	Beskrivning
Polygoner	`INPUT`	[vektor: polygon]	Vektorlager för inmatad polygon
** Delar som ska behållas**	`PARTS`	[numerisk: heltal] Standard: 1	Antal delar som ska behållas. Om 1, kommer endast den största delen av funktionen att behållas.
Parts (delar)	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående polygonvektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Parts (delar)	`OUTPUT`	[vektor: polygon]	Det utgående polygonvektorlagret med de N största delarna av varje egenskap

Python-kod

Algoritm ID: `qgis:keepnbiggestparts`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Linje delsträng

Returnerar den del av en linje (eller kurva) som ligger mellan de angivna start- och slutavstånden (mätt från linjens början).

Z- och M-värdena interpoleras linjärt från befintliga värden.

Om en geometri med flera delar påträffas beaktas endast den första delen vid beräkningen av delsträngen.

Fig. 24.89: Substringslinje med startavståndet inställt på 0 meter och slutavståndet på 250 meter.

Tillåter *funktioner på plats modifiering* av linjefunktioner

Se även

Förlänga linjer

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje]	Ingångslinje vektorlager
Startavstånd	`START_DISTANCE`	[numerisk: dubbel] Standard: 0,0	Avstånd längs inmatningslinjen till startpunkten för utmatningsfunktionen
Slutavstånd	`END_DISTANCE`	[numerisk: dubbel] Standard: 1,0	Avstånd längs inmatningslinjen till slutpunkten för utmatningsfunktionen
Substräng	`OUTPUT`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange utdatans linjevektorlager. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Substräng	`OUTPUT`	[vektor: linje]	Det utgående linjevektorlagret.

Python-kod

Algoritm-ID: `native:linjesubsträng`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Denna algoritm skapar ett vektorlager med ett rutnät som täcker en angiven utsträckning. Elementen i rutnätet kan vara punkter, linjer eller polygoner. Storleken och/eller placeringen av varje elementen i rutnätet bestäms av de horisontella och vertikala avstånden. Koordinatsystemet i utdata måste anges. Rutnätets utsträckning och avståndsvärden måste anges i koordinaterna och enheterna för koordinatsystemet. Den övre vänstra punkten (minX, maxY) används som referenspunkt. Det betyder att, vid den punkten, är ett element garanterat att placeras. Om inte bredden och höjden av den valda utsträckningen är en multipel av de valda avstånden, är detta inte sant för de andra punkterna som definierar den utsträckningen.

Genererar ett polygonlager med linjerna från ett inmatat linjelager som polygonringar.

Attributtabellen för det utgående lagret är densamma som för det ingående lagret.

Standardmeny: *Vektor ► Geometriverktyg*

Se även

Polygoner till linjer, *Polygonisera*, *Konvertera geometrityp*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje]	Ingångslinje vektorlager
Polygoner	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående polygonvektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Polygoner	`OUTPUT`	[vektor: polygon]	Det utgående polygonvektorlagret.

Python-kod

Algoritm-ID: `qgis:linestopolygoner`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Sammanfoga linjer

Sammanfogar alla anslutna delar av MultiLineString-geometrier till enkla LineString-geometrier.

Om några delar av de inmatade MultiLineString-geometrierna inte är anslutna, kommer den resulterande geometrin att vara en MultiLineString som innehåller alla linjer som kan slås samman och alla icke anslutna linjedelar.

Tillåter *funktioner på plats modifiering* av linjefunktioner

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje]	Ingångslinje vektorlager
Sammanslagna	`OUTPUT`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange utdatans linjevektorlager. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Sammanslagna	`OUTPUT`	[vektor: linje]	Det utgående (sammanslagna) linjevektorlagret.

Python-kod

Algoritm-ID: `native:mergelines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Minsta begränsande geometri

Skapar geometrier som omsluter objekten från ett inmatningslager. Funktionerna kan grupperas med ett fält. Utmatningslagret kommer sedan att innehålla en funktion per gruppvärde med en geometri (MBB) som täcker geometrierna för funktionerna med matchande värde.

Följande typer av omslutande geometrier stöds:

- begränsande låda (kuvert)
- orienterad rektangel
- cirkel
- konvex skrov

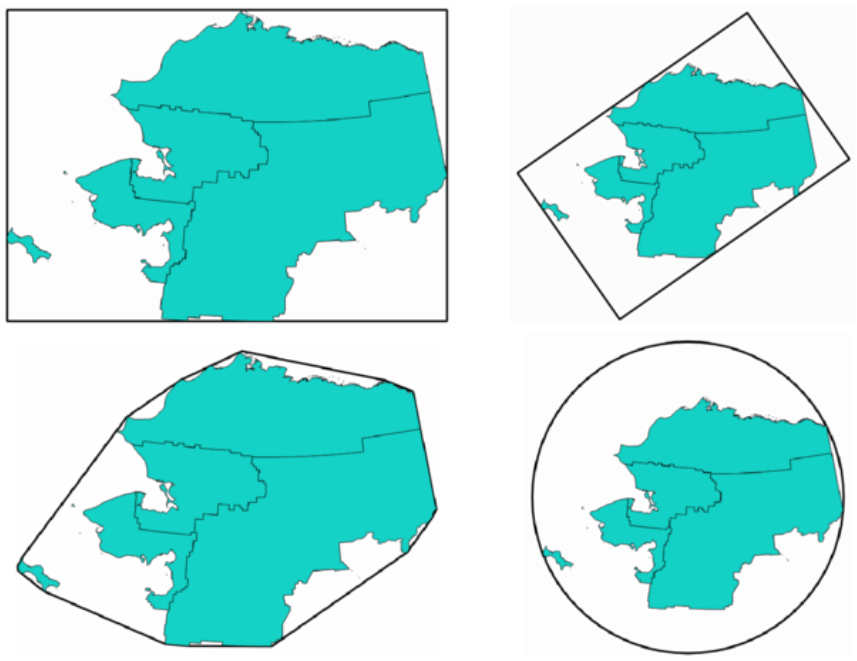


Fig. 24.90: Medurs från övre vänstra hörnet: kuvert, orienterad rektangel, cirkel, konvex kropp

Se även

Minsta omslutande cirklar

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Textfält Valfritt	`FIELD`	[tabellfält: alla]	Funktioner kan grupperas med ett fält. Om detta anges kommer utdatalagret att innehålla en funktion per grupperat värde med en minimal geometri som endast täcker funktionerna med matchande värden.
Typ av geometri	`TYPE`	[uppräkning] Standard: 0	Omslutande geometrityper. En av: • 0 — Kuvert (avgränsande box) • 1 — Minsta orienterade rektangel • 2 — Minsta omslutande cirkel • 3 — Konvex skrov
Avgränsande geometri	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående polygonvektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Avgränsande geometri	`OUTPUT`	[vektor: polygon]	Det utgående (avgränsande) polygonvektorlagret. Det har fälten från det inkommande lagret och, beroende på den valda parametern `TYPE`, har det utgående lagret även följande attribut: • `bredd`, `höjd`, `area` och `omkrets` för kuvertet (begränsningsrutan) • `bredd`, `höjd`, `area`, `vinkel` och `omkrets` för den minsta orienterade rektangeln • `radie` och `area` för den minsta omslutande cirkeln • `area` och `perimeter` för den konvexa höljet

Python-kod

Algoritm-ID: `qgis:minimumboundinggeometry`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Minsta omslutande cirklar

Beräknar de minsta omslutande cirklarna för objekten i inmatningslagret.

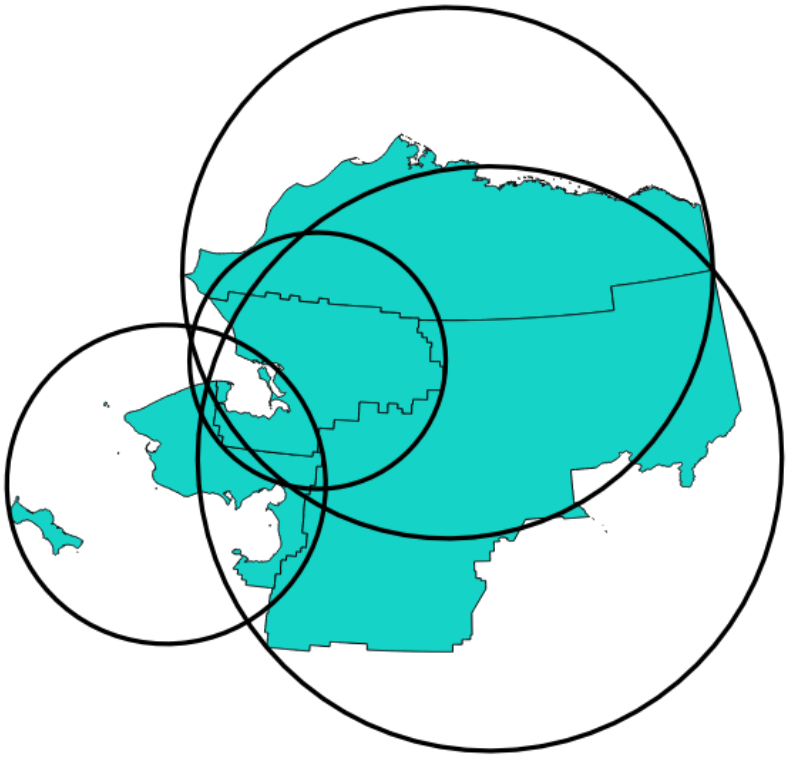


Fig. 24.91: Omslutande cirklar för varje funktion

Tillåter *features in-place modification* av polygonfeatures

Se även

Minsta begränsande geometri

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Antal segment i cirklar	`SEGMENTS`	[numerisk: heltal] Standard: 72	Antalet segment som används för att approximera en cirkel. Minimum 8, maximum 100000.
Minsta omslutande cirklar	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående polygonvektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Minsta omslutande cirklar	`OUTPUT`	[vektor: polygon]	Det utgående polygonvektorlagret. Förutom inmatningsattributen innehåller utdatalagret även följande fält: `area` och `radie` för den genererade polygonen.

Python-kod

Algoritm-ID: `native:minimumenclosingcircle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Buffert med flera ringar (konstant avstånd)

Beräknar en buffert med flera ringar (*donut*) för funktionerna i inmatningslagret med hjälp av ett fast eller dynamiskt avstånd och antal ringar.

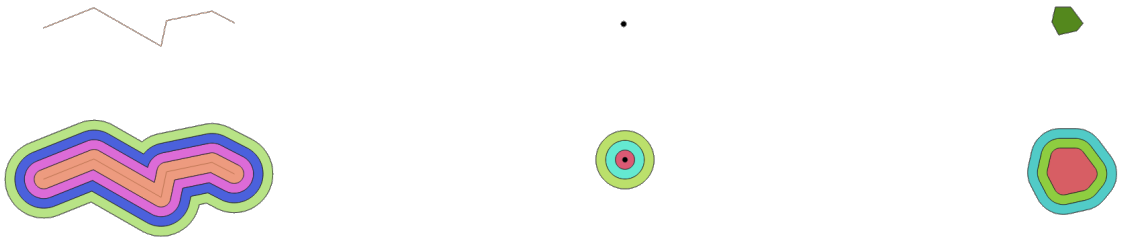


Fig. 24.92: Buffert med flera ringar för ett linje-, punkt- och polygonlager

Tillåter *features in-place modification* av polygonfeatures

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Buffra, *Buffert med variabelt avstånd*, *Rektanglar, ovaler, diamanter*, *Enkelsidig buffert*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Antal ringar	`RINGS`	[numerisk: heltal] Standard: 1	Antalet ringar. Det kan vara ett unikt värde (samma antal ringar för alla funktioner) eller så kan det hämtas från funktionsdata (antalet ringar beror på funktionsvärden).
Avstånd mellan ringarna	`DISTANCE`	[numerisk: dubbel] Standard: 1,0	Avstånd mellan ringarna. Det kan vara ett unikt värde (samma avstånd för alla funktioner) eller så kan det hämtas från funktionsdata (avståndet beror på funktionsvärden).
Buffert med flera ringar (konstant avstånd)	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående polygonvektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Buffert med flera ringar (konstant avstånd)	`OUTPUT`	[vektor: polygon]	Det utgående polygonvektorlagret. Förutom ingångsattributen innehåller utdatalagret även följande fält: `ringId` och `distance` för den genererade buffertpolygonen.

Python-kod

Algoritm-ID: `native:multiringconstantbuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Flera delar till enstaka delar

Delar upp flerdelade funktioner i inmatningslagret i endelade funktioner.

Attributen i utmatningslagret är desamma som de ursprungliga men uppdelade i enskilda funktioner.

Fig. 24.93: Till vänster källagret med flera delar och till höger utmatningsresultatet med en enda del

Tillåter *features in-place modification* av punkt-, linje- och polygonfeatures

Standardmeny: *Vektor* ► *Geometriverktyg*

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Samla in geometrier, *Befordra till multipart*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Enskilda delar	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående polygonvektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Enskilda delar	`OUTPUT`	[samma som inmatning]	Utgångsvektorlagret.

Python-kod

Algoritm-ID: `` `native:multiparttosingleparts` ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Offset-linjer

Förskjuter linjer med ett angivet avstånd. Positiva avstånd förskjuter linjerna åt vänster och negativa avstånd förskjuter dem åt höger.

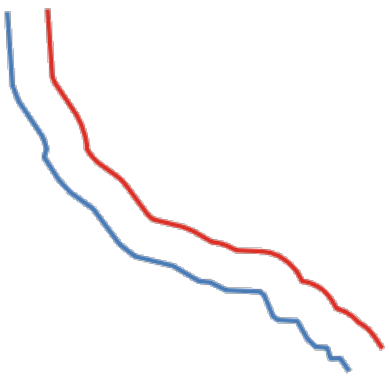


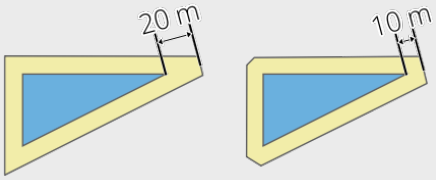
Fig. 24.94: I blått källagret, i rött det förskjutna lagret

Tillåter *funktioner på plats modifiering* av linjefunktioner

Se även

Array av förskjutna (parallella) linjer, *Översätt*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje]	Ingångslinje vektorlager
Avstånd inom vilket ett objekt kommer att snäppa till dina placerade stödlinjer. Efter att du har ställt in detta måste du starta om Scribus för att ställa in denna inställning.	`DISTANCE`	[numerisk: dubbel] Standard: 10.0	Offset-avstånd. Du kan använda knappen Data Defined till höger för att välja ett fält från vilket radien ska beräknas. På så sätt kan du ha olika radier för varje objekt (se *Buffert med variabelt avstånd*).
Segment	`SEGMENTS`	[numerisk: heltal] Standard: 8	Styr antalet linjesegment som ska användas för att approximera en kvartscirkel när man skapar avrundade offsets.
Join style	`JOIN_STYLE`	[uppräkning] Standard: 0	Anger om runda, geringade eller avfasade skarvar ska användas vid förskjutning av hörn i en linje. Alternativen är: • 0 — Omgång • 1 — Miter • 2 — Avfasning Fig. 24.95: Runda, geringade och fasade fogningsstilar
Mitergräns	`MITER_LIMIT`	[numerisk: dubbel] Standard: 2,0	Ställer in det maximala avståndet från offsetgeometrin som ska användas när man skapar en geringad fog som en faktor av offsetavståndet (gäller endast för geringade fogstilar). Minsta värde: 1.0  Fig. 24.96: En 10 m buffert med en gräns på 2 och en 10 m buffert med en gräns på 1

continues on next page

Table 24.168 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Offset	`OUTPUT`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret (offset). *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Offset	`OUTPUT`	[vektor: linje]	Utgående (offset) linjelager

Python-kod

Algoritm-ID: `native:offsetline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Orienterad minsta avgränsande box

Beräknar den roterade rektangeln med minsta yta för varje funktion i inmatningslagret.

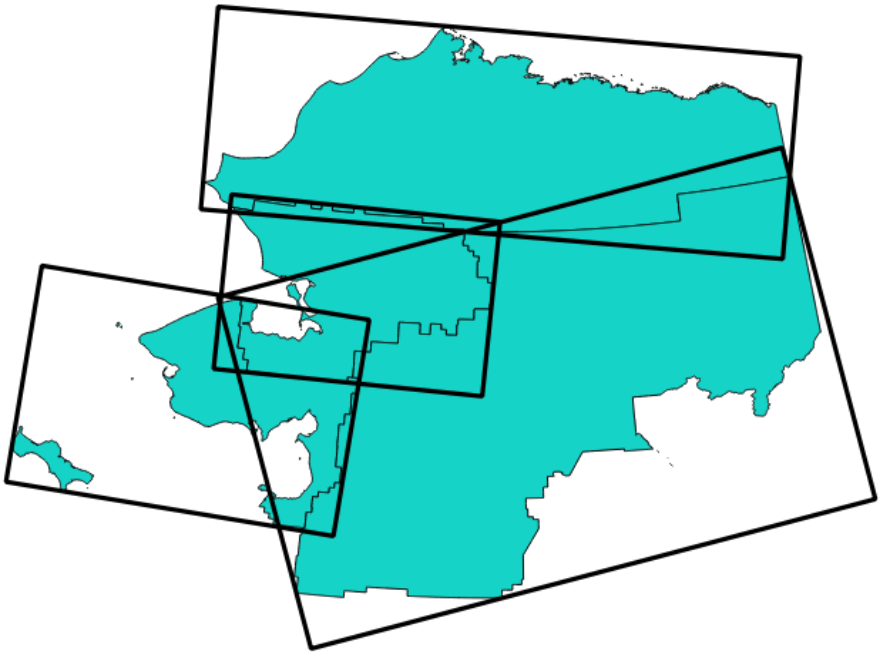


Fig. 24.97: Orienterad minsta avgränsande box

Tillåter *features in-place modification* av polygonfeatures

Se även

Minsta begränsande geometri

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Begränsande boxar	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående polygonvektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Begränsande boxar	`OUTPUT`	[vektor: polygon]	Det utgående polygonvektorlagret. Förutom ingångsattributen innehåller utdatalagret också följande fält: `bredd`, `höjd`, `vinkel`, `area` och `perimeter` för den genererade polygonen.

Python-kod

Algoritm-ID: `native:orientedminimumboundingbox`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Ortogonalisera

Försöker ortogonalisera geometrierna i det inmatade linje- eller polygonlagret. Denna process flyttar geometrins hörnpunkter för att försöka göra varje vinkel i geometrin antingen till en rät vinkel eller en rak linje.

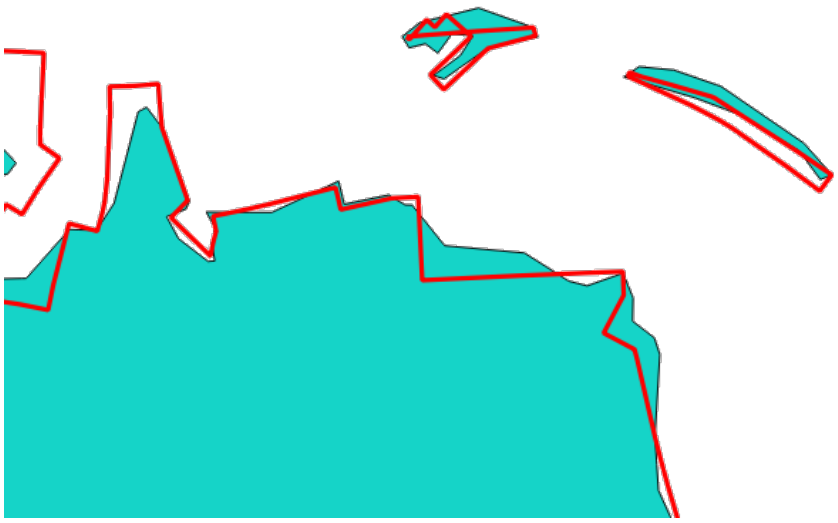


Fig. 24.98: I blått källagret och i rött ortogonaliserat resultat

Tillåter *features in-place modification* av linje- och polygonfeatures

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje, polygon]	Vektorlager för linje- eller polygoninmatning
Maximal vinkeltolerans (grader)	`ANGLE_TOLERANC`	[numerisk: dubbel] Standard: 15,0	Ange den maximala avvikelsen från en rät vinkel eller rak linje som en toppunkt kan ha för att den ska justeras. Mindre toleranser innebär att endast vertexar som redan är närmare räta vinklar kommer att justeras, och större toleranser innebär att vertexar som avviker mer från räta vinklar också kommer att justeras.
Maximala iterationer för algoritmen	`MAX_ITERATIONS`	[numerisk: heltal] Standard: 1000	Om du anger ett högre värde för det maximala antalet iterationer får du en mer ortogonal geometri till priset av extra bearbetningstid.
Orthogonaliserad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående polygonvektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Orthogonaliserad	`OUTPUT`	[samma som inmatning]	Det utgående polygonvektorlagret med justerade vinklar.

Python-kod

Algoritm-ID: `native:orthogonalize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Punkt på ytan

För varje funktion i inmatningslagret returneras en punkt som garanterat ligger på ytan av funktionens geometri.

Tillåter *features in-place modification* av punktfeatures

Varning

Den här algoritmen kan ta bort befintliga primärnycklar eller FID-värden och återskapa dem i utdatalagren, beroende på indataparametrarna.

Se även

Centroider

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Skapa en punkt på ytan för varje del	`ANGLE_TOLERANC`	[boolean]	Om den är markerad skapas en punkt för varje del av geometrin.
Punkt	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange utmatningspunktens vektorlager. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Punkt	`OUTPUT`	[vektor: punkt]	Vektorlagret för utmatningspunkten.

Python-kod

Algoritm-ID: `native:pointonsurface`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Punkter längs geometrin

Skapar punkter med regelbundna intervall längs linje- eller polygongeometrier. Skapade punkter får nya attribut som läggs till för avståndet längs geometrin och linjens vinkel vid punkten.

En valfri start- och slutoffset kan anges, som styr hur långt från geometrins start och slut punkterna ska skapas.

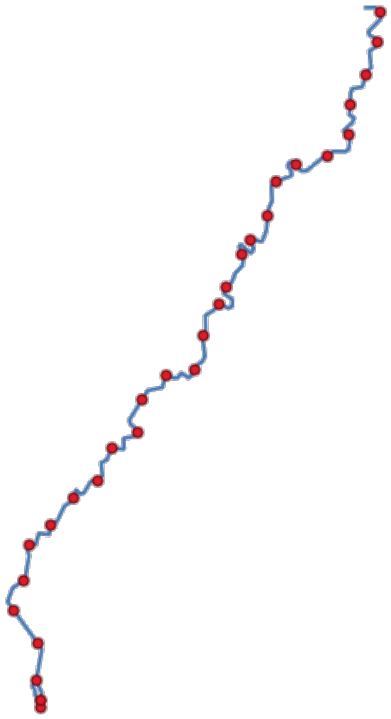


Fig. 24.99: Punkter skapade längs källinje-lagret

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Interpolera punkt på linje

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje, polygon]	Vektorlager för linje- eller polygoninmatning

continues on next page

Table 24.170 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Avstånd inom vilket ett objekt kommer att snäppa till dina placerade stödlinjer. Efter att du har ställt in detta måste du starta om Scribus för att ställa in denna inställning.	`DISTANCE`	[numerisk: dubbel] Standard: 1,0	Avstånd mellan två på varandra följande punkter längs linjen
Startförskjutning	`START_OFFSET`	[numerisk: dubbel] Standard: 0,0	Avstånd från början av inmatningsraden, som representerar positionen för den första punkten.
Slutavräkning	`END_OFFSET`	[numerisk: dubbel] Standard: 0,0	Avstånd från slutet av inmatningslinjen, som representerar den position bortom vilken ingen punktfunktion ska skapas.
Interpolerade punkter	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Interpolerade punkter	`OUTPUT`	[vektor: punkt]	Punktvektorlager med funktioner placerade längs linjer eller polygongränser i inmatningslagret. Förutom indataattributen för varje relaterad punkt innehåller utdatalagret också följande fält: `avstånd` och `vinkel` för den genererade punkten.

Python-kod

Algoritm-ID: `native:punkter längs linjer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Förskjutning av punkter

Givet ett avstånd identifierar närliggande punktfunktioner och fördelar dem radiellt över en cirkel vars centrum representerar deras barycenter. Ett bekvämt verktyg för att sprida överlappande funktioner.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: punkt]	Vektorlager för inmatningspunkt
Minsta avstånd till andra punkter	`PROXIMITY`	[numerisk: dubbel] Standard: 1,0	Avstånd under vilket punkter anses vara nära. Nära funktioner fördelas helt och hållet.
Förskjutningsavstår	`DISTANCE`	[numerisk: dubbel] Standard: 1,0	Radie för den cirkel där närliggande funktioner placeras
Horisontell fördelning för tvåpunktshus	`HORIZONTAL`	[boolean] Standard: Falsk	När endast två punkter identifieras som nära varandra, justeras de horisontellt på cirkeln i stället för vertikalt.
Displaced	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Displaced	`OUTPUT`	[vektor: punkt]	Utgångspunkt vektorlager

Python-kod

Algoritm-ID: `qgis:poängförskjutning`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Pol av otillgänglighet

Beräknar polen för otillgänglighet för ett polygonlager, vilket är den mest avlägsna interna punkten från ytans gräns.

Denna algoritm använder ”polylabel”-algoritmen (Vladimir Agafonkin, 2016), som är en iterativ metod som garanterat hittar den verkliga polen för otillgänglighet inom en angiven tolerans. En mer exakt tolerans (lägre värde) kräver fler iterationer och tar längre tid att beräkna.

Fig. 24.100: Pol av otillgänglighet

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: polygon]	Ingångsvektorlager
Tolerans	`TOLERANCE`	[numerisk: dubbel] Standard: 1,0	Ställ in toleransen för beräkningen
Punkt	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange det utgående polygonvektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Punkt	`OUTPUT`	[vektor: punkt]	Vektorlagret för utmatningspunkten. Det kommer att innehålla ett nytt attribut med avståndet `dist_pole` från den beräknade polen till polygongränsen.

Python-kod

Algoritm ID: `native:poleofinaccessibility`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Polygonisera

Skapar ett polygonlager vars gränser genereras från ett linjelager med **slutna** objekt.

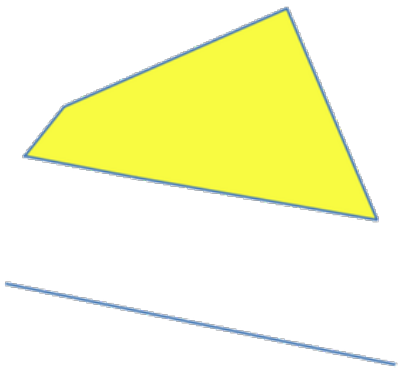


Fig. 24.101: De gula polygonerna som genereras från de slutna linjerna

Observera

Linjelagret måste ha slutna former för att kunna omvandlas till en polygon.

Se även

Polygoner till linjer, *Denna algoritm skapar ett vektorlager med ett rutnät som täcker en angiven utsträckning. Elementen i rutnätet kan vara punkter, linjer eller polygoner. Storleken och/eller placeringen av varje elementen i rutnätet bestäms av de horisontella och vertikala avstånden. Koordinatsystemet i utdata måste anges. Rutnätets utsträckning och avståndsvärden måste anges i koordinaterna och enheterna för koordinatsystemet. Den övre vänstra punkten (minX, maxY) används som referenspunkt. Det betyder att, vid den punkten, är ett element garanterat att placeras. Om inte bredden och höjden av den valda utsträckningen är en multipel av de valda avstånden, är detta inte sant för de andra punkterna som definierar den utsträckningen.*, *Konvertera geometrityp*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje]	Ingångslinje vektorlager
Behåll fält från inmatningslagret	`KEEP_FIELDS`	[boolean] Standard: Falsk	Kontrollera att fälten (endast tabellstrukturen, inte värdena) i inmatningslagret behålls
Polygoner från linjer	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående polygonvektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Polygoner från linjer	`OUTPUT`	[vektor: polygon]	Det utgående polygonvektorlagret från linjer

Python-kod

Algoritm-ID: `native:polygonize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Polygoner till linjer

Tar ett polygonlager och skapar ett linjelager med linjer som representerar gränserna för polygonerna i inmatningslagret.

Attributtabellen för det utgående lagret är densamma som för det ingående lagret.

Fig. 24.102: Svarta linjer som resultat av algoritmen

Standardmeny: *Vektor* ► *Geometriverktyg*

Se även

Denna algoritm skapar ett vektorlager med ett rutnät som täcker en angiven utsträckning. Elementen i rutnätet kan vara punkter, linjer eller polygoner. Storleken och/eller placeringen av varje elementen i rutnätet bestäms av de horisontella och vertikala avstånden. Koordinatsystemet i utdata måste anges. Rutnätets utsträckning och avståndsvärden måste anges i koordinaterna och enheterna för koordinatsystemet. Den övre vänstra punkten (minX, maxY) används som referenspunkt. Det betyder att, vid den punkten, är ett element garanterat att placeras. Om inte bredden och höjden av den valda utsträckningen är en multipel av de valda avstånden, är detta inte sant för de andra punkterna som definierar den utsträckningen., *Polygonisera*, *Konvertera geometrityp*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: polygon]	Vektorlager för inmatad polygon
Linjer:	`OUTPUT`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange utdatans linjevektorlager. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Linjer:	`OUTPUT`	[vektor: linje]	Linjevektorlagret för utdata från polygoner

Python-kod

Algoritm-ID: `native:polygonstoliner`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Projektpunkter (kartesiska)

Projicerar punktgeometrier med ett angivet avstånd och bäring (azimut).

Tillåter *features in-place modification* av punktfeatures

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: punkt]	Vektorlager för inmatningspunkt
Bearing (grader från norr)	`BEARING`	[numerisk: dubbel] Standard: 0,0	Medsols vinkel från norr, i enheten grad (°)
Avstånd inom vilket ett objekt kommer att snäppa till dina placerade stödlinjer. Efter att du har ställt in detta måste du starta om Scribus för att ställa in denna inställning.	`DISTANCE`	[numerisk: dubbel] Standard: 1,0	Avstånd till offsetgeometrier, i lagerenheter
Projicerad	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange utmatningspunktens vektorlager. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Projicerad	`OUTPUT`	[vektor: punkt]	Det utgående (projicerade) punktvektorlagret

Python-kod

Algoritm-ID: `native:projektpunktcartesian`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Befordra till multipart

Tar ett vektorlager med enpartsgeometrier och genererar ett nytt där alla geometrier är flerparts.

Inmatade funktioner som redan är flerdelade funktioner kommer att förbli oförändrade.

Denna algoritm kan användas för att tvinga geometrier till multiparttyper för att vara kompatibla med dataleverantörer som kräver multipartfunktioner.

Tillåter *features in-place modification* av punkt-, linje- och polygonfeatures

Se även

Sammanslagna, *Samla in geometrier*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Multiparts	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det multiparta vektorlagret för utdata. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Multiparts	`OUTPUT`	[samma som inmatning]	Det utgående flerdelade vektorlagret

Python-kod

Algoritm-ID: `native:promotetomulti`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rektanglar, ovaler, diamanter

Skapar ett buffertområde med rektangel-, oval- eller diamantform för varje egenskap i det inmatade punktlagret.

Formparametrarna kan vara fasta för alla funktioner eller dynamiska med hjälp av ett fält eller ett uttryck.

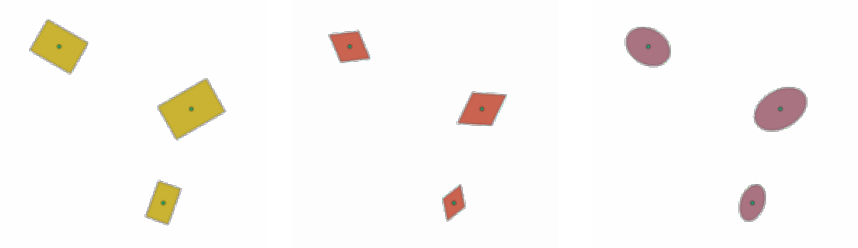


Fig. 24.103: Olika buffertformer med dynamiska parametrar

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: punkt]	Vektorlager för inmatningspunkt
Buffertens form	`SHAPE`	[uppräkning]	Den form som ska användas. En av: • 0 — Rektanglar • 1 — Ovaler • 2 — Diamanter
Bredd	`WIDTH`	[numerisk: dubbel] Standard: 1,0	Bredd på buffertformen
Höjd	`HEIGHT`	[numerisk: dubbel] Standard: 1,0	Buffertformens höjd
Rotation Valfritt	`ROTATION`	[numerisk: dubbel] Standard: 0,0	Rotation av buffertformen
Antal segment	`SEGMENTS`	[numerisk: heltal] Standard: 36	Antal segment för en hel cirkel (*Ovalform*)
&Spara utdata…	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
&Spara utdata…	`OUTPUT`	[vektor: polygon]	Det utgående vektorlagret (med buffertformerna)

Python-kod

Algoritm-ID: ”original:rektanglarovalsdiamanter

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Ta bort dubbletter av toppar

Tar bort duplicerade hörn från objekt, om borttagandet av hörnen inte resulterar i en degenererad geometri.

Parametern tolerance anger toleransen för koordinater när man avgör om hörnpunkterna är identiska.

Som standard beaktas inte Z-värden vid detektering av dubbletter av hörnpunkter. T.ex. kommer två hörn med samma X- och Y-koordinat men olika Z-värden fortfarande att betraktas som duplikat och ett av dem kommer att tas bort. Om parametern *Use Z Value* är true, testas även Z-värdena och hörn med samma X- och Y-koordinater men olika Z-värden behålls.

Tillåter *features in-place modification* av punkt-, linje- och polygonfeatures

Observera

Duplicerade vertikaler testas inte mellan olika delar av en geometri med flera delar, t.ex. kommer en geometri med flera punkter och överlappande punkter inte att ändras med den här metoden.

Se även

Extrahera toppar, *Extrahera specifika toppar*, *Ta bort duplicerade geometrier*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Tolerans	`TOLERANCE`	[numerisk: dubbel] Standard: 0,000001	Toppar som är närmare än det angivna avståndet betraktas som dubbletter
Använd Z-värde	`USE_Z_VALUE`	[boolean] Standard: Falsk	Om parametern *Use Z Value* är true, testas även Z-värdena och toppar med samma X och Y men olika Z bibehålls.

continues on next page

Table 24.174 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Renad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Renad	`OUTPUT`	[samma som inmatning]	Det utgående vektorlagret (utan dubbla toppar)

Python-kod

Algoritm-ID: `native:removeduplicatevertices`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Ta bort nollgeometrier

Tar bort alla objekt som inte har en geometri från ett vektorlager. Alla andra funktioner kopieras oförändrade.

Funktioner med nollgeometrier kan sparas i ett separat lager.

Om *Also remove empty geometries* är markerat tar algoritmen bort objekt vars geometrier inte har några koordinater, dvs. geometrier som är tomma. I så fall kommer även null-utdata att återspegla detta alternativ, eftersom de innehåller både null- och tomma geometrier.

Se även

Ta bort duplicerade geometrier

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager (med geometrier som inte är NULL)
Ta även bort tomma geometrier	`REMOVE_EMPTY`	[boolean]	
Geometrier som inte är noll	`OUTPUT` Valfritt	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange utdatavektorlagret för geometrierna som inte är NULL (och inte tomma). *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Null geometrier Valfritt	`NULL_OUTPUT`	[samma som inmatning] Standard: `[Hoppa över utdata]`	Ange utmatningsvektorlagret för NULL (och tomma) geometrier. *Ett av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Null geometrier	`NULL_OUTPUT`	[samma som inmatning]	Vektorlager för utdata (för NULL och, om så väljs, tomma geometrier)
Geometrier som inte är noll	`OUTPUT`	[samma som inmatning]	Det utgående vektorlagret (utan NULL och, om så väljs, tomma geometrier)

Python-kod

Algoritm-ID: `native:removenullgeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Omvänd riktning på linjen

Inverterar riktningen för ett linjelager.

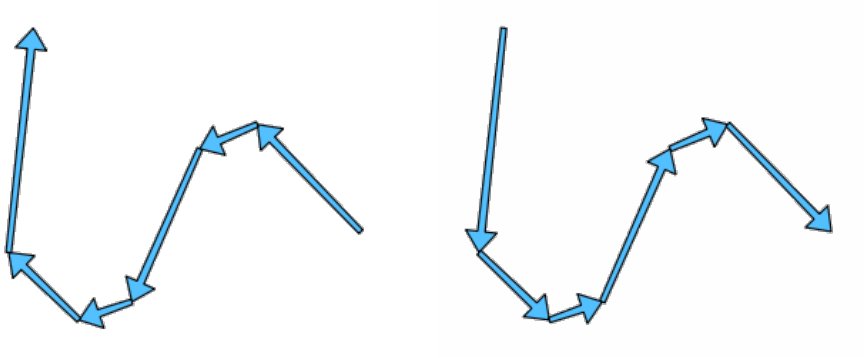


Fig. 24.104: Före och efter inverteringen av riktningen

Tillåter *funktioner på plats modifiering* av linjefunktioner

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje]	Ingångslinje vektorlager
Omvänd	`OUTPUT`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange utdatans linjevektorlager. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Omvänd	`OUTPUT`	[vektor: linje]	Vektorlagret för utdatalinjen (med omvända linjer)

Python-kod

Algoritm-ID: `` `native:reverselinedirection ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rotera

Roterar objektgeometrier med den angivna vinkeln medurs. Rotationen sker runt varje objekts centroid, eller valfritt runt en unik förinställd punkt.

Tillåter *features in-place modification* av punkt-, linje- och polygonfeatures

Se även

Översätt, *Byt X- och Y-koordinater*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Rotation (grader medurs)	`ANGLE`	[numerisk: dubbel] Standard: 0,0	Rotationens vinkel i grader
Rotationsförankring (x, y) Valfritt	`ANCHOR`	[punkt] Standard: Ej inställd	X,Y-koordinater för den punkt som funktionerna ska roteras runt. Om den inte är inställd sker rotationen runt varje objekts centroid.
Roterad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret (med roterade geometrier). *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Roterad	`OUTPUT`	[samma som inmatning]	Utgångsvektorlagret med roterade geometrier

Python-kod

Algoritm-ID: `native:rotatefeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rundhet

Beräknar rundheten för varje funktion och lagrar den som ett nytt fält `roundness`. Ingångsvektorlagret måste innehålla polygoner.

Rundheten hos en polygon definieras som $4\pi \times$ polygonens area / omkrets^2. Rundhetsvärdet varierar mellan 0 och 1. En perfekt cirkel har en rundhet på 1, medan en helt platt polygon har en rundhet på 0.

Observera

Algoritmen returnerar NULL för flerdelade polygonfunktioner.

Tillåter *features in-place modification* av polygonfeatures

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: polygon]	Ingångsvektorlager
Rundhet	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange utmatningsvektorlagret (med rundhetsfält). *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Rundhet	`OUTPUT`	[samma som inmatning]	The output vector layer with `roundness` value in a field

Python-kod

Algoritm-ID: `native:roundness`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Segmentiera med maximal vinkel

Segmentiserar en geometri genom att omvandla krökta sektioner till linjära sektioner.

Segmentiseringen utförs genom att ange den maximalt tillåtna radievinkeln mellan hörn på den rätade geometrin (t.ex. vinkeln på den båge som skapas från den ursprungliga bågens mittpunkt till på varandra följande utgångshörn på den linjäriserade geometrin). Icke-krökta geometrier kommer att behållas utan ändring.

Se även

Segmentiera efter maximalt avstånd, *Förenkla*, *Smidig*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje, polygon]	Vektorlager för linje- eller polygoninmatning
Maximal vinkel mellan topparna (grader)	`ANGLE`	[numerisk: dubbel] Standard: 5,0	Maximal tillåten radievinkel mellan hörn på den uträtade geometrin
Segmentiserad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret (med segmentiserade geometrier). *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Segmentiserad	`OUTPUT`	[samma som inmatning]	Det utgående vektorlagret med segmentiserade geometrier

Python-kod

Algoritm-ID: `native:segmentizebymaxangle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Segmentiera efter maximalt avstånd

Segmentiserar en geometri genom att omvandla krökta sektioner till linjära sektioner.

Segmentiseringen utförs genom att ange det maximalt tillåtna offsetavståndet mellan originalkurvan och den segmentiserade representationen. Icke-krökta geometrier behålls utan förändring.

Se även

Segmentiera med maximal vinkel, *Förenkla*, *Smidig*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje, polygon]	Vektorlager för linje- eller polygoninmatning
Maximalt offsetavstånd	`DISTANCE`	[numerisk: dubbel] Standard: 1,0	Maximalt tillåtet offsetavstånd mellan originalkurvan och den segmentiserade representationen, i lagerenheter.
Segmentiserad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret (med segmentiserade geometrier). *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Segmentiserad	`OUTPUT`	[samma som inmatning]	Det utgående vektorlagret med segmentiserade geometrier

Python-kod

Algoritm-ID: `native:segmentizebymaxdistance`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Ställ in M-värde

Ställer in M-värdet för geometrier i ett lager.

Om det redan finns M-värden i lagret skrivs de över med det nya värdet. Om inga M-värden finns kommer geometrin att uppgraderas så att den innehåller M-värden och det angivna värdet kommer att användas som det ursprungliga M-värdet för alla geometrier.

Tillåter *features in-place modification* av punkt-, linje- och polygonfunktioner med M aktiverat

Tips

Använd knappen

för att kontrollera det extra M-värdet: resultaten finns tillgängliga i dialogrutan *Identifiera resultat*.

Se även

Ställ in M-värde från raster, *Ställ in Z-värde*, *Släpp M/Z-värden*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
M Värde	`M_VALUE`	[numerisk: dubbel] Standard: 0,0	M-värde som ska tilldelas geometrierna
M Tillagd	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
M Tillagd	`OUTPUT`	[samma som inmatning]	Utgångsvektorlagret (med M-värden som tilldelats geometrierna)

Python-kod

Algoritm-ID: `native:setmvalue`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Ställ in M-värde från raster

Använder värden som samplas från ett band i ett rasterlager för att ställa in M-värdet för varje överlappande vertex i objektets geometri. Rastervärdena kan eventuellt skalas med en förinställd mängd.

Om det redan finns M-värden i lagret skrivs de över med det nya värdet. Om det inte finns några M-värden kommer geometrin att uppgraderas för att inkludera M-värden.

Tillåter *features in-place modification* av punkt-, linje- och polygonfunktioner med M aktiverat

Se även

Drape (ställ in Z-värde från raster), *Ställ in M-värde*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Ett rasterlager	`RASTER`	[raster]	Rasterlager med M-värden
Bandnummer	`BAND`	[rasterband] Standard: 1	Det rasterband från vilket M-värdena hämtas
Värde för NoData eller icke korsande hörn	`NODATA`	[numerisk: dubbel] Standard: 0,0	Värde att använda om vertexen inte korsar (en giltig pixel i) rastret
Skalfaktor	`SCALE`	[numerisk: dubbel] Standard: 1,0	Skalningsvärde: bandvärdena multipliceras med detta värde.
Offset	`OFFSET`	[numerisk: dubbel] Standard: 0,0	Offset-värde: det läggs algebraiskt till bandvärdena efter tillämpning av ”Scale factor”.
Uppdaterad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret (med uppdaterade M-värden). *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Uppdaterad	`OUTPUT`	[samma som inmatning]	Utgångsvektorlagret (med uppdaterade M-värden)

Python-kod

Algoritm-ID: `native:setmfromraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Ställ in Z-värde

Ställer in Z-värdet för geometrier i ett lager.

Om det redan finns Z-värden i lagret skrivs de över med det nya värdet. Om inga Z-värden finns kommer geometrin att uppgraderas så att Z-värden inkluderas och det angivna värdet används som det initiala Z-värdet för alla geometrier.

Tillåter *features in-place modification* av punkt-, linje- och polygonfeatures med Z aktiverat

Tips

Använd knappen

för att kontrollera det extra Z-värdet: resultaten finns tillgängliga i dialogrutan *Identifiera resultat*.

Se även

Drape (ställ in Z-värde från raster), *Ställ in M-värde*, *Släpp M/Z-värden*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Z-värde	`Z_VALUE`	[numerisk: dubbel] Standard: 0,0	Z-värde som ska tilldelas geometrierna för objekten
Z Tillagd	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Z Tillagd	`OUTPUT`	[samma som inmatning]	Utgångsvektorlagret (med tilldelade Z-värden)

Python-kod

Algoritm-ID: `native:setzvalue`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Förenkla

Förenklar geometrierna i ett linje- eller polygonlager. Det skapar ett nytt lager med samma funktioner som i inmatningslagret, men med geometrier som innehåller ett lägre antal hörn.

Algoritmen ger ett urval av förenklingsmetoder, inklusive avståndsbaserad (Douglas-Peucker-algoritmen), områdesbaserad (Visvalingam-algoritmen) och snäppning av geometrier till rutnätet.

Fig. 24.105: Från vänster till höger, källager och ökande förenklingstoleranser

Tillåter *features in-place modification* av linje- och polygonfeatures

Standardmeny: *Vektor* ► *Geometriverktyg*

Se även

Förenkla täckningen, *Smidig*, *Förtätning efter antal*, *Förtätning med intervall*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje, polygon]	Vektorlager för linje- eller polygoninmatning
Förenklingsmetod	`METHOD`	[uppräkning] Standard: 0	Förenklingsmetod. En av: • 0 — Avstånd (Douglas-Peucker) • 1 — Snäpp till rutnätet • 2 — Område (Visvalingam)
Tolerans	`TOLERANCE`	[numerisk: dubbel] Standard: 1,0	Tröskeltolerans (i lagrets enheter): om avståndet mellan två noder är mindre än toleransvärdet förenklas segmentet och topparna tas bort.
Förenklat	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående (förenklade) vektorlagret. *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Förenklat	`OUTPUT`	[samma som inmatning]	Det utgående (förenklade) vektorlagret

Python-kod

Algoritm-ID: `native:simplifygeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Enkelsidig buffert

Beräknar en buffert på linjer med ett angivet avstånd på endast en sida av linjen.

Buffert resulterar alltid i ett polygonlager.

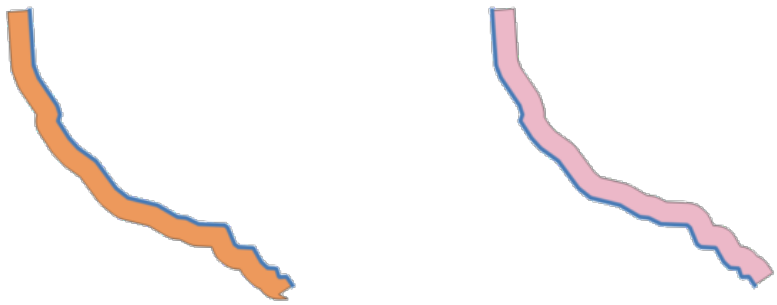


Fig. 24.106: Vänster- och högersidig buffert på samma vektorlinjelager

Se även

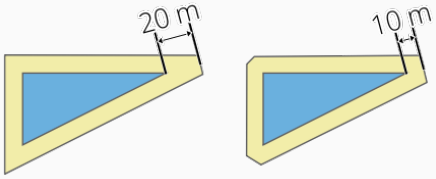
Buffra

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje]	Ingångslinje vektorlager

continues on next page

Table 24.178 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Avstånd inom vilket ett objekt kommer att snäppa till dina placerade stödlinjer. Efter att du har ställt in detta måste du starta om Scribus för att ställa in denna inställning.	`DISTANCE`	[numerisk: dubbel] Standard: 10.0	Buffertavstånd.
Sida	`SIDE`	[uppräkning] Standard: 0	Vilken sida som bufferten ska skapas på. En av: • 0 – Vänster • 1 – Höger
Segment	`SEGMENTS`	[numerisk: heltal] Standard: 8	Styr antalet linjesegment som ska användas för att approximera en kvartscirkel när man skapar avrundade offsets.
Join style	`JOIN_STYLE`	[uppräkning] Standard: 0	Anger om runda, geringade eller avfasade skarvar ska användas vid förskjutning av hörn i en linje. Alternativen är: • 0 — Omgång • 1 — Miter • 2 — Avfasning Fig. 24.107: Runda, geringade och fasade fogningsstilar
Mitergräns	`MITER_LIMIT`	[numerisk: dubbel] Standard: 2,0	Ställer in det maximala avståndet från offsetgeometrin som ska användas när man skapar en geringad fog som en faktor av offsetavståndet (gäller endast för geringade fogstilar). Minsta värde: 1.0  Fig. 24.108: En 10 m buffert med en gräns på 2 och en 10 m buffert med en gräns på 1
Buffert	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret (buffert). *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Buffert	`OUTPUT`	[vektor: polygon]	Utgående (buffert) polygonlager

Python-kod

Algoritm-ID: `native:singlesidedbuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Smidig

Utjämnar geometrierna i ett linje- eller polygonlager genom att lägga till fler **vertikaler och hörn** till geometrierna.

Parametern iterationer anger hur många utjämningsiterationer som ska tillämpas på varje geometri. Ett högre antal iterationer resulterar i jämnare geometrier, men till priset av ett större antal noder i geometrierna.

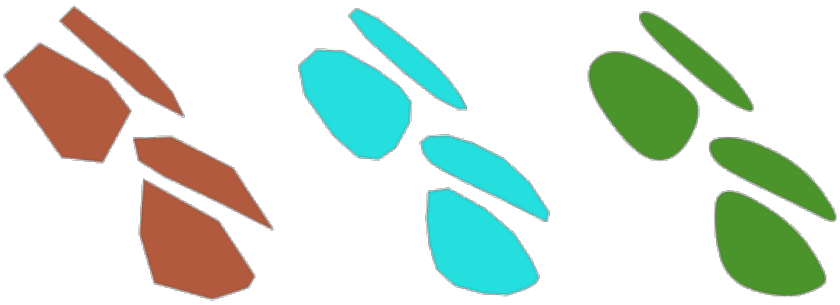


Fig. 24.109: Ökat antal iterationer ger jämnare geometrier

Offset-parametern styr hur ”tätt” de utjämnade geometrierna följer originalgeometrierna. Mindre värden ger en snävare passform och större värden ger en lösare passform.

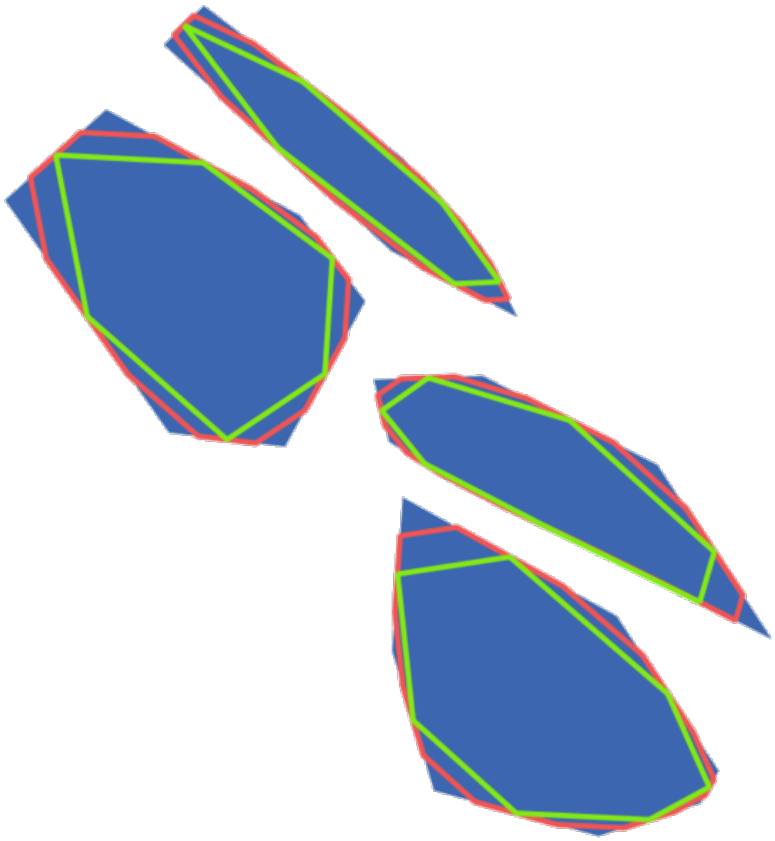


Fig. 24.110: Blå: inmatningslagret. Offset 0,25 ger den röda linjen, medan offset 0,50 ger den gröna linjen.

Parametern för maximal vinkel kan användas för att förhindra utjämning av noder med stora vinklar. Alla noder där vinkeln mellan segmenten på vardera sidan är större än detta kommer inte att utjämnas. Om du t.ex. ställer in den maximala vinkeln till 90 grader eller lägre bevaras räta vinklar i geometrin.

☑ Tillåter *features in-place modification* av linje- och polygonfeatures

Se även

Förenkla, *Förenkla täckningen*, *Förtätning efter antal*, *Förtätning med intervall*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje, polygon]	Vektorlager för linje- eller polygoninmatning
Iterationer	`ITERATIONS`	[numerisk: heltal] Standard: 1	Om du ökar antalet iterationer får du jämnare geometrier (och fler hörn).
Offset	`OFFSET`	[numerisk: dubbel] Standard: 0,25	Ökande värden kommer att *flytta* de utjämnade linjerna/gränserna längre bort från de inmatade linjerna/gränserna.

continues on next page

Table 24.179 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Maximal nodvinkel för att släta ut	`MAX_ANGLE`	[numerisk: dubbel] Standard: 180,0	Varje nod under detta värde kommer att jämnas ut
Smoothed	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående (utjämnade) lagret. *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Smoothed	`OUTPUT`	[samma som inmatning]	Utgående (utjämnat) vektorlager

Python-kod

Algoritm-ID: `native:jämngeometri`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Fäst geometrier till lager

Snäpper geometrierna i ett lager antingen till geometrierna från ett annat lager eller till geometrier inom samma lager.

Matchningen görs baserat på ett toleransavstånd och hörnpunkter infogas eller tas bort efter behov för att geometrierna ska stämma överens med referensgeometrierna.

Tillåter *features in-place modification* av punkt-, linje- och polygonfeatures

Se även

Fäst punkter i rutnätet

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	INPUT	[vektor: geometri]	Ingångsvektorlager
Reference layer	REFERENCE_LAYE	[vektor: geometri]	Vektorlager att snäppa till
Tolerans	TOLERANCE	[numerisk: dubbel] Standard: 10.0	Kontrollera hur nära ingångsvertices måste vara referenslagrets geometrier innan de snäppas.

continues on next page

Table 24.180 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Beteende	BEHAVIOR	[uppräkning] Standard: 0	Snapping kan göras till en befintlig nod eller ett segment (dess närmaste punkt till den vertex som ska flyttas). Tillgängliga snapping-alternativ: • 0 — Företräcker att noderna är i linje, infoga extra hörn där det behövs Snäpp helst till noder, även när ett segment kan vara närmare än en nod. Nya noder kommer att infogas för att få geometrierna att följa varandra exakt inom tillåtna toleranser. • 1 — Föredrar närmaste punkt, infogar extra hörn där så krävs Snäpp till närmaste punkt, oavsett om det är en nod eller ett segment. Nya noder kommer att infogas för att få geometrierna att följa varandra exakt inom tillåtna toleranser. • 2 — Företräcker att rikta in noder, infoga inte nya hörn Företräcker att fästa vid noder, även om ett segment kan vara närmare än en nod. Inga nya noder kommer att infogas. • 3 — Föredrar närmaste punkt, infogar inte nya hörn Snäpp till närmaste punkt, oavsett om det är en nod eller ett segment. Inga nya noder kommer att infogas. • 4 — Flytta endast slutpunkter, föredrar att rikta in noder Snäpp bara start- och slutpunkter för linjer (punktfunktioner snäppas också, polygonfunktioner ändras inte), föredrar att snäppa till noder. • 5 — Flytta endast slutpunkter, välj närmaste punkt Snäpp bara start- och slutpunkter för linjer (punktfunktioner snäppas också, polygonfunktioner ändras inte), snäpp till närmaste punkt • 6 — Snäpp ihop ändpunkter till enbart ändpunkter Snäpp bara linjernas start-/slutpunkter mot andra linjers start-/slutpunkter • 7 — Snap till ankarnoder (endast ett lager)

continues on next page

Table 24.180 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Snapped geometry	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående (snäppta) lagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Snapped geometry	`OUTPUT`	[samma som inmatning]	Utgång (snappad) vektorlager

Python-kod

Algoritm-ID: `` `native:snapgeometrier` ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Fäst punkter i rutnätet

Ändrar koordinaterna för geometrier i ett vektorlager, så att alla punkter eller hörn snappas till den närmaste punkten i ett rutnät.

Om den snäppta geometrin inte kan beräknas (eller om den är helt kollapsad) kommer objektets geometri att rensas.

Snapping kan utföras på X-, Y-, Z- eller M-axeln. Ett rutnätsavstånd på 0 för någon axel inaktiverar snäppning för den axeln.

☑ Tillåter *features in-place modification* av punkt-, linje- och polygonfeatures

Varning

Snapping till rutnätet kan generera en ogiltig geometri i vissa hörnfall.

Se även

Fäst geometrier till lager

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
*avstånd mellan rutnätet och *X**	`HSPACING`	[numerisk: dubbel] Standard: 1,0	Rasteravstånd på X-axeln
Avstånd mellan Y-rutorna	`VSPACING`	[numerisk: dubbel] Standard: 1,0	Rasteravstånd på Y-axeln
Z Avstånd mellan rutnät	`ZSPACING`	[numerisk: dubbel] Standard: 0,0	Rasteravstånd på Z-axeln
M Rasteravstånd	`MSPACING`	[numerisk: dubbel] Standard: 0,0	Rasteravstånd på M-axeln
Snapped	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående (snäppta) lagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Snapped	`OUTPUT`	[samma som inmatning]	Utgång (snappad) vektorlager

Python-kod

Algoritm-ID: `native:snappointstogrid`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Dela upp linjerna efter maximal längd

Tar ett linje- (eller kurv-) lager och delar upp varje funktion i flera delar, där varje del har en angiven maximal längd. Z- och M-värdena i början och slutet av de nya linjesubstraten interpoleras linjärt från befintliga värden.

Tillåter *funktioner på plats modifiering* av linjefunktioner

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje]	Det ingående linjevektorlagret
Maximal linjelängd	`LENGTH`	[numerisk: dubbel] Standard: 10.0	Den maximala längden på en rad i utdata.
Split	`OUTPUT`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange utdatans linjevektorlager. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Split	`OUTPUT`	[vektor: linje]	Det nya linjevektorlagret. Längden på funktionsgeometrierna är mindre än eller lika med den längd som anges i parametern LENGTH. Ett ytterligare fält som innehåller `order` kommer att läggas till.

Python-kod

Algoritm-ID: `native:splitlinesbylength`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Dela upp

Delar upp geometrin. Den returnerade geometrin kommer att vara en samling som innehåller indelade delar från den ursprungliga geometrin, där ingen del har mer än det angivna maximala antalet noder.

Detta är användbart för att dela upp en komplex geometri i mindre komplexa delar, lättare att spatialt indexera och snabbare att utföra spatiala operationer. Krökta geometrier kommer att segmentiseras före subdivision.

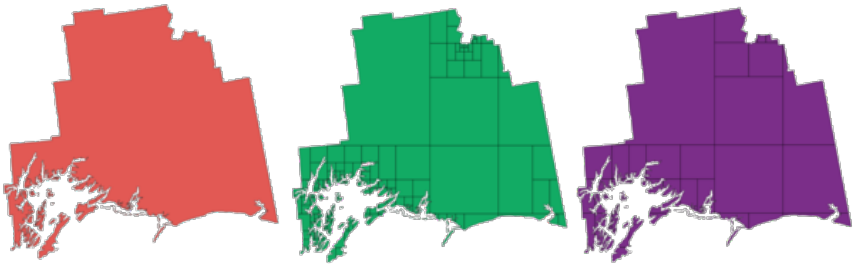


Fig. 24.111: Vänster inmatningslager, mitten maximalt nodvärde är 100 och höger maximalt värde är 200

Tillåter *features in-place modification* av punkt-, linje- och polygonfeatures

Observera

Att dela upp en geometri kan generera geometridelar som kanske inte är giltiga och som kan innehålla självskärningar.

Se även

Explodera linjer, *Linje delsträng*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlagret
Maximala noder i delar	`MAX_NODES`	[numerisk: heltal] Standard: 256	Maximalt antal hörnpunkter som varje ny geometridel får ha. Färre *sub-parts* för högre värden.
Subdividerat	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående (indelade) vektorlagret. *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Subdividerat	`OUTPUT`	[samma som inmatning]	Utgående vektorlager

Python-kod

Algoritm-ID: `native:underindelning`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Byt X- och Y-koordinater

Växlar X- och Y-koordinatvärden i inmatade geometrier.

Den kan användas för att reparera geometrier som av misstag har fått sina latitud- och longitudvärden omvända.

Tillåter *features in-place modification* av punkt-, linje- och polygonfeatures

Se även

Översätt, *Rotera*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlagret
Utbytt	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Utbytt	`OUTPUT`	[samma som inmatning]	Utgående (swappat) vektorlager

Python-kod

Algoritm-ID: `native:swapxy`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Avsmalnande buffertar

Skapar avsmalnande buffert längs linjegeometrier med hjälp av en angiven start- och slutdiameter för bufferten.

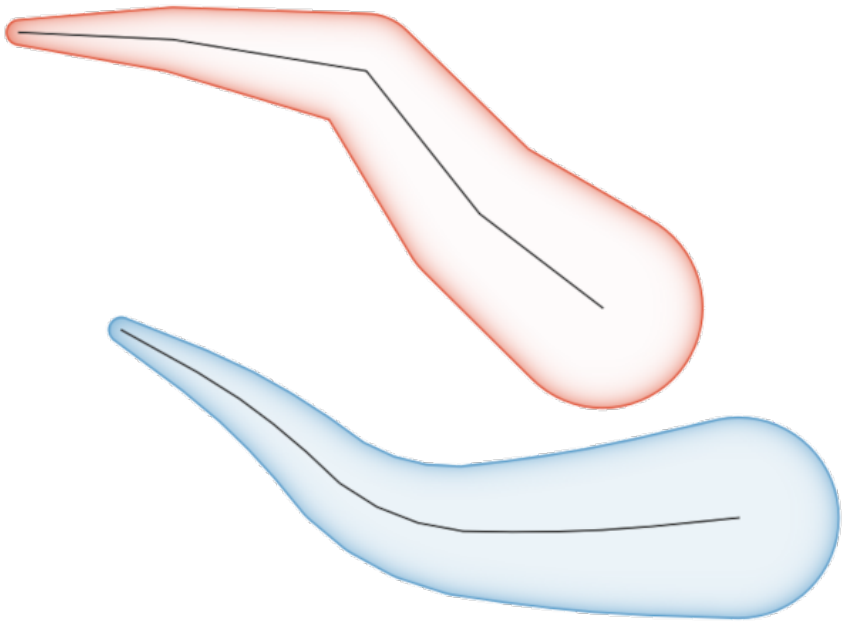


Fig. 24.112: Exempel på avsmalnande buffert

Se även

Buffert med variabel bredd (av M-värde), *Buffra*, *Skapa kilbuffertar*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje]	Ingångslinje vektorlager
Startbredd	`START_WIDTH`	[numerisk: dubbel] Standard: 0,0	Representerar radien för den buffert som används vid startpunkten för linjefunktionen
Slutbredd	`END_WIDTH`	[numerisk: dubbel] Standard: 0,0	Representerar radien för den buffert som appliceras vid slutpunkten för linjefunktionen.
Segment	`SEGMENTS`	[numerisk: heltal] Standard: 16	Styr antalet linjesegment som ska användas för att approximera en kvartscirkel när man skapar avrundade offsets.
Buffrad	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret (buffert). *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Buffrad	`OUTPUT`	[vektor: polygon]	Utgående (buffert) polygonlager

Python-kod

Algoritm-ID: `native:taperedbuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Tessellat

Tessellerar ett lager med polygongeometrier genom att dela upp geometrierna i triangulära komponenter.

Utmatningslagret består av multipolygongeometrier för varje inmatad funktion, där varje multipolygon består av flera polygoner med triangelkomponenter.

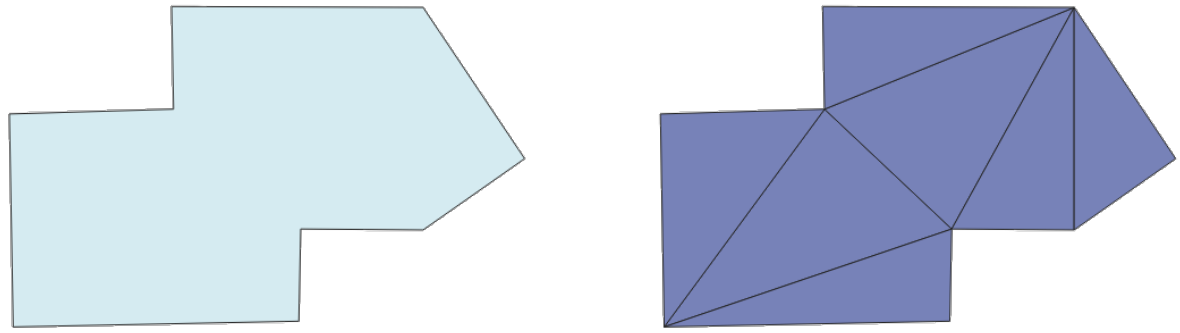


Fig. 24.113: Tessellerad polygon (höger)

Tillåter *features in-place modification* av polygonfeatures

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: polygon]	Vektorlager för inmatad polygon
Tesselated	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Tesselated	`OUTPUT`	[vektor: polygon]	Utmatning av multipolygonZ-lager

Python-kod

Algoritm ID: `3d:tessellate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Transekt

Skapar transekter på hörn för (multi)linjestring.

En transekt är en linje som är orienterad från en vinkel (som standard vinkelrät) mot de inmatade polylinjerna (vid hörnen).

Fält från objekt returneras i transekten med dessa nya fält:

- TR_FID: ID för den ursprungliga funktionen
- TR_ID: ID för transekten. Varje transekt har ett unikt ID
- TR_SEGMENT: ID för segmentet av linjesträngen
- TR_ANGLE: Vinkel i grader från den ursprungliga linjen vid toppunktet
- TR_LENGTH: Total längd på den returnerade transekten
- TR_ORIENT: Transektens sida (endast till vänster eller höger om linjen, eller båda sidorna)

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

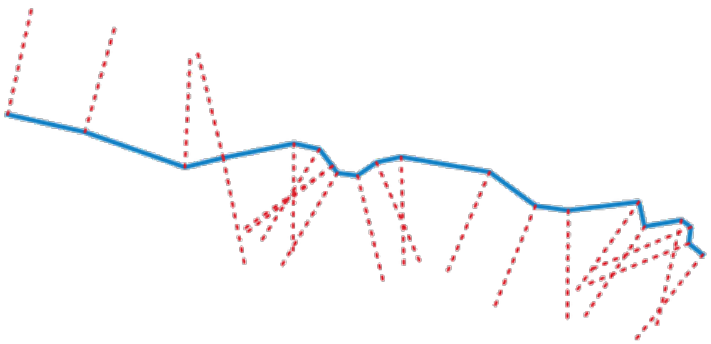


Fig. 24.114: Streckade röda linjer representerar transekten för det ingående linjelagret

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje]	Ingångslinje vektorlager
Transektens längd	`LENGTH`	[numerisk: dubbel] Standard: 5,0	Transektens längd i kartanhet
Vinkel i grader från den ursprungliga linjen vid hörnen	`ANGLE`	[numerisk: dubbel] Standard: 90,0	Ändra vinkeln på transekten

continues on next page

Table 24.183 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Sida för att skapa transekten	`SIDE`	[uppräkning]	Välj sida av transekten. Tillgängliga alternativ är: • 0 — Vänster • 1 — Höger • 2 — Båda
Transekt	`OUTPUT`	[vektor: linje] Standard: `[Skapa tillfälligt lager]`	Ange utmatningens linjelager. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Transekt	`OUTPUT`	[vektor: linje]	Utgående linje lager

Python-kod

Algoritm-ID: `original:transekt`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Översätt

Flyttar geometrierna inom ett lager genom att förskjuta dem med en fördefinierad X- och Y-förskjutning.

Z- och M-värden som finns i geometrin kan också översättas.

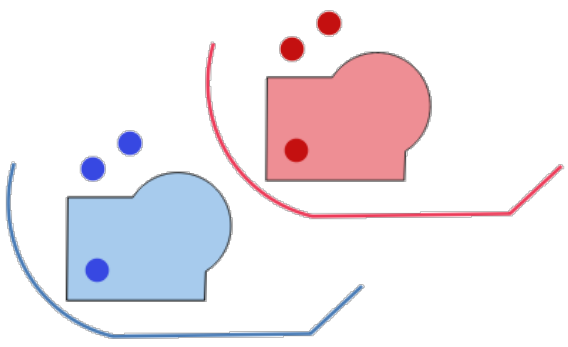


Fig. 24.115: Funktioner i rött representerar den översatta geometrin för inmatningslagret (blått)

Tillåter *features in-place modification* av punkt-, linje- och polygonfeatures

Se även

En rad översatta funktioner, *Offset-linjer*, *Rotera*, *Byt X- och Y-koordinater*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Förskjutet avstånd (x-axel)	`DELTA_X`	[numerisk: dubbel] Standard: 0,0	Förskjutning som ska tillämpas på X-axeln
Offset-avstånd (y-axeln)	`DELTA_Y`	[numerisk: dubbel] Standard: 0,0	Förskjutning som ska tillämpas på Y-axeln
Offset-avstånd (z-axel)	`DELTA_Z`	[numerisk: dubbel] Standard: 0,0	Förskjutning som ska tillämpas på Z-axeln
Offset-avstånd (m-värden)	`DELTA_M`	[numerisk: dubbel] Standard: 0,0	Förskjutning som ska appliceras på M-axeln
Översatt	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Översatt	`OUTPUT`	[samma som inmatning]	Utgående vektorlager

Python-kod

Algoritm-ID: `native:translategeometry`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Buffert med variabel bredd (av M-värde)

Skapar buffertar med variabel bredd längs linjerna och använder M-värdet för linjegeometrierna som buffertens diameter vid varje toppunkt.

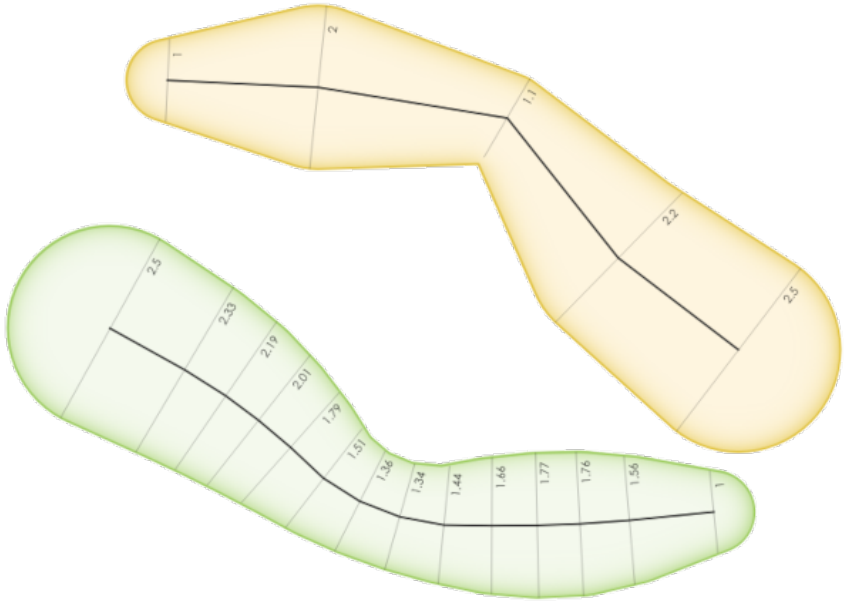


Fig. 24.116: Exempel på variabel buffert

Se även

Avsmalnande buffertar, *Buffra*, *Ställ in M-värde*, *Buffert med variabelt avstånd*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje]	Ingångslinje vektorlager
Segment	`SEGMENTS`	[numerisk: heltal] Standard: 16	Antal buffertsegment per kvartscirkel. Det kan vara ett unikt värde (samma värde för alla objekt), eller så kan det hämtas från objektdata (värdet kan bero på objektattribut).
Buffrad	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret (buffert). *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Buffrad	`OUTPUT`	[vektor: polygon]	Polygonlager med variabel buffert

Python-kod

Algoritm-ID: `native:bufferbym`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Voronoi-polygoner

Tar ett punktlager och genererar ett polygonlager som innehåller Voronoi-polygonerna (även kända som Thiessen-polygoner) som motsvarar dessa inmatningspunkter.

Varje plats inom en Voronoi-polygon är närmare den associerade punkten än någon annan punkt.

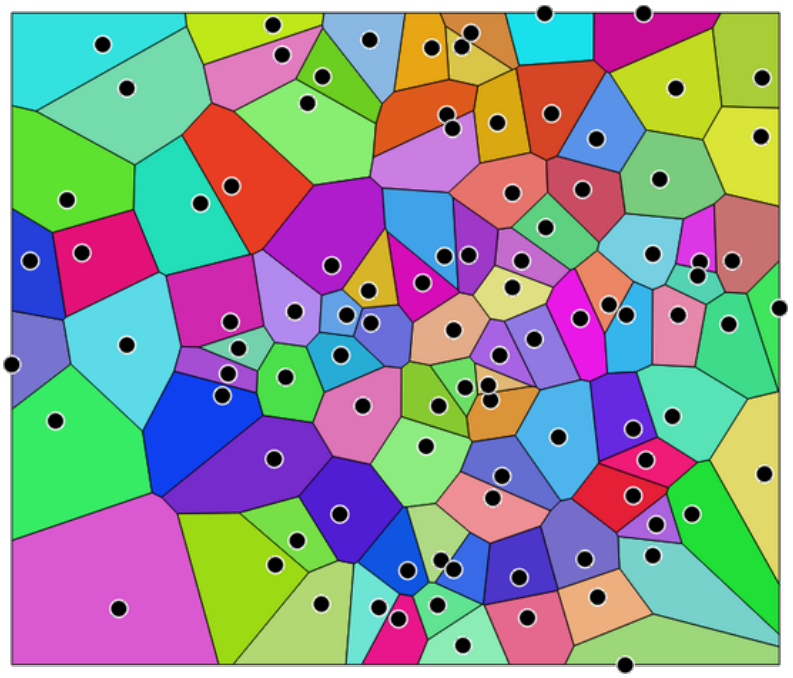


Fig. 24.117: Voronoi-polygoner

Standardmeny: *Vektor ► Geometriverktyg*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: punkt]	Vektorlager för inmatningspunkt
Buffertregion (% of utsträckning)	`BUFFER`	[numerisk: dubbel] Standard: 0,0	Utmatningslagrets omfattning kommer att vara så här mycket större än inmatningslagrets omfattning
Tolerans Valfritt	`TOLERANCE`	[numerisk: dubbel] Standard: 0,0	Anger en valfri snäpptolerans som kan användas för att förbättra robustheten hos voronoi.
Kopiera attribut från inmatade funktioner	`COPY_ATTRIBUTE`	[boolean] Standard: Sann	Anger om fält som lagrar ID för involverade punktfunktioner ska läggas till i utdata. Om False, skapas ett fält med ”id” för att identifiera polygonerna.
Voronoi-polygoner	`OUTPUT`	[vektor: polygon] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret (med Voronoi-polygonerna). *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Voronoi-polygoner	`OUTPUT`	[vektor: polygon]	Voronoi-polygoner för inmatningspunktens vektorlager

Python-kod

Algoritm-ID: `native:voronoipolygoner`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.24 Vektoröverlägg

Klipp

Klipper ett vektorlager med hjälp av funktionerna i ett extra polygonlager.

Endast de delar av objekten i inmatningslagret som faller inom polygonerna i överlagringslagret läggs till i det resulterande lagret.

Varning

Endast geometriändring

Denna operation modifierar endast objektens geometri. Attributvärdena för objekten **ändras inte**, även om egenskaper som area eller längd för objekten kommer att ändras av overlay-operationen. Om sådana egenskaper är lagrade som attribut måste dessa uppdateras manuellt.

Den här algoritmen använder spatiala index på datakällorna, förberedda geometrier och tillämpar en klippningsoperation om geometrin inte är helt innesluten i maskgeometrin.

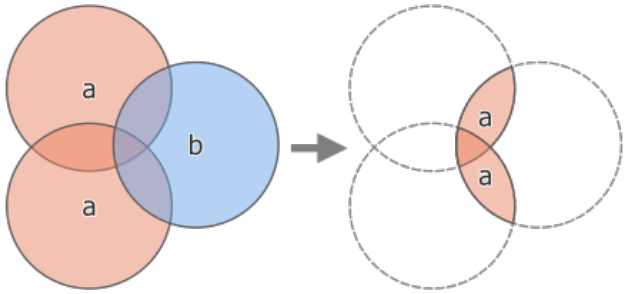


Fig. 24.118: Klippning mellan ett inmatningslager med två funktioner ”a” och ett överlagringslager med en funktion ”b” (vänster) - resulterar i ett nytt lager med de modifierade funktionerna från ”a” (höger)

Tillåter *features in-place modification* av punkt-, linje- och polygonfunktioner

Standardmeny: *Vektor* ► *Geobearbetningsverktyg*

Se även

Korsningar, *Skillnad*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Lager som innehåller de funktioner som ska klippas
Överläggslager	`OVERLAY`	[vektor: polygon]	Lager som innehåller klippfunktionerna
Klippt	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det lager som ska innehålla de funktioner från inmatningslagret som finns i överlagringslagret (klippning). *One of*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Klippt	`OUTPUT`	[samma som inmatning]	Lager som innehåller funktioner från inmatningslagret som delats av överlagringslagret.

Python-kod

Algoritm ID: `qgis:clip`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skillnad

Extraherar funktioner från inmatningslagret som inte faller inom gränserna för överlagringslagret.

Inmatningslagerfunktioner som delvis överlappar överlagringslagerfunktionerna delas längs gränsen för dessa funktioner och endast de delar som ligger utanför överlagringslagerfunktionerna behålls.

Varning

Endast geometriändring

Denna operation modifierar endast objektens geometri. Attributvärdena för objekten **ändras inte**, även om egenskaper som area eller längd för objekten kommer att ändras av overlay-operationen. Om sådana egenskaper är lagrade som attribut måste dessa uppdateras manuellt.

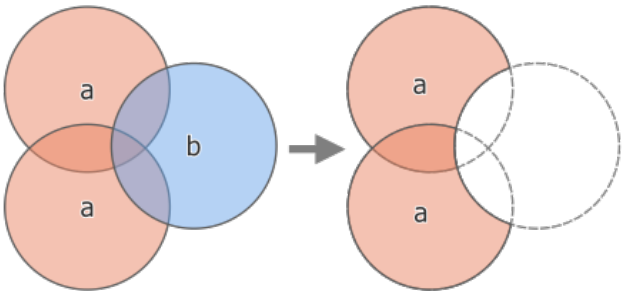


Fig. 24.119: Skillnadsoperation mellan ett inmatningslager med två funktioner ”a” och ett överlagringslager med en funktion ”b” (vänster) - resulterar i ett nytt lager med de modifierade funktionerna från ”a” (höger)

Tillåter *features in-place modification* av punkt-, linje- och polygonfunktioner

Standardmeny: *Vektor* ► *Geobearbetningsverktyg*

Se även

Skillnad (flera), *Symmetrisk skillnad*, *Klipp*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Lager att extrahera (delar av) funktioner från.
Överläggslager	`OVERLAY`	[vektor: geometri]	Lager som innehåller de geometrier som ska subtraheras från geometrierna i inmatningslagret. Det förväntas ha minst lika många dimensioner (punkt: 0D, linje: 1D, polygon: 2D, volym: 3D) som geometrierna i inmatningslagret.
Skillnad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det lager som ska innehålla de (delar av) funktioner från inmatningslagret som inte finns i överlagringslagret. *One of*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Rutnätsstorlek Valfritt	`GRID_SIZE`	[numerisk: dubbel] Standard: Ej inställd	Om det anges, snappas indatageometrierna till ett rutnät med den angivna storleken och resultatets hörn beräknas på samma rutnät. Kräver GEOS 3.9.0 eller senare.

Utgångar

Etikett	Namn	Typ	Beskrivning
Skillnad	`OUTPUT`	[samma som inmatning]	Lager som innehåller (delar av) funktioner från inmatningslagret som inte överlappar överlagringslagret.

Python-kod

Algoritm-ID: `qgis:difference`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skillnad (flera)

Extraherar funktioner från inmatningslagret som ligger helt utanför eller endast delvis överlappar funktionerna från något av överlagringslagren.

För varje överlagringslager beräknas skillnaden mellan resultatet av alla tidigare differensoperationer och detta överlagringslager. Ingångslagrets funktioner som delvis överlappar funktioner i överlagringslagret delas längs dessa funktioners gräns och endast de delar som ligger utanför överlagringslagrets funktioner behålls.

Varning

Endast geometriändring

Denna operation modifierar endast objektens geometri. Attributvärdena för objekten **ändras inte**, även om egenskaper som area eller längd för objekten kommer att ändras av overlay-operationen. Om sådana egenskaper är lagrade som attribut måste dessa uppdateras manuellt.

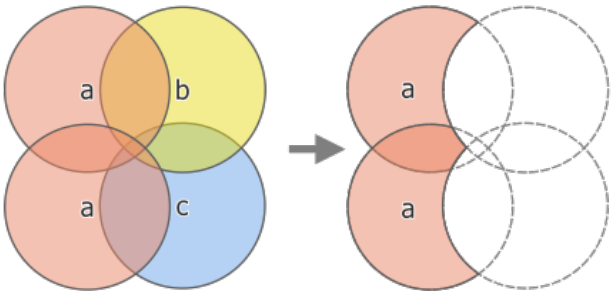


Fig. 24.120: Skillnadsoperation mellan ett inmatningslager med två funktioner ”a” och överlagringslager med en funktion ”b” och ”c” (vänster) - resulterar i ett nytt lager med de modifierade funktionerna ”a” (höger)

Se även

Skillnad, *Symmetrisk skillnad*, *Klipp*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Lager att extrahera (delar av) funktioner från.
Överläggslager	`OVERLAYS`	[vektor: geometri] [lista]	Lista över lager som innehåller de geometrier som ska subtraheras från geometrierna i indatalagret. De förväntas ha minst lika många dimensioner (punkt: 0D, linje: 1D, polygon: 2D, volym: 3D) som geometrierna i inmatningslagret.
Skillnad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det lager som ska innehålla de (delar av) funktioner från inmatningslagret som inte överlappar funktioner i överlagringslagren. *One of*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Skillnad	`OUTPUT`	[samma som inmatning]	Lager som innehåller (delar av) funktioner från inmatningslagret som inte överlappar funktioner från överlagringslagren.

Python-kod

Algoritm-ID: `qgis:multidifference`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Extrahera/klippa efter omfattning

Skapar ett nytt vektorlager som endast innehåller objekt som faller inom en angiven utsträckning. Alla objekt som korsar utbredningen inkluderas som standard.

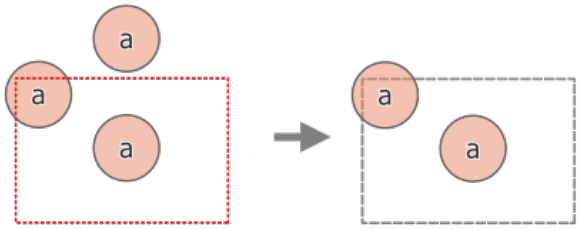


Fig. 24.121: Extraheringsoperation mellan ett inmatningslager med tre funktioner ”a” och en streckad utsträckning (vänster) - resulterande funktioner med streckad utsträckning som referens (höger)

Eventuellt kan geometrier för objekten också klippas till omfattningen.

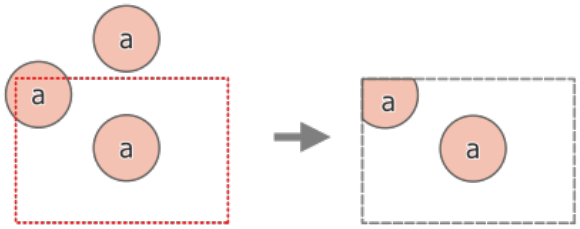


Fig. 24.122: Extraheringsoperation mellan ett inmatningslager med tre funktioner ”a” och en streckad utsträckning (vänster) - resulterande funktioner med streckad utsträckning som referens (höger)

Se även

Klipp

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Lager att extrahera (delar av) funktioner från.
Extent (xmin, xmax, ymin, ymax)	`EXTENT`	[omfattning]	Utsträckning för klippning. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Klipp funktioner till omfattning	`CLIP`	[boolean] Standard: Falsk	Om detta markeras kommer geometrierna att klippas i den utsträckning som valts istället för att hela geometrin tas med i utdata. Dessutom konverteras utdatageometrierna automatiskt till multigeometrier för att säkerställa enhetliga utdatatyper.
Extraherad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det lager som ska innehålla de funktioner från inmatningslagret som ligger inom klippets utsträckning. *One of*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Extraherad	`OUTPUT`	[samma som inmatning]	Lager som innehåller de klippta objekten.

Python-kod

Algoritm-ID: `native:extraherabyextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Korsningar

Extraherar de delar av funktioner från inmatningslagret som överlappar funktioner i överlagringslagret.

Objekten i intersektionslagret tilldelas attributen för de överlappande objekten från både inmatnings- och överlagringslagret.

Varning

Endast geometriändring

Denna operation modifierar endast objektens geometri. Attributvärdena för objekten **ändras inte**, även om egenskaper som area eller längd för objekten kommer att ändras av overlay-operationen. Om sådana egenskaper är lagrade som attribut måste dessa uppdateras manuellt.

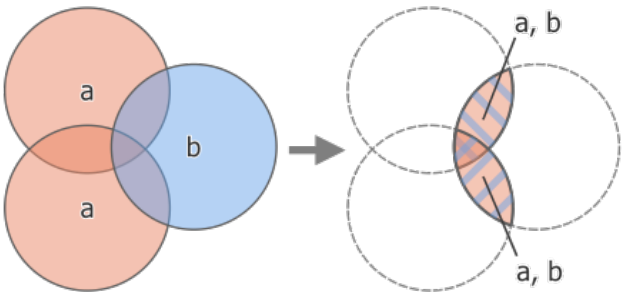


Fig. 24.123: Intersektion mellan ett inmatningslager med två funktioner ”a” och ett överlagringslager med en funktion ”b” (vänster) - överlappande områden blir ett nytt lager med två funktioner med båda lagrens attribut (höger)

Standardmeny: *Vektor ► Geobearbetningsverktyg*

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Korsning (flera), *Klipp*, *Skillnad*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Lager att extrahera (delar av) funktioner från.
Överläggslager	`OVERLAY`	[vektor: geometri]	Lager som innehåller de funktioner som ska kontrolleras för överlappning. Dess geometri förväntas ha minst lika många dimensioner (punkt: 0D, linje: 1D, polygon: 2D, volym: 3D) som inputlagrets.
Inmatningsfält som ska behållas (lämna tomt för att behålla alla fält) Valfritt	`INPUT_FIELDS`	[tabellfält: valfritt] [lista] Standard: Ej inställd	Inmatningslagrets fält som ska behållas i utmatningen. Om inga fält väljs tas alla fält med.
Overlay-fält som ska behållas (lämna tomt för att behålla alla fält) Valfritt	`OVERLAY_FIELDS`	[tabellfält: valfritt] [lista] Standard: Ej inställd	Fält för det överliggande lagret som ska behållas i utdata. Om inga fält väljs tas alla fält med. Duplicerade fältnamn kommer att få ett räknesuffix för att undvika kollision.
Snitt	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det lager som ska innehålla (de delar av) funktionerna från inmatningslagret som överlappar en eller flera funktioner från överlagringslagret. *One of*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Prefix för överlagringsfält Valfritt	`OVERLAY_FIELDS_`	[sträng]	Lägg till ett prefix för att identifiera fält i överlagringslagret. Duplicerade fältnamn kommer att få ett räknesuffix för att undvika kollisioner.
Rutnätsstorlek Valfritt	`GRID_SIZE`	[numerisk: dubbel] Standard: Ej inställd	Om det anges, snappas indatageometrierna till ett rutnät med den angivna storleken och resultatets hörn beräknas på samma rutnät. Kräver GEOS 3.9.0 eller senare.

Utgångar

Etikett	Namn	Typ	Beskrivning
Snitt	`OUTPUT`	[samma som inmatning]	Lager som innehåller (delar av) funktioner från inmatningslagret som överlappar överlagringslagret.

Python-kod

Algoritm-ID: `qgis:intersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Korsning (flera)

Extraherar de överlappande delarna av funktioner i indata och alla överliggande lager.

Funktionerna i utdatalagret tilldelas attributen för de överlappande funktionerna från både inmatnings- och överlagringslagret.

Varning

Endast geometriändring

Denna operation modifierar endast objektens geometri. Attributvärdena för objekten **ändras inte**, även om egenskaper som area eller längd för objekten kommer att ändras av overlay-operationen. Om sådana egenskaper är lagrade som attribut måste dessa uppdateras manuellt.

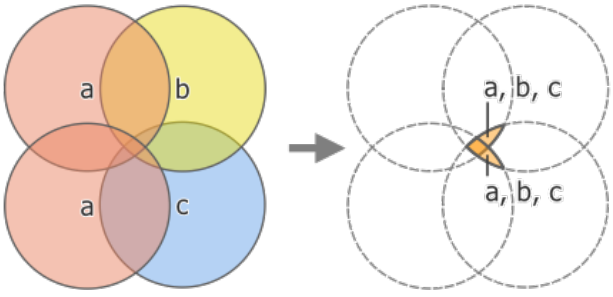


Fig. 24.124: Skärningsoperation mellan ett inmatningslager med två funktioner ”a” och överlagringslager med en funktion ”b” och ”c” (vänster) - överlappande områden blir ett nytt lager med två funktioner med alla lagrens attribut (höger)

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Korsningar, *Klipp*, *Skillnad*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Lager att extrahera (delar av) funktioner från.
Överläggslager	`OVERLAYS`	[vektor: geometri] [lista]	Lager som innehåller de funktioner som ska kontrolleras för överlappning. Objektens geometri förväntas ha minst lika många dimensioner (punkt: 0D, linje: 1D, polygon: 2D, volym: 3D) som inmatningslagrets.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Prefix för överlagringsfält Valfritt	`OVERLAY_FIELDS_`	[sträng]	Lägg till ett prefix för att identifiera fält i överlagringslagren. Duplicerade fältnamn kommer att få ett räknesuffix för att undvika kollisioner.

Utgångar

Etikett	Namn	Typ	Beskrivning
Snitt	`OUTPUT`	[samma som inmatning]	Lager som innehåller (delar av) funktioner från inmatningslagret som överlappar alla överlagringslager.

Python-kod

Algoritm ID: `qgis:multiintersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Korsningar mellan linjer

Skapar punktfunktioner där linjerna från de två lagren korsar varandra.

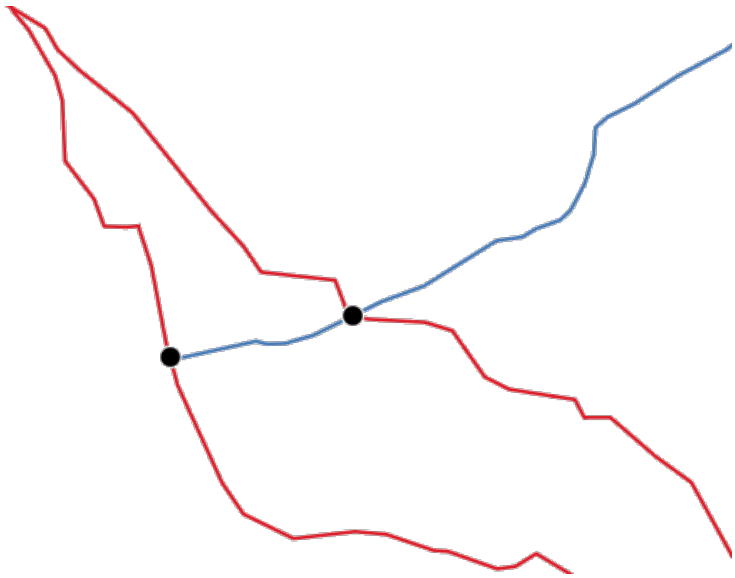


Fig. 24.125: Skärningspunkter

Standardmeny: *Vektor* ► *Analysverktyg*

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje]	Inmatningslinje lager.
Snittlager	`INTERSECT`	[vektor: linje]	Lager att använda för att hitta linjers skärningspunkter.
Inmatningsfält som ska behållas (lämna tomt för att behålla alla fält) Valfritt	`INPUT_FIELDS`	[tabellfält: valfritt] [lista] Standard: Ej inställd	Inmatningslagrets fält som ska behållas i utmatningen. Om inga fält väljs tas alla fält med.
Intersektera fält som ska behållas (lämna tomt om alla fält ska behållas) Valfritt	`INTERSECT_FIEL`	[tabellfält: valfritt] [lista] Standard: Ej inställd	Fält i intersect-lagret som ska behållas i utdata. Om inga fält väljs tas alla fält med. Duplicerade fältnamn kommer att få ett räknesuffix för att undvika kollision.

continues on next page

Table 24.193 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Snitt	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange det lager som ska innehålla skärningspunkterna för linjerna från indata- och överlagringslagren. *One of*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Intersektera fält prefix Valfritt	`INTERSECT_FIEL`	[sträng]	Lägg till ett prefix för att identifiera fält i intersect-lagret.

Utgångar

Etikett	Namn	Typ	Beskrivning
Snitt	`OUTPUT`	[vektor: punkt]	Punktvektorlager av linjernas skärningspunkter, med båda lagrens attribut.

Python-kod

Algoritm-ID: `qgis:linjeskärningar`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Delad med linjer

Delar linjerna eller polygonerna i ett lager med hjälp av linjerna eller polygonringarna i ett annat lager för att definiera brytpunkterna. Skärningspunkter mellan geometrier i båda lagren betraktas som delningspunkter.

Utdata kommer att innehålla flera geometrier för delade funktioner.

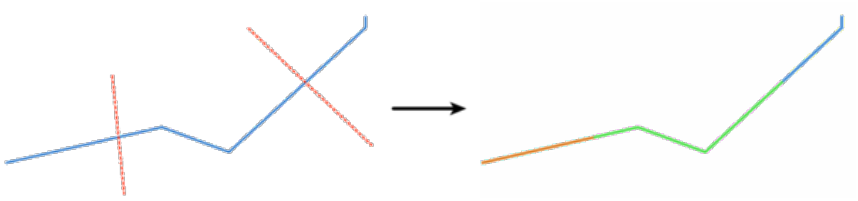


Fig. 24.126: Delade linjer

Tillåter *features in-place modification* av linje- och polygonfeatures

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje, polygon]	Lager som innehåller de linjer eller polygoner som ska delas.
Delat lager	`LINES`	[vektor: linje, polygon]	Lager vars linjer eller ringar används för att definiera brytpunkterna.
Dela	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det lager som ska innehålla de delade (om de korsas av en linje i det delade lagret) linje-/polygonfunktionerna från inmatningslagret. *One of*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Dela	`OUTPUT`	[samma som inmatning]	Utgående vektorlager med delade linjer eller polygoner från inmatningslagret.

Python-kod

Algoritm ID: `qgis:splitwithlines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Symmetrisk skillnad

Skapar ett lager som innehåller funktioner från både indata- och överlagringslagret, men där de överlappande områdena mellan de två lagren har tagits bort.

Attributtabellen för lagret med symmetrisk skillnad innehåller attribut och fält från både inmatnings- och överlagringslagret.

Varning

Endast geometriändring

Denna operation modifierar endast objektens geometri. Attributvärdena för objekten **ändras inte**, även om egenskaper som area eller längd för objekten kommer att ändras av overlay-operationen. Om sådana egenskaper är lagrade som attribut måste dessa uppdateras manuellt.

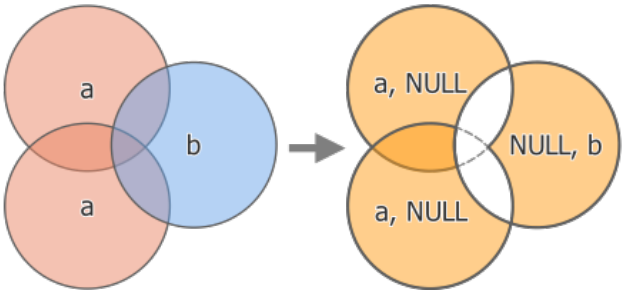


Fig. 24.127: Symmetrisk differensoperation mellan ett inmatningslager med två funktioner ”a” och ett överlagringslager med en funktion ”b” (vänster) - resulterande lager med tre funktioner med båda lagrens attribut (höger)

Standardmeny: *Vektor ► Geobearbetningsverktyg*

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Skillnad, *Klipp*, *Korsningar*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Första lagret att extrahera (delar av) funktioner från.
Överläggslager	`OVERLAY`	[vektor: geometri]	Andra lagret att extrahera (delar av) funktioner från. Helst bör geometritypen vara densamma som i inmatningslagret.
Symmetrisk skillnad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det lager som ska innehålla (de delar av) funktionerna från indata- och överlagringslagren som inte överlappar funktioner från det andra lagret. *One of*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Prefix för överlagringsfält Valfritt	`OVERLAY_FIELDS_`	[sträng]	Lägg till ett prefix för att identifiera fält i överlagringslagret. Duplicerade fältnamn kommer att få ett räknesuffix för att undvika kollisioner.
Rutnätsstorlek Valfritt	`GRID_SIZE`	[numerisk: dubbel] Standard: Ej inställd	Om det anges, snappas indatageometrierna till ett rutnät med den angivna storleken och resultatets hörn beräknas på samma rutnät. Kräver GEOS 3.9.0 eller senare.

Utgångar

Etikett	Namn	Typ	Beskrivning
Symmetrisk skillnad	`OUTPUT`	[samma som inmatning]	Lager som innehåller (delar av) egenskaper från varje lager som inte överlappar det andra lagret, med båda lagrens attribut.

Python-kod

Algoritm-ID: `qgis:symmetriskskillnad`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Förening

Kontrollerar överlappningar mellan funktioner i inmatningslagret och skapar separata funktioner för överlappande och icke-överlappande delar. Överlappningsområdet kommer att skapa lika många identiska överlappande features som det finns features som deltar i överlappningen.

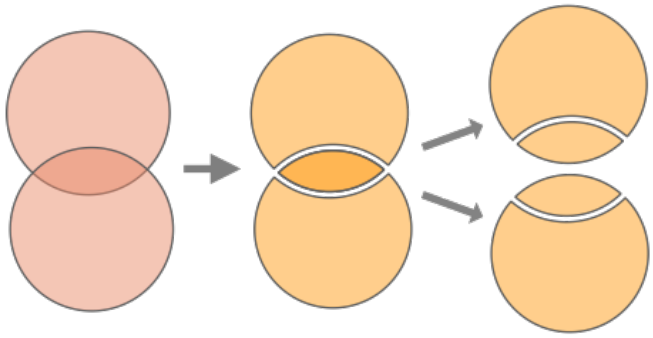


Fig. 24.128: Unionsoperation med ett enda inmatningslager med två överlappande funktioner (vänster) - resulterar i fyra funktioner (mitten) - funktioner flyttade för tydlighetens skull (höger)

Ett överlagringslager kan också användas, i vilket fall funktioner från varje lager delas upp vid överlappningen med funktioner från det andra lagret, vilket skapar ett lager som innehåller alla delar från både inmatnings- och överlagringslagren. Funktioner i samma lager kommer inte att dela varandra. Attributtabellen i unionslagret fylls med attributvärden från respektive originallager för icke-överlappande funktioner och med attributvärden från båda lagren för överlappande funktioner.

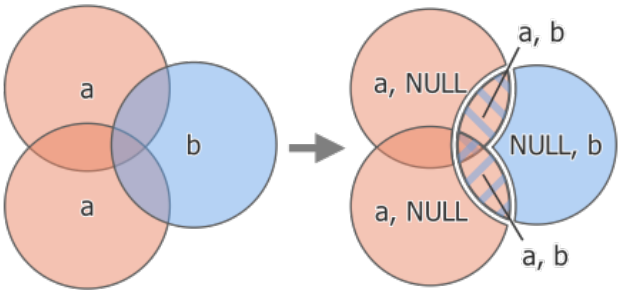


Fig. 24.129: Unionsoperation mellan ett inmatningslager med två funktioner ”a” och ett överlagringslager med en funktion ”b” (vänster) - resulterande lager med fem funktioner med attribut från båda lagren (höger)

Observera

Med ett överlagringslager kommer funktioner på samma lager inte att dela varandra. Om du vill dela upp överlappningar på samma lager samt andra lager kör du först algoritmen med flera lager och sedan algoritmen

igen med endast den tidigare utmatningen.

Standardmeny: *Vektor ► Geobearbetningsverktyg*

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Union (flera), *Klipp*, *Skillnad*, *Korsningar*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager för att dela vid alla korsningar.
Överläggslager Valfritt	`OVERLAY`	[vektor: geometri]	Lager som kommer att kombineras med det första lagret. Helst ska geometritypen vara densamma som input-lagret.
Union	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det lager som ska innehålla (delade och duplicerade) funktioner från inmatningslagret och överlagringslagret. *One of*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Prefix för överlagringsfält Valfritt	`OVERLAY_FIELDS_`	[sträng]	Lägg till ett prefix för att identifiera fält i överlagringslagret. Duplicerade fältnamn kommer att få ett räknesuffix för att undvika kollisioner.
Rutnätsstorlek Valfritt	`GRID_SIZE`	[numerisk: dubbel] Standard: Ej inställd	Om det anges, snappas indatageometrierna till ett rutnät med den angivna storleken och resultatets hörn beräknas på samma rutnät. Kräver GEOS 3.9.0 eller senare.

Utgångar

Etikett	Namn	Typ	Beskrivning
Union	`OUTPUT`	[samma som inmatning]	Lager som innehåller alla överlappande och icke-överlappande delar från det/de bearbetade lagret/lagren.

Python-kod

Algoritm-ID: `qgis:union`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Union (flera)

Kontrollerar överlappningar mellan funktioner i inmatningslagret och skapar separata funktioner för överlappande och icke-överlappande delar. Överlappningsområdet kommer att skapa lika många identiska överlappande features som det finns features som deltar i överlappningen.

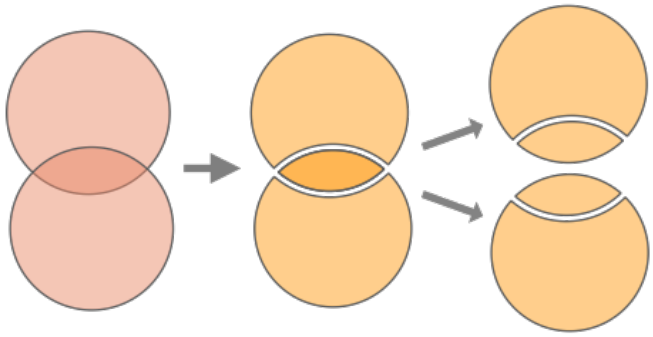


Fig. 24.130: Unionsoperation med ett enda inmatningslager med två överlappande funktioner (vänster) - resulterar i fyra funktioner (mitten) - funktioner flyttade för tydlighetens skull (höger)

Flera överlagringslager kan också användas, i vilket fall funktioner från varje lager delas vid överlappningen med funktioner från alla andra lager, vilket skapar ett lager som innehåller alla delar från både inmatnings- och överlagringslager. Funktioner i samma lager kommer inte att dela varandra. Attributtabellen i Union-lagret fylls med attributvärden från respektive originallager för icke-överlappande funktioner och attributvärden från överlagringslager för överlappande funktioner.

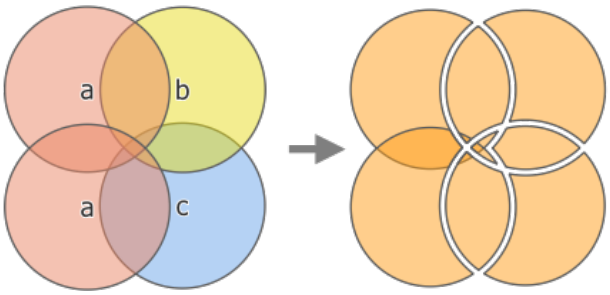


Fig. 24.131: Unionsoperation mellan ett inmatningslager med två funktioner ”a” och överlagringslager med en funktion ”b” och ”c” (vänster) - resulterande lager med elva funktioner med attribut från alla lager (höger)

Observera

Med ett överlagringslager kommer funktioner på samma lager inte att dela varandra. Om du vill dela upp överlappningar på samma lager samt andra lager kör du först algoritmen med flera lager och sedan algoritmen igen med endast den tidigare utmatningen.

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Förening, *Klipp*, *Skillnad*, *Korsningar*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager för att dela vid alla korsningar.
Överläggslager Valfritt	`OVERLAYS`	[vektor: geometri] [lista]	Lager som kommer att kombineras med det första lagret. Helst ska geometritypen vara densamma som input-lagret.
Union	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det lager som ska innehålla (delade och duplicerade) funktioner från inmatningslagret och överlagringslagren. *One of*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Prefix för överlagringsfält Valfritt	`OVERLAY_FIELDS_`	[sträng]	Lägg till ett prefix för att identifiera fält i överlagringslagren. Duplicerade fältnamn kommer att få ett räknesuffix för att undvika kollisioner.

Utgångar

Etikett	Namn	Typ	Beskrivning
Union	`OUTPUT`	[samma som inmatning]	Lager som innehåller alla överlappande och icke-överlappande delar från det/de bearbetade lagret/lagren, med alla lagrens attribut.

Python-kod

Algoritm-ID: `qgis:multiunion`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.25 Val av vektor

Extrahera per attribut

Skapar två vektorlager från ett inmatningslager: det ena innehåller endast matchande funktioner medan det andra innehåller alla icke-matchande funktioner.

Kriterierna för att lägga till funktioner i det resulterande lagret baseras på värdena för ett attribut från inmatningslagret.

Se även

Välj efter attribut

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Lager att extrahera funktioner från.
Val av attribut	`FIELD`	[tabellfält: alla]	Lagrets filtreringsfält
Operator	`OPERATOR`	[uppräkning] Standard: 0	Många olika operatörer finns tillgängliga: • 0 — = • 1 — ≠ • 2 — >> • 3 — >= • 4 — < • 5 — <= • 6 — börjar med • 7 — innehåller • 8 — är noll • 9 — är inte noll • 10 — innehåller inte
Värde Valfritt	`VALUE`	[sträng]	Värde som ska utvärderas
Extraherad (attribut)	`OUTPUT`	[samma som inmatning] Standard: `[Create Temporary Layer]`	Ange utdatavektorlagret för matchande funktioner. *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Extraherad (ej matchande)	`FAIL_OUTPUT`	[samma som inmatning] Standard: `[Skip output]`	Ange utdatavektorlagret för icke-matchande funktioner. *Ett av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Extraherad (attribut)	`OUTPUT`	[samma som inmatning]	Vektorlager med matchande funktioner från inmatningslagret
Extraherad (ej matchande)	`FAIL_OUTPUT`	[samma som inmatning]	Vektorlager med icke-matchande funktioner från inmatningslagret

Python-kod

Algoritm-id: `native:extractbyattribute`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Utdrag per uttryck

Skapar två vektorlager från ett inmatningslager: det ena innehåller endast matchande funktioner medan det andra innehåller alla icke-matchande funktioner.

Kriterierna för att lägga till funktioner i det resulterande lagret baseras på ett QGIS-uttryck. För mer information om uttryck, se *Uttryck*.

Se även

Välj efter uttryck

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlager
Uttryck	`EXPRESSION`	[uttryck]	Uttryck för att filtrera vektorlagret
Matchande funktioner	`OUTPUT`	[samma som inmatning] Standard: `[Create Temporary Layer]`	Ange utdatavektorlagret för matchande funktioner. *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Inte matchande	`FAIL_OUTPUT`	[samma som inmatning] Standard: `[Skip output]`	Ange utdatavektorlagret för icke-matchande funktioner. *Ett av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Matchande funktioner	`OUTPUT`	[samma som inmatning]	Vektorlager med matchande funktioner från inmatningslagret
Inte matchande	`FAIL_OUTPUT`	[samma som inmatning]	Vektorlager med icke-matchande funktioner från inmatningslagret

Python-kod

Algoritm-id: `native:extractbyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Utdrag per plats

Skapar ett nytt vektorlager som endast innehåller matchande funktioner från ett inmatningslager.

Kriterierna för att lägga till funktioner i det resulterande lagret baseras på det spatiala förhållandet mellan varje funktion och funktionerna i ett ytterligare lager.

Se även

Välj efter plats, *Extrahera inom avstånd*

Utforska spatiala relationer

Geometriska predikat är booleska funktioner som används för att bestämma den spatiala relationen mellan en egenskap och en annan genom att jämföra om och hur deras geometrier delar en del av utrymmet.

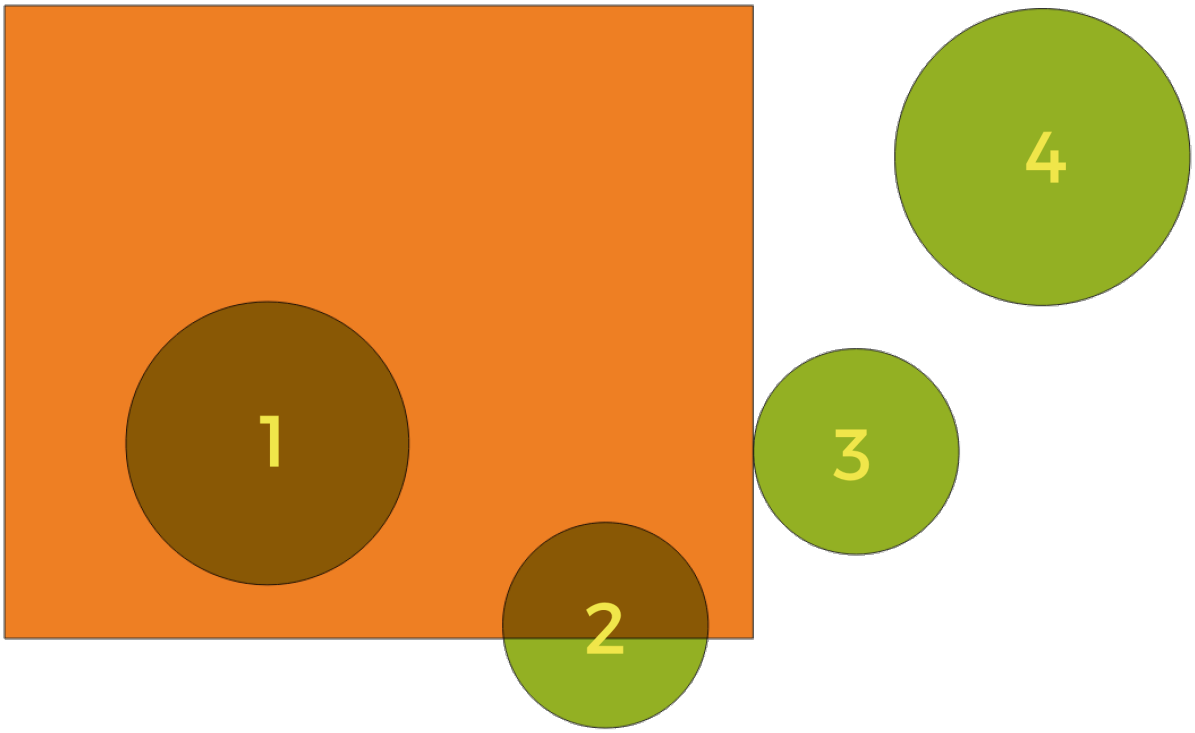


Fig. 24.132: Letar efter spatiala relationer mellan lager

I figuren ovan letar vi efter de gröna cirklarna genom att spatialt jämföra dem med den orangefärgade rektangeln. Tillgängliga geometriska predikat är:

Korsningspunkt
Testar om en geometri korsar en annan. Returnerar 1 (sant) om geometrierna korsar varandra spatialt (delar någon del av utrymmet - överlappar eller rör vid varandra) och 0 om de inte gör det. I bilden ovan kommer detta att returnera cirklarna 1, 2 och 3.

Innehåller
Returnerar 1 (sant) om och endast om inga punkter i b ligger i utsidan av a, och minst en punkt i insidan av b ligger i insidan av a. I bilden returneras ingen cirkel, men rektangeln skulle göra det om man letade efter den tvärtom, eftersom den innehåller cirkel 1 helt och hållet. Detta är motsatsen till *är inom*.

Disjunkt
Returnerar 1 (sant) om geometrierna inte delar någon del av utrymmet (ingen överlappning, ingen beröring). Endast cirkel 4 returneras.

Lika
Returnerar 1 (true) om och endast om geometrierna är exakt likadana. Inga cirklar kommer att returneras.

Berör
Testar om en geometri berör en annan. Returnerar 1 (true) om geometrierna har minst en gemensam punkt, men deras inre delar inte korsar varandra. Endast cirkel 3 returneras.

Överlappning
Testar om en geometri överlappar en annan. Returnerar 1 (true) om geometrierna delar utrymme, har samma dimension, men inte är helt inneslutna av varandra. Endast cirkel 2 returneras.

Är inom
Testar om en geometri är inuti en annan. Returnerar 1 (true) om geometri a är helt inuti geometri b. Endast cirkel 1 returneras.

Kors
Returnerar 1 (true) om de angivna geometrierna har vissa, men inte alla, inre punkter gemensamt och den faktiska korsningen har en lägre dimension än den högsta angivna geometrin. Till exempel kommer en linje

som korsar en polygon att korsas som en linje (true). Två linjer som korsar varandra kommer att korsas som en punkt (sant). Två polygoner korsar varandra som en polygon (falsk). I bilden kommer inga cirklar att returneras.

Parametrar

Etikett	Namn	Typ	Beskrivning
Extrahera funktioner från	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Var funktionerna (geometriskt predikat)	`PREDICATE`	[uppräkning] [lista] Standard: [0]	Typ av spatial relation som indatafunktionen bör ha med en intersect-funktion för att den ska kunna väljas. En eller flera av: • 0 — korsa • 1 — innehåller • 2 — åtskilda • 3 — lika • 4 — beröring • 5 — överlappning • 6 — är inom • 7 — kors Om mer än ett villkor väljs måste minst ett av dem (OR-operationen) uppfyllas för att en funktion ska kunna extraheras.
Genom att jämföra med funktionerna från	`INTERSECT`	[vektor: geometri]	Vektorlager för korsning
Utdragna (plats)	`OUTPUT`	[samma som inmatning] Standard: `[Create temporary layer]`	Ange utmatningsvektorlagret för de funktioner som har de valda spatiala relationerna med en eller flera funktioner i jämförelselagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Utdragna (plats)	`OUTPUT`	[samma som inmatning]	Vektorlager med funktioner från inmatningslagret som har det eller de valda spatiala förhållandena med en eller flera funktioner i jämförelselagret.

Python-kod

Algoritm-id: `native:extractbylocation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Extrahera inom avstånd

Skapar ett nytt vektorlager som endast innehåller matchande funktioner från ett inmatningslager. Funktionerna kopieras om de ligger inom det angivna maximala avståndet från funktionerna i ett extra referenslager.

Se även

Välj inom avstånd, *Utdrag per plats*

Parametrar

Etikett	Namn	Typ	Beskrivning
Extrahera funktioner från	`INPUT`	[vektor: geometri]	Ingångsvektorlager för att kopiera funktioner från
Genom att jämföra med funktionerna från	`REFERENCE`	[vektor: geometri]	Vektorlager vars egenskaper närhet används
Var funktionerna finns inom	`DISTANCE`	[numerisk: dubbel] Standard: 100.0	Det maximala avståndet runt referensobjekten för att välja inmatningsobjekt inom
Ändra aktuellt urval genom	`METHOD`	[uppräkning] Standard: 0	Hur valet av algoritm ska hanteras. En av: • 0 — skapa nytt urval • 1 — lägga till aktuellt val • 2 — välja inom aktuellt val • 3 — ta bort från aktuellt val
Utdragna (plats)	`OUTPUT`	[samma som inmatning] Standard: `[Create temporary layer]`	Ange utmatningsvektorlagret för de funktioner som ligger inom det inställda avståndet från referensfunktionerna. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Utdragna (plats)	`OUTPUT`	[samma som inmatning]	Vektorlager med funktioner från inmatningslagret som matchar villkoret för avstånd från referensfunktioner

Python-kod

Algoritm-id: `native:extractwithindistance`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Filtrera efter geometrityp

Filtrerar funktioner efter deras geometrityp. Inkommande objekt styrs till olika utdata beroende på om de har en punkt-, linje- eller polygongeometri.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Lager att utvärdera

Utmatningar

Etikett	Namn	Typ	Beskrivning
Punktfunktioner Valfritt	`POINTS`	[vektor: punkt]	Lager med punkter
Linjefunktioner Valfritt	`LINES`	[vektor: linje]	Lager med linjer
Polygonfunktioner Valfritt	`POLYGONS`	[vektor: polygon]	Lager med polygoner
Funktioner utan geometri Valfritt	`NO_GEOMETRY`	[vektor: tabell]	Geometrilöst vektorlager

Python-kod

Algoritm-id: `native:filterbygeometry`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Slumpmässigt utdrag

Tar ett vektorlager och genererar ett nytt som bara innehåller en delmängd av funktionerna i inmatningslagret.

Undergruppen definieras slumpmässigt, baserat på funktions-ID:n, med hjälp av en procentsats eller ett räknevärde för att definiera det totala antalet funktioner i undergruppen.

Varning

Den här algoritmen tar bort befintliga primärnycklar eller FID-värden och återskapar dem i utdatalagren.

Se även

Slumpmässigt val

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Källvektorlager för att välja funktioner från
Metod	`METHOD`	[uppräkning] Standard: 0	Metoder för slumpmässigt urval. En av: • 0 — Antal utvalda funktioner • 1 — Procentandel av utvalda funktioner
Antal/procentandel av valda funktioner	`NUMBER`	[numerisk: heltal] Standard: 10	Antal eller procentandel av funktioner som ska väljas
Utdragna (slumpmässigt)	`OUTPUT`	[samma som inmatning] Standard: `[Create temporary layer]`	Ange utdatavektorlagret för de slumpmässigt valda funktionerna. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Utdragna (slumpmässigt)	`OUTPUT`	[samma som inmatning]	Vektorlager som innehåller slumpmässigt utvalda funktioner från inmatningslagret

Python-kod

Algoritm-id: `native:randomextract`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Slumpmässigt utdrag inom delmängder

Tar ett vektorlager och genererar ett nytt som bara innehåller en delmängd av funktionerna i inmatningslagret.

Undergruppen definieras slumpmässigt, baserat på funktionens ID, med hjälp av ett procent- eller räknevärde för att definiera det totala antalet funktioner i undergruppen. Procentvärdet/antalet tillämpas inte på hela lagret, utan på varje kategori. Kategorierna definieras utifrån ett givet attribut.

Se även

Slumpmässigt urval inom delmängder

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Vektorlager för att välja funktioner från
ID-fält	`FIELD`	[tabellfält: alla]	Kategori för källvektorlagret som funktionerna ska väljas från
Metod	`METHOD`	[uppräkning] Standard: 0	Metod för slumpmässigt urval. En av: • 0 — Antal utvalda funktioner • 1 — Procentandel av utvalda funktioner
Antal/procentandel av valda funktioner	`NUMBER`	[numerisk: heltal] Standard: 10	Antal eller procentandel av funktioner som ska väljas

continues on next page

Table 24.202 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Extraherad (slumpmässigt stratifierad)	`OUTPUT`	[samma som inmatning] Standard: `[Create temporary layer]`	Ange utdatavektorlagret för de slumpmässigt valda funktionerna. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utmatningar

Etikett	Namn	Typ	Beskrivning
Extraherad (slumpmässigt stratifierad)	`OUTPUT`	[samma som inmatning]	Vektorlager som innehåller slumpmässigt utvalda funktioner från inmatningslagret

Python-kod

Algoritm-ID: `qgis:randomextraheringmedundergrupper`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Slumpmässigt val

Tar ett vektorlager och väljer en delmängd av dess funktioner. Inget nytt lager genereras av denna algoritm.

Undergruppen definieras slumpmässigt, baserat på funktions-ID:n, med hjälp av en procentsats eller ett räknevärde för att definiera det totala antalet funktioner i undergruppen.

Standardmeny: *Vektor ► Forskningsverktyg*

Se även

Slumpmässigt utdrag

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Vektorlager för urvalet
Metod	`METHOD`	[uppräkning] Standard: 0	Metod för slumpmässigt urval. En av: • 0 — Antal utvalda funktioner • 1 — Procentandel av utvalda funktioner
Antal/procentandel av valda funktioner	`NUMBER`	[numerisk: heltal] Standard: 10	Antal eller procentandel av funktioner som ska väljas

Utmatningar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[samma som inmatning]	Inmatningslagret med valda funktioner

Python-kod

Algoritm-id: `qgis:randomselection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Slumpmässigt urval inom delmängder

Tar ett vektorlager och väljer en delmängd av dess funktioner. Inget nytt lager genereras av denna algoritm.

Undergruppen definieras slumpmässigt, baserat på funktions-ID:n, med hjälp av en procentsats eller ett räknevärde för att definiera det totala antalet funktioner i undergruppen.

Värdet för procent/antal tillämpas inte på hela lagret, utan på varje kategori.

Kategorierna definieras utifrån ett visst attribut, som också anges som en inmatningsparameter för algoritmen.

Inga nya utgångar skapas.

Standardmeny: *Vektor ► Forskningsverktyg*

Se även

Slumpmässigt utdrag inom delmängder

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Vektorlager för att välja funktioner i
ID-fält	`FIELD`	[tabellfält: alla]	Kategori för inmatningslagret att välja funktioner från
Metod	`METHOD`	[uppräkning] Standard: 0	Metod för slumpmässigt urval. En av: • 0 — Antal utvalda funktioner • 1 — Procentandel av utvalda funktioner
Antal/procentandel av valda funktioner	`NUMBER`	[numerisk: heltal] Standard: 10	Antal eller procentandel av funktioner som ska väljas

Utmatningar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[samma som inmatning]	Inmatningslagret med valda funktioner

Python-kod

Algoritm-id: `qgis:randomselectionwithinsubsets`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Välj efter attribut

Skapar ett urval i ett vektorlager.

Kriterierna för att välja ut funktioner baseras på värdena för ett attribut från inmatningslagret.

Se även

Extrahera per attribut

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Vektorlager för att välja funktioner i
Val av attribut	`FIELD`	[tabellfält: alla]	Lagrets filtreringsfält
Operator	`OPERATOR`	[uppräkning] Standard: 0	Många olika operatörer finns tillgängliga: • 0 — = • 1 — ≠ • 2 — >> • 3 — >= • 4 — < • 5 — <= • 6 — börjar med • 7 — innehåller • 8 — är noll • 9 — är inte noll • 10 — innehåller inte
Värde Valfritt	`VALUE`	[sträng]	Värde som ska utvärderas
Ändra aktuellt urval genom	`METHOD`	[uppräkning] Standard: 0	Hur valet av algoritm ska hanteras. En av: • 0 — skapa nytt urval • 1 — lägga till aktuellt val • 2 — borttagning från aktuellt val • 3 — välja inom aktuellt val

Utmatningar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[samma som inmatning]	Inmatningslagret med valda funktioner

Python-kod

Algoritm ID: `qgis:selectbyattribute`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Välj efter uttryck

Skapar ett urval i ett vektorlager.

Kriterierna för att välja ut funktioner baseras på ett QGIS-uttryck. Mer information om uttryck finns i *Uttryck*.

Se även

Utdrag per uttryck

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlager
Uttryck	`EXPRESSION`	[uttryck]	Uttryck för att filtrera inmatningslagret
Ändra aktuellt urval genom	`METHOD`	[uppräkning] Standard: 0	Hur valet av algoritm ska hanteras. En av: • 0 — skapa nytt urval • 1 — lägga till aktuellt val • 2 — borttagning från aktuellt val • 3 — välja inom aktuellt val

Utmatningar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[samma som inmatning]	Inmatningslagret med valda funktioner

Python-kod

Algoritm-ID: `qgis:selectbyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Välj efter plats

Skapar ett urval i ett vektorlager.

Kriterierna för att välja ut funktioner baseras på det spatiala förhållandet mellan varje funktion och funktionerna i ett ytterligare lager.

Standardmeny: *Vektor* ► *Forskningsverktyg*

Se även

Utdrag per plats, *Välj inom avstånd*

Utforska spatiala relationer

Geometriska predikat är booleska funktioner som används för att bestämma den spatiala relationen mellan en egenskap och en annan genom att jämföra om och hur deras geometrier delar en del av utrymmet.

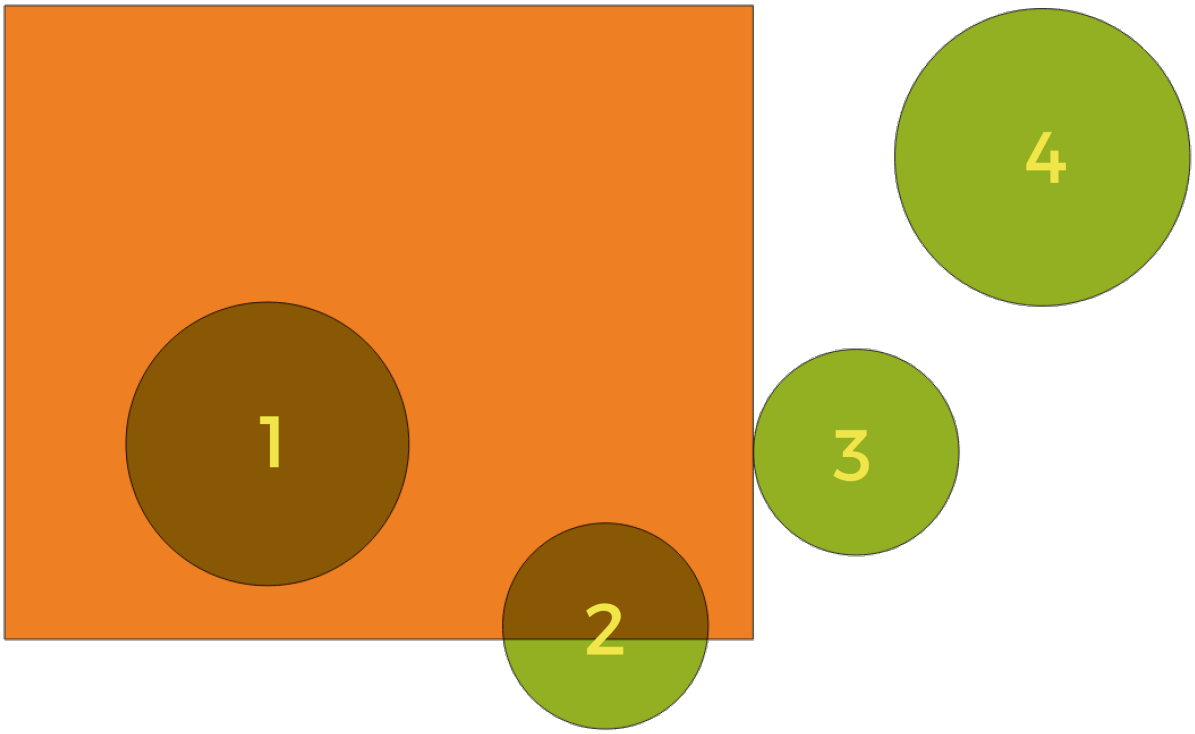


Fig. 24.133: Letar efter spatiala relationer mellan lager

I figuren ovan letar vi efter de gröna cirklarna genom att spatialt jämföra dem med den orangefärgade rektangeln. Tillgängliga geometriska predikat är:

Korsningspunkt
: Testar om en geometri korsar en annan. Returnerar 1 (sant) om geometrierna korsar varandra spatialt (delar någon del av utrymmet - överlappar eller rör vid varandra) och 0 om de inte gör det. I bilden ovan kommer detta att returnera cirklarna 1, 2 och 3.

Innehåller
: Returnerar 1 (sant) om och endast om inga punkter i b ligger i utsidan av a, och minst en punkt i insidan av b ligger i insidan av a. I bilden returneras ingen cirkel, men rektangeln skulle göra det om man letade efter den tvärtom, eftersom den innehåller cirkel 1 helt och hållet. Detta är motsatsen till *är inom*.

Disjunkt
Returnerar 1 (sant) om geometrierna inte delar någon del av utrymmet (ingen överlappning, ingen beröring). Endast cirkel 4 returneras.

Lika
Returnerar 1 (true) om och endast om geometrierna är exakt likadana. Inga cirklar kommer att returneras.

Berör
Testar om en geometri berör en annan. Returnerar 1 (true) om geometrierna har minst en gemensam punkt, men deras inre delar inte korsar varandra. Endast cirkel 3 returneras.

Överlappning
Testar om en geometri överlappar en annan. Returnerar 1 (true) om geometrierna delar utrymme, har samma dimension, men inte är helt inneslutna av varandra. Endast cirkel 2 returneras.

Är inom
Testar om en geometri är inuti en annan. Returnerar 1 (true) om geometri a är helt inuti geometri b. Endast cirkel 1 returneras.

Kors
Returnerar 1 (true) om de angivna geometrierna har vissa, men inte alla, inre punkter gemensamt och den faktiska korsningen har en lägre dimension än den högsta angivna geometrin. Till exempel kommer en linje som korsar en polygon att korsas som en linje (true). Två linjer som korsar varandra kommer att korsas som en punkt (sant). Två polygoner korsar varandra som en polygon (falsk). I bilden kommer inga cirklar att returneras.

Parametrar

Etikett	Namn	Typ	Beskrivning
Välj funktioner från	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Var funktionerna (geometriskt predikat)	`PREDICATE`	[uppräkning] [lista] Standard: [0]	Typ av spatial relation som indatafunktionen bör ha med en intersect-funktion för att den ska kunna väljas. En eller flera av: • 0 — korsa • 1 — innehåller • 2 — åtskilda • 3 — lika • 4 — beröring • 5 — överlappning • 6 — är inom • 7 — kors Om mer än ett villkor väljs måste minst ett av dem (OR-operationen) uppfyllas för att en funktion ska kunna extraheras.
Genom att jämföra med funktionerna från	`INTERSECT`	[vektor: geometri]	Vektorlager för korsning
Ändra aktuellt urval genom	`METHOD`	[uppräkning] Standard: 0	Hur valet av algoritm ska hanteras. En av: • 0 — skapa nytt urval • 1 — lägga till aktuellt val • 2 — välja inom aktuellt val • 3 — ta bort från aktuellt val

Utmatningar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[samma som inmatning]	Inmatningslagret med valda funktioner

Python-kod

Algoritm-id: `native:selectbylocation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Välj inom avstånd

skapar en markering i ett vektorlager. Objekten väljs ut när de ligger inom det angivna maximala avståndet från objekten i ett extra referenslager.

Se även

Extrahera inom avstånd, *Välj efter plats*

Parametrar

Etikett	Namn	Typ	Beskrivning
Välj funktioner från	`INPUT`	[vektor: geometri]	Ingångsvektorlager för att välja funktioner från
Genom att jämföra med funktionerna från	`REFERENCE`	[vektor: geometri]	Vektorlager vars egenskaper närhet används
Var funktionerna finns inom	`DISTANCE`	[numerisk: dubbel] Standard: 100.0	Det maximala avståndet runt referensobjekt för att välja inmatningsobjekt
Ändra aktuellt urval genom	`METHOD`	[uppräkning] Standard: 0	Hur valet av algoritm ska hanteras. En av: • 0 — skapa nytt urval • 1 — lägga till aktuellt val • 2 — välja inom aktuellt val • 3 — ta bort från aktuellt val

Utmatningar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[samma som inmatning]	Inmatningslagret med valda funktioner

Python-kod

Algoritm-id: `native:selectwithindistance`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

Algoritm-id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.26 Vektortabell

Lägg till autoinkrementellt fält

Lägger till ett nytt heltalsfält till ett vektorlager, med ett sekventiellt värde för varje objekt.

Detta fält kan användas som ett unikt ID för objekten i lagret. Det nya attributet läggs inte till i inmatningslagret utan ett nytt lager genereras istället.

Det initiala startvärdet för den inkrementella serien kan specificeras. Som tillval kan den inkrementella serien baseras på grupperingsfält och en sorteringsordning för funktioner kan också anges.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlagret.
Fältets namn	`FIELD_NAME`	[sträng] Standard: ”AUTO	Namn på fältet med autoinkrementella värden
Startvärden vid Valfritt	`START`	[numerisk: heltal] Standard: 0	Välj det inledande numret för den stegvisa räkningen
Modulusvärde Valfritt	`MODULUS`	[numerisk: heltal] Standard: 0	Om du anger ett valfritt modulusvärde startar räkningen om till START när fältvärdet når modulusvärdet. `0` innebär ingen omstart.
Gruppera värden efter Valfritt	`GROUP_FIELDS`	[tabellfält: valfritt] [lista]	Välj grupperingsfält: istället för en enda räkning för hela lagret, görs en separat räkning för varje värde som returneras genom kombinationen av dessa fält.
Sortera uttryck Valfritt	`SORT_EXPRESSIO`	[uttryck]	Använd ett uttryck för att sortera objekten i lagret antingen globalt eller, om det är inställt, baserat på gruppfält.

continues on next page

Table 24.206 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Sortera i stigande ordning	`SORT_ASCENDING`	[boolean] Standard: Sann	När ett ”sorteringsuttryck” har angetts kan du använda det här alternativet för att styra i vilken ordning funktioner tilldelas värden.
Sortera nollor först	`SORT_NULLS_FIR`	[boolean] Standard: Falsk	När ett ”sorteringsuttryck” har angetts kan du använda det här alternativet för att ange om *Null*-värden ska räknas först eller sist.
Inkrementellt	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange utmatningsvektorskiktet med fältet för automatisk ökning. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Inkrementellt	`OUTPUT`	[samma som inmatning]	Vektorlager med automatiskt inkrementellt fält

Python-kod

Algoritm-ID: `` `native:addautoincrementalfield ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Lägg till fält i attributtabellen

Lägger till ett nytt fält till ett vektorlager.

Attributets namn och egenskaper definieras som parametrar.

Det nya attributet läggs inte till i inmatningslagret utan ett nytt lager genereras i stället.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Det ingående lagret
Fältets namn	`FIELD_NAME`	[sträng]	Namn på det nya fältet
Typ av fält	`FIELD_TYPE`	[uppräkning] Standard: 0	Typ av det nya fältet. Du kan välja mellan: • 0 — Heltal (32 bitar) • 1 — Decimal (dubbel) • 2 — Text (sträng) • 3 — Boolean • 4 — Datum • 5 — Tid • 6 — Datum och tid • 7 — Binärt objekt (BLOB) • 8 — Stränglista • 9 — Heltalslista • 10 — Decimal (dubbel) Lista
Fältets längd	`FIELD_LENGTH`	[numerisk: heltal] Standard: 10	Längd på fältet
Fältets precision	`FIELD_PRECISIO`	[numerisk: heltal] Standard: 0	Precision för fältet. Användbar med fälttypen Float.
alias för fält Valfritt	`FIELD_ALIAS`	[sträng]	Ange ett namn som ska användas som alias för fältet. Stöds inte av alla formattyper.
Fältkommentar Valfritt	`FIELD_COMMENT`	[sträng]	Lagra en kommentar som beskriver fältet. Stöds inte av alla formattyper.
Tillägg	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Tillägg	`OUTPUT`	[samma som inmatning]	Vektorlager med nytt fält tillagt

Python-kod

Algoritm-ID: `native:addfieldtoattributestable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Lägg till indexfält för unikt värde

Tar ett vektorlager och ett attribut och lägger till ett nytt numeriskt fält.

Värdena i det här fältet motsvarar värdena i det angivna attributet, så funktioner med samma värde för attributet kommer att ha samma värde i det nya numeriska fältet.

Detta skapar en numerisk motsvarighet till det angivna attributet, som definierar samma klasser.

Det nya attributet läggs inte till i inmatningslagret utan ett nytt lager genereras i stället.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Det ingående lagret.
Klassfält	`FIELD`	[tabellfält: alla]	Funktioner som har samma värde för detta fält kommer att få samma index.
Namn på fält för utdata	`FIELD_NAME`	[sträng] Standard: 'NUM_FIELD'	Namn på det nya fältet som innehåller indexen.
Lager med indexfält	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Vektorlager med det numeriska fältet som innehåller index. *En av*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.
Klassöversikt	`SUMMARY_OUTPUT`	[vektor: tabell] Standard: `[Hoppa över utdata]`	Ange den tabell som ska innehålla sammanfattningen av klassfältet som mappats till motsvarande unika värde. *One of*: • Hoppa över utmatning • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Lager med indexfält	`OUTPUT`	[samma som inmatning]	Vektorlager med det numeriska fältet som innehåller index.
Klassöversikt	`SUMMARY_OUTPUT`	[vektor: tabell]	Tabell med sammanfattning av klassfältet som mappats till motsvarande unika värde.

Python-kod

Algoritm-ID: `native:adduniquevalueindexfield`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Lägg till X/Y-fält i lagret

Lägger till X- och Y-fält (eller latitud/longitud) till ett punktlager. X/Y-fälten kan beräknas i ett annat CRS än lagret (t.ex. genom att skapa latitud/longitud-fält för ett lager i ett projicerat CRS).

Tillåter *features in-place modification* av punktfeatures

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: punkt]	Det ingående lagret.
Koordinatsystem	`CRS`	[crs] Standard: ”EPSG:4326”	Referenssystem för koordinater som ska användas för de genererade x- och y-fälten.
Fältprefix Valfritt	`PREFIX`	[sträng]	Prefix som ska läggas till de nya fältnamnen för att undvika namnkollisioner med fält i inmatningslagret.
Tillagda fält	`OUTPUT`	[vektor: punkt] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Tillagda fält	`OUTPUT`	[vektor: punkt]	Utmatningslagret - identiskt med inmatningslagret men med två nya dubbla fält, `x` och `y`.

Python-kod

Algoritm-ID: `` `native:addxyfields` ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Avancerad fältkalkylator i Python

Lägger till ett nytt attribut till ett vektorlager, med värden som erhålls genom att tillämpa ett uttryck på varje egenskap.

Uttrycket definieras som en Python-funktion.

Varning

Denna algoritm är en potentiell säkerhetsrisk om den körs med okontrollerade indata och kan leda till systemskador eller dataläckage.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlager
Namn på fält för resultat	`FIELD_NAME`	[sträng] Standard: 'Nytt fält'	Namn på det nya fältet
Typ av fält	`FIELD_TYPE`	[uppräkning] Standard: 0	Typ av det nya fältet. Ett av: • 0 — Heltal (32 bitar) • 1 — Decimal (dubbel) • 2 — Text (sträng) • 3 — Boolean • 4 — Datum • 5 — Tid • 6 — Datum och tid • 7 — Binärt objekt (BLOB) • 8 — Stränglista • 9 — Heltalslista • 10 — Decimal (dubbel) Lista

continues on next page

Table 24.210 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Fältets längd	`FIELD_LENGTH`	[numerisk: heltal] Standard: 10	Längd på fältet
Fältets precision	`FIELD_PRECISIO`	[numerisk: heltal] Standard: 3	Precision för fältet. Användbar med fälttypen Float.
Globalt uttryck Valfritt	`GLOBAL`	[sträng]	Koden i avsnittet med det globala uttrycket kommer bara att köras en gång innan räknaren börjar iterera genom alla funktioner i inmatningslagret. Därför är detta rätt plats för att importera nödvändiga moduler eller beräkna variabler som ska användas i efterföljande beräkningar.
Formula	`FORMULA`	[sträng]	Python-formeln som ska utvärderas. Exempel::: För att beräkna arean av ett polygonlager kan du lägga till: `value = $geom.area()`
Beräknad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange vektorlagret med det nya beräknade fältet. Ett av följande: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Beräknad	`OUTPUT`	[samma som inmatning]	Vektorlager med det nya beräknade fältet

Python-kod

Algoritm ID: `qgis:advancedpythonfieldcalculator`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Släpp fält(en)

Tar ett vektorlager och genererar ett nytt som har samma egenskaper men utan de valda kolumnerna.

Se även

Behålla fält

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlager för att ta bort fält(en) från
Fält att släppa	`COLUMN`	[tabellfält: valfritt] [lista]	Det eller de fält som ska tas bort
Resterande fält	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange utdatavektorlagret med de återstående fälten. *Ett av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Resterande fält	`OUTPUT`	[samma som inmatning]	Vektorlager med de återstående fälten

Python-kod

Algoritm-ID: `native:deletecolumn`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Explodera HStore-fältet

Skapar en kopia av inmatningslagret och lägger till ett nytt fält för varje unik nyckel i HStore-fältet.

Listan över förväntade fält är en valfri kommaseparerad lista. Om den här listan anges läggs endast dessa fält till och HStore-fältet uppdateras. Som standard läggs alla unika nycklar till.

PostgreSQL HStore är en enkel nyckelvärdesbutik som används i PostgreSQL och GDAL (när du läser en OSM-fil med fältet `other_tags`.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Store fält	`FIELD`	[tabellfält: alla]	Det eller de fält som ska tas bort
Väntad lista över fält separerade med ett kommatecken Valfritt	`EXPECTED_FIELD`	[sträng] Standard: ”	Kommaseparerad lista över fält som ska extraheras. HStore-fältet kommer att uppdateras genom att dessa nycklar tas bort.
Exploded	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange det utgående vektorlagret. Ett av följande: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Exploded	`OUTPUT`	[samma som inmatning]	Utgående vektorlager

Python-kod

Algoritm-id: `native:explodehstorefield`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Extrahera binärt fält

Extraherar innehåll från ett binärt fält och sparar det i enskilda filer. Filnamnen kan genereras med hjälp av värden som hämtas från ett attribut i källtabellen eller baserat på ett mer komplext uttryck.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlager som innehåller binära data
Binärt fält	`FIELD`	[tabellfält: alla]	Fält som innehåller binära data
Filnamn	`FILENAME`	[uttryck]	Fält eller uttrycksbaserad text för att namnge varje utdatafil
Destinationsmapp	`FOLDER`	mapp Standard: `[Spara till tillfällig mapp]`	Mapp där utdatafilerna ska lagras. *En av*: • Spara till en temporär katalog • Spara till katalog

Utgångar

Etikett	Namn	Typ	Beskrivning
Mapp	`FOLDER`	mapp	Den mapp som innehåller utdatafilerna.

Python-kod

Algoritm-ID: `native:extraherabinär`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Fältkalkylator

Öppnar fältkalkylatorn (se *Uttryck*). Du kan använda alla uttryck och funktioner som stöds.

Ett nytt lager skapas med resultatet av uttrycket.

Fältkalkylatorn är mycket användbar när den används i *Modellkonstruktören*.

Observera

Den här algoritmen använder ellipsoidbaserade mätningar och respekterar de aktuella *ellipsoidinställningarna*.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Lagret att beräkna på
Namn på fält för utdata	`FIELD_NAME`	[sträng]	Namnet på fältet för resultaten
Typ av utdatafält	`FIELD_TYPE`	[uppräkning] Standard: 0	Typ av fält. En av: • 0 — Decimal (dubbel) • 1 — Heltal (32 bitar) • 2 — Text (sträng) • 3 — Datum • 4 — Tid • 5 — Datum och tid • 6 — Boolean • 7 — Binärt objekt (BLOB) • 8 — Stränglista • 9 — Heltalslista • 10 — Decimal (dubbel) Lista
Utmatningsfältets bredd	`FIELD_LENGTH`	[numerisk: heltal] Standard: 0	Resultatfältets längd (minst 0)
Fältets precision	`FIELD_PRECISIO`	[numerisk: heltal] Standard: 0	Resultatfältets precision (minimum 0, maximum 15)
Skapa nytt fält	`NEW_FIELD`	[boolean] Standard: Sann	Bör resultatfältet vara ett nytt fält
Formula	`FORMULA`	[uttryck]	Formeln som ska användas för att beräkna resultatet
Beräknad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Specifikation av det utgående lagret. • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Beräknad	`OUTPUT`	[samma som inmatning]	Utmatningslager med de beräknade fältvärdena

Python-kod

Algoritm-ID: `native:fältberäknare`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Refaktorisera fält

Gör det möjligt att redigera strukturen i attributtabellen för ett vektorlager.

Fälten kan ändras i fråga om typ och namn med hjälp av en fältmappning.

Det ursprungliga lagret ändras inte. Ett nytt lager genereras, som innehåller en modifierad attributtabell, enligt den tillhandahållna fältmappningen.

Observera

När du använder ett mallager med *constraints* på fält visas informationen i widgeten med en färgad bakgrund och verktygstips. Behandla denna information som en ledtråd under konfigurationen. Inga begränsningar kommer att läggas till i ett utdatalager och de kommer inte heller att kontrolleras eller upprätthållas av algoritmen.

Refactor-fältalgoritmen gör det möjligt att:

- Ändra fältnamn och typer
- Lägg till och ta bort fält
- Ordna om fält
- Beräkna nya fält baserat på uttryck
- Ladda fältlista från ett annat lager

Observera

Den här algoritmen använder ellipsoidbaserade mätningar och respekterar de aktuella *ellipsoidinställningarna*.

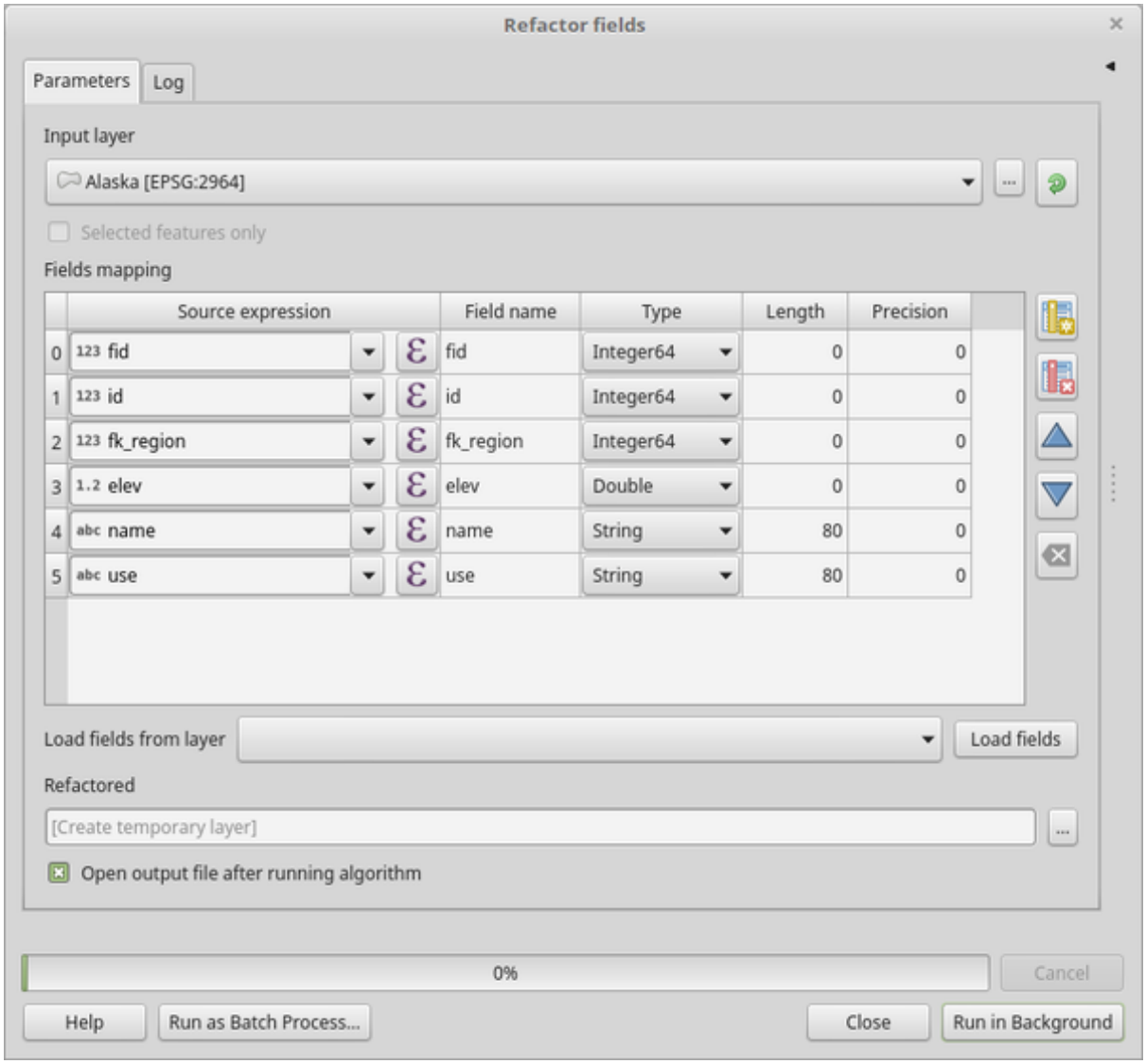


Fig. 24.134: Dialog Refactor-fält

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Lagret som ska modifieras

continues on next page

Table 24.213 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Mappning av fält	`FIELDS_MAPPING`	Konfigurationens id %1 finns redan i listan	Lista över utdatafält med deras definitioner. Den inbäddade tabellen visar alla fält i källagret och gör det möjligt att redigera dem: • Klicka på för att skapa ett nytt fält. • Klicka på för att ta bort ett fält. • Använd och för att ändra den valda fältordningen. • Klicka på för att återställa till standardvyn. • Klicka på för att invertera urvalet i fältlistan. För vart och ett av de fält som du vill återanvända måste du fylla i följande alternativ: ***Källa uttryck* (`uttryck`) [uttryck]** Fält eller uttryck från inmatningslagret. ***Namn* (`namn`) [sträng]** Namn på fältet i utdatalagret. Som standard behålls namnet på inmatningsfältet. ***Type* (`typ`) [uppräkning]** Datatyp för utdatafältet. Tillgängliga typer kanske inte är kompatibla med leverantören av utdatalagret. En av följande: **Uppmärksamma** För vissa fälttyper, t.ex. listor, hjälper en extra parameter `sub_type` till att förfina den specifika typen av data. Den ställs in automatiskt i det grafiska gränssnittet men kan behövas om du kör algoritmen i Python eller från kommandoraden. • 1 — Boolean • 2 — Heltal (32bit) • 4 — Heltal (64bit) • 6 — Decimal (dubbel) • 9 — Heltalslista • 9 — Heltal (64bit) lista • 9 — Decimal (dubbel) lista • 10 — Text (sträng) • 11 — Stränglista • 12 — Binärt objekt (BLOB) • 14 — Datum • 15 — Tid • 16 — Datum och tid ***Sub-type* (`sub_type`) [uppräkning]** För vissa fälttyper, t.ex. listor, hjälper den här parametern till att avgränsa [illegible] ställs in automatiskt i det grafiska gränssnittet men kan behövas om du kör algoritmen i Python eller från

Table 24.213 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Refaktoriserad	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Specifikation av utdatalagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Refaktoriserad	`OUTPUT`	[samma som inmatning]	Utmatningslager med omarbetade fält

Python-kod

Algoritm-ID: `` `native:refactorfields` ``

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Byt namn på fält

Byter namn på ett befintligt fält från ett vektorlager.

Det ursprungliga lagret ändras inte. Ett nytt lager genereras där attributtabellen innehåller det omdöpta fältet.

Se även

Refaktorisera fält

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlagret
Fält att byta namn på	`FIELD`	[tabellfält: alla]	Det fält som ska ändras
Nytt fältnamn	`NEW_NAME`	[sträng]	Det nya fältnamnet

continues on next page

Table 24.214 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Nytt namn	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Specifikation av utdatalagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Nytt namn	`OUTPUT`	[samma som inmatning]	Utmatningslager med det omdöpta fältet

Python-kod

Algoritm ID: `qgis:renametablefield`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Behålla fält

Tar ett vektorlager och genererar ett nytt som endast behåller de markerade fälten. Alla andra fält kommer att tas bort.

Se även

Släpp fält(en)

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlagret
Fält som ska behållas	`FIELDS`	[tabellfält: valfritt] [lista]	Lista över fält som ska behållas i lagret

continues on next page

Table 24.215 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Retained fields	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Specifikation av utdatalagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Retained fields	`OUTPUT`	[samma som inmatning]	Utgående lager med de bibehållna fälten

Python-kod

Algoritm-ID: `native:behålla fält`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Text som ska flyta

Ändrar typen av ett givet attribut i ett vektorlager och konverterar ett textattribut som innehåller numeriska strängar till ett numeriskt attribut (t.ex. ’1’ till `1.0`).

Algoritmen skapar ett nytt vektorlager så att källan inte ändras.

Om konverteringen inte är möjlig kommer den valda kolumnen att ha `NULL`-värden.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlagret.
Textattribut som ska konverteras till float	`FIELD`	[tabellfält: sträng]	Strängfältet för inmatningslagret som ska konverteras till ett floatfält.
Flata från text	`OUTPUT`	[samma som inmatning] Standard: `[Skapa tillfälligt lager]`	Ange utdatalagret. *En av*: • Skapa ett tillfälligt lager (`TEMPORARY_OUTPUT`) • Spara till fil… • Spara till Geopackage… • Spara till databastabell… • Lägg till lager… Här kan du också ändra filkodningen.

Utgångar

Etikett	Namn	Typ	Beskrivning
Flata från text	`OUTPUT`	[samma som inmatning]	Utgående vektorlager med strängfältet omvandlat till ett floatfält

Python-kod

Algoritm ID: `qgis:textofloat`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.1.27 Vektorplattor

Ladda ner vektorplattor

Hämtar vektorplattor från ett lager med inmatade vektorplattor och sparar dem i en lokal vektorplattfil.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektorplattor]	Ett vektorlager för att extrahera några plattor från
Extent	`EXTENT`	[omfattning]	Ange den spatiala omfattningen av det nedladdade området. Den kommer internt att utökas till en multipel av plattstorleken. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Maximal zoomnivå för nedladdning	`MAX_ZOOM`	[numerisk: heltal] Standard: 10	Definierar hur långt man ska zooma in och hämta data från plattorna
Plattgräns	`TILE_LIMIT`	[numerisk: heltal] Standard: 100	Maximalt antal brickor att ladda ner, med hänsyn till zoomnivåerna och omfattningen.
Utdata	`OUTPUT`	[vektorplattor] Standard: [Spara till temporär fil]	Specifikation av utdatafilen för vektorplattor. *En av*: • Spara till en temporär fil • Spara till fil…

Utmatningar

Etikett	Namn	Typ	Beskrivning
Utdata	`OUTPUT`	[vektorplattor]	En lokal vektorfil som lagrar de nedladdade brickorna.

Python-kod

Algoritm ID: `native:ladda ner vektortiler`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skriva vektorplattor (MBTiles)

Exporterar ett eller flera vektorlager till vector tiles, ett dataformat som är optimerat för snabb kartrendering och liten datastorlek.

MBTiles är en specifikation för lagring av kartdata i SQLite-databaser för omedelbar användning och för överföring. MBTiles-filer är kända som tilesets.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: valfri] [lista]	En lista över lager som ska kombineras för att generera vektorplattorna
Minsta zoomnivå	`MIN_ZOOM`	[numerisk: heltal] Standard: 0	Den lägsta zoomnivån för vilken tileset tillhandahåller data. Ställs in mellan 0 och 24.
Maximala zoomnivån	`MAX_ZOOM`	[numerisk: heltal] Standard: 3	Den högsta zoomnivån för vilken tileset tillhandahåller data. Ställs in mellan 0 och 24.
Extent Valfritt	`EXTENT`	[omfattning] Standard: Inte inställd	Den maximala omfattningen av det återgivna kartområdet. Gränserna måste definiera ett område som täcks av alla zoomnivåer.
Metadata: Namn Valfritt	`META_NAME`	[sträng]	Namn på tileset
Metadata: Beskrivning Valfritt	`META_DESCRIPTI`	[sträng]	En beskrivning av innehållet i tileset
Metadata: Förtydligande Valfritt	`META_ATTRIBUTI`	[sträng]	En attributionssträng som förklarar källorna till data och/eller stilen för kartan.
Metadata: Version Valfritt	`META_VERSION`	[sträng]	Versionen av tileset. Detta hänvisar till en revidering av själva tileset, inte av MBTiles-specifikationen.
Metadata: Typ Valfritt	`META_TYPE`	[sträng]	Typ av tileset. Möjliga värden är `overlay` eller `baselayer`.
Metadata: Centrum Valfritt	`META_CENTER`	[sträng]	Centrum (sträng av kommaseparerade siffror: longitud, latitud och zoomnivå) för standardvyn av kartan. Example: `-122.1906,37.7599,11`
Destination MBTiles	`OUTPUT`	[vektorplattor] Standard: [Spara till temporär fil]	Specifikation av MBTiles-utdatafilen. *En av*: • Spara till en temporär fil • Spara till fil…

Utmatningar

Etikett	Namn	Typ	Beskrivning
Destination MBTiles	`OUTPUT`	[file]	Utmatning av vektorplattor `.mbtiles` file.

Python-kod

Algoritm-ID: `native:skrivvektortiler_mbtiles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skriva vektorplattor (XYZ)

Exporterar ett eller flera vektorlager till vector tiles, ett dataformat som är optimerat för snabb kartrendering och liten datastorlek.

Parametrar

Etikett	Namn	Typ	Beskrivning
Filmall	`XYZ_TEMPLATE`	[sträng] Standard: '{z}/{x}/{y}.pbf'	Mall för att generera vektorplattor url
Inmatningslager	`INPUT`	[vektor: valfri] [lista]	En lista över lager som ska kombineras för att generera vektorplattorna
Minsta zoomnivå	`MIN_ZOOM`	[numerisk: heltal] Standard: 0	Den lägsta zoomnivån för vilken tileset tillhandahåller data. Ställs in mellan 0 och 24.
Maximala zoomnivån	`MAX_ZOOM`	[numerisk: heltal] Standard: 3	Den högsta zoomnivån för vilken tileset tillhandahåller data. Ställs in mellan 0 och 24.
Extent Valfritt	`EXTENT`	[omfattning] Standard: Inte inställd	Den maximala omfattningen av det återgivna kartområdet. Gränserna måste definiera ett område som täcks av alla zoomnivåer.
Utmatningskatalog	`OUTPUT_DIRECTO`	[folder] Standard: [Spara i tillfällig mapp]	Specifikation av mappen för utdatavektorplattor. *En av*: • Spara till en temporär katalog • Spara till katalog

Utmatningar

Etikett	Namn	Typ	Beskrivning
Utmatningskatalog	`OUTPUT_DIRECTO`	[folder]	En mapp som innehåller olika delmängder av vektorplattorna (`.pbf`) lagrade i undermappar som motsvarar zoomnivåerna.

Python-kod

Algoritm-id: `native:writevectortiles_xyz`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.2 Leverantör av GDAL-algoritm

GDAL (Geospatial Data Abstraction Library) är ett översättningsbibliotek för raster- och vektorformat för geospatiala data. Algoritmer i Processing Framework härleds från GDAL rasterprogram och GDAL vektorprogram.

24.2.1 Rasteranalys

Bildförhållande

Genererar en aspektkarta från alla höjdraster med GDAL-stöd. Aspect är den kompassriktning som en sluttning vetter mot. Pixlarna kommer att ha ett värde från 0-360° mätt i grader från norr som anger azimut. På norra halvklotet är den norra sidan av sluttningar ofta skuggad (liten azimut från 0°-90°), medan den södra sidan får mer solstrålning (högre azimut från 180°-270°).

Denna algoritm är hämtad från GDAL DEM-verktyget.

Standardmeny: *Raster* ► *Analys*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatat höjdrasterlager
Bandnummer	`BAND`	[rasterband] Standard: 1	Numret på bandet som ska användas som höjd
Returnera trigonometrisk vinkel istället för azimut	`TRIG_ANGLE`	[boolean] Standard: Falsk	Om du aktiverar den trigonometriska vinkeln får du olika kategorier: 0° (öst), 90° (nord), 180° (väst), 270° (syd).

continues on next page

Table 24.216 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Returnera 0 för platt i stället för -9999	`ZERO_FLAT`	[boolean] Standard: Falsk	Om du aktiverar detta alternativ kommer ett 0-värde att infogas för värdet -9999 på plana ytor.
Beräkna kanter	`COMPUTE_EDGES`	[boolean] Standard: Falsk	Skapar kanter från höjdrastret
Använd Zevenbergen&Thorne formeln istället för Horns formel	`ZEVENBERGEN`	[boolean] Standard: Falsk	Aktiverar Zevenbergen&Thorne formula för mjuka landskap
Aspekt	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Rasterlager för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Aspekt	`OUTPUT`	[raster]	Utmatningsraster med vinkelvärden i grader

Python-kod

Algoritm ID: `gdal:aspekt`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Färgavlastning

Genererar en reliefkarta i färg från alla höjdraster som stöds av GDAL. Färgreliefer kan särskilt användas för att beskriva höjder. Algoritmen matar ut ett 4-bandsraster med värden som beräknats från höjden och en textbaserad färgkonfigurationsfil. Som standard blandas färgerna mellan de angivna höjdvärdena smidigt och resultatet är ett snyggt färgat höjdraster.

Denna algoritm är hämtad från GDAL DEM-verktyget.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatat höjdrasterlager
Bandnummer	`BAND`	[rasterband] Standard: 1	Numret på bandet som ska användas som höjd
Beräkna kanter	`COMPUTE_EDGES`	[boolean] Standard: Falsk	Skapar kanter från höjdrastret
Färgkonfigurationsf	`COLOR_TABLE`	&Arkiv	En textbaserad färgkonfigurationsfil
Matchningsläge	`MATCH_MODE`	[uppräkning] Standard: 2	En av: • 0 — Använd strikt färgmatchning • 1 — Använd närmaste RGBA-kvadrupler • 2 — Använd mjukt blandade färger
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ
Färgrelief	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Rasterlager för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Färgrelief	`OUTPUT`	[raster]	Ett 4-bands utmatningsraster

Python-kod

Algoritm-ID: `gdal:färgavlastning`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Fylla NoData

Fyll rasterregioner med NoData-värden genom interpolering från kanterna. Värdena för NoData-regionerna beräknas utifrån de omgivande pixelvärdena med hjälp av invers avståndsviktning. Efter interpoleringen sker en utjämning av resultaten. Indata kan vara vilket GDAL-stöttat rasterlager som helst. Den här algoritmen är i allmänhet lämplig för interpolering av saknade regioner i raster som varierar ganska kontinuerligt (t.ex. höjdmodeller). Den är också lämplig för att fylla små hål och sprickor i mer oregelbundet varierande bilder (t.ex. flygfoton). Det är i allmänhet inte så bra för att interpolerar ett raster från glesa punktdata.

Denna algoritm är hämtad från verktyget GDAL fillnodata.

Standardmeny: *Raster* ► *Analys*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatat rasterlager
Bandnummer	`BAND`	[rasterband] Standard: 1	Det band som ska användas. NoData-värden måste representeras av värdet 0.
Maximalt avstånd (i pixlar) för att söka efter värden att interpolera	`DISTANCE`	[numerisk: heltal] Standard: 10	Antalet pixlar som ska sökas i alla riktningar för att hitta värden att interpoleras från
Antal utjämningsiteration som ska köras efter interpoleringen	`ITERATIONS`	[numerisk: heltal] Standard: 0	Antalet 3x3-filterpass som ska köras (0 eller fler) för att jämna ut resultatet av interpoleringen.
Validitetsmask	`MASK_LAYER`	[raster]	Ett rasterlager som definierar de områden som ska fyllas.
Befylld	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av rasterlagret för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Befylld	`OUTPUT`	[raster]	Utgångsraster

Python-kod

Algoritm ID: `gdal:fillnodata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Grid (datamätningar)

Beräknar vissa datamått med hjälp av den angivna geometrin för fönster och utdatagaller.

Den här algoritmen är hämtad från GDAL grid utility.

Standardmeny: *Raster* ► *Analys*

Se även

GDAL grid handledning

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Denna algoritm genererar ett nytt lager baserat på ett befintligt, med en annan typ av geometri.	`INPUT`	[vektor: punkt]	Vektorlager för inmatningspunkt
Datamätning som ska användas	`METRIC`	[uppräkning] Standard: 0	En av: • 0 — Minimum, minsta värde som hittats i ellipsen för sökning av gridnod • 1 — Maximum, maximalt värde som hittats i ellipsen för sökning av gridnod • 2 — Range, en skillnad mellan minimi- och maximivärdena i ellipsen för sökning av gridnod • 3 — Count, ett antal datapunkter som hittats i ellipsen för sökning av gridnod • 4 — Medelavstånd, ett medelavstånd mellan gridnoden (mitten av sökellipsen) och alla datapunkter som finns i gridnodens sökellips • 5 — Genomsnittligt avstånd mellan punkter, ett genomsnittligt avstånd mellan de datapunkter som finns i ellipsen för gridnodssökning. Avståndet mellan varje punktpar inom ellipsen beräknas och genomsnittet av alla avstånd anges som ett värde för gridnoden
Den första radien för sökellips	`RADIUS_1`	[numerisk: dubbel] Standard: 0,0	Den första radien (X-axeln om rotationsvinkeln är 0) för sökellipsens
Den andra radien för sökellips	`RADIUS_2`	[numerisk: dubbel] Standard: 0,0	Den andra radien (Y-axeln om rotationsvinkeln är 0) för sökellipsens
Rotationsvinkel för sökellips i grader (moturs)	`ANGLE`	[numerisk: dubbel] Standard: 0,0	Vinkel för ellipsens rotation i grader. Ellipsen roterad moturs.
Minsta antal datapunkter som ska användas	`MIN_POINTS`	[numerisk: dubbel] Standard: 0,0	Minsta antal datapunkter att beräkna medelvärdet för. Om ett mindre antal punkter hittas anses rutnätsnoden vara tom och fylls med NoData-markören.
Inga uppgifter	`NODATA`	[numerisk: dubbel] Standard: 0,0	NoData-markör för att fylla tomma punkter
Interpolerad (datamätningar)	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange det utgående rasterlagret med interpolerade värden. *En av*: • Spara till en temporär fil • Spara till fil…

continues on next page

Table 24.221 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Z-värde från fältet Valfritt	`Z_FIELD`	[tabellfält: numerisk]	Fält för interpolering
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ
Typ av utdata	`DATA_TYPE`	[uppräkning] Standard: 5	Definierar datatypen för den utgående rasterfilen. Alternativ för: • 0 — Byte (Åtta bitars osignerat heltal (quint8)) • 1 — Int16 (Sexton bitars signerat heltal (qint16)) • 2 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 3 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 4 — Int32 (trettio tvåbitars signerade heltal (qint32)) • 5 — Float32 (trettio tvåbitars flyttal (float)) • 6 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 7 — CInt16 (Komplex Int16) • 8 — CInt32 (Komplex Int32) • 9 — CFloat32 (komplex float32) • 10 — CFloat64 (komplex float64) • 11 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)

Utgångar

Etikett	Namn	Typ	Beskrivning
Interpolerad (datamätningar)	`OUTPUT`	[raster]	Utmatningsraster med interpolerade värden

Python-kod

Algoritm ID: `gdal:griddatametrics`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rutnät (IDW med närmaste granne-sökning)

Beräknar det omvända avståndet till en Power gridding i kombination med närmaste granne-metoden. Idealisk när ett maximalt antal datapunkter ska användas.

Den här algoritmen är hämtad från GDAL grid utility.

Se även

GDAL grid handledning

Parametrar

Etikett	Namn	Typ	Beskrivning
Denna algoritm genererar ett nytt lager baserat på ett befintligt, med en annan typ av geometri.	`INPUT`	[vektor: punkt]	Vektorlager för inmatningspunkt
Vägningskraft	`POWER`	[numerisk: dubbel] Standard: 2,0	Viktningseffekt
Utjämning	`SMOOTHING`	[numerisk: dubbel] Standard: 0,0	Parameter för utjämning
Radien för sökcirkeln	`RADIUS`	[numerisk: dubbel] Standard: 1,0	Radien för sökcirkeln
Maximalt antal datapunkter som ska användas	`MAX_POINTS`	[numerisk: heltal] Standard: 12	Sök inte efter fler poäng än detta antal.
Minsta antal datapunkter som ska användas	`MIN_POINTS`	[numerisk: heltal] Standard: 0	Minsta antal datapunkter att beräkna medelvärdet för. Om ett mindre antal punkter hittas anses rutnätsnoden vara tom och fylls med NoData-markören.

continues on next page

Table 24.223 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Inga uppgifter	`NODATA`	[numerisk: dubbel] Standard: 0,0	NoData-markör för att fylla tomma punkter
Interpolerad (IDW med NN-sökning)	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange det utgående rasterlagret med interpolerade värden. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Z-värde från fältet Valfritt	`Z_FIELD`	[tabellfält: numerisk]	Fält för interpolering
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ
Typ av utdata	`DATA_TYPE`	[uppräkning] Standard: 5	Definierar datatypen för den utgående rasterfilen. Alternativ för: • 0 — Byte (Åtta bitars osignerat heltal (quint8)) • 1 — Int16 (Sexton bitars signerat heltal (qint16)) • 2 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 3 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 4 — Int32 (trettio tvåbitars signerade heltal (qint32)) • 5 — Float32 (trettio tvåbitars flyttal (float)) • 6 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 7 — CInt16 (Komplex Int16) • 8 — CInt32 (Komplex Int32) • 9 — CFloat32 (komplex float32) • 10 — CFloat64 (komplex float64) • 11 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)

Utgångar

Etikett	Namn	Typ	Beskrivning
Interpolerad (IDW med NN-sökning)	`OUTPUT`	[raster]	Utmatningsraster med interpolerade värden

Python-kod

Algoritm-ID: `gdal:gridinversedistancenearestneighbor`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Grid (omvänt avstånd till en kraft)

Gridningsmetoden Inverse Distance to a Power är en viktad medelvärdesinterpolator.

Du bör förse indmatningsarrayerna med de spridda datavärdena, inklusive koordinater för varje datapunkt och geometri för utmatningsgallret. Funktionen beräknar ett interpolerat värde för den givna positionen i utmatningsgallret.

Den här algoritmen är hämtad från GDAL grid utility.

Standardmeny: *Raster* ► *Analys*

Se även

GDAL grid handledning

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Denna algoritm genererar ett nytt lager baserat på ett befintligt, med en annan typ av geometri.	`INPUT`	[vektor: punkt]	Vektorlager för inmatningspunkt
Vägningskraft	`POWER`	[numerisk: dubbel] Standard: 2,0	Viktningseffekt
Smygande	`SMOOTHING`	[numerisk: dubbel] Standard: 0,0	Parameter för utjämning
Den första radien för sökellips	`RADIUS_1`	[numerisk: dubbel] Standard: 0,0	Den första radien (X-axeln om rotationsvinkeln är 0) för sökellipsens

continues on next page

Table 24.225 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Den andra radien för sökellips	`RADIUS_2`	[numerisk: dubbel] Standard: 0,0	Den andra radien (Y-axeln om rotationsvinkeln är 0) för sökellipsens
Rotationsvinkel för sökellips i grader (moturs)	`ANGLE`	[numerisk: dubbel] Standard: 0,0	Vinkel för ellipsens rotation i grader. Ellipsen roterad moturs.
Maximalt antal datapunkter som ska användas	`MAX_POINTS`	[numerisk: heltal] Standard: 0	Sök inte efter fler poäng än detta antal.
Minsta antal datapunkter som ska användas	`MIN_POINTS`	[numerisk: heltal] Standard: 0	Minsta antal datapunkter att beräkna medelvärdet för. Om ett mindre antal punkter hittas anses rutnätsnoden vara tom och fylls med NoData-markören.
Inga uppgifter	`NODATA`	[numerisk: dubbel] Standard: 0,0	NoData-markör för att fylla tomma punkter
Interpolerad (IDW)	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange det utgående rasterlagret med interpolerade värden. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Z-värde från fältet Valfritt	`Z_FIELD`	[tabellfält: numerisk]	Fält för interpolering
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ

continues on next page

Table 24.226 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Typ av utdata	`DATA_TYPE`	[uppräkning] Standard: 5	Definierar datatypen för den utgående rasterfilen. Alternativ för: • 0 — Byte (Åtta bitars osignerat heltal (quint8)) • 1 — Int16 (Sexton bitars signerat heltal (qint16)) • 2 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 3 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 4 — Int32 (trettio tvåbitars signerade heltal (qint32)) • 5 — Float32 (trettio tvåbitars flyttal (float)) • 6 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 7 — CInt16 (Komplex Int16) • 8 — CInt32 (Komplex Int32) • 9 — CFloat32 (komplex float32) • 10 — CFloat64 (komplex float64) • 11 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)

Utgångar

Etikett	Namn	Typ	Beskrivning
Interpolerad (IDW)	`OUTPUT`	[raster]	Utmatningsraster med interpolerade värden

Python-kod

Algoritm-ID: `gdal:gridinversedistance`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rutnät (linjärt)

Linear-metoden utför linjär interpolation genom att beräkna en Delaunay-triangulering av punktmolnet, ta reda på i vilken triangel i trianguleringen punkten befinner sig och utföra linjär interpolation från dess barycentriska koordinater i triangeln. Om punkten inte ligger i någon triangel, beroende på radien, kommer algoritmen att använda värdet för den närmaste punkten eller NoData-värdet.

Den här algoritmen är hämtad från GDAL grid utility.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Denna algoritm genererar ett nytt lager baserat på ett befintligt, med en annan typ av geometri.	`INPUT`	[vektor: punkt]	Vektorlager för inmatningspunkt
Sökavstånd	`RADIUS`	[numerisk: dubbel] Standard: -1,0	Om den punkt som ska interpoleras inte passar in i en triangel i Delaunay-trianguleringen används det maximala avståndet för att söka efter en närmaste granne, annars används NoData. Om värdet är satt till `-1` är sökavståndet oändligt. Om det sätts till `0` kommer NoData-värdet att användas.
Inga uppgifter	`NODATA`	[numerisk: dubbel] Standard: 0,0	NoData-markör för att fylla tomma punkter
Interpolerad (linjär)	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange det utgående rasterlagret med interpolerade värden. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Z-värde från fältet Valfritt	`Z_FIELD`	[tabellfält: numerisk]	Fält för interpolering
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

continues on next page

Table 24.228 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ
Typ av utdata	`DATA_TYPE`	[uppräkning] Standard: 5	Definierar datatypen för den utgående rasterfilen. Alternativ för: • 0 — Byte (Åtta bitars osignerat heltal (quint8)) • 1 — Int16 (Sexton bitars signerat heltal (qint16)) • 2 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 3 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 4 — Int32 (trettio tvåbitars signerade heltal (qint32)) • 5 — Float32 (trettio tvåbitars flyttal (float)) • 6 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 7 — CInt16 (Komplex Int16) • 8 — CInt32 (Komplex Int32) • 9 — CFloat32 (komplex float32) • 10 — CFloat64 (komplex float64) • 11 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)

Utgångar

Etikett	Namn	Typ	Beskrivning
Interpolerad (linjär)	`OUTPUT`	[raster]	Utmatningsraster med interpolerade värden

Python-kod

Algoritm-ID: `gdal:gridlinear`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rutnät (glidande medelvärde)

Moving Average är en enkel algoritm för medelvärdesberäkning av data. Den använder ett rörligt fönster av elliptisk form för att söka värden och beräknar medelvärdet för alla datapunkter inom fönstret. Sökellipsen kan roteras med angiven vinkel, ellipsens centrum ligger vid rutnätsnoden. Det minsta antalet datapunkter som ska medelvärdesbildas kan också ställas in. Om det inte finns tillräckligt många punkter i fönstret betraktas rutnätsnoden som tom och fylls med det angivna NoData-värdet.

Den här algoritmen är hämtad från GDAL grid utility.

Standardmeny: *Raster* ► *Analys*

Se även

GDAL grid handledning

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Denna algoritm genererar ett nytt lager baserat på ett befintligt, med en annan typ av geometri.	`INPUT`	[vektor: punkt]	Vektorlager för inmatningspunkt
Den första radien för sökellips	`RADIUS_1`	[numerisk: dubbel] Standard: 0,0	Den första radien (X-axeln om rotationsvinkeln är 0) för sökellipsens
Den andra radien för sökellips	`RADIUS_2`	[numerisk: dubbel] Standard: 0,0	Den andra radien (Y-axeln om rotationsvinkeln är 0) för sökellipsens
Rotationsvinkel för sökellips i grader (moturs)	`ANGLE`	[numerisk: dubbel] Standard: 0,0	Vinkel för ellipsens rotation i grader. Ellipsen roterad moturs.
Minsta antal datapunkter som ska användas	`MIN_POINTS`	[numerisk: heltal] Standard: 0	Minsta antal datapunkter att beräkna medelvärdet för. Om ett mindre antal punkter hittas anses rutnätsnoden vara tom och fylls med NoData-markören.
Inga uppgifter	`NODATA`	[numerisk: dubbel] Standard: 0,0	NoData-markör för att fylla tomma punkter
Interpolerad (glidande medelvärde)	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange rasterlagret för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Z-värde från fältet Valfritt	`Z_FIELD`	[tabellfält: numerisk]	Fält för interpolering
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ
Typ av utdata	`DATA_TYPE`	[uppräkning] Standard: 5	Definierar datatypen för den utgående rasterfilen. Alternativ för: • 0 — Byte (Åtta bitars osignerat heltal (quint8)) • 1 — Int16 (Sexton bitars signerat heltal (qint16)) • 2 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 3 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 4 — Int32 (trettio tvåbitars signerade heltal (qint32)) • 5 — Float32 (trettio tvåbitars flyttal (float)) • 6 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 7 — CInt16 (Komplex Int16) • 8 — CInt32 (Komplex Int32) • 9 — CFloat32 (komplex float32) • 10 — CFloat64 (komplex float64) • 11 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)

Utgångar

Etikett	Namn	Typ	Beskrivning
Interpolerad (glidande medelvärde)	`OUTPUT`	[raster]	Utmatningsraster med interpolerade värden

Python-kod

Algoritm-ID: `gdal:gridaverage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Grid (Närmaste granne)

Metoden Nearest Neighbor utför ingen interpolering eller utjämning, den tar bara värdet för den närmaste punkten som hittas i ellipsen för sökning av rutnätsnoden och returnerar det som ett resultat. Om inga punkter hittas returneras det angivna NoData-värdet.

Den här algoritmen är hämtad från GDAL grid utility.

Standardmeny: *Raster* ► *Analys*

Se även

GDAL grid handledning

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Denna algoritm genererar ett nytt lager baserat på ett befintligt, med en annan typ av geometri.	`INPUT`	[vektor: punkt]	Vektorlager för inmatningspunkt
Den första radien för sökellips	`RADIUS_1`	[numerisk: dubbel] Standard: 0,0	Den första radien (X-axeln om rotationsvinkeln är 0) för sökellipsens
Den andra radien för sökellips	`RADIUS_2`	[numerisk: dubbel] Standard: 0,0	Den andra radien (Y-axeln om rotationsvinkeln är 0) för sökellipsens
Rotationsvinkel för sökellips i grader (moturs)	`ANGLE`	[numerisk: dubbel] Standard: 0,0	Vinkel för ellipsens rotation i grader. Ellipsen roterad moturs.

continues on next page

Table 24.231 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Inga uppgifter	`NODATA`	[numerisk: dubbel] Standard: 0,0	NoData-markör för att fylla tomma punkter
Interpolerad (närmaste granne)	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange det utgående rasterlagret med interpolerade värden. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Z-värde från fältet Valfritt	`Z_FIELD`	[tabellfält: numerisk]	Fält för interpolering
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ
Typ av utdata	`DATA_TYPE`	[uppräkning] Standard: 5	Definierar datatypen för den utgående rasterfilen. Alternativ för: • 0 — Byte (Åtta bitars osignerat heltal (quint8)) • 1 — Int16 (Sexton bitars signerat heltal (qint16)) • 2 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 3 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 4 — Int32 (trettio tvåbitars signerade heltal (qint32)) • 5 — Float32 (trettio tvåbitars flyttal (float)) • 6 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 7 — CInt16 (Komplex Int16) • 8 — CInt32 (Komplex Int32) • 9 — CFloat32 (komplex float32) • 10 — CFloat64 (komplex float64) • 11 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)

Utgångar

Etikett	Namn	Typ	Beskrivning
Interpolerad (närmaste granne)	`OUTPUT`	[raster]	Utmatningsraster med interpolerade värden

Python-kod

Algoritm-ID: `gdal:gridnearestneighbor`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Terrängskuggning

Utmatar ett raster med en fin skuggad relief-effekt. Det är mycket användbart för att visualisera terrängen. Du kan eventuellt ange ljuskällans azimut och höjd, en vertikal överdimensioneringsfaktor och en skalningsfaktor för att ta hänsyn till skillnader mellan vertikala och horisontella enheter.

Denna algoritm härrör från verktyget GDAL DEM .

Standardmeny: *Raster* ► *Analys*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Ingång Höjd rasterlager
Bandnummer	`BAND`	[rasterband] Standard: 1	Band som innehåller information om höjden
Z-faktor (vertikal överdrift)	`Z_FACTOR`	[numerisk: dubbel] Standard: 1,0	Faktorn överdriver höjden på det utgående höjdrastret
Skala (förhållandet mellan vertikala och horisontella enheter)	`SCALE`	[numerisk: dubbel] Standard: 1,0	Förhållandet mellan vertikala enheter och horisontella enheter
Ljusets azimut	`AZIMUTH`	[numerisk: dubbel] Standard: 315.0	Definierar azimuten för ljuset som skiner på höjdrastret i grader. Om det kommer från toppen av rastret är värdet 0, om det kommer från öster är det 90 a.s.o.
Ljusets höjd över havet	`ALTITUDE`	[numerisk: dubbel] Standard: 45,0	Definierar ljusets höjd i grader. 90 om ljuset kommer från ovanför höjdrastret, 0 om det är rakljus.
Beräkna kanter	`COMPUTE_EDGES`	[boolean] Standard: Falsk	Skapar kanter från höjdrastret

continues on next page

Table 24.233 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Använd Zevenbergen&Thor formeln (i stället för Horns formel)	`ZEVENBERGEN`	[boolean] Standard: Falsk	Aktiverar Zevenbergen&Thorne formula för mjuka landskap
Kombinerad skuggning	`COMBINED`	[boolean] Standard: Falsk	
Skuggning i flera riktningar	`MULTIDIRECTION`	[boolean] Standard: Falsk	
Terrängskuggning	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange det utgående rasterlagret med interpolerade värden. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Terrängskuggning	`OUTPUT`	[raster]	Utmatningsraster med interpolerade värden

Python-kod

Algoritm-id: `gdal:hillshade`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Nära svart

Konverterar nästan svartvita kanter till svarta.

Den här algoritmen skannar en bild och försöker ställa in alla pixlar som är nästan eller exakt svarta, vita eller en eller flera anpassade färger runt kragen till svart eller vitt. Detta används ofta för att ”fixa till” komprimerade luftfotografier med förlust så att färgpixlar kan behandlas som transparenta vid mosaikläggning.

Den här algoritmen är hämtad från GDAL nearblack utility.

Standardmeny: *Raster* ► *Analys*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Ingång Höjd rasterlager
Hur långt från svart (vitt)	`NEAR`	[numerisk: heltal] Standard: 15	Välj hur långt från svart, vitt eller anpassade färger pixelvärdena kan vara och fortfarande anses vara nära svart, vitt eller anpassad färg.
Sök efter nästan vita pixlar istället för nästan svarta	`WHITE`	[boolean] Standard: Falsk	Sök efter nästan vita (255) pixlar i stället för nästan svarta pixlar
Nästan svart	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange rasterlagret för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Nästan svart	`OUTPUT`	[raster]	Utgångsraster

Python-kod

Algoritm ID: `gdal:nära svart`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Närhet (rasteravstånd)

Skapar en närhetskarta för raster som anger avståndet från mitten av varje pixel till mitten av den närmaste pixel som identifierats som en målpixel. Målpixlar är de i källrastret för vilka rasterpixelvärdet finns i uppsättningen av målpixelvärden.

Den här algoritmen är hämtad från GDAL proximity utility.

Standardmeny: *Raster* ► *Analys*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Ingång Höjd rasterlager
Bandnummer	`BAND`	[rasterband] Standard: 1	Band som innehåller information om höjden
En lista över pixelvärden i källbilden som ska betraktas som målpixlar Valfritt	`VALUES`	[sträng] Standard: ”	En lista med målpixelvärden i källbilden som ska betraktas som målpixlar. Om inget anges kommer alla pixlar som inte är noll att betraktas som målpixlar.
Avståndsenheter	`UNITS`	[uppräkning] Standard: 1	Ange om de genererade avstånden ska vara i pixel- eller georefererade koordinater. Ett av följande: • 0 — Georefererade koordinater • 1 — Pixelkoordinater

continues on next page

Table 24.237 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Det maximala avstånd som ska genereras Valfritt	`MAX_DISTANCE`	[numerisk: dubbel] Standard: 0,0	Det maximala avstånd som ska genereras. NoData-värdet kommer att användas för pixlar bortom detta avstånd. Om ett NoData-värde inte anges kommer utgångsbandet att tillfrågas om sitt NoData-värde. Om utmatningsbandet inte har något NoData-värde, används värdet 65535. Avståndet tolkas enligt värdet för *Distance units*.
Värde som ska tillämpas på alla pixlar som ligger inom maxdist från målpixlarna Valfritt	`REPLACE`	[numerisk: dubbel] Standard: 0,0	Ange ett värde som ska tillämpas på alla pixlar som är närmare än det maximala avståndet från målpixlarna (inklusive målpixlarna) i stället för ett avståndsvärde.
NoData-värde att använda för destinationens närhetsraster Valfritt	`NODATA`	[numerisk: dubbel] Standard: 0,0	Ange det NoData-värde som ska användas för utdatarastern
Närhetskarta	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange rasterlagret för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ

continues on next page

Table 24.238 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Typ av utdata	`DATA_TYPE`	[uppräkning] Standard: 5	Definierar datatypen för den utgående rasterfilen. Alternativ för: • 0 — Byte (Åtta bitars osignerat heltal (quint8)) • 1 — Int16 (Sexton bitars signerat heltal (qint16)) • 2 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 3 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 4 — Int32 (trettio tvåbitars signerade heltal (qint32)) • 5 — Float32 (trettio tvåbitars flyttal (float)) • 6 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 7 — CInt16 (Komplex Int16) • 8 — CInt32 (Komplex Int32) • 9 — CFloat32 (komplex float32) • 10 — CFloat64 (komplex float64) • 11 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)

Utgångar

Etikett	Namn	Typ	Beskrivning
Närhetskarta	`OUTPUT`	[raster]	Utgångsraster

Python-kod

Algoritm-ID: `gdal:närhet`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Grovhet

Utmatar ett raster med enstaka band med värden som beräknats från höjden. Grovhet är graden av oregelbundenhet hos ytan. Den beräknas genom den största skillnaden mellan cellerna i en central pixel och dess omgivande cell. Bestämningen av ojämnheten spelar en roll i analysen av terränghöjdsdata, den är användbar för beräkningar av flodmorfologin, i klimatologi och fysisk geografi i allmänhet.

Denna algoritm är hämtad från GDAL DEM-verktyget.

Standardmeny: *Raster* ► *Analys*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatat höjdrasterlager
Bandnummer	`BAND`	[rasterband] Standard: 1	Numret på bandet som ska användas som höjd
Beräkna kanter	`COMPUTE_EDGES`	[boolean] Standard: Falsk	Skapar kanter från höjdrastret
Roughness	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange rasterlagret för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Roughness	`OUTPUT`	[raster]	Raster för ojämnhet i utmatning med en bandbredd. Värdet -9999 används som NoData-värde.

Python-kod

Algoritm-ID: `gdal:ojämnhet`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Sikt

Tar bort rasterpolygoner som är mindre än en angiven tröskelstorlek (i pixlar) och ersätter dem med pixelvärdet för den största grannpolygonen. Det är användbart om du har en stor mängd små områden på din rasterkarta.

Den här algoritmen är hämtad från GDAL sieve utility.

Standardmeny: *Raster* ► *Analys*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatat höjdrasterlager
Tröskelvärde	`THRESHOLD`	[numerisk: heltal] Standard: 10	Endast rasterpolygoner som är mindre än denna storlek kommer att tas bort
Använd 8-anslutning	`EIGHT_CONNECTE`	[boolean] Standard: Falsk	Om False, betraktas endast de (4) pixlar som förbinder kanterna horisontellt eller vertikalt som grannar. Om True, betraktas varje pixel som vidrör en kant eller ett hörn som granne.
Använd inte standardvaliditetsm för inmatningsbandet	`NO_MASK`	[boolean] Standard: Falsk	
Validitetsmask Valfritt	`MASK_LAYER`	[raster]	Giltighetsmask som ska användas istället för standardvärdet
Siktad	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange rasterlagret för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Siktad	`OUTPUT`	[raster]	Utgående rasterlager.

Python-kod

Algoritm ID: `gdal:sieve`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Lutning

Genererar en lutningskarta från alla GDAL-stödda höjdraster. Lutning är lutningsvinkeln mot horisontalplanet. Du har möjlighet att ange vilken typ av lutningsvärde du vill ha: grader eller procentuell lutning.

Denna algoritm är hämtad från GDAL DEM-verktyget.

Standardmeny: *Raster* ► *Analys*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Ingång Höjd rasterlager
Bandnummer	`BAND`	[rasterband] Standard: 1	Band som innehåller information om höjden
Förhållandet mellan vertikala enheter och horisontella	`SCALE`	[numerisk: dubbel] Standard: 1,0	Förhållandet mellan vertikala enheter och horisontella enheter
Lutning uttryckt i procent (i stället för grader)	`AS_PERCENT`	[boolean] Standard: Falsk	Uttryck lutning som procent istället för grader

continues on next page

Table 24.243 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Beräkna kanter	`COMPUTE_EDGES`	[boolean] Standard: Falsk	Skapar kanter från höjdrastret
Använd Zevenbergen&Thorn formeln (i stället för Horns formel)	`ZEVENBERGEN`	[boolean] Standard: Falsk	Aktiverar Zevenbergen&Thorne formula för mjuka landskap
Slope	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange rasterlagret för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Slope	`OUTPUT`	[raster]	Utgångsraster

Python-kod

Algoritm-ID: `gdal:sluttning`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Index för terrängens robusthet (TRI)

Ger ett raster med enstaka band med värden som beräknats från höjden. TRI står för Terrain Ruggedness Index, som definieras som den genomsnittliga skillnaden mellan en central pixel och dess omgivande celler.

Denna algoritm är hämtad från GDAL DEM-verktyget.

Standardmeny: *Raster* ► *Analys*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatat höjdrasterlager
Bandnummer	`BAND`	[rasterband] Standard: 1	Numret på bandet som ska användas som höjd
Beräkna kanter	`COMPUTE_EDGES`	[boolean] Standard: Falsk	Skapar kanter från höjdrastret
Terrains robusthetsindex	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange rasterlagret för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Terrains robusthetsindex	`OUTPUT`	[raster]	Raster för utmatningens robusthet. Värdet -9999 används som NoData-värde.

Python-kod

Algoritm ID: `gdal:triterrainruggednessindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Index för topografisk position (TPI)

Utmatar ett raster med enstaka band med värden som beräknats från höjden. TPI står för Topographic Position Index, som definieras som skillnaden mellan en central pixel och medelvärdet av dess omgivande celler.

Denna algoritm är hämtad från GDAL DEM-verktyget.

Standardmeny: *Raster* ► *Analys*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatat höjdrasterlager
Bandnummer	`BAND`	[rasterband] Standard: 1	Numret på det band som ska användas för höjdvärden
Beräkna kanter	`COMPUTE_EDGES`	[boolean] Standard: Falsk	Skapar kanter från höjdrastret
Terrains robusthetsindex	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange rasterlagret för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Terrains robusthetsindex	`OUTPUT`	[raster]	Utgångsraster.

Python-kod

Algoritm-ID: `gdal:tpitopographicpositionindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.2.2 Rasterkonvertering

gdal2xyz

Konverterar rasterdata till XYZ ASCII-filformat.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Rasterlager att konvertera
Bandnummer	`BAND`	[rasterband] Standard: Det första bandet i inmatningslagret	Om rastret är flerbandigt väljer du det band du vill konvertera
Inmatat pixelvärde som ska behandlas som NoData Valfritt	`NODATA_INPUT`	[numerisk: heltal] Standard: Ej inställd	Inmatat pixelvärde som ska behandlas som ”NoData” (GDAL >= 3.7).
Destination NoData Valfritt	`NODATA_OUTPUT`	[numerisk: heltal] Standard: Ej inställd	Tilldela specificerat ”NoData”-värde till utgången (GDAL >= 3.7).
Skriv inte ut NoData-värden	`SKIP_NODATA`	[boolean] Standard: Falsk	Skicka inte ut ”NoData”-värden (GDAL >= 3.3).
Utdata kommaseparerade värden	`CSV`	[boolean] Standard: Falsk	Anger om utdatafilen ska vara av typen csv (comma-separated values).
XYZ ASCII-fil	`OUTPUT`	&Arkiv Standard: `[Spara till temporär fil]`	Specifikation av utdatafilen. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
XYZ ASCII-fil	`INPUT`	[tabell]	Tabellfil som innehåller de värden som exporterats från rasterbandet.

Python-kod

Algoritm-ID: `gdal:gdal2xyz`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

PCT till RGB

Konverterar en 8 bitars paletterad bild till 24 bitars RGB. Den konverterar ett pseudofärgband från indatafilen till en RGB-fil i önskat format.

Denna algoritm är hämtad från GDAL pct2rgb utility.

Standardmeny: *Raster* ► *Konvertering*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatning 8 bitars rasterbild
Bandnummer	`BAND`	[rasterband] Standard: Det första bandet i inmatningslagret	Om rastret är flerbandigt väljer du det band du vill konvertera
Genererar en RGBA-fil	`RGBA`	[boolean] Standard: Falsk	Anger om utdatafilen ska vara av typen RGBA.
PCT till RGB	`OUTPUT`	&Arkiv Standard: `[Spara till temporär fil]`	Specifikation av utdatafilen. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
PCT till RGB	`OUTPUT`	[raster]	24 bitars RGB-rasterbild

Python-kod

Algoritm-ID: `gdal:pcttorgb`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Polygonisera (raster till vektor)

Skapar vektorpolygoner för alla anslutna regioner av pixlar i rastret som delar ett gemensamt pixelvärde. Varje polygon skapas med ett attribut som anger polygonens pixelvärde.

Den här algoritmen är hämtad från GDAL polygonize utility.

Standardmeny: *Raster* ► *Konvertering*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatat rasterlager
Bandnummer	`BAND`	[rasterband] Standard: Det första bandet i inmatningslagret	Om rastret är multibandigt väljer du det band du vill använda
Namn på det fält som ska skapas	`FIELD`	[sträng] Standard: ”DN	Ange fältnamnet för attributen för de anslutna regionerna.
Använd 8-anslutning	`EIGHT_CONNECTE`	[boolean] Standard: Falsk	Om den inte är inställd måste rastercellerna ha en gemensam gräns för att anses vara anslutna (*4-connected*). Om den är inställd anses även rasterceller som berör varandra vara anslutna (*8-connected*).
Vektoriserad	`OUTPUT`	[vektor: polygon] Standard: `[Spara till temporär fil]`	Specifikation av det utgående (polygon-) vektorlagret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Vektoriserad	`OUTPUT`	[vektor: polygon]	Utgående vektorlager

Python-kod

Algoritm-ID: `gdal:polygonize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Omorganisera band

Skapar ett nytt raster med hjälp av valda band från ett givet rasterlager. Algoritmen gör det också möjligt att omordna banden för det nyskapade rastret.

Denna algoritm är hämtad från GDAL translate utility.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatat rasterlager
Utvalda band	`BANDS`	[rasterband] [lista] Standard: Ej inställd	Ordnad lista över de band som ska användas för att skapa det nya rastret
Konverterad	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	OPTIONS	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).
Typ av utdata	DATA_TYPE	[uppräkning] Standard: 0	Definierar datatypen för den utgående rasterfilen. Alternativ för: • 0 — Använd datatyp för inmatningslager • 1 — Byte (Åtta bitars osignerat heltal (quint8)) • 2 — Int16 (Sexton bitars signerat heltal (qint16)) • 3 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 4 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 5 — Int32 (Trettio tvåbitars signerade heltal (qint32)) • 6 — Float32 (trettio tvåbitars flyttal (float)) • 7 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 8 — CInt16 (Komplex Int16) • 9 — CInt32 (Komplex Int32) • 10 — CFloat32 (komplex float32) • 11 — CFloat64 (komplex float64) • 12 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)

Utgångar

Etikett	Namn	Typ	Beskrivning
Konverterad	OUTPUT	[raster]	Utmatning av rasterlager med omarrangerade band.

Python-kod

Algoritm-ID: `gdal:rearrange_bands`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

RGB till PCT

Konverterar en 24 bitars RGB-bild till en 8 bitars palett. Beräknar en optimal pseudofärgtabell för den givna RGB-bilden med hjälp av en medianskärningsalgoritm på ett nedsamplat RGB-histogram. Sedan konverteras bilden till en pseudofärgad bild med hjälp av färgtabellen. Vid konverteringen används Floyd-Steinberg-dithering (feldiffusion) för att maximera den visuella kvaliteten på den utgående bilden.

Om du vill klassificera en rasterkarta och vill minska antalet klasser kan det vara bra att nedprova din bild med den här algoritmen innan.

Denna algoritm är hämtad från GDAL rgb2pct utility.

Standardmeny: *Raster* ► *Konvertering*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Ingång (RGB) rasterlager
Antal färger	`NCOLORS`	[numerisk: heltal] Standard: 2	Antalet färger som den resulterande bilden kommer att innehålla. Ett värde från 2-256 är möjligt.
RGB till PCT	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsrastret. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
RGB till PCT	`OUTPUT`	[raster]	Utgående rasterlager.

Python-kod

Algoritm-ID: `gdal:rgbtopct`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Översätt (konvertera format)

Konverterar rasterdata mellan olika format.

Denna algoritm är hämtad från GDAL translate utility.

Standardmeny: *Raster* ► *Konvertering*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatat rasterlager
Åsidosätt projiceringen av utdatafilen Valfritt	`TARGET_CRS`	[crs]	Ange en projektion för utdatafilen
Tilldela ett specificerat NoData-värde till utgångsbanden Valfritt	`NODATA`	[numerisk: dubbel] Standard: Ej inställd	Definierar det värde som ska användas för NoData i utdatarastern
Kopiera alla underdataset i den här filen till enskilda utdatafiler	`COPY_SUBDATASE`	[boolean] Standard: Falsk	Skapa enskilda filer för underdatauppsättningar
Konverterad	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av det utgående (översatta) rasterlagret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ
Typ av utdata	`DATA_TYPE`	[uppräkning] Standard: 0	Definierar datatypen för den utgående rasterfilen. Alternativ för: • 0 — Använd datatyp för inmatningslager • 1 — Byte (Åtta bitars osignerat heltal (quint8)) • 2 — Int16 (Sexton bitars signerat heltal (qint16)) • 3 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 4 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 5 — Int32 (Trettio tvåbitars signerade heltal (qint32)) • 6 — Float32 (trettio tvåbitars flyttal (float)) • 7 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 8 — CInt16 (Komplex Int16) • 9 — CInt32 (Komplex Int32) • 10 — CFloat32 (komplex float32) • 11 — CFloat64 (komplex float64) • 12 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)

Utgångar

Etikett	Namn	Typ	Beskrivning
Konverterad	`OUTPUT`	[raster]	Utgående (översatt) rasterlager.

Python-kod

Algoritm-ID: `gdal:translate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.2.3 Extrahering av raster

Klipp raster efter utsträckning

Klipper ut en GDAL-stödd rasterfil till en viss utsträckning.

Denna algoritm är hämtad från GDAL translate utility.

Standardmeny: *Raster* ► *Extraktion*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Det ingående rastret

continues on next page

Table 24.255 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Klippningsgrad	`EXTENT`	[omfattning]	Utsträckning som ska användas för utdatarastret. Endast pixlar inom den angivna avgränsningsrutan kommer att ingå i utdata. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Overstyra projektionen för utdatafilen	`OVERCRS`	[boolean] Standard: Falsk	Om markerat tilldelas utdatafilen CRS för inmatningslagret.
Tilldela ett specificerat NoData-värde till utgångsbanden Valfritt	`NODATA`	[numerisk: dubbel] Standard: Ej inställd	Definierar ett värde som ska infogas för NoData-värdena i utdatarastern
Klippt (omfattning)	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av rasterlagret för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).
Typ av utdata	`DATA_TYPE`	[uppräkning] Standard: 0	Definierar formatet för den utgående rasterfilen. Alternativ: • 0 — Använd datatyp för inmatningslager • 1 — Byte (Åtta bitars osignerat heltal (quint8)) • 2 — Int16 (Sexton bitars signerat heltal (qint16)) • 3 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 4 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 5 — Int32 (Trettio tvåbitars signerade heltal (qint32)) • 6 — Float32 (trettio tvåbitars flyttal (float)) • 7 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 8 — CInt16 (Komplex Int16) • 9 — CInt32 (Komplex Int32) • 10 — CFloat32 (komplex float32) • 11 — CFloat64 (komplex float64) • 12 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Klippt (omfattning)	`OUTPUT`	[raster]	Utmatning av rasterlager som klippts av med den angivna omfattningen

Python-kod

Algoritm-ID: `gdal:cliprasterbyextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Klipp raster genom masklager

Klipper alla GDAL-stödda raster med ett vektormasklager.

Den här algoritmen är hämtad från GDAL warp utility.

Standardmeny: *Raster* ► *Extraktion*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Det ingående rastret
Masklager	`MASK`	[vektor: polygon]	Vektormask för klippning av rastret
Källa CRS	`SOURCE_CRS`	[crs]	Ange den koordinatreferens som ska användas för inmatningsrastret
Mål CRS	`TARGET_CRS`	[crs]	Ange den koordinatreferens som ska användas för masklagret

continues on next page

Table 24.256 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
** Målets omfattning** Valfritt	TARGET_EXTENT	[omfattning]	Omfattning av den utdatafil som ska skapas Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Tilldela ett specificerat NoData-värde till utgångsbanden Valfritt	NODATA	[numerisk: dubbel] Standard: Ej inställd	Definierar ett värde som ska infogas för NoData-värdena i utdatarastern
Skapa ett alfaband för utdata	ALPHA_BAND	[boolean] Standard: Falsk	Skapar ett alfaband för resultatet. Alfabandet innehåller sedan pixlarnas transparensvärden.
Matcha omfattningen av det klippta rastret med omfattningen av masklagret	CROP_TO_CUTLIN	[boolean] Standard: Sann	Tillämpar vektorlagrets utsträckning på utdatarastern om detta är markerat.
Behåll upplösningen på inmatningsrastret	KEEP_RESOLUTIO	[boolean] Standard: Falsk	Upplösningen på utmatningsrastret kommer inte att ändras
Ställ in upplösning för utdatafil	SET_RESOLUTION	[boolean] Standard: Falsk	Ska utmatningsupplösningen (cellstorlek) specificeras
X Upplösning till utmatningsband Valfritt	X_RESOLUTION	[numerisk: dubbel] Standard: Ej inställd	Bredden på cellerna i utdatarastret
Y Upplösning till utgångsband Valfritt	Y_RESOLUTION	[numerisk: dubbel] Standard: Ej inställd	Höjden på cellerna i utdatarastret
Använda flertrådig warping-implementering	MULTITHREADING	[boolean] Standard: Falsk	Två trådar kommer att användas för att bearbeta delar av bilden och utföra in- och utmatningsoperationer samtidigt. Observera att själva beräkningen inte är flertrådad.

continues on next page

Table 24.256 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Klippt (mask)	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av rasterlagret för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).
Typ av utdata	`DATA_TYPE`	[uppräkning] Standard: 0	Definierar formatet för den utgående rasterfilen. Alternativ: • 0 — Använd datatyp för inmatningslager • 1 — Byte (Åtta bitars osignerat heltal (quint8)) • 2 — Int16 (Sexton bitars signerat heltal (qint16)) • 3 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 4 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 5 — Int32 (Trettio tvåbitars signerade heltal (qint32)) • 6 — Float32 (trettio tvåbitars flyttal (float)) • 7 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 8 — CInt16 (Komplex Int16) • 9 — CInt32 (Komplex Int32) • 10 — CFloat32 (komplex float32) • 11 — CFloat64 (komplex float64) • 12 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Klippt (mask)	`OUTPUT`	[raster]	Utgående rasterlager som klippts av vektorlagret

Python-kod

Algoritm-ID: `gdal:cliprasterbymasklayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Söm/kontur

Extraherar höjdkurvor från alla GDAL-stödda höjdraster.

Denna algoritm är hämtad från GDAL contour utility.

Standardmeny: *Raster* ► *Extraktion*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Ingångsraster
Bandnummer	`BAND`	[rasterband] Standard: 1	Rasterband för att skapa konturerna från
Intervall mellan höjdkurvor	`INTERVAL`	[numerisk: dubbel] Standard: 10.0	Definierar intervallet mellan höjdkurvorna i de angivna enheterna i höjdrastret (minsta värde 0)
Attributets namn (om det inte anges är inget höjdattribut kopplat) Valfritt	`FIELD_NAME`	[sträng] Standard: ”ELEV	Ger ett namn på det attribut där förhöjningen ska placeras.
Offset från noll i förhållande till vilket intervallen ska tolkas Valfritt	`OFFSET`	[numerisk: dubbel] Standard: 0,0	

continues on next page

Table 24.257 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Contours	`OUTPUT`	[vektor: linje] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsvektorlagret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Producera 3D-vektor	`CREATE_3D`	[boolean] Standard: Falsk	Tvingar fram 3D-vektorer i stället för 2D-vektorer. Inkluderar elevation vid varje vertex.
Behandla alla rastervärden som giltiga	`IGNORE_NODATA`	[boolean] Standard: Falsk	Ignorerar eventuella NoData-värden i datasetet.
Inmatat pixelvärde som ska behandlas som ”NoData” Valfritt	`NODATA`	[numerisk: dubbel] Standard: Ej inställd	Definierar ett värde som ska infogas för NoData-värdena i utdatarastern
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägga till extra GDAL-kommandoradsalternativ. Se dokumentationen för motsvarande GDAL-verktyg.

Utgångar

Etikett	Namn	Typ	Beskrivning
Contours	`OUTPUT`	[vektor: linje]	Utgående vektorlager med konturlinjer

Python-kod

Algoritm-ID: `gdal:kontur`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Konturpolygoner

Extraherar konturpolygoner från alla GDAL-stödda höjdraster.

Denna algoritm är hämtad från GDAL contour utility.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Ingångsraster
Bandnummer	`BAND`	[rasterband] Standard: 1	Rasterband för att skapa konturerna från
Intervall mellan höjdkurvor	`INTERVAL`	[numerisk: dubbel] Standard: 10.0	Definierar intervallet mellan höjdkurvorna i de angivna enheterna i höjdrastret (minsta värde 0)
Offset från noll i förhållande till vilket intervallen ska tolkas Valfritt	`OFFSET`	[numerisk: dubbel] Standard: 0,0	
Attributnamn för minsta höjd för konturpolygon Valfritt	`FIELD_NAME_MIN`	[sträng] Standard: ”ELEV_MIN	Anger ett namn på det attribut i vilket den lägsta höjden för konturpolygonen ska anges. Om det inte anges bifogas inget attribut för lägsta höjd.
Attributnamn för maximal höjd för konturpolygon Valfritt	`FIELD_NAME_MAX`	[sträng] Standard: ”ELEV_MAX	Anger ett namn på det attribut i vilket den maximala höjden för konturpolygonen ska anges. Om det inte anges bifogas inget attribut för maximal höjd.
Contours	`OUTPUT`	[vektor: polygon] Standard: `[Spara till temporär fil]`	Specifikation av utmatningsvektorlagret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Producera 3D-vektor	`CREATE_3D`	[boolean] Standard: Falsk	Tvingar fram 3D-vektorer i stället för 2D-vektorer. Inkluderar elevation vid varje vertex.
Behandla alla rastervärden som giltiga	`IGNORE_NODATA`	[boolean] Standard: Falsk	Ignorerar eventuella NoData-värden i datasetet.
Inmatat pixelvärde som ska behandlas som ”NoData” Valfritt	`NODATA`	[numerisk: dubbel] Standard: Ej inställd	Definierar ett värde som ska infogas för NoData-värdena i utdatarastern
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägga till extra GDAL-kommandoradsalternativ. Se dokumentationen för motsvarande GDAL-verktyg.

Utgångar

Etikett	Namn	Typ	Beskrivning
Contours	`OUTPUT`	[vektor: polygon]	Vektorlager för utdata med konturpolygoner

Python-kod

Algoritm-ID: `gdal:kontur_polygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.2.4 Raster diverse

Bygga översikter (pyramider)

För att påskynda renderingstiden för rasterlager kan översikter (pyramider) skapas. Översikter är kopior med lägre upplösning av de data som QGIS använder beroende på zoomnivån.

Den här algoritmen är hämtad från GDAL addo utility.

Standardmeny: *Raster* ► *Diverse*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatat rasterlager
Ta bort alla befintliga översikter	`CLEAN`	[boolean] Standard: Falsk	Tar bort befintliga översikter från rastret. Som standard tas dessa inte bort.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Överblicksnivåer	`LEVELS`	[sträng] Standard: '2 4 8 16'	Definierar antalet översiktsnivåer som beräknas utifrån den ursprungliga upplösningen för det ingående rasterlagret. Som standard kommer 4 nivåer att tas i beaktande.
Metod för omprovtagning Valfritt	`RESAMPLING`	[uppräkning] Standard: 0	Beräknar översikterna med en definierad omsamplingsmetod. Möjliga omsamplingsmetoder är: • 0 – Närmaste granne (`närmaste`) • 1 – Genomsnitt (`genomsnitt`) • 2 – Gaussisk (`gauss`) • 3 – Kubisk (4x4-kärna) (`cubic`) • 4 – Kubisk B-Spline (4x4-kärna) (`cubicspline`) • 5 – Lanczos (6x6 kernel) (`lanczos`) • 6 – Genomsnittlig MP (`average_mp`) • 7 – Medelvärde i Mag/Fas-utrymme (`average_magphase`) • 8 – Läge (`mode`)
Översikt format Valfritt	`FORMAT`	[uppräkning] Standard: 0	Översikterna kan lagras internt eller externt som GTiff- eller ERDAS Imagine-fil. Som standard lagras översikterna i utdatarastret. Möjliga formatmetoder är: • 0 – Intern (om möjligt) • 1 – Extern (GTiff .ovr) • 2 – Extern (ERDAS Imagine .aux)
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Pyramidisering	`OUTPUT`	[raster]	Utmatning av rasterlager med översikter

Python-kod

Algoritm ID: `gdal:overviews`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skapa virtuell raster

Bygger en VRT (Virtual Dataset) som är en mosaik av listan med inmatade GDAL-stödda raster. Med en mosaik kan du slå samman flera rasterfiler.

Denna algoritm är hämtad från GDAL buildvrt utility.

Standardmeny: *Raster* ► *Diverse*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster] [lista]	GDAL-stödda rasterlager.
Föredragen upplösning	`RESOLUTION`	[uppräkning] Standard: 0	Mosaikens utgångsupplösning. Som standard kommer den genomsnittliga upplösningen för rasterfilerna att väljas. Alternativ: • 0 — Genomsnitt (`genomsnitt`) • 1 — Högsta (`högsta`) • 2 — Lägsta (`lågsta`)
Placera varje inmatningsfil i ett separat band	`SEPARATE`	[boolean] Standard: Falsk	Med ”True” kan du definiera att varje rasterfil läggs in i ett separat staplat band i VRT-bandet.
Allow projection difference	`PROJ_DIFFERENC`	[boolean] Standard: Falsk	Tillåter att de utgående banden har olika projektioner som härrör från projektionen av de ingående rasterlagren.
Virtuell	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av rasterlagret för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Lägga till alfamaskband till VRT när källraster inte har något	`ADD_ALPHA`	[boolean] Standard: Falsk	Lägger till ett alfamaskband till VRT när källrastret inte har något.
Åsidosätta projicering för utdatafilen Valfritt	`ASSIGN_CRS`	[crs] Standard: Ej inställd	Åsidosätter projektionen för utdatafilen. Ingen återprojektion görs.
Algoritm för omsampling	`RESAMPLING`	[uppräkning] Standard: 0	Den omsamplingsalgoritm <https://gdal.org/en/latest/programs/gdalwarp.html#cmdoption-gdalwarp-r>`_ som ska användas. Alternativ: • 0 — Närmaste granne (`nearest`) • 1 — Bilinear (2x2-kärna) (`bilinear`) • 2 — Kubisk (4x4-kärna) (`cubic`) • 3 — Kubisk B-Spline (4x4-kärna) (`cubicspline`) • 4 — Lanczos (6x6 kernel) (`lanczos`) • 5 — Genomsnitt (`average`) • 6 — Läge (`mode`)
Inga datavärden för inmatningsband (mellanslagsseparer Valfritt	`SRC_NODATA`	[sträng] Standard: Ej inställd	Mellanslagsseparerat NoData-värde(n) för inmatningsband(en)
Ytterligare kommandoradspara	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Virtuell	`OUTPUT`	[raster]	Utgående rasterlager

Python-kod

Algoritm-ID: `gdal:buildvirtualraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

gdal2tiles

Genererar en katalog med små plattor och metadata, i enlighet med OSGeo Tile Map Service Specification. Se även OpenGIS Web Map Tile Service Implementation Standard. Enkla webbsidor med visare baserade på Google Maps, OpenLayers och Leaflet genereras också. För att utforska dina kartor online i webbläsaren behöver du bara ladda upp den genererade katalogen till en webbserver.

Denna algoritm skapar också de metadata som krävs för Google Earth (KML SuperOverlay), om den levererade kartan använder projektionen `EPSG:4326`.

ESRI World-filer och inbäddad georeferering används vid generering av plattor, men du kan också publicera en bild utan korrekt georeferering.

Den här algoritmen är hämtad från verktyget GDAL gdal2tiles.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	GDAL-stöttat rasterlager.
Profil för kakelskärning	`PROFILE`	[uppräkning] Standard: 0	En av: • 0 — Mercator (`mercator`) • 1 — Geodetisk (`geodetic`) • 2 — Raster (`raster`)
Zoom-nivåer för rendering Valfritt	`ZOOM`	[sträng] Standard: ”	
Webbvisare för att generera	`VIEWER`	[uppräkning] Standard: 0	En av: • 0 — Alla (`all`) • 1 — GoogleMaps (`google`) • 2 — OpenLayers (`openlayers`) • 3 — Broschyr (`leaflet`) • 4 — Ingen (`none`)
Titel på kartan Valfritt	`TITLE`	[sträng] Standard: ”	
Upphovsrätt till kartan	`COPYRIGHT`	[sträng] Standard: ”	
Utmatningskatalog	`OUTPUT`	mapp Standard: `[Spara till tillfällig mapp]`	Ange utdatamappen för plattorna. *En av*: • Spara till en temporär katalog • Spara till katalog

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Metod för omprovtagning	`RESAMPLING`	[uppräkning] Standard: 0	Den omsamplingsalgoritm som ska användas. Alternativ: • 0 — Genomsnitt (`genomsnitt`) • 1 — Närmaste granne (`near`) • 2 — Bilineär (2x2-kärna) (`bilinear`) • 3 — Kubisk (4x4-kärna) (`cubic`) • 4 — Kubisk B-Spline (4x4-kärna) (`cubicspline`) • 5 — Lanczos (6x6 kernel) (`lanczos`) • 6 — Antialias (`antialias`)
Det spatiala referenssystem som används för källans indata Valfritt	`SOURCE_CRS`	[crs] Standard: Ej inställd	
Transparensvärde som ska tilldelas indata Valfritt	`NODATA`	[numerisk: dubbel] Standard: 0,0	
URL-adress där de genererade plattorna kommer att publiceras Valfritt	`URL`	[sträng] Standard: ”	
Google Maps API-nyckel (http://code.google.c Valfritt	`GOOGLE_KEY`	[sträng] Standard: ”	Din API-nyckel för Google Maps.
Bing Maps API-nyckel (https://www.bingm Valfritt	`BING_KEY`	[sträng] Standard: ”	Din API-nyckel för Bing Maps.
Genererar endast filer som saknas	`RESUME`	[boolean] Standard: Falsk	
Genererar KML för Google Earth	`KML`	[boolean] Standard: Falsk	
Undvik automatisk generering av KML-filer för EPSG:4326	`NO_KML`	[boolean] Standard: Falsk	

Utgångar

Etikett	Namn	Typ	Beskrivning
Utmatningskatalog	`OUTPUT`	mapp	Utdatamappen (för brickorna)

Python-kod

Algoritm ID: `gdal:gdal2tiles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Sammanfoga

Sammanfogar rasterfiler på ett enkelt sätt. Här kan du använda en pseudofärgtabell från ett inmatningsraster och definiera rastertypen för utmatningen. Alla bilder måste vara i samma koordinatsystem.

Denna algoritm är hämtad från sammanslagningsverktyget GDAL.

Standardmeny: *Raster* ► *Diverse*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster] [lista]	Inmatade rasterlager
Grab pseudocolor tabell från första lagret	`PCT`	[boolean] Standard: Falsk	Pseudocolor-tabellen från det första lagret kommer att användas för färgläggningen
Placera varje inmatningsfil i ett separat band	`SEPARATE`	[boolean] Standard: Falsk	Placera varje inmatningsfil i ett separat band

continues on next page

Table 24.262 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Typ av utdata	`DATA_TYPE`	[uppräkning] Standard: 5	Definierar formatet för den utgående rasterfilen. Alternativ för: • 0 — Byte (Åtta bitars osignerat heltal (quint8)) • 1 — Int16 (Sexton bitars signerat heltal (qint16)) • 2 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 3 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 4 — Int32 (trettio tvåbitars signerade heltal (qint32)) • 5 — Float32 (trettio tvåbitars flyttal (float)) • 6 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 7 — CInt16 (Komplex Int16) • 8 — CInt32 (Komplex Int32) • 9 — CFloat32 (komplex float32) • 10 — CFloat64 (komplex float64) • 11 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)
Sammanslagna	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av rasterlagret för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Inmatat pixelvärde som ska behandlas som ”NoData” Valfritt	`NODATA_INPUT`	[numerisk: dubbel] Standard: Ej inställd	Ignorerar pixlar från filer som slås samman med detta pixelvärde
Tilldela utdata ett specificerat ”NoData”-värde Valfritt	`NODATA_OUTPUT`	[numerisk: dubbel] Standard: Ej inställd	Tilldelar det angivna NoData-värdet till utgångsbanden.
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).
Ytterligare kommandoradspara	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Sammanslagna	`OUTPUT`	[raster]	Utgående rasterlager

Python-kod

Algoritm ID: `gdal:merge`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Skärpning av pannor

Utför en pan-sharpening-operation. Den kan skapa en ”klassisk” utdataset (t.ex. GeoTIFF) eller en VRT-dataset som beskriver pan-sharpening-operationen.

Se GDAL Pansharpen.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Spektral dataset	`SPECTRAL`	[raster]	Inmatat (spektralt) rasterlager
Panchromatisk dataset	`PANCHROMATIC`	[raster]	Inmatat (pankromatiskt) rasterlager
Utdata	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange det utgående (skärpta) rasterlagret. *Ett av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Algoritm för omsampling	`RESAMPLING`	[uppräkning] Standard: 2	Den omsamplingsalgoritm <https://gdal.org/en/latest/programs/gdalwarp.html#cmdoption-gdalwarp-r>`_ som ska användas. Alternativ: • 0 — Närmaste granne (`nearest`) • 1 — Bilineär (2x2-kärna) (`bilinear`) • 2 — Kubisk (4x4-kärna) (`cubic`) • 3 — Kubisk B-Spline (4x4-kärna) (`cubicspline`) • 4 — Lanczos (6x6 kernel) (`lanczos`) • 5 — Genomsnitt (`average`)
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (`|`).
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Utdata	`OUTPUT`	[raster]	Utgående (skärpt) rasterlager

Python-kod

Algoritm-ID: `gdal:pansharp`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rasterkalkylator

Rasterkalkylator för kommandoraden med numpy-syntax. Använd all grundläggande aritmetik som stöds av numpy-arrayer, t.ex. +, -, * och / tillsammans med logiska operatorer, t.ex. >. Observera att alla inmatningsraster måste ha samma dimensioner, men ingen projektionskontroll utförs.

Se dokumentationen för verktyget GDAL Raster Calculator.

Se även

Rasterkalkylator, *Rasterkalkylator (virtuell)*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager A	`INPUT_A`	[raster]	Första rasterlagret för inmatning (obligatoriskt)
Nummer på rasterband för A	`BAND_A`	[rasterband]	Band för inmatningslager A (obligatoriskt)
Inmatningslager B Valfritt	`INPUT_B`	[raster] Standard: Ej inställd	Andra inmatade rasterlagret
Nummer på rasterband för B Valfritt	`BAND_B`	[rasterband]	Band för inmatningslager B
Inmatningslager C Valfritt	`INPUT_C`	[raster] Standard: Ej inställd	Tredje rasterlagret för inmatning
Nummer på rasterband för C Valfritt	`BAND_C`	[rasterband]	Band för inmatningslagret C

continues on next page

Table 24.264 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Inmatningslager D Valfritt	`INPUT_D`	[raster] Standard: Ej inställd	Fjärde rasterlagret för inmatning
Nummer på rasterband för D Valfritt	`BAND_D`	[rasterband]	Band för inmatningslagret D
Inmatningslager E Valfritt	`INPUT_E`	[raster] Standard: Ej inställd	Femte rasterlagret för inmatning
Nummer på rasterband för E Valfritt	`BAND_E`	[rasterband]	Band för inmatningslagret E
Inmatningslager F Valfritt	`INPUT_F`	[raster]	Sjätte rasterlagret för inmatning
Nummer på rasterband för F Valfritt	`BAND_F`	[rasterband] Standard: Ej inställd	Band för inmatningslagret F
Beräkning i gdalnumerisk syntax med hjälp av +-/* eller någon numpy array-funktion (t.ex. logical_and())	`FORMULA`	[sträng] Standard: ”	Beräkningsformeln. Exempel på detta: • `A*(A>0)` — matar ut värdet på rastret A om värdet på A är större än 0. Om inte, matas 0 ut. • `A*(A>0 och A>B)` — matar ut värdet för A om det värdet är större än 0 och större än värdet för B. Om inte, matas 0 ut. • `A*logical_or(A<=177, A>=185)` — ger värdet av A om A <= 177 eller A >= 185. Om inte, matas 0 ut. • `sqrt(A*A+B*B)` — Utmatar kvadratroten av summan av värdet av A i kvadrat och värdet av B i kvadrat.
Ställ in NoData-värde för utdata Valfritt	`NO_DATA`	[numerisk: dubbel] Standard: Ej inställd	Värde som ska användas för NoData
Hantering av skillnader i omfattning	`EXTENT_OPT`	[uppräkning] Standard: 0	Bestämmer hur man ska hantera raster med olika utsträckning. Endast tillgängligt med GDAL 3.3+. alternativ som stöds <https://gdal.org/en/latest/programs/gdal_calc.html#cmdoption-extent>`_ är: • 0 — Ignorera • 1 — Fel • 2 — Union • 3 — Korsningspunkt

continues on next page

Table 24.264 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Utmatning omfattning Valfritt	`INPUT`	[omfattning]	Anpassad utsträckning av utmatningsrastret. Endast tillgänglig med GDAL 3.3+. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Rastertyp för utdata	`RTYPE`	[uppräkning] Standard: 5	Definierar datatypen för den utgående rasterfilen. Alternativ för: • 0 — Byte (Åtta bitars osignerat heltal (quint8)) • 1 — Int16 (Sexton bitars signerat heltal (qint16)) • 2 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 3 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 4 — Int32 (trettio tvåbitars signerade heltal (qint32)) • 5 — Float32 (trettio tvåbitars flyttal (float)) • 6 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 7 — CInt16 (Komplex Int16) • 8 — CInt32 (Komplex Int32) • 9 — CFloat32 (komplex float32) • 10 — CFloat64 (komplex float64) • 11 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)
Beräknad	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Ange det utgående (beräknade) rasterlagret. *Ett av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	OPTIONS	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).
Ytterligare kommandoradspara Valfritt	EXTRA	[sträng] Standard: ”	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Beräknad	OUTPUT	[raster]	Utgående (beräknat) rasterlager

Python-kod

Algoritm ID: `gdal:rasterberäknare`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rasterinformation

Programmet gdalinfo listar olika uppgifter om en rasterdatauppsättning som stöds av GDAL.

Denna algoritm är hämtad från GDAL info utility.

Standardmeny: *Raster* ► *Diverse*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	INPUT	[raster]	Inmatat rasterlager

continues on next page

Table 24.266 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
tvinga fram en beräkning av de faktiska min/max-värdena för varje band	`MIN_MAX`	[boolean] Standard: Falsk	Kraftberäkning av de faktiska min/max-värdena för varje band i datasetet
Läsa och visa bildstatistik (tvinga fram beräkningar vid behov)	`STATS`	[boolean] Standard: Falsk	Läser och visar bildstatistik. Forcerar beräkningen om ingen statistik finns lagrad i en bild.
Undertrycka GCP-information	`NO_GCP`	[boolean] Standard: Falsk	Undertrycker utskrift av listan med markkontrollpunkter. Det kan vara användbart för dataset med en stor mängd GCP:er, t.ex. L1B AVHRR eller HDF4 MODIS som innehåller tusentals sådana.
Undertryck metadata-information	`NO_METADATA`	[boolean] Standard: Falsk	Undertrycker utskrift av metadata. Vissa dataset kan innehålla många metadatasträngar.
Lagerinformation	`OUTPUT`	[html] Standard: `[Spara till temporär fil]`	Ange HTML-filen för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Lagerinformation	`OUTPUT`	[html]	HTML-filen som innehåller information om det ingående rasterlagret

Python-kod

Algoritm ID: `gdal:gdalinfo`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Retile

Retilerar en uppsättning inmatade plattor. Alla inmatade plattor måste georefereras i samma koordinatsystem och ha ett matchande antal band. Eventuellt genereras pyramidnivåer.

Denna algoritm är hämtad från GDAL Retile utility.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsfiler	`INPUT`	[raster] [lista]	De ingående rasterfilerna
Kakelplattans bredd	`TILE_SIZE_X`	[numerisk: heltal] Standard: 256	Bredd på plattorna i pixlar (minimum 0)
Kakelplattans höjd	`TILE_SIZE_Y`	[numerisk: heltal] Standard: 256	Höjd på brickorna i pixlar (minimum 0)
Överlappning i pixlar mellan på varandra följande plattor	`OVERLAP`	[numerisk: heltal] Standard: 0	
Antal pyramidnivåer att bygga	`LEVELS`	[numerisk: heltal] Standard: 1	Minimum: 0
Utmatningskatalog	`OUTPUT`	mapp Standard: `[Spara till tillfällig mapp]`	Ange utdatamappen för plattorna. *En av*: • Spara till en temporär katalog • Spara till katalog
CSV-fil som innehåller information om georeferenser för plattan/plattorna	`OUTPUT_CSV`	&Arkiv Standard: `[Hoppa över utdata]`	Ange utdatafilen för plattorna. *En av*: • Hoppa över utmatning • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Referenssystem för källkoordinater Valfritt	`SOURCE_CRS`	[crs] Standard: Ej inställd	

continues on next page

Table 24.269 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Metod för omprovtagning	`RESAMPLING`	[uppräkning] Standard: 0	Den omsamplingsalgoritm <https://gdal.org/en/latest/programs/gdalwarp.html#cmdoption-gdalwarp-r>`_ som ska användas. Alternativ: • 0 — Närmaste granne (`nearest`) • 1 — Bilineär (2x2-kärna) (`bilinear`) • 2 — Kubisk (4x4-kärna) (`cubic`) • 3 — Kubisk B-Spline (4x4-kärna) (`cubicspline`) • 4 — Lanczos (6x6 kernel) (`lanczos`)
Kolumnavgränsare som används i CSV-filen Valfritt	`DELIMITER`	[sträng] Standard: ';'	Avgränsare som ska användas i CSV-filen som innehåller georeferensinformation för plattan/plattorna
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ''	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: ''	Lägg till extra GDAL-kommandoradsalternativ
Typ av utdata	`DATA_TYPE`	[uppräkning] Standard: 5	Definierar formatet för den utgående rasterfilen. Alternativ för: • 0 — Byte (Åtta bitars osignerat heltal (quint8)) • 1 — Int16 (Sexton bitars signerat heltal (qint16)) • 2 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 3 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 4 — Int32 (trettio tvåbitars signerade heltal (qint32)) • 5 — Float32 (trettio tvåbitars flyttal (float)) • 6 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 7 — CInt16 (Komplex Int16) • 8 — CInt32 (Komplex Int32) • 9 — CFloat32 (komplex float32) • 10 — CFloat64 (komplex float64) • 11 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)

continues on next page

Table 24.269 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Bygg bara pyramiderna	`ONLY_PYRAMIDS`	[boolean] Standard: Falsk	
Använd en separat katalog för varje kakelrad	`DIR_FOR_ROW`	[boolean] Standard: Falsk	

Utgångar

Etikett	Namn	Typ	Beskrivning
Utmatningskatalog	`OUTPUT`	mapp	Utdatamappen för plattorna.
CSV-fil som innehåller information om georeferenser för plattan/plattorna	`OUTPUT_CSV`	&Arkiv	CSV-filen med information om georeferenser för plattorna.

Python-kod

Algoritm ID: `gdal:retile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Kakelindex

Skapar ett vektorlager med en post för varje ingående rasterfil, ett attribut som innehåller filnamnet och en polygongeometri som avgränsar rastret. Denna utdata är lämplig för användning med MapServer som ett raster-tileindex.

Denna algoritm är hämtad från verktyget GDAL Tile Index.

Standardmeny: *Raster* ► *Diverse*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsfiler	`LAYERS`	[raster] [lista]	De ingående rasterfilerna. Kan vara flera filer.

continues on next page

Table 24.270 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Fältnamn som innehåller filsökvägen till de indexerade rastren	`PATH_FIELD_NAM` Valfritt	[sträng] Standard: ”location” (plats)	Namnet på utdatafältet som innehåller filsökvägen/platsen till de indexerade rastren.
Lagra absolut sökväg till de indexerade rastren	`ABSOLUTE_PATH`	[boolean] Standard: Falsk	Ställ in om den absoluta sökvägen till rasterfilerna ska lagras i filen för kakelindex. Som standard kommer rasterfilnamnen att läggas in i filen exakt som de anges i kommandot.
Skippa filer med annan projektionsreferens	`PROJ_DIFFERENC`	[boolean] Standard: Falsk	Endast filer med samma projektion som filer som redan är infogade i kakelindexet kommer att infogas. Standard kontrollerar inte projektion och accepterar alla inmatningar.
Kakelindex	`OUTPUT`	[vektor: polygon] Standard: `[Spara till temporär fil]`	Ange det polygonvektorlager som indexet ska skrivas till. *Ett av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Transformera geometrier till den angivna CRS Valfritt	`TARGET_CRS`	[crs]	Geometrierna i indatafilerna omvandlas till det angivna referenssystemet för målkoordinater. Standard skapar enkla rektangulära polygoner i samma koordinatreferenssystem som de ingående rastrerna.
Namnet på det fält som ska lagra SRS för varje bricka Valfritt	`CRS_FIELD_NAME`	[sträng]	Namnet på det fält som lagrar SRS för varje kakelplatta
Det format i vilket CRS för varje bricka måste skrivas	`CRS_FORMAT`	[uppräkning] Standard: 0	Format för CRS. En av: • 0 – Auto (`AUTO`) • 1 – Välkänd text (`WKT`) • 2 – EPSG (`EPSG`) • 3 – Proj.4 (`PROJ`)

Utgångar

Etikett	Namn	Typ	Beskrivning
Kakelindex	`OUTPUT`	[vektor: polygon]	Polygonvektorlagret med kakelindexet.

Python-kod

Algoritm ID: `gdal:tileindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Viewshed

Beräknar ett utsiktsraster från ett inmatningsraster DEM med hjälp av metoden som definieras i Wang2000 för en användardefinierad punkt.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatat höjdrasterlager
Bandnummer	`BAND`	[rasterband] Standard: 1	Numret på bandet som ska användas som höjd
Observerplats	`OBSERVER`	[punkt]	Observatörens plats
Observatörens höjd	`OBSERVER_HEIGH`	[numerisk: dubbel] Standard: 1,0	Observatörens höjd, i DEM-enheter
Målhöjd	`TARGET_HEIGHT`	[numerisk: dubbel] Standard: 1,0	Höjden för målelementet, i DEM-enheter
Maximalt avstånd från observatören för att beräkna synlighet	`MAX_DISTANCE`	[numerisk: dubbel] Standard: 100.0	Maximalt avstånd från observatören för att beräkna synlighet, i DEM-enheter
Utdata	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Rasterlager för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	OPTIONS	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).
Ytterligare kommandoradspara	EXTRA	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Utdata	OUTPUT	[raster]	Rasterlagret som visar utsiktshorisonten.

Python-kod

Algoritm-ID: `gdal:viewshed`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.2.5 Rasterprojektioner

Tilldela projektion

Tillämpar ett koordinatsystem på en rasterdatauppsättning.

Den här algoritmen är hämtad från redigeringsverktyget GDAL.

Standardmeny: *Raster* ► *Projektioner*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT_LAYER`	[raster]	Inmatat rasterlager
Önskad CRS	`CRS`	[crs]	Projektionen (CRS) för det utgående lagret

Utgångar

Etikett	Namn	Typ	Beskrivning
Lager med projektion	`OUTPUT`	[raster]	Det utgående rasterlagret (med den nya projektionsinformationen)

Python-kod

Algoritm ID: `gdal:assignprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Extrahera projektion

Extraherar projektionen av en rasterfil och skriver in den i en *world*-fil med tillägget `.wld`.

Denna algoritm är hämtad från verktyget GDAL srsinfo.

Standardmeny: *Raster* ► *Projektioner*

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsfil	`INPUT_LAYER`	[raster]	Input raster Rasterlagret måste vara filbaserat, eftersom algoritmen använder sökvägen till rasterfilen som plats för den genererade `.wld`-filen. Om du använder ett rasterlager som inte är en fil kommer det att leda till ett fel.
Skapa även .prj-fil	`PRJ_FILE_CREAT`	[boolean] Standard: Falsk	Om detta är aktiverat skapas även en fil med projektionsinformation (`.prj`).

Utgångar

Etikett	Namn	Typ	Beskrivning
Världsfil	`WORLD_FILE`	&Arkiv	Textfil med filnamnstillägget `.wld` som innehåller transformationsparametrar för rasterfilen.
ESRI Shapefile prj-fil	`PRJ_FILE`	&Arkiv	Textfil med tillägget `.prj` som beskriver CRS. Kommer att vara `None` om :guilabel:`Create also .prj file` är False.

Python-kod

Algoritm-ID: `gdal:extraktprojektion`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Warp (omprojektering)

Återprojicerar ett rasterlager till ett annat koordinatreferenssystem (CRS). Upplösningen på utdatafilen och omsamplingsmetoden kan väljas.

Den här algoritmen är hämtad från GDAL warp utility.

Standardmeny: *Raster* ► *Projektioner*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[raster]	Inmatat rasterlager att återprojicera
Källa CRS Valfritt	`SOURCE_CRS`	[crs]	Definierar CRS för det ingående rasterlagret
Mål CRS Valfritt	`TARGET_CRS`	[crs] Standard: `EPSG:4326`	CRS för det utgående lagret

continues on next page

Table 24.274 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Metod för omprovtagning som ska användas	`RESAMPLING`	[uppräkning] Standard: 0	Metod för omsampling av pixelvärden som ska användas. Alternativ: • 0 — Närmaste granne • 1 — Bilineär (2x2 kernel) • 2 — Kubisk (4x4-kärna) • 3 — Kubisk B-Spline (4x4 kernel) • 4 — Lanczos (6x6 kärnor) • 5 — Genomsnitt • 6 — Läge • 7 — Maximalt • 8 — Minimum • 9 — Median • 10 — Första kvartilen (Q1) • 11 — Tredje kvartilen (Q3)
Inget datavärde för utgångsband Valfritt	`NODATA`	[numerisk: dubbel] Standard: Ej inställd	Ställer in NoData-värde för utdataband. Om det inte anges kommer NoData-värden att kopieras från källdatasetet.
Utdatafilens upplösning i georefererade målenheter Valfritt	`TARGET_RESOLUT`	[numerisk: dubbel] Standard: Ej inställd	Definierar upplösningen på utdatafilen för reprojektionsresultatet
Omprojekterad	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av rasterlagret för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).

continues on next page

Table 24.275 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Typ av utdata	`DATA_TYPE`	[uppräkning] Standard: 0	Definierar formatet för den utgående rasterfilen. Alternativ för: • 0 — Använd datatyp för inmatningslager • 1 — Byte (Åtta bitars osignerat heltal (quint8)) • 2 — Int16 (Sexton bitars signerat heltal (qint16)) • 3 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 4 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 5 — Int32 (Trettio tvåbitars signerade heltal (qint32)) • 6 — Float32 (trettio tvåbitars flyttal (float)) • 7 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 8 — CInt16 (Komplex Int16) • 9 — CInt32 (Komplex Int32) • 10 — CFloat32 (komplex float32) • 11 — CFloat64 (komplex float64) • 12 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)
Georefererade extenter för utdatafilen som ska skapas Valfritt	`TARGET_EXTENT`	[omfattning]	Anger den georefererade utbredningen för den utdatafil som ska skapas (i *Target CRS* som standard. I *CRS för målrasterutbredningen*, om detta anges). Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
CRS för målrasterutbredning Valfritt	`TARGET_EXTENT_`	[crs]	Anger det CRS i vilket de angivna koordinaterna för utdatafilens omfattning ska tolkas. Detta får inte förväxlas med det CRS som är målet för utdatasetet. Det är istället en bekvämlighet, t.ex. när man känner till utdatakoordinaterna i ett geodetiskt CRS för längd/lat men vill ha ett resultat i ett projicerat koordinatsystem.

continues on next page

Table 24.275 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Använda flertrådig warping-implementering	`MULTITHREADING`	[boolean] Standard: Falsk	Två trådar kommer att användas för att bearbeta delar av bilden och utföra in- och utmatningsoperationer samtidigt. Observera att själva beräkningen inte är flertrådad.
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: Ej inställd	Lägg till extra GDAL-kommandoradsalternativ.

Utgångar

Etikett	Namn	Typ	Beskrivning
Omprojekterad	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Reprojicerat rasterlager för utdata

Python-kod

Algoritm ID: `gdal:warpreproject`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.2.6 Vektoromvandling

Konvertera format

Konverterar ett OGR-stöttat vektorlager till ett annat OGR-stöttat format.

Den här algoritmen är hämtad från verktyget ogr2ogr.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	Ingångsvektorlager
** Konvertera alla lager från dataset**	`CONVERT_ALL_LA`	[boolean] Standard: Falsk	Konverterar hela datasetet. Utdataformat som stöds för detta alternativ är `GPKG` och `GML`.
Konverterad	`OUTPUT`	[samma som inmatning]	Specifikation av utmatningsvektorlagret. *En av*: • Spara till en temporär fil • Spara till fil… För `Spara till fil` måste utdataformatet anges. Alla GDAL-vektorformat stöds. För `Spara till en temporär fil` kommer QGIS standardvektorformat att användas.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ” (inga ytterligare alternativ)	Ytterligare alternativ för skapande av GDAL.

Utgångar

Etikett	Namn	Typ	Beskrivning
Konverterad	`OUTPUT`	[samma som inmatning]	Utgångsvektorlagret

Python-kod

Algoritm ID: `gdal:konverteringsformat`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rasterisera (skriv över med attribut)

Skriver över ett rasterlager med värden från ett vektorlager. Nya värden tilldelas baserat på attributvärdet för den överlappande vektorfunktionen.

Denna algoritm är hämtad från GDAL rasterize utility.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Rasterlager för inmatning	`INPUT_RASTER`	[raster]	Inmatat rasterlager
Fält att använda för ett inbränningsvärde Valfritt	`FIELD`	[tabellfält: numerisk]	Definierar det attributfält som ska användas för att ställa in pixlarnas värden

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Lägg till inbränningsvärden till befintliga rastervärden	`ADD`	[boolean] Standard: Falsk	Om False, tilldelas pixlarna det valda fältets värde. Om True, läggs det markerade fältets värde till värdet för det ingående rasterlagret.
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: ”	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Rasteriserad	`OUTPUT`	[raster]	Det överskrivna ingående rasterlagret

Python-kod

Algoritm-ID: `gdal:rasterize_over`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rasterize (överskriv med fast värde)

Skriver över delar av ett rasterlager med ett fast värde. De pixlar som ska skrivas över väljs utifrån det medföljande (överlappande) vektorlagret.

Denna algoritm är hämtad från GDAL rasterize utility.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Rasterlager för inmatning	`INPUT_RASTER`	[raster]	Inmatat rasterlager
Ett fast värde att bränna	`BURN`	[numerisk: dubbel] Standard: 0,0	Värdet att bränna

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Lägg till inbränningsvärden till befintliga rastervärden	`ADD`	[boolean] Standard: Falsk	Om False, tilldelas pixlarna det fasta värdet. Om True, läggs det fasta värdet till värdet för det ingående rasterlagret.
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: ”	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Rasteriserad	`OUTPUT`	[raster]	Det överskrivna ingående rasterlagret

Python-kod

Algoritm-ID: `gdal:rasterize_over_fixed_value`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Rasterize (vektor till raster)

Konverterar vektorgeometrier (punkter, linjer och polygoner) till en rasterbild.

Denna algoritm är hämtad från GDAL rasterize utility.

Standardmeny: *Raster* ► *Konvertering*

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Fält att använda för ett inbränningsvärde Valfritt	`FIELD`	[tabellfält: numerisk]	Definierar det attributfält från vilket attributen för pixlarna ska väljas
Ett fast värde att bränna Valfritt	`BURN`	[numerisk: dubbel] Standard: 0,0	Ett fast värde att bränna in i ett band för alla funktioner.
Brännvärde extraherat från ”Z”-värdena för objektet	`USE_Z`	[boolean] Standard: Falsk	Anger att ett brännvärde ska extraheras från funktionens ”Z”-värden. Fungerar med punkter och linjer (linjär interpolering längs varje segment). För polygoner fungerar det bara korrekt om de är plana (samma Z-värde för alla hörn)
Utmatning av rasterstorleksenhete	`UNITS`	[uppräkning] Standard: 0	Enheter som ska användas för att definiera rasterstorlek/upplösning för utdata. En av: • 0 — Pixlar • 1 — Georefererade enheter
Bredd/horisontell upplösning	`WIDTH`	[numerisk: dubbel] Standard: 0,0	Ställer in bredden (om storleksenheter är ”Pixels”) eller den horisontella upplösningen (om storleksenheter är ”Georefererade enheter”) för utmatningsrastret. Minsta värde: 0,0.
Höjd/Vertikal upplösning	`HEIGHT`	[numerisk: dubbel] Standard: 0,0	Ställer in höjden (om storleksenheter är ”Pixels”) eller den vertikala upplösningen (om storleksenheter är ”Georefererade enheter”) för utmatningsrastret.

continues on next page

Table 24.278 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Utmatning omfattning Valfritt	`EXTENT`	[omfattning]	Utsträckning för det utgående rasterlagret. Om utsträckning inte anges kommer den minsta utsträckning som täcker det/de valda referenslagret/lagren att användas. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Tilldela ett specificerat NoData-värde till utgångsbanden Valfritt	`NODATA`	[numerisk: dubbel] Standard: 0,0	Tilldelar ett specificerat NoData-värde till utgångsbanden
Rasteriserad	`OUTPUT`	[raster] Standard: `[Spara till temporär fil]`	Specifikation av rasterlagret för utdata. *En av*: • Spara till en temporär fil • Spara till fil… För `Spara till fil` måste utdataformatet anges. Alla GDAL-rasterformat stöds. För `Spara till en temporär fil` kommer QGIS standardrasterformat att användas.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ”	För att lägga till ett eller flera skapandealternativ som styr det raster som ska skapas (färger, blockstorlek, filkomprimering …). För enkelhetens skull kan du förlita dig på fördefinierade profiler (se *GDAL driver options section*). Batch Process and Model Designer: separera flera alternativ med ett pipe-tecken (|).
Typ av utdata	`DATA_TYPE`	[uppräkning] Standard: 5	Definierar formatet för den utgående rasterfilen. Alternativ: • 0 — Byte (Åtta bitars osignerat heltal (quint8)) • 1 — Int16 (Sexton bitars signerat heltal (qint16)) • 2 — UInt16 (Sexton bitars osignerat heltal (quint16)) • 3 — UInt32 (Trettio tvåbitars osignerat heltal (quint32)) • 4 — Int32 (trettio tvåbitars signerade heltal (qint32)) • 5 — Float32 (trettio tvåbitars flyttal (float)) • 6 — Float64 (Sextiofyra bitars flyttal (dubbel)) • 7 — CInt16 (Komplex Int16) • 8 — CInt32 (Komplex Int32) • 9 — CFloat32 (komplex float32) • 10 — CFloat64 (komplex float64) • 11 — Int8 (Åtta bitars signerat heltal (qint8)) Tillgängliga alternativ beror på vilken GDAL-version som byggts med QGIS (se *Hjälp* ► *Om*-menyn)
Förinitiera utdatabilden med värde Valfritt	`INIT`	[numerisk: dubbel]	Förinitialiserar de utgående bildbanden med detta värde. Inte markerat som NoData-värde i utdatafilen. Samma värde används i alla band.
Invertera rasterisering	`INVERT`	[boolean] Standard: Falsk	Bränner det fasta brännvärdet eller det brännvärde som är associerat med den första funktionen i alla delar av bilden som inte ligger inom den angivna polygonen.
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: ”	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Rasteriserad	`OUTPUT`	[raster]	Utgående rasterlager

Python-kod

Algoritm-ID: `gdal:rasterize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.2.7 Geoprocessning av vektorer

Buffertvektorer

Skapa buffertar runt funktionerna i ett vektorlager.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlagret
Namn på geometrikolumn	`GEOMETRY`	[sträng] Standard: ”geometry	Namnet på den geometriska kolumn för inmatningslagret som ska användas
Buffertavstånd	`DISTANCE`	[numerisk: dubbel] Standard: 10.0	Minimum: 0.0
Löses upp genom attribut Valfritt	`FIELD`	[tabellfält: alla] Standard: Ej inställd	Fält att använda för upplösning
Lösa upp resultaten	`DISSOLVE`	[boolean] Standard: Falsk	Om den är inställd upplöses resultatet. Om inget fält anges för dissolving, löses alla buffertar upp till en funktion.
Skapa en funktion för varje geometri i alla typer av geometrisamlingar i källfilen	`EXPLODE_COLLEC`	[boolean] Standard: Falsk	
Buffert	`OUTPUT`	[vektor: polygon] Standard: `[Spara till temporär fil]`	Ange buffertlagret för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ” (inga ytterligare alternativ)	Ytterligare alternativ för skapande av GDAL.

Utgångar

Etikett	Namn	Typ	Beskrivning
Buffert	`OUTPUT`	[vektor: polygon]	Det utgående buffertlagret

Python-kod

Algoritm-ID: `gdal:buffervektorer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Klipp ut vektor efter omfattning

Klipper en OGR-stödd vektorfil till en given omfattning.

Denna algoritm är hämtad från verktyget GDAL ogr2ogr.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlagret

continues on next page

Table 24.281 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Clip utsträckning	`EXTENT`	[omfattning]	Definierar den avgränsande box som ska användas för den utgående vektorfilen. Den måste definieras i CRS-målkoordinater. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Klippt (omfattning)	`OUTPUT`	[samma som inmatning] Standard: `[Spara till temporär fil]`	Ange det utgående (klippta) lagret. *Ett av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ” (inga ytterligare alternativ)	Ytterligare alternativ för skapande av GDAL.

Utgångar

Etikett	Namn	Typ	Beskrivning
Klippt (omfattning)	`OUTPUT`	[samma som inmatning]	Det utgående (klippta) lagret. Standardformatet är ”ESRI Shapefile”.

Python-kod

Algoritm-ID: `gdal:clipvectorbyextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Clip vektor med masklager

Klipper ett OGR-stöttat vektorlager med ett maskpolygonlager.

Denna algoritm är hämtad från verktyget GDAL ogr2ogr.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlagret
Masklager	`MASK`	[vektor: polygon]	Lager som ska användas som klippningsomfattning för inputvektorlagret.
Klippt (mask)	`OUTPUT`	[samma som inmatning] Standard: `[Spara till temporär fil]`	Det utgående (maskerade) lagret. *Ett av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ” (inga ytterligare alternativ)	Ytterligare alternativ för skapande av GDAL.

Utgångar

Etikett	Namn	Typ	Beskrivning
Klippt (mask)	`OUTPUT`	[samma som inmatning]	Det utgående (maskerade) lagret. Standardformatet är ”ESRI Shapefile”.

Python-kod

Algoritm-ID: `gdal:clipvectorbypolygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Lös upp

Lös upp (kombinera) geometrier som har samma värde för ett givet attribut/fält. Utdatageometrierna är flerdelade.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Inmatningslagret för att lösa upp
Lös upp fältet Valfritt	`FIELD`	[tabellfält: alla]	Fältet för inmatningslagret som ska användas för upplösning
Namn på geometrikolumn	`GEOMETRY`	[sträng] Standard: ”geometry	Namnet på den geometriska kolumn i inmatningslagret som ska användas för upplösning.
Upplöst	`OUTPUT`	[samma som inmatning] Standard: `[Spara till temporär fil]`	Ange utdatalagret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Skapa en funktion för varje geometri i alla typer av geometrisamlingar i källfilen	`EXPLODE_COLLEC`	[boolean] Standard: Falsk	Ta fram en funktion för varje geometri i alla typer av geometrisamlingar i källfilen
Behåll inmatningsattribut	`KEEP_ATTRIBUTE`	[boolean] Standard: Falsk	Behåll alla attribut från inmatningslagret
Räkna upplösta funktioner	`COUNT_FEATURES`	[boolean] Standard: Falsk	Räkna de upplösta funktionerna och inkludera dem i utdatalagret.
Beräkna area och omkrets för upplösta objekt	`COMPUTE_AREA`	[boolean] Standard: Falsk	Beräkna arean och omkretsen för upplösta objekt och inkludera dem i utdatalagret
Beräkna min/max/summa/m för attribut	`COMPUTE_STATIS`	[boolean] Standard: Falsk	Beräkna statistik (min, max, summa och medelvärde) för det angivna numeriska attributet och inkludera dem i utdatalagret

continues on next page

Table 24.283 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Numeriskt attribut att beräkna statistik på Valfritt	`STATISTICS_ATT`	[tabellfält: numerisk]	Det numeriska attribut som statistiken ska beräknas på
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ” (inga ytterligare alternativ)	Ytterligare alternativ för skapande av GDAL.

Utgångar

Etikett	Namn	Typ	Beskrivning
Upplöst	`OUTPUT`	[samma som inmatning]	Geometrilager med flera delar som utdata (med upplösta geometrier)

Python-kod

Algoritm ID: `gdal:upplösning`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Offset-kurva

Förskjuter linjer med ett angivet avstånd. Positiva avstånd förskjuter linjerna åt vänster och negativa avstånd förskjuter dem åt höger.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje]	Det ingående linjelagret
Namn på geometrikolumn	`GEOMETRY`	[sträng] Standard: ”geometry	Namnet på den geometriska kolumn för inmatningslagret som ska användas
Förskjutningsavstånd (vänstersidigt: positivt, högersidigt: negativt)	`DISTANCE`	[numerisk: dubbel] Standard: 10.0	

continues on next page

Table 24.284 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Offset-kurva	`OUTPUT`	[vektor: linje] Standard: `[Spara till temporär fil]`	Ange utmatningens linjelager. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ” (inga ytterligare alternativ)	Ytterligare alternativ för skapande av GDAL.

Utgångar

Etikett	Namn	Typ	Beskrivning
Offset-kurva	`OUTPUT`	[vektor: linje]	Lagret för utmatningens offsetkurva

Python-kod

Algoritm-ID: `gdal:offsetkurva`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Buffert på ena sidan

Skapar en buffert på ena sidan (höger eller vänster) av linjerna i ett linjevektorlager.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje]	Det ingående linjelagret
Namn på geometrikolumn	`GEOMETRY`	[sträng] Standard: ”geometry	Namnet på den geometriska kolumn för inmatningslagret som ska användas
Buffertavstånd	`DISTANCE`	[numerisk: dubbel] Standard: 10.0	

continues on next page

Table 24.285 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Buffert sida	`BUFFER_SIDE`	[uppräkning] Standard: 0	En av: • 0 — Höger • 1 — vänster
Löses upp genom attribut Valfritt	`FIELD`	[tabellfält: alla] Standard: Ej inställd	Fält att använda för upplösning
Lös upp alla resultat	`DISSOLVE`	[boolean] Standard: Falsk	**Om den är inställd blir resultatet upplöst.** Om inget fält anges för upplösning, upplöses alla buffertar till en funktion.
Skapa en funktion för varje geometri i alla typer av geometrisamlingar i källfilen	`EXPLODE_COLLEC`	[boolean] Standard: Falsk	
Ensidig buffert	`OUTPUT`	[vektor: polygon] Standard: `[Spara till temporär fil]`	Ange buffertlagret för utdata. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ” (inga ytterligare alternativ)	Ytterligare alternativ för skapande av GDAL.

Utgångar

Etikett	Namn	Typ	Beskrivning
Ensidig buffert	`OUTPUT`	[vektor: polygon]	Det utgående buffertlagret

Python-kod

Algoritm ID: `gdal:onesidebuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Punkter längs linjer

Genererar en punkt på varje linje i ett linjevektorlager på ett avstånd från start. Avståndet anges som en bråkdel av linjens längd.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: linje]	Det ingående linjelagret
Namn på geometrikolumn	`GEOMETRY`	[sträng] Standard: ”geometry	Namnet på den geometriska kolumn för inmatningslagret som ska användas
Avstånd från linjestart representerat som en bråkdel av linjelängden	`DISTANCE`	[numerisk: dubbel] Standard: 0,5	
Pekar längs linjerna	`OUTPUT`	[vektor: punkt] Standard: `[Spara till temporär fil]`	Ange det utgående punktlagret. *En av*: • Spara till en temporär fil • Spara till fil…

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ” (inga ytterligare alternativ)	Ytterligare alternativ för skapande av GDAL.

Utgångar

Etikett	Namn	Typ	Beskrivning
Pekar ut längs linjen	`OUTPUT`	[vektor: punkt]	Det utgående punktlagret

Python-kod

Algoritm-ID: `gdal:punkter längs linjer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

24.2.8 Vektor diverse

Bygg virtuell vektor

Skapar ett virtuellt vektorlager som innehåller en uppsättning vektorlager.

Denna algoritm är särskilt användbar om en annan algoritm behöver flera lager men endast accepterar en `vrt` där lagren specificeras.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningsdatakäll	`INPUT`	[vektor: valfri] [lista]	Välj de vektorlager som du vill använda för att bygga upp den virtuella vektorn
Skapa en ”sammanfogad” VRT	`UNIONED`	[boolean] Standard: Falsk	Kontrollera om du vill samla alla vektorer i en enda `vrt`-fil
Virtuell vektor	`OUTPUT`	[samma som inmatning] Standard: `[Spara till temporär fil]`	Ange utdatalagret som endast innehåller duplikaten. *En av*: • Spara till en temporär fil • Spara till fil…

Utgångar

Etikett	Namn	Typ	Beskrivning
Virtuell vektor	`OUTPUT`	[vektor: vilken som helst]	Den virtuella vektorn för utdata som skapas från de valda källorna

Python-kod

Algoritm-id: `gdal:buildvirtualvector`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exekvera SQL

Kör en enkel eller komplex fråga med SQL-syntax på källagret. Resultatet av frågan kommer att läggas till som ett nytt lager.

Denna algoritm är hämtad från verktyget GDAL ogr2ogr.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	OGR-stödd inmatningsvektorlager
SQL-uttryck	`SQL`	[sträng]	Definierar SQL-frågan, t.ex. `SELECT * FROM my_table WHERE name is not null`.
SQL-dialekt	`DIALECT`	[uppräkning] Standard: 0	SQL-dialekt som ska användas. En av: • 0 — Ingen • 1 — OGR SQL • 2 — SQLite
SQL-resultat	`OUTPUT`	[vektor: vilken som helst]	Specifikation av utdatalagret. *En av*: • Spara till en temporär fil • Spara till fil… För `Spara till fil` måste utdataformatet anges. Alla GDAL-vektorformat stöds. För `Spara till en temporär fil` används standardformatet för utdatavektorlager.

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ” (inga ytterligare alternativ)	Ytterligare alternativ för skapande av GDAL.

Utgångar

Etikett	Namn	Typ	Beskrivning
SQL-resultat	`OUTPUT`	[vektor: vilken som helst]	Vektorlager som skapats av frågan

Python-kod

Algoritm ID: `gdal:executesql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exportera till PostgreSQL (tillgängliga anslutningar)

Importerar vektorlager i en PostgreSQL-databas på grundval av en tillgänglig anslutning. Anslutningen måste *definieras korrekt* i förväg. Var uppmärksam på att kryssrutorna 'Spara användarnamn' och 'Spara lösenord' är aktiverade. Sedan kan du använda algoritmen.

Denna algoritm är hämtad från verktyget GDAL ogr2ogr.

Parametrar

Etikett	Namn	Typ	Beskrivning
Databas (anslutningsnamn)	`DATABASE`	[sträng]	PostgreSQL-databasen att ansluta till
Inmatningslager	`INPUT`	[vektor: vilken som helst]	OGR-stött vektorlager för export till databasen
Formkodning Valfritt	`SHAPE_ENCODING`	[sträng] Standard: ''	Anger den kodning som ska tillämpas på data

continues on next page

Table 24.288 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Typ av geometri för utdata	`GTYPE`	[uppräkning] Standard: 0	Definierar typ av geometri för utdata. En av: • 0 — • 1 — INGEN • 2 — GEOMETRI • 3 — PUNKT • 4 — LINJESTRING • 5 — POLYGON • 6 — GEOMETRISAMLING • 7 — MULTIPUNKT • 8 — MULTIPOLYGON • 9 — MULTILINSTRÄNG • 10 — CIRKULÄRSTRÄNG • 11 — KOUMPUNDSKURVA • 12 — KURVAPOLYGON • 13 — FLERKURVA • 14 — FLERA YTOR • 15 — KONVERTERA_TILL_LINJÄR • 16 — KONVERTERA_TILL_KURVA
Tilldela en utgång CRS Valfritt	`A_SRS`	[crs] Standard: Ej inställd	Definierar CRS-utmatningen för databastabellen
Reprojicera till denna CRS på utdata Valfritt	`T_SRS`	[crs] Standard: Ej inställd	Reprojicerar/transformerar till denna CRS vid utmatning
Åsidosättande av källan CRS Valfritt	`S_SRS`	[crs] Standard: Ej inställd	Åsidosätter inmatningslagrets CRS
Schema (schemats namn) Valfritt	`SCHEMA`	[sträng] Standard: ”offentlig	Definierar schemat för databastabellen
Tabell att exportera till (lämna tomt för att använda lagernamn) Valfritt	`TABLE`	[sträng] Standard: ”	Definierar ett namn för den tabell som ska importeras till databasen. Som standard är tabellnamnet namnet på den ingående vektorfilen.
Primärnyckel (nytt fält) Valfritt	`PK`	[sträng] Standard: ”id	Definierar vilket attributfält som ska vara primärnyckel i databastabellen
Primary Key (befintligt fält, används om ovanstående alternativ lämnas tomt) Valfritt	`PRIMARY_KEY`	[tabellfält: alla] Standard: Ej inställd	Definierar vilket attributfält i det exporterade lagret som ska vara primärnyckel i databastabellen
Namn på geometrikolumn Valfritt	`GEOCOLUMN`	[sträng] Standard: ’geom’	Definierar i vilket attributfält i databasen som geometriinformationen ska finnas

continues on next page

Table 24.288 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
** Vektordimensioner** Valfritt	`DIM`	[uppräkning] Standard: 0 (2D)	Definierar koordinatdimensionerna för de importerade vektordata. En av: • 0 — 2 (`XY`) • 1 — 3 (`XYZ`) • 2 — 4 (`XYZM`)
Avståndstolerans för förenkling Valfritt	`SIMPLIFY`	[sträng] Standard: ”	Definierar en avståndstolerans för förenkling av de vektorgeometrier som ska importeras. Som standard sker ingen förenkling.
Maximalt avstånd mellan 2 noder (förtätning) Valfritt	`SEGMENTIZE`	[sträng] Standard: ”	Det maximala avståndet mellan två noder. Används för att skapa mellanliggande punkter. Som standard sker ingen förtätning.
Välj objekt efter omfattning (definierad i CRS för inmatningslagret) Valfritt	`SPAT`	[omfattning] Standard: Ej inställd	Du kan välja funktioner från en given omfattning som ska finnas i utdatatabellen. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`
Klipp in inmatningslagret med hjälp av ovanstående (rektangel) utsträckning	`CLIP`	[boolean] Standard: Falsk	Inmatningslagret kommer att klippas med den utsträckning som du definierade tidigare
Välj funktioner med hjälp av en SQL ”WHERE”-sats (t.ex. kolumn=”värde”) Valfritt	`WHERE`	[sträng] Standard: ”	Definierar med en SQL ”WHERE”-sats vilka funktioner som ska väljas från inmatningslagret
Grupp N funktioner per transaktion (standard: 2000) Valfritt	`GT`	[sträng] Standard: ”	Du kan gruppera inmatningsfunktionerna i transaktioner där N definierar storleken. Som standard begränsar N transaktionsstorleken till 20000 funktioner.
Overwrite existing table (Skriv över befintlig tabell)	`OVERWRITE`	[boolean] Standard: Sann	Om det finns en tabell med samma namn i databasen och om detta alternativ är inställt på True, skrivs tabellen över.

continues on next page

Table 24.288 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Lägg till i befintlig tabell	APPEND	[boolean] Standard: Falsk	Om markerad / True kommer vektordata att läggas till i en befintlig tabell. Nya fält som hittas i inmatningslagret ignoreras. Som standard kommer en ny tabell att skapas.
Lägga till och lägga till nya fält i en befintlig tabell	ADDFIELDS	[boolean] Standard: Falsk	Om den är aktiverad kommer vektordata att läggas till i en befintlig tabell, ingen ny tabell kommer att skapas. Nya fält som hittas i inmatningslagret läggs till i tabellen. Som standard kommer en ny tabell att skapas.
Tvätta inte namn på kolumner/tabeller	LAUNDER	[boolean] Standard: Falsk	Om det här alternativet är markerat kan du förhindra standardbeteendet (konvertera kolumnnamn till gemener, ta bort mellanslag och andra ogiltiga tecken).
Skapa inte ett spatialt index	INDEX	[boolean] Standard: Falsk	Förhindrar att ett spatialt index för utdatatabellen skapas. Som standard läggs ett spatialt index till.
Fortsätt efter ett misslyckande, hoppa över den misslyckade funktionen	SKIPFAILURES	[boolean] Standard: Falsk	Fortsätt efter ett fel, hoppa över den felaktiga funktionen.
Validera geometrier baserat på Simple Features-specifikationen	MAKEVALID	[boolean] Standard: Falsk	Tillämpar rensningsoperationer på geometrier för att säkerställa att de är giltiga enligt reglerna i Simple Features-specifikationen.
Promote till Multipart	PROMOTETOMULTI	[boolean] Standard: Sann	Slår samman geometritypen för funktioner till multipart i utmatningstabellen
Behåll bredd och precision i inmatningsattribute	PRECISION	[boolean] Standard: Sann	Undviker att ändra kolumnattribut för att överensstämma med indata
Andra alternativ för skapande Valfritt	OPTIONS	[sträng] Standard: ” (inga ytterligare alternativ)	Ytterligare alternativ för skapande av GDAL.

Utgångar

Denna algoritm har ingen utgång.

Python-kod

Algoritm-id: `gdal:importvectorintopostgisdatabaseavailableconnections`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Exportera till PostgreSQL (ny anslutning)

Importerar vektorlager i en PostgreSQL-databas. En ny anslutning till PostGIS-databasen måste skapas.

Denna algoritm är hämtad från verktyget GDAL ogr2ogr.

Parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: vilken som helst]	OGR-stött vektorlager för export till databasen
Formkodning Valfritt	`SHAPE_ENCODING`	[sträng] Standard: ”	Anger den kodning som ska tillämpas på data
Typ av geometri för utdata	`GTYPE`	[uppräkning] Standard: 0	Definierar typ av geometri för utdata. En av: • 0 — • 1 — INGEN • 2 — GEOMETRI • 3 — PUNKT • 4 — LINJESTRING • 5 — POLYGON • 6 — GEOMETRISAMLING • 7 — MULTIPUNKT • 8 — MULTIPOLYGON • 9 — MULTILINSTRÄNG • 10 — CIRKULÄRSTRÄNG • 11 — KOUMPUNDSKURVA • 12 — KURVAPOLYGON • 13 — FLERKURVA • 14 — FLERA YTOR • 15 — KONVERTERA_TILL_LINJÄR • 16 — KONVERTERA_TILL_KURVA
Tilldela en utgång CRS Valfritt	`A_SRS`	[crs] Standard: Ej inställd	Definierar CRS-utmatningen för databastabellen
Reprojicera till denna CRS på utdata Valfritt	`T_SRS`	[crs] Standard: Ej inställd	Reprojicerar/transformerar till denna CRS vid utmatning
Åsidosättande av källan CRS Valfritt	`S_SRS`	[crs] Standard: Ej inställd	Åsidosätter inmatningslagrets CRS
Värd Valfritt	`HOST`	[sträng] Standard: ”localhost	Namn på databasens värd
Port Valfritt	`PORT`	[sträng] Standard: ’5432’	Portnummer som PostgreSQL-databasservern lyssnar på
Användarnamn Valfritt	`USER`	[sträng] Standard: ”	Användarnamn som används för att logga in i databasen
Databasens namn Valfritt	`DBNAME`	[sträng] Standard: ”	Namn på databasen
Lösenord Valfritt	`PASSWORD`	[sträng] Standard: ”	Lösenord som används tillsammans med användarnamnet för att ansluta till databasen

continues on next page

Table 24.289 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Schema (schemats namn) Valfritt	`SCHEMA`	[sträng] Standard: ”offentlig	Definierar schemat för databastabellen
Tabellnamn, lämna tomt för att använda inmatningsnamn Valfritt	`TABLE`	[sträng] Standard: ”	Definierar ett namn för den tabell som ska importeras till databasen. Som standard är tabellnamnet namnet på den ingående vektorfilen.
Primärnyckel (nytt fält) Valfritt	`PK`	[sträng] Standard: ”id	Definierar vilket attributfält som ska vara primärnyckel i databastabellen
Primary Key (befintligt fält, används om ovanstående alternativ lämnas tomt) Valfritt	`PRIMARY_KEY`	[tabellfält: alla] Standard: Ej inställd	Definierar vilket attributfält i det exporterade lagret som ska vara primärnyckel i databastabellen
Namn på geometrikolumn Valfritt	`GEOCOLUMN`	[sträng] Standard: ’geom’	Definierar i vilket attributfält geometriinformationen ska lagras
** Vektordimensioner** Valfritt	`DIM`	[uppräkning] Standard: 0 (2D)	Definierar koordinatdimensionerna för de importerade vektordata. En av: • 0 — 2 (`XY`) • 1 — 3 (`XYZ`) • 2 — 4 (`XYZM`)
Avståndstolerans för förenkling Valfritt	`SIMPLIFY`	[sträng] Standard: ”	Definierar en avståndstolerans för förenkling av de vektorgeometrier som ska importeras. Som standard sker ingen förenkling.
Maximalt avstånd mellan 2 noder (förtätning) Valfritt	`SEGMENTIZE`	[sträng] Standard: ”	Det maximala avståndet mellan två noder. Används för att skapa mellanliggande punkter. Som standard sker ingen förtätning.
Välj objekt efter omfattning (definierad i CRS för inmatningslagret) Valfritt	`SPAT`	[omfattning] Standard: Ej inställd	Du kan välja funktioner från en given omfattning som ska finnas i utdatatabellen. Tillgängliga metoder är: • Beräkna från lager…: använder omfattningen av ett lager som laddats i det aktuella projektet • Beräkna från layoutkarta…: använder omfattningen av ett *layoutkartobjekt* i det aktiva projektet • Beräkna från bokmärke…: använder omfattningen av ett sparat *bokmärke* • Använd kartans omfattning • Rita på duken: klicka och dra en rektangel som avgränsar det område som ska tas med i beräkningen • Ange koordinaterna som `xmin, xmax, ymin, ymax`

continues on next page

Table 24.289 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Klipp in inmatningslagret med hjälp av ovanstående (rektangel) utsträckning	CLIP	[boolean] Standard: Falsk	Inmatningslagret kommer att klippas med den utsträckning som du definierade tidigare
Fält att inkludera (lämna tomt för att använda alla fält) Valfritt	FIELDS	[sträng] [lista] Standard: []	Definierar fält som ska behållas från den importerade vektorfilen. Om ingen är markerad importeras alla fält.
Välj funktioner med hjälp av en SQL ”WHERE”-sats (t.ex. kolumn=”värde”) Valfritt	WHERE	[sträng] Standard: ”	Definierar med en SQL ”WHERE”-sats vilka funktioner som ska väljas för utdatatabellen
Grupp N funktioner per transaktion (standard: 2000) Valfritt	GT	[sträng] Standard: ”	Du kan gruppera inmatningsfunktionerna i transaktioner där N definierar storleken. Som standard begränsar N transaktionsstorleken till 20000 funktioner.
Overwrite existing table (Skriv över befintlig tabell)	OVERWRITE	[boolean] Standard: Sann	Om det finns en tabell med samma namn i databasen och om detta alternativ är inställt på True, skrivs tabellen över.
Lägg till i befintlig tabell	APPEND	[boolean] Standard: Falsk	Om markerad / True kommer vektordata att läggas till i en befintlig tabell. Nya fält som hittas i inmatningslagret ignoreras. Som standard kommer en ny tabell att skapas.
Lägga till och lägga till nya fält i en befintlig tabell	ADDFIELDS	[boolean] Standard: Falsk	Om den är aktiverad kommer vektordata att läggas till i en befintlig tabell, det kommer inte att skapas en ny tabell. Nya fält som hittas i inmatningslagret läggs till i tabellen. Som standard kommer en ny tabell att skapas.
Tvätta inte namn på kolumner/tabeller	LAUNDER	[boolean] Standard: Falsk	Om det här alternativet är markerat kan du förhindra standardbeteendet (konvertera kolumnnamn till gemener, ta bort mellanslag och andra ogiltiga tecken).
Skapa inte ett spatialt index	INDEX	[boolean] Standard: Falsk	Förhindrar att ett spatialt index för utdatatabellen skapas. Som standard läggs ett spatialt index till.
Fortsätt efter ett misslyckande, hoppa över den misslyckade funktionen	SKIPFAILURES	[boolean] Standard: Falsk	Fortsätt efter ett fel, hoppa över den felaktiga funktionen.
Validera geometrier baserat på Simple Features-specifikationen	MAKEVALID	[boolean] Standard: Falsk	Tillämpar rensningsoperationer på geometrier för att säkerställa att de är giltiga enligt reglerna i Simple Features-specifikationen.
Promote till Multipart	PROMOTETOMULTI	[boolean] Standard: Sann	Slår samman geometritypen för funktioner till multipart i utmatningstabellen

continues on next page

Table 24.289 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Behåll bredd och precision i inmatningsattribute	`PRECISION`	[boolean] Standard: Sann	Undviker att ändra kolumnattribut för att överensstämma med indata
Andra alternativ för skapande Valfritt	`OPTIONS`	[sträng] Standard: ” (inga ytterligare alternativ)	Ytterligare alternativ för skapande av GDAL.

Utgångar

Denna algoritm har ingen utgång.

Python-kod

Algoritm ID: `gdal:importvektorintopostgisdatabasenewconnection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Vektorinformation

Skapar en informationsfil som listar information om en OGR-stödd datakälla. Resultatet visas i ett ”Result”-fönster och kan skrivas in i en HTML-fil. Informationen omfattar geometrityp, antal objekt, spatial utsträckning, projektionsinformation, lista över attribut och relaterade typer, lista över relationer och fältdomäner, och mycket mer.

Denna algoritm är hämtad från verktyget GDAL ogrinfo.

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Aktivera listning av alla lager i datasetet	`ALL_LAYERS`	[boolean] Standard: Falsk	Om markerad kommer QGIS att mata ut information om alla lager i indatadatasetet istället för information om endast det första lagret. Lämpligt när ett format med flera lager (GeoPackage, GML, KML, SpatiaLite, SQLite, …) används som indata.
Endast sammanfattning av utdata	`SUMMARY_ONLY`	[boolean] Standard: Sann	Undertryck listning av enskilda funktioner och visa endast sammanfattande information som projektion, schema, antal funktioner och utsträckning.

continues on next page

Table 24.290 – fortsättning från föregående sida

Etikett	Namn	Typ	Beskrivning
Undertryck metadata-information	`NO_METADATA`	[boolean] Standard: Falsk	Undertryck utskrift av metadata. Vissa dataset kan innehålla många metadatasträngar.
Lagerinformation	`OUTPUT`	[html] Standard: `[Spara till temporär fil]`	Ange HTML-filen för utdata som innehåller filinformationen. *En av*: • Spara till en temporär fil • Spara till fil… Om ingen HTML-fil har definierats kommer utdata att skrivas till en temporär fil

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: ” (inga ytterligare alternativ)	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Lagerinformation	`OUTPUT`	[html]	Den utgående HTML-filen som innehåller filinformationen.

Python-kod

Algoritm ID: `gdal:ogrinfo`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

Vektorinformation (JSON)

Skapar en informationsfil som listar information om en OGR-stödd datakälla. Resultatet visas i ett ”Result”-fönster och kan skrivas till en `.JSON`-fil. Informationen omfattar geometrityp, antal objekt, spatial utsträckning, projektionsinformation, en lista över attribut och relaterade typer, en lista över relationer och fältdomäner och mycket mer.

Denna algoritm är hämtad från verktyget GDAL ogrinfo.

Kräver version av GDAL >= 3.7

Parametrar

Grundläggande parametrar

Etikett	Namn	Typ	Beskrivning
Inmatningslager	`INPUT`	[vektor: geometri]	Ingångsvektorlager
Aktivera listning av alla lager i datasetet	`ALL_LAYERS`	[boolean] Standard: Falsk	Om markerad kommer QGIS att mata ut information om alla lager i indatadatasetet istället för information om endast det första lagret. Lämpligt när ett format med flera lager (GeoPackage, GML, KML, SpatiaLite, SQLite, …) används som indata.
Aktivera listning av funktioner	`FEATURES`	[boolean] Standard: Falsk	
Undertryck metadata-information	`NO_METADATA`	[boolean] Standard: Falsk	Undertryck utskrift av metadata. Vissa dataset kan innehålla många metadatasträngar.
Lagerinformation	`OUTPUT`	&Arkiv Standard: `[Spara till temporär fil]`	Ange JSON-filen för utdata som innehåller filinformationen. *En av*: • Spara till en temporär fil • Spara till fil… Om ingen JSON-fil har definierats skrivs utdata till en tillfällig fil

Avancerade parametrar

Etikett	Namn	Typ	Beskrivning
Ytterligare kommandoradspara Valfritt	`EXTRA`	[sträng] Standard: ” (inga ytterligare alternativ)	Lägg till extra GDAL-kommandoradsalternativ

Utgångar

Etikett	Namn	Typ	Beskrivning
Lagerinformation	`OUTPUT`	&Arkiv	Utdatafilen *.JSON* som innehåller filinformationen.

Python-kod

Algoritm ID: `gdal:ogrinfojson`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritm id visas när du håller muspekaren över algoritmen i verktygslådan Processing Toolbox. I *parameter dictionary* finns parameternamn och värden. Se *Använda bearbetningsalgoritmer från konsolen* för information om hur du kör bearbetningsalgoritmer från Python-konsolen.

KAPITEL 25

Tillägg

25.1 QGIS tillägg

QGIS har utformats med en plugin-arkitektur. Detta gör att många nya egenskaper och funktioner enkelt kan läggas till i programmet. Några av funktionerna i QGIS är faktiskt implementerade som tillägg.

25.1.1 Kärn- och externa tillägg

QGIS-tillägg implementeras antingen som **Core Tillägg** eller **External Tillägg**.

Core Tillägget underhålls av QGIS Development Team och är automatiskt en del av varje QGIS-distribution. De är skrivna i ett av två språk: **C++** eller **Python**.

De flesta av de externa tilläggen är för närvarande skrivna i Python. De lagras antingen i det ”officiella” QGIS-arkivet på https://tillägg.qgis.org/tillägg/ eller i externa arkiv och underhålls av de enskilda författarna. Detaljerad dokumentation om användning, minsta QGIS-version, hemsida, författare och annan viktig information finns för tilläggen i det officiella arkivet. För andra externa arkiv kan dokumentation finnas tillgänglig med de externa tilläggen själva. Dokumentationen för externa tillägg ingår inte i den här handboken.

För att installera eller aktivera ett plugin, gå till *Tillägg* menyn och välj *Manage and install tillägg…*. Installerade externa pythonplugins placeras i mappen `python/tillägg` i den aktiva sökvägen *användarprofil*.

Sökvägar till bibliotek med anpassade C++-tillägg kan också läggas till under *Inställningar* ► *Alternativ* ► *System*.

25.1.2 Dialogrutan för tillägg

Fliken Inställningar

Fliken *Inställningar* längst ned i den vänstra panelen är den huvudsakliga platsen där du kan konfigurera vilka tillägg som kan visas i din applikation. Du kan använda följande alternativ:

- *Kontrollera efter uppdateringar vid start*. När ett installerat plugin har en tillgänglig uppdatering kommer QGIS att informera dig *Varje gång QGIS startar*, *En gång om dagen*, *Var tredje dag*, *Var vecka*, *Varannan vecka* eller *Var månad*.
- *Visa även experimentella tillägg*. QGIS kommer att visa dig tillägg i tidiga utvecklingsstadier, som i allmänhet är olämpliga för produktionsanvändning. För dessa tillägg kan du installera antingen den stabila eller den experimentella versionen, och när som helst byta från den ena till den andra.
- *Visa även utfasade tillägg*. Dessa tillägg underhålls vanligtvis inte eftersom de har ersättningsfunktioner i QGIS, brist på underhållare, de förlitar sig på funktioner som inte längre är tillgängliga i QGIS… De är i allmänhet olämpliga för produktionsanvändning och visas i grått i listan över tillägg.

Som standard, i avsnittet *Plugin Repositories*, ger QGIS dig sitt officiella plugin-arkiv med URL `https://tillägg.qgis.org/tillägg/tillägg.xml?qgis=version` (där `<version>` representerar den exakta QGIS-versionen du kör). För att lägga till externa författararkiv, klicka på *Lägg till…* och fyll i formuläret *Repository Details* med ett namn och URL. URL:en kan vara av protokolltyp `http://` eller `file://`.

Standard QGIS-arkivet är ett öppet arkiv och du behöver inte någon autentisering för att komma åt det. Du kan dock distribuera ditt eget plugin-arkiv och kräva en autentisering (grundläggande autentisering, PKI). Du kan få mer information om QGIS autentiseringsstöd i kapitlet *Autentisering*.

Om du inte vill ha ett eller flera av de tillagda arkiven kan de inaktiveras från fliken Inställningar via knappen *Redigera…*, eller tas bort helt med knappen *Ta bort*.

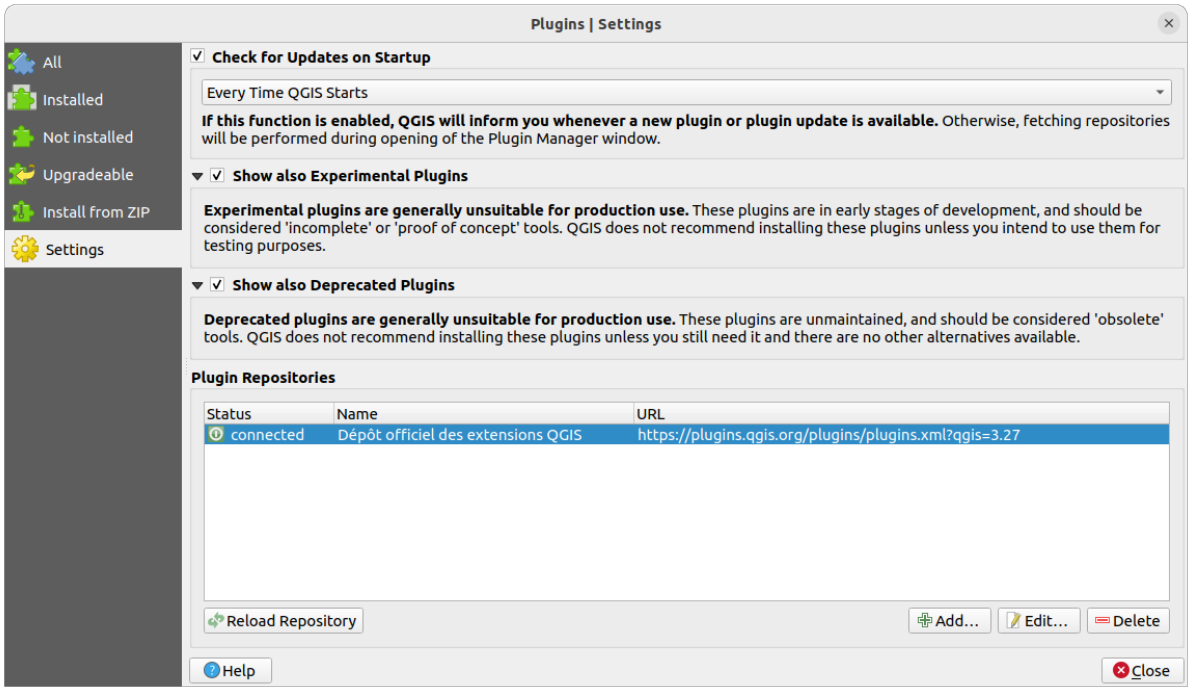


Fig. 25.1: Fliken *Inställningar*

Bläddra bland tilläggen

Flikarna

De övre flikarna i dialogrutan *Tillägg* ger dig listor över tillägg baserat på deras installations-, skapande- eller uppdateringsstatus. Beroende på inställningarna för tillägget kan de tillgängliga flikarna vara:

- *Alla*: visar alla tillgängliga tillägg i de aktiverade lagringsplatserna
- *Installerade*: visar både de tillägg du installerat och de kärntillägg som är installerade som standard och som du inte kan avinstallera
- *Inte installerade*: visar avinstallerade eller ännu inte installerade tillägg i de aktiverade arkiven
- *Nya*: visar tillägg som släppts sedan den senaste *Leta efter uppdateringar vid uppstart*
- *Uppgraderingsbara*: visar installerade tillägg som har publicerat en nyare version i repository
- *Ogiltiga*: visar alla installerade tillägg som för närvarande är trasiga av någon anledning (saknat beroende, fel vid laddning, inkompatibla funktioner med QGIS-versionen …)

Längst upp på flikarna finns en *Sök*-funktion som hjälper dig att hitta ett plugin med hjälp av metadatainformation (författare, namn, beskrivning, tagg, …).

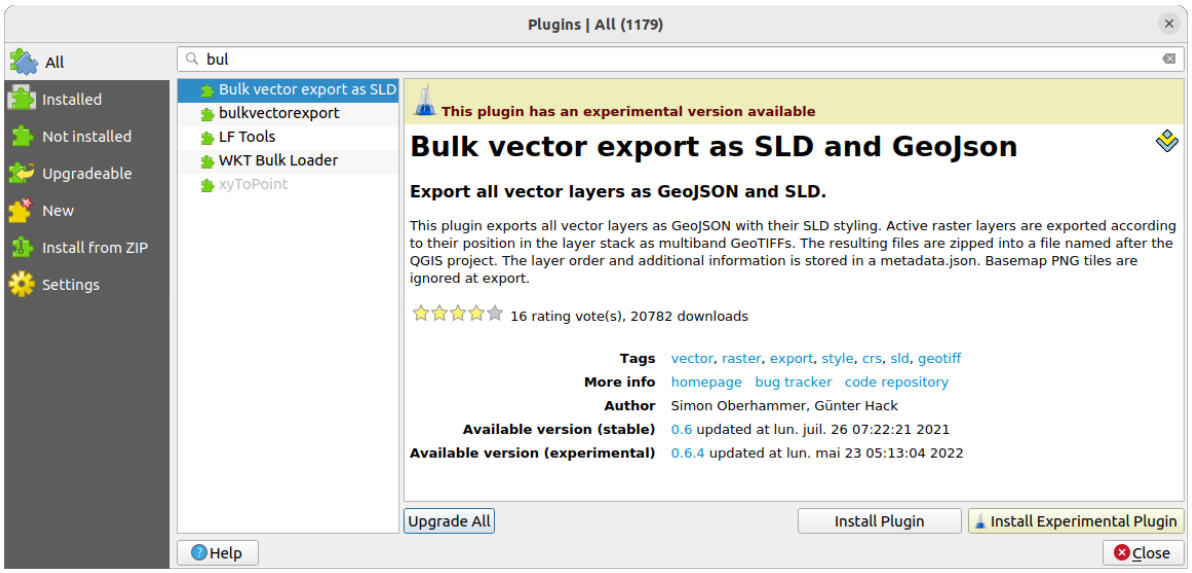


Fig. 25.2: Söka efter ett tillägg från fliken *Alla*

Tillägg

Välj ett tillägg och du kommer att få se metadata i den högra panelen:

- information om huruvida tillägget är experimentellt eller har en experimentell version tillgänglig (om *Visa även experimentella tillägg* är markerat)
- sammanfattning och beskrivning
- betygsröst(er) (du kan rösta på det tillägg du föredrar!)
- tags
- några användbara länkar till hemsidan, spåraren och kodförvaret

- författare(n)
- tillgängliga version(er) med länk till nedladdningssida i arkivet, eller sökväg till lokal mapp för installerade tillägg

I dialogrutan *Tilläggshanterare* kan du interagera med den senaste versionen av tillägg. När detta är aktiverat kan den experimentella versionen endast visas om den är nyare än den senaste stabila versionen. Beroende på den aktiva fliken, om det valda tillägget är installerat, kommer du att få några av följande alternativ:

- *Install*: installerar den senaste stabila versionen av det valda tillägget
- *Install Experimental Plugin*: installerar den experimentella versionen av det valda tillägget
- *Installera om tillägg*: installerar samma stabila version av tillägget, t.ex. efter att det inte har laddats
- *Reinstall Experimental Plugin*: installerar samma experimentella version av tillägget, t.ex. efter att det inte har laddats
- *Uppgradera tillägg*: uppgraderar valt tillägg till den senaste stabila versionen
- *Upgrade Experimental Plugin*: uppgraderar valt plugin till dess experimentella version
- *Upgrade All*: uppgraderar alla installerade tillägg till den senaste stabila eller experimentella versionen (beroende på om den tidigare installerade versionen var stabil eller experimentell).
- *Downgrade Plugin*: flyttar från den experimentella versionen av tillägget till dess tidigare stabila version
- *Downgrade Experimental Plugin*: flyttar från en experimentell version av tillägget till dess senaste publicerade experimentella version. Detta kan inträffa när man spelar med en ännu inte publicerad version.
- *Avinstallera tillägg*: tar bort det installerade tillägget från användarprofilen

Ett installerat tillägg visar en kryssruta till vänster. Avmarkera den för att tillfälligt inaktivera tillägget.

Högerklicka på ett tillägg i listan så kan du sortera tilläggen efter olika metadata. Den nya ordningen gäller för alla flikar. Sorteringsalternativen är:

- *Sortera efter namn*
- *Sortera efter nedladdningar*
- *Sortera efter röst*
- *Sortera efter status*
- *Sortera efter datum för skapande*
- *Sortera efter datum för uppdatering*

Fliken Installera från ZIP

Fliken *Install from ZIP* tillhandahåller en filväljarwidget för att importera tillägg i ett zippat format, t.ex. tillägg som laddats ner direkt från deras arkiv. Krypterade filer stöds.

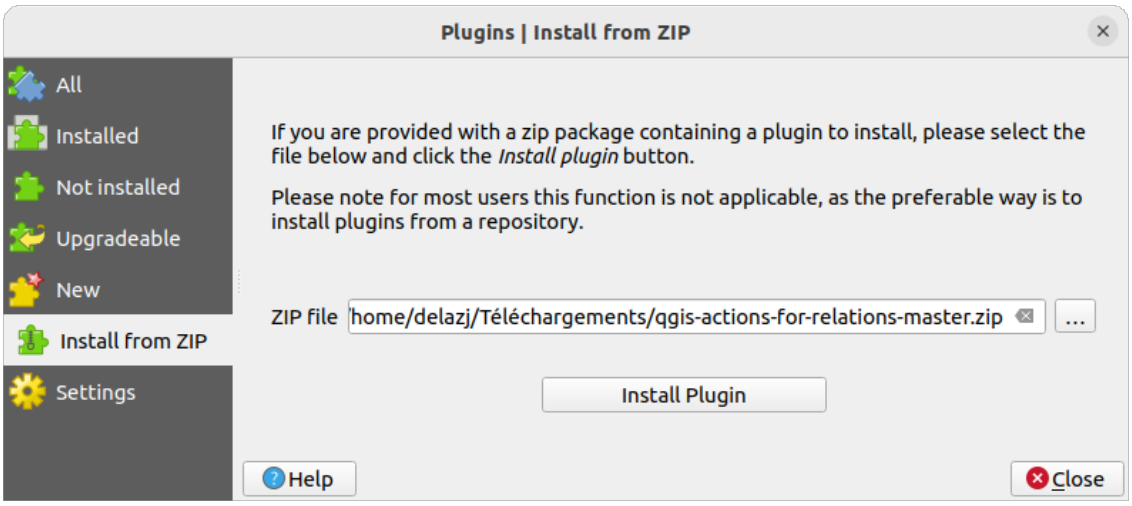


Fig. 25.3: Fliken *Installera från zip*

25.2 Använda QGIS Core Tillägg

25.2.1 DB Manager Plugin

DB Manager-pluginprogrammet är avsett att vara huvudverktyget för att integrera och hantera spatiala databasformat som stöds av QGIS (PostGIS, SpatiaLite, GeoPackage, Oracle Spatial, Virtual layers) i ett användargränssnitt. Tillägget DB Manager innehåller flera funktioner. Du kan dra lager från QGIS Browser till DB Manager, som sedan importerar lagret till din spatiala databas. Du kan dra och släppa tabeller mellan spatiala databaser och de kommer att importeras.

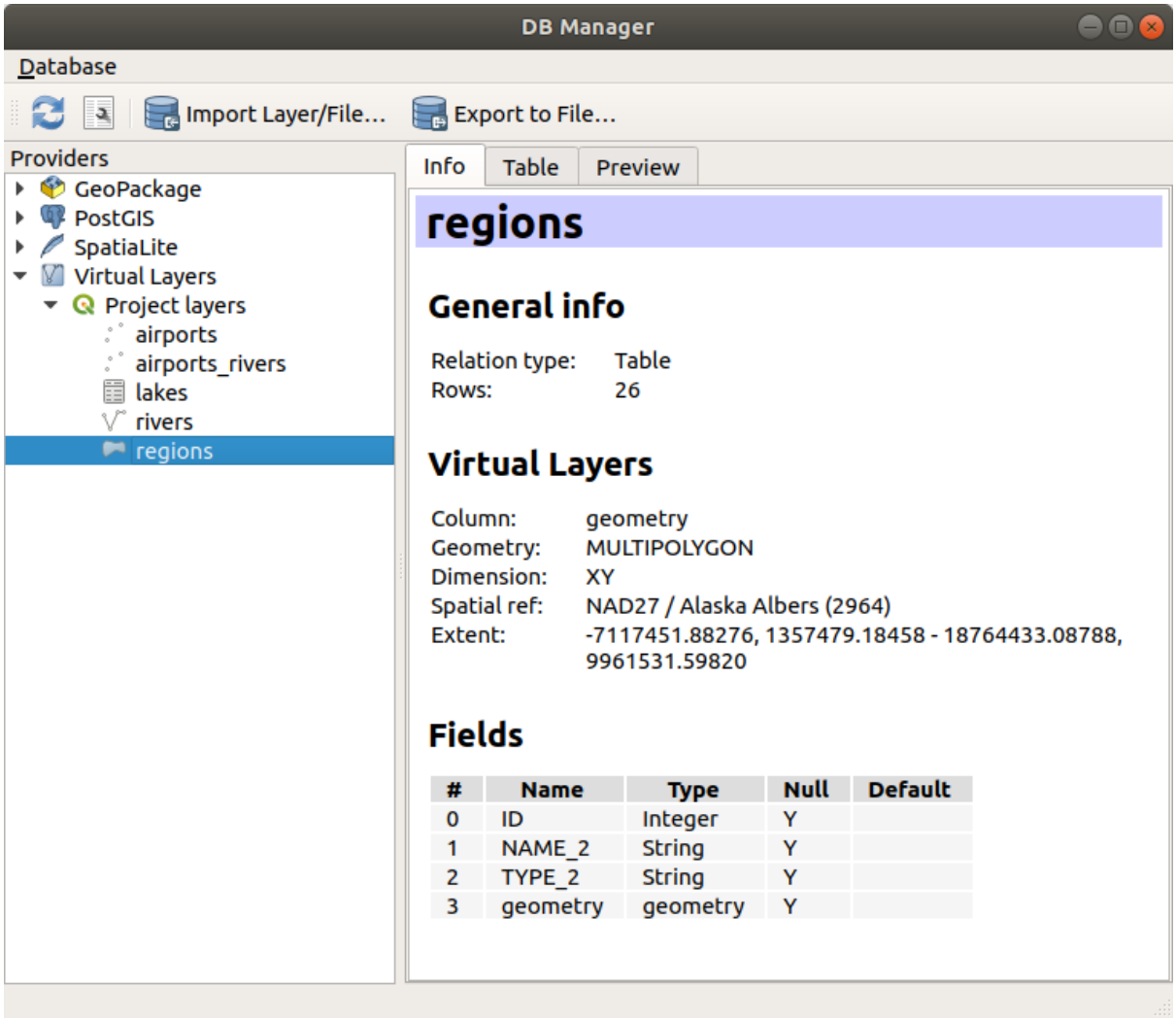


Fig. 25.4: Dialogrutan DB Manager

Menyn *Database* låter dig ansluta till en befintlig databas, starta SQL-fönstret och avsluta DB Manager Plugin. När du har anslutit till en befintlig databas visas menyerna *Schema* (relevant för DBMS, t.ex. PostGIS / PostgreSQL) och *Table*.

Menyn *Schema* innehåller verktyg för att skapa och ta bort (endast om de är tomma) scheman och, om topologi finns tillgänglig (t.ex. med PostGIS-topologi), för att starta en *TopoViewer*.

Med menyn *Table* kan du skapa och redigera tabeller samt ta bort tabeller och vyer. Det är också möjligt att tömma tabeller och att flytta tabeller mellan scheman. Du kan *Köra Vacuum Analyze* för den valda tabellen. *Vacuum* återtar utrymme och gör det tillgängligt för återanvändning, och *analyze* uppdaterar statistik som används för att avgöra det mest effektiva sättet att utföra en fråga. *Change Logging…* låter dig lägga till stöd för ändringsloggning i en tabell. Slutligen kan du använda *Import Layer/File…* och *Export to File…*.

Observera

Med hjälp av DB Manager är det möjligt att lägga till kommentarer för tabeller och kolumner i en PostgreSQL-databas.

I fönstret *Datakällor* listas alla befintliga databaser som stöds av QGIS. Med ett dubbelklick kan du ansluta till databasen. Med höger musknapp kan du byta namn på och ta bort befintliga scheman och tabeller. Tabeller kan också läggas till i QGIS-rutan med hjälp av snabbmenyn.

Om du är ansluten till en databas finns det fyra flikar i fönstret **huvud** i DB Manager. Fliken *Info* innehåller information om tabellen och dess geometri, samt om befintliga fält, begränsningar och index. Här kan du skapa ett spatialt index på den valda tabellen. På fliken *Table* visas tabellen och på fliken *Preview* visas geometrierna som förhandsgranskning. När du öppnar ett *SQL Window*, kommer det att placeras i en ny flik.

Arbeta med SQL-fönstret

Du kan använda DB Manager för att köra SQL-frågor mot din spatiala databas. Frågor kan sparas och laddas, och där hjälper *SQL Query Builder* dig att formulera dina frågor. Du kan även visa spatiala utdata genom att markera *Load as new layer* och ange *Column(s) with unique values* (IDs), *Geometry column* och *Layer name (prefix)*. Det är möjligt att markera en del av SQL för att bara köra den delen när man trycker på `Ctrl+R` eller klickar på *Execute*-knappen.

När du har kört din fråga kan du välja specifika celler i resultatuppsättningen. Använd kortkommandot `Ctrl+C` för att kopiera de markerade cellerna till urklipp. De kopierade uppgifterna är tillgängliga som en formaterad tabell. Detta gör att du kan klistra in data i andra program, t.ex. kalkylblad, där de visas som en tabell.

Knappen *Query History* lagrar de senaste 20 förfrågningarna för varje databas och leverantör.

Dubbelklicka på en post för att lägga till strängen i SQL-fönstret.

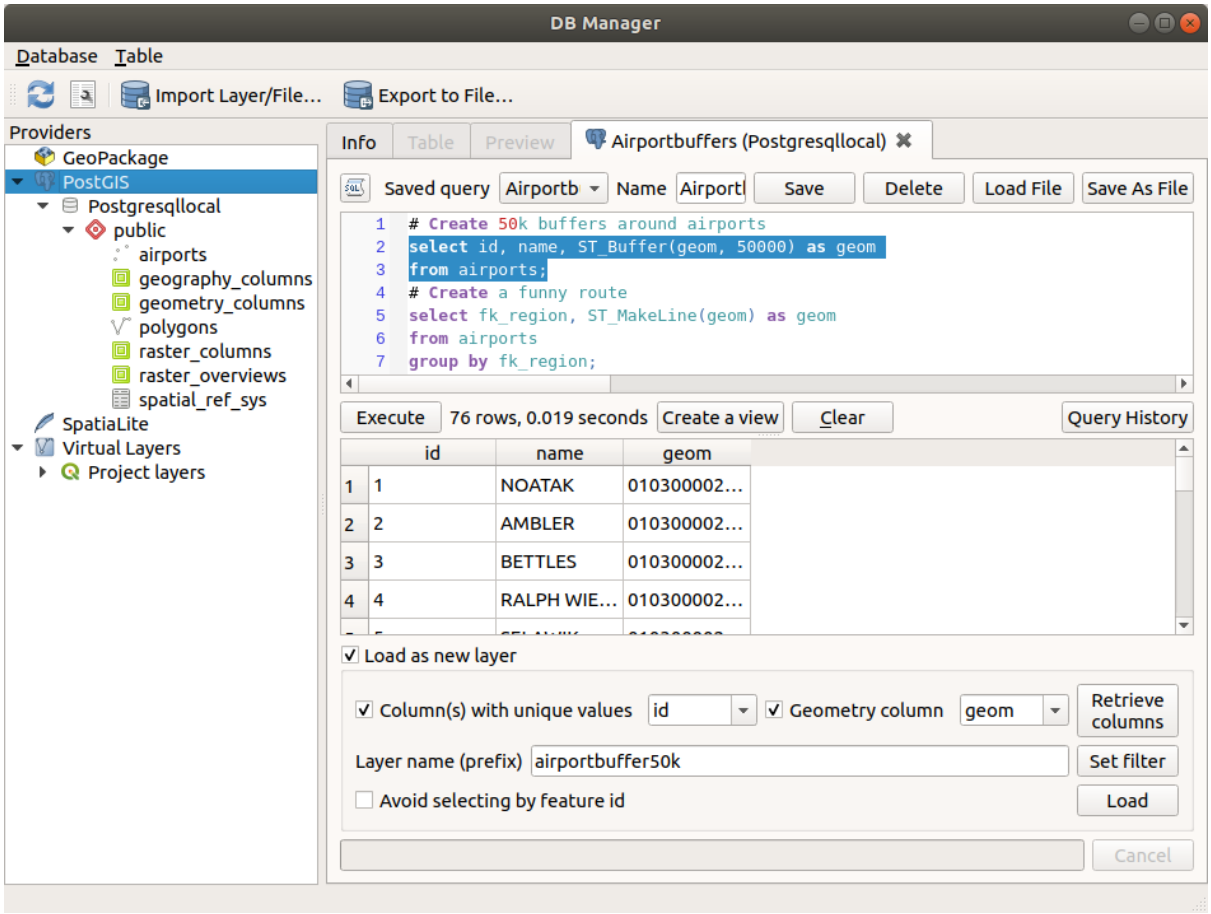


Fig. 25.5: Exekvera SQL-frågor i SQL-fönstret i DB Manager

Observera

SQL-fönstret kan också användas för att skapa virtuella lager. I så fall väljer du **QGIS Layers** under **Virtual Layers** innan du öppnar SQL-fönstret, istället för att välja en databas. Se *Skapa virtuella lager* för instruktioner om vilken SQL-syntax som ska användas.

25.2.2 Tillägget Geometrikontroll

Geometry Checker är ett kraftfullt kärntillägg för att kontrollera och fixa geometrigiltigheten för ett lager. Det är tillgängligt från menyn *Vektor* (*Kontrollera geometrier…*).

Konfigurera kontrollerna

I dialogrutan *Check Geometries* visas olika grupperade inställningar i den första fliken (*Setup*):

- *Input vector layers*: för att välja de lager som ska kontrolleras. En kryssruta *Only selected features* kan användas för att begränsa kontrollen till geometrierna för de valda funktionerna.
- *Tillåtna geometrityper* ger möjlighet att begränsa geometritypen för inmatningslagret/lagren till:
  - Punkt
  - Multipoint
  - Linje
  - Flerradig
  - Polygon
  - Multipolygon
- *Geometrins giltighet*. Beroende på geometrityp kan du välja mellan:
  - *Självskärningspunkter*
  - *Duplicera noder*
  - *Självkontakter*
  - *Polygon med mindre än 3 noder*.
- *Geometriegenskaper*. Beroende på geometrityp finns olika alternativ tillgängliga:
  - *Polygoner och multipolygoner får inte innehålla några hål*
  - *Multipartsobjekt måste bestå av mer än en del*
  - *Linjer får inte ha danglar*
- *Geometrivillkor*. Här kan du lägga till ett villkor som geometrierna ska valideras med:
  - *Minimal segmentlängd (kartans enheter)*
  - *Minsta vinkel mellan segmenten (grad)*
  - *Minimal polygonområde (kartenheter sqr.)*
  - *Inga sliverpolygoner* med en *Maximal tunnhet* och en *Max. area (kartenheter sqr.)*
- *Topologikontroller*. Beroende på geometrityp finns många olika alternativ tillgängliga:
  - *Kontrollerar för dubbletter*
  - *Kontrollerar för funktioner inom andra funktioner*

  - *Kontrollerar överlappningar som är mindre än* 1,00
  - *Kontrollerar för luckor som är mindre än* 1,00
  - *Punkter måste täckas av linjer*
  - *Punkter måste ligga korrekt inuti en polygon*
  - *Linjer får inte korsa några andra linjer*
  - *Linjer får inte korsa egenskaper i lagret*
  - *Polygonerna måste följa lagrets gränser*
- *Tolerans*. Du kan definiera toleransen för kontrollen i kartlagerenheter.
- *Output vector layer* ger möjlighet att välja mellan:
  - *Modifiera inmatningslagret*
  - *Skapa nya lager*

När du är nöjd med konfigurationen kan du klicka på knappen *Run*.

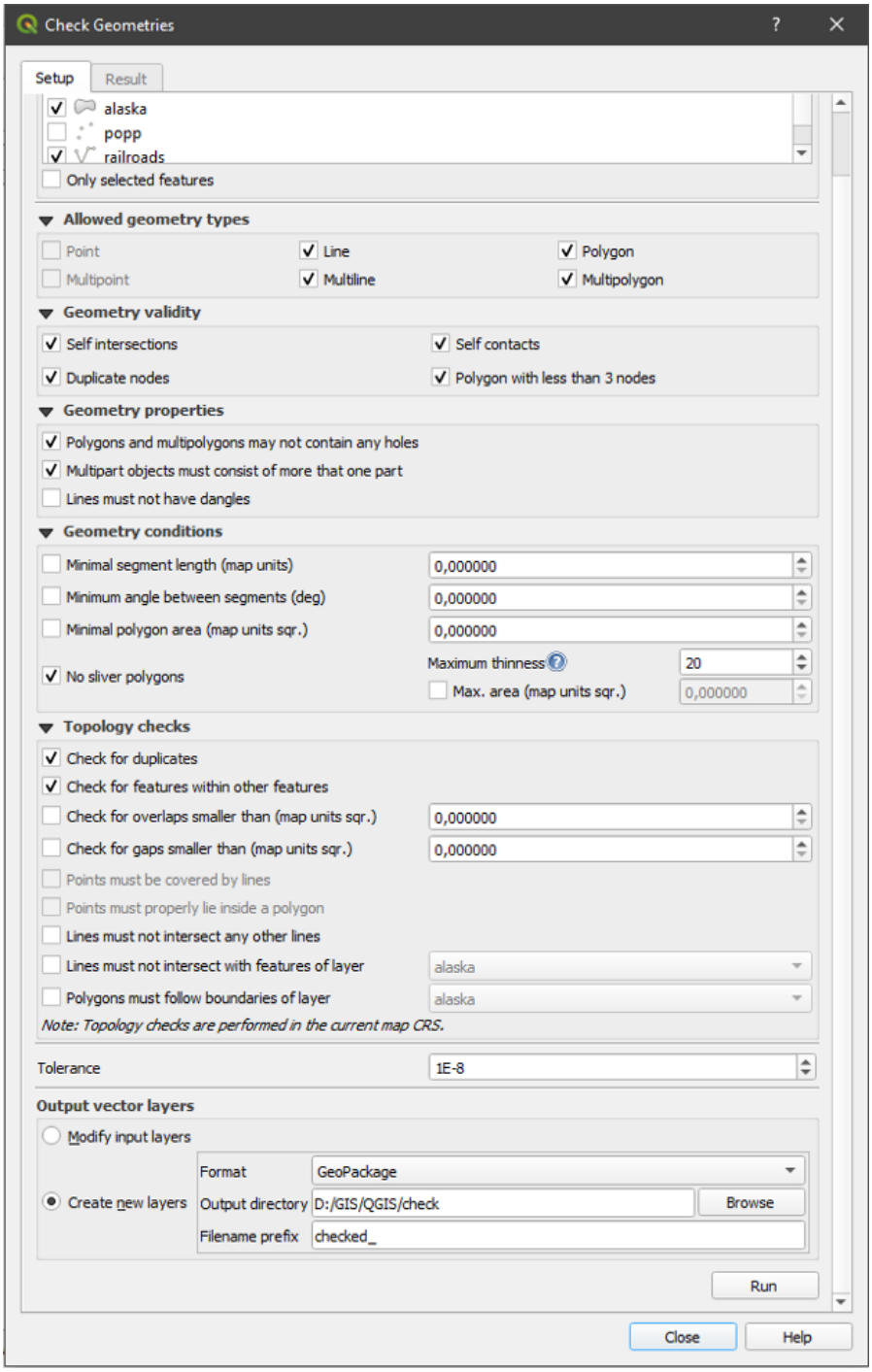


Fig. 25.6: Tillägget Geometrikontroll

Tillägget *Geometry Checker* kan hitta följande fel:

- Self intersections: en polygon med en self intersection
- Duplicerade noder: två duplicerade noder i ett segment
- Hål: hål i en polygon
- Segmentlängd: en segmentlängd som är lägre än ett tröskelvärde
- Minsta vinkel: två segment med en vinkel som är lägre än ett tröskelvärde
- Minimiyta: polygonområde lägre än ett tröskelvärde

- Silverpolygon: detta fel kommer från en mycket liten polygon (med liten yta) med stor omkrets
- Duplicerar funktioner
- Funktion inom funktion
- Överlappningar: överlappning av polygoner
- Gaps: mellanrum mellan polygoner

Följande bild visar de olika kontrollerna som görs av tillägget.

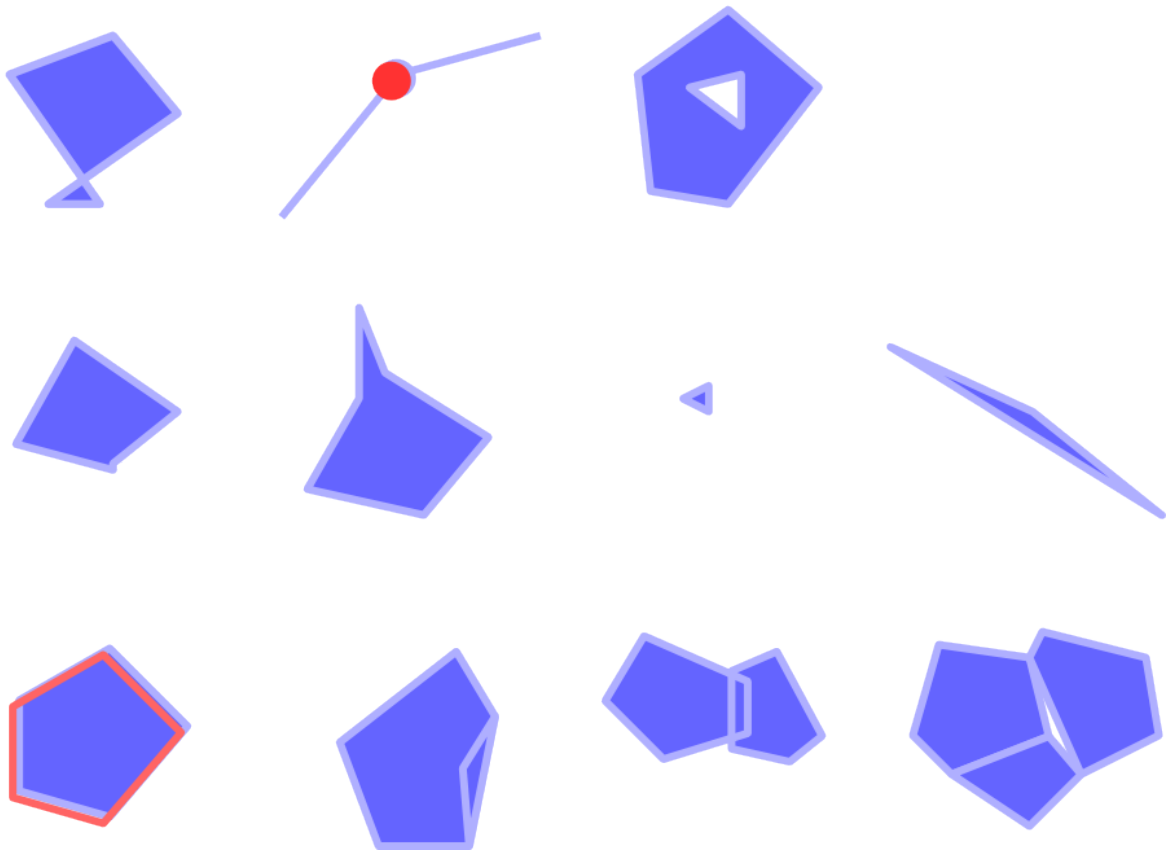


Fig. 25.7: Några kontroller som stöds av tillägget

Analys av resultaten

Resultaten visas på den andra fliken (*Result*) och som ett översiktslager över felen i Canvas (dess namn har standardprefixet `checked_`). En tabell visar *Geometry check result* med ett fel per rad och kolumner som innehåller: lagrets namn, ett ID, feltypen, sedan koordinaterna för felet, ett värde (beroende på feltyp) och slutligen kolumnen resolution som anger upplösningen av felet. Längst ner i tabellen kan du *Exportera* felet till olika filformat. Du har också en räknare som visar antalet totala fel och antalet åtgärdade fel.

Du kan välja en rad för att se var felet finns. Du kan ändra detta beteende genom att välja en annan åtgärd mellan *Error* (standard), *Feature*, *Don't move* och *Highlight selected features*.

Under zoomåtgärden när du klickar på tabellraden kan du:

- *Visa utvalda funktioner i attributtabellen*
- *Fixa utvalda fel med standardlösning*

- *Fixa valda fel, fråga efter lösningsmetod* Du kommer att se ett fönster för att välja lösningsmetod bland vilka:
  - Sammanfoga med grannpolygon med längsta gemensamma kant
  - Slå samman med grannpolygonen med störst yta
  - Sammanfoga med angränsande polygon med identiskt attributvärde, om något, eller lämna som det är
  - Ta bort funktion
  - Ingen åtgärd
- *Inställningar för felsökning* gör att du kan ändra standardmetoden för felsökning beroende på feltyp

Tips

Rätta flera fel

Du kan åtgärda flera fel genom att markera mer än en rad i tabellen med åtgärden *CTRL + klicka*.

Slutligen kan du välja vilket *Attribut som ska användas när funktioner slås samman efter attributvärde*.

25.2.3 MetaSearch katalogklient

Introduktion

MetaSearch är ett QGIS-tillägg för att interagera med metadatakatalogtjänster, som stöder både OGC API - Records och OGC Catalog Service for the Web (CSW) standarder.

MetaSearch ger en enkel och intuitiv metod och ett användarvänligt gränssnitt för att söka i metadatakataloger inom QGIS.

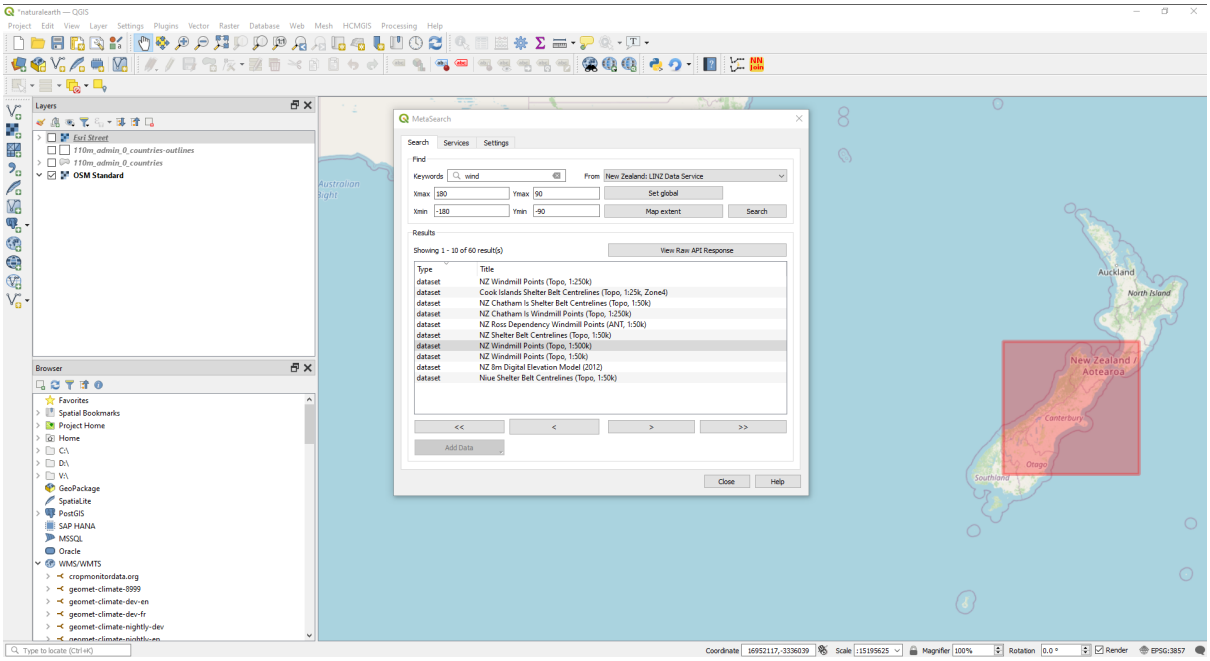


Fig. 25.8: Sökning och resultat av tjänster i MetaSearch

Arbeta med metadatakataloger i QGIS

MetaSearch ingår som standard i QGIS, med alla dess beroenden, och kan aktiveras från QGIS tilläggshanterare.

OGC API - Register

OGC API - Records är en standard från OGC (Open Geospatial Consortium) för att hitta geospatiala resurser på webben.

CSW (katalogtjänst för webben)

CSW (Catalog Service for the Web) är en specifikation från OGC (Open Geospatial Consortium) som definierar gemensamma gränssnitt för att upptäcka, bläddra i och fråga efter metadata om data, tjänster och andra potentiella resurser.

Uppstart

MetaSearch startas genom att klicka på ikonen eller genom att välja *Webb* ► *MetaSearch* ► *MetaSearch* via QGIS huvudmeny. Dialogrutan MetaSearch kommer att visas. Det grafiska huvudgränssnittet består av tre flikar: *Tjänster*, *Sök* och *Inställningar*.

Hantera katalogtjänster

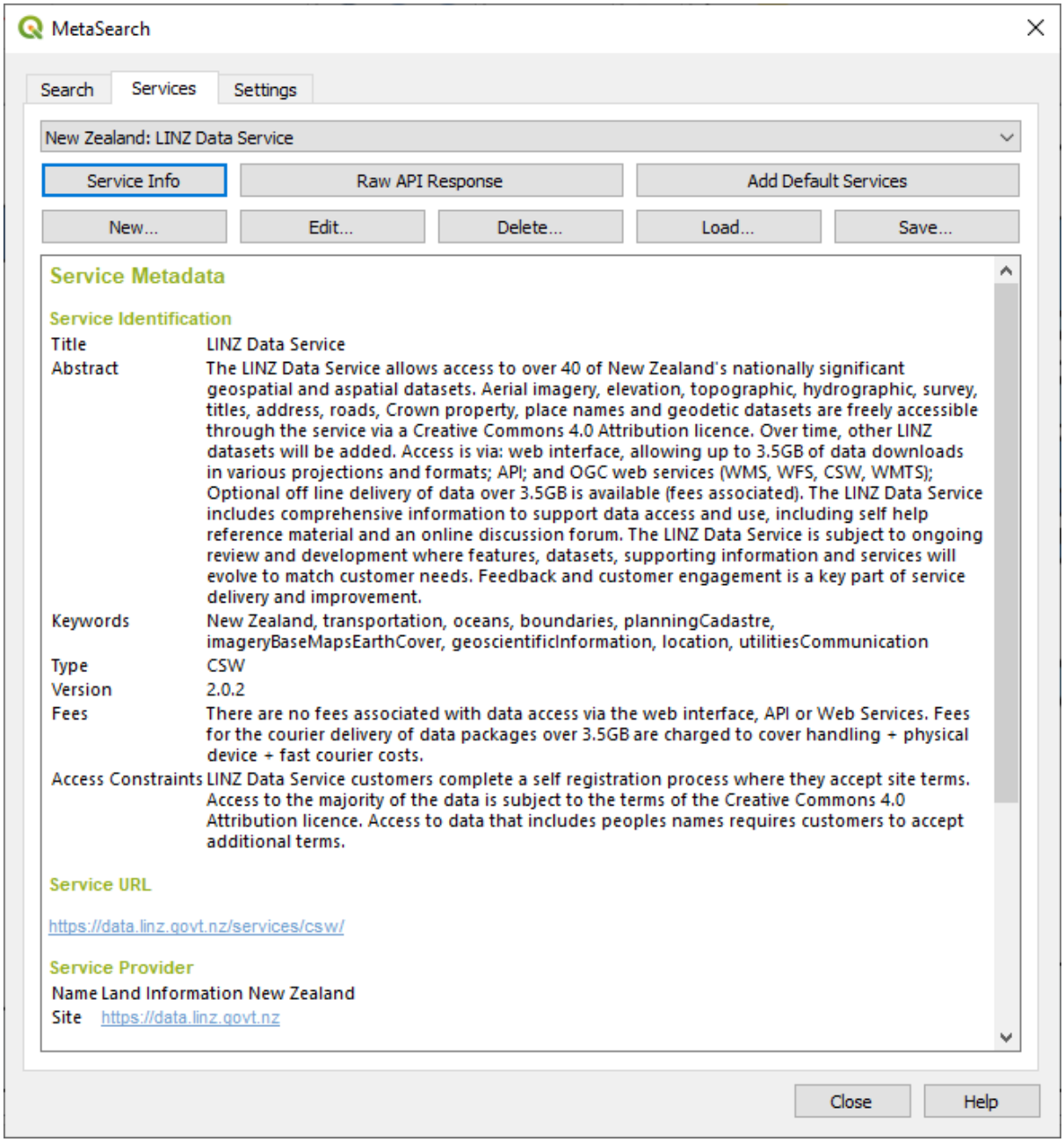


Fig. 25.9: Hantera katalogtjänster

På fliken *Services* kan användaren hantera alla tillgängliga katalogtjänster. MetaSearch tillhandahåller en standardlista med katalogtjänster, som kan läggas till genom att trycka på knappen *Add Default Services*.

Om du vill hitta alla listade katalogtjänstposter klickar du på rullgardinsmenyn.

Så här lägger du till en katalogtjänstpost:

1. Klicka på knappen *New* (ny)
2. Ange ett *Name* för tjänsten, samt *URL* (endpoint). Observera att för OGC CSW 2.0.2 Catalogs krävs endast bas-URL:en (inte en fullständig GetCapabilities-URL). För OGC API - Records Catalogs ska URL:en vara sökvägen till samlingens slutpunkt
3. Om katalogen kräver autentisering anger du lämpliga uppgifter för *Användarnamn* och *Lösenord*.

4. Klicka på *OK* för att lägga till tjänsten i listan över poster.

Så här redigerar du en befintlig katalogtjänstpost:

1. Välj den post som du vill redigera
2. Klicka på *Edit*-knappen
3. Och ändra värdena *Name* eller *URL*
4. Klicka på *OK*.

För att radera en katalogtjänstpost, välj den post du vill radera och klicka på knappen *Delete*. Du kommer att bli ombedd att bekräfta borttagningen av posten.

MetaSearch gör det möjligt att ladda och spara anslutningar till en XML-fil. Detta är användbart när du behöver dela inställningar mellan applikationer. Nedan visas ett exempel på XML-filformatet.

```
<?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>
```

För att ladda en lista med poster:

1. Klicka på knappen *Load*. Ett nytt fönster kommer att visas.
2. Klicka på knappen *Bläddra* och navigera till den XML-fil med poster som du vill ladda.
3. Klicka på *Öppna*. Listan över poster kommer att visas.
4. Välj de poster som du vill lägga till från listan och klicka på *Load*.

Klicka på knappen *Service Info* för att visa information om den valda katalogtjänsten, t.ex. tjänsteidentifiering, tjänsteleverantör och kontaktinformation. Om du vill visa det råa API-svaret klickar du på knappen *Raw API Response*. Ett separat fönster öppnas och visar serverinformation i rå JSON- eller XML-format.

Sökning i katalogtjänster

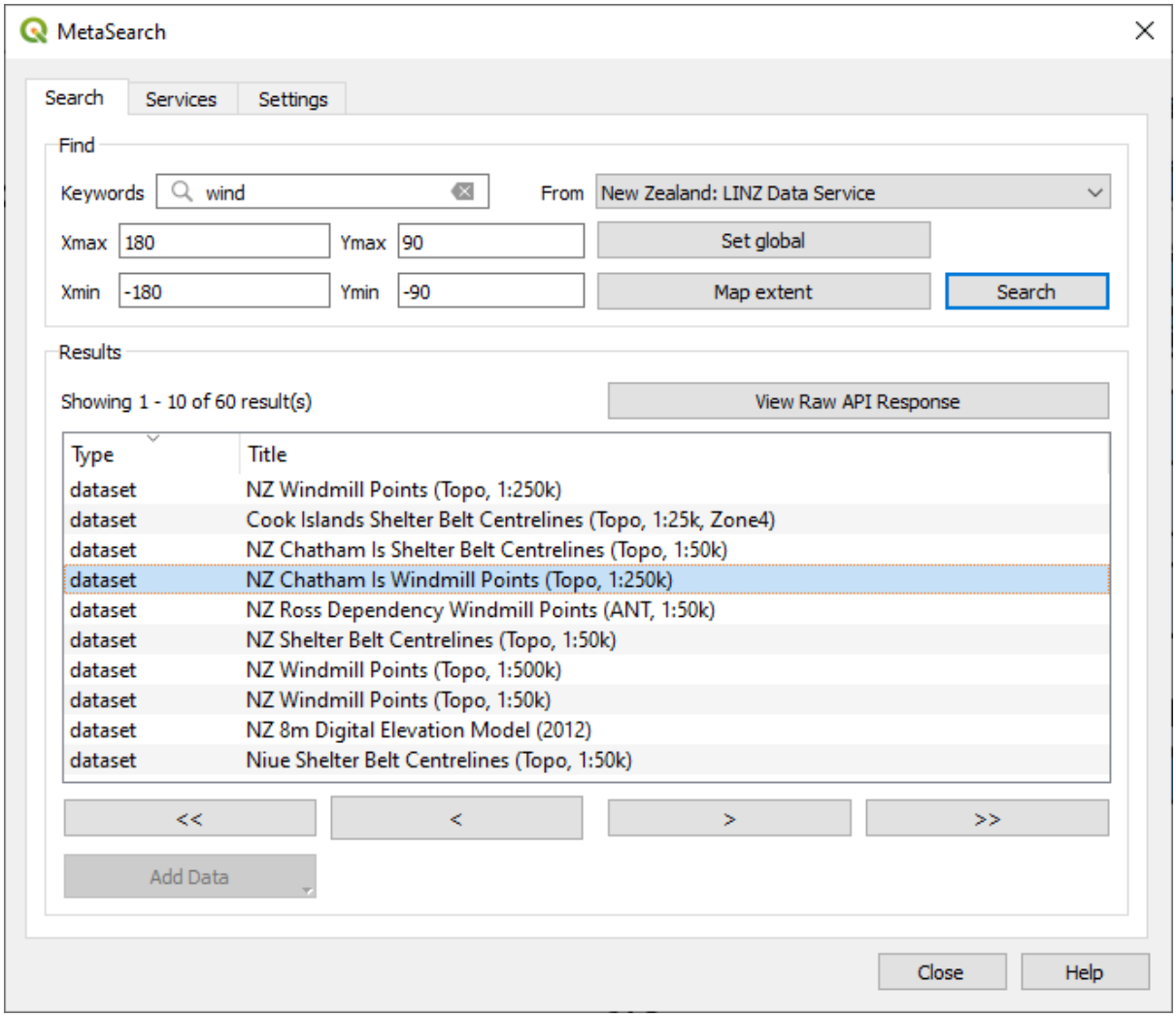


Fig. 25.10: Söka katalogtjänster

Fliken *Search* ger användaren möjlighet att söka efter data och tjänster i Catalog Services, ställa in olika sökparametrar och visa resultat.

Följande sökparametrar är tillgängliga:

- *Nyckelord*: nyckelord för fritextsökning;
- *From*: den katalogtjänst som frågan ska ställas mot;
- **Avgränsningsruta**: det spatiala område som är intressant att filtrera, definierat av *Xmax*, *Xmin*, *Ymax* och *Ymin*. Klicka på *Set Global* för att göra en global sökning, klicka på *Map Extent* för att göra en sökning i det synliga området, eller ange värden manuellt.

Om du klickar på knappen *Search* görs en sökning i den valda metadatakatalogen. Sökresultaten visas i en lista och kan sorteras genom att klicka på kolumnrubriken. Du kan navigera genom sökresultaten med hjälp av riktningsknapparna under sökresultaten.

Välj ett resultat och:

- Klicka på knappen *View Raw API Response* för att öppna ett fönster med tjänstens svar i rå JSON- eller XML-format.
- Om metadataposten har en tillhörande avgränsningsbox visas ett fotavtryck av avgränsningsboxen på kartan.

- Dubbelklicka på posten för att visa postens metadata med eventuella tillhörande åtkomstlänkar. Om du klickar på en länk öppnas länken i användarens webbläsare.
- Om posten är en webbtjänst som stöds (WMS/WMTS, WFS, WCS, ArcGIS REST Service, etc.) eller en GIS-fil, kommer knappen *Add Data* att aktiveras. När du klickar på den här knappen kommer MetaSearch att verifiera om detta är en giltig OWS. Tjänsten kommer sedan att läggas till i lämplig QGIS-anslutningslista och lämplig anslutningsdialog visas. Om du väljer *Add GIS File* kommer den länkade GIS-filen att läggas till i det aktuella projektet.

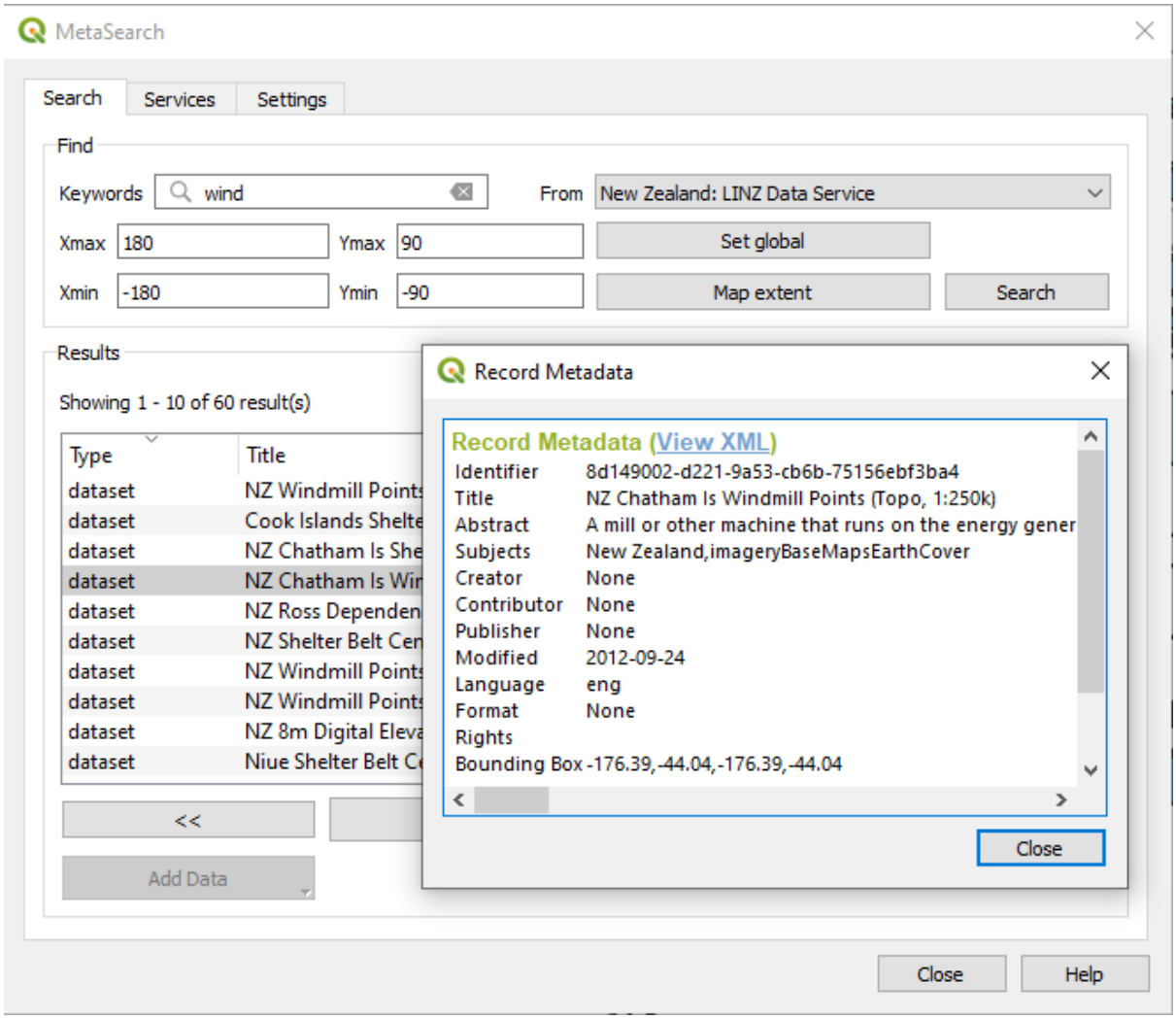


Fig. 25.11: Visning av metadataposter

Inställningar

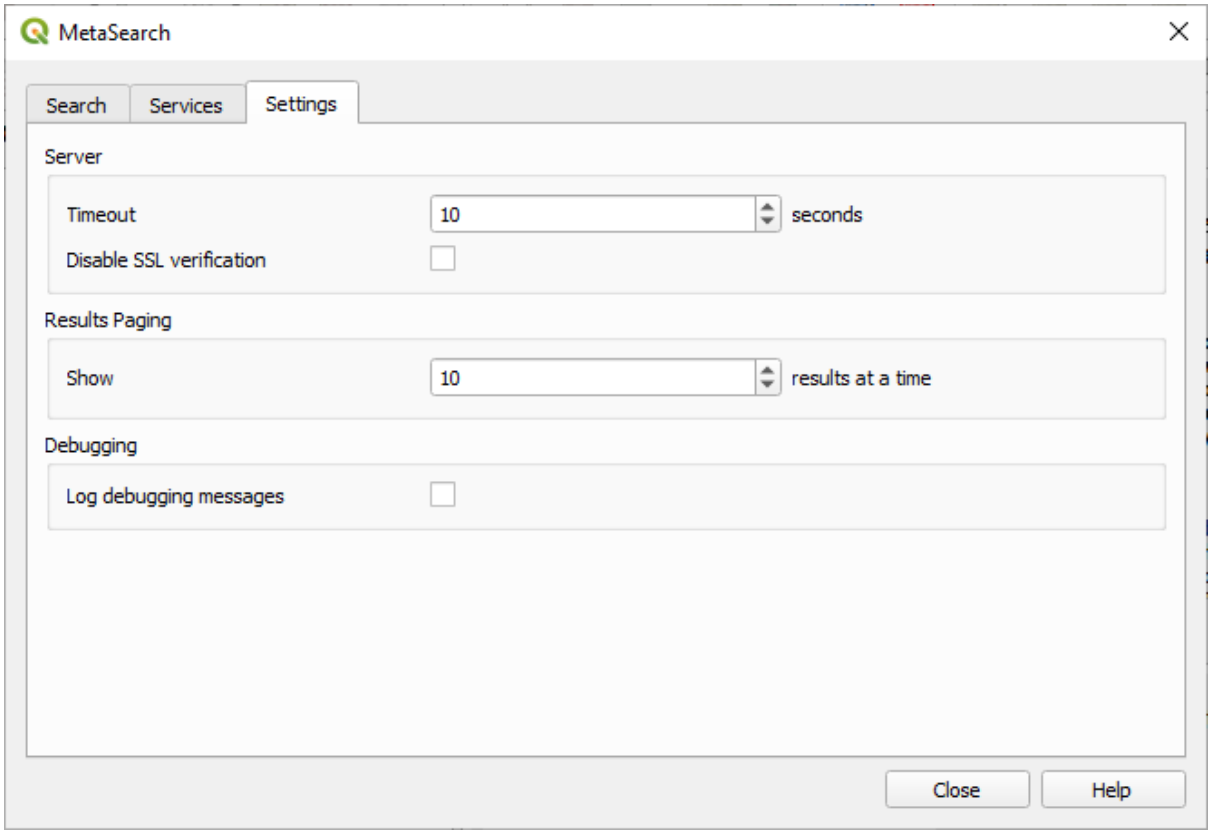


Fig. 25.12: MetaSearch-inställningar

Du kan finjustera MetaSearch med följande *Settings*:

- *Server Timeout*: vid sökning i metadatakataloger, antal sekunder för blockering av anslutningsförsök. Standardvärdet är 10.
- *Disable SSL verification*: alternativ för att stänga av ssl-verifiering.
- *Results paging*: vid sökning i metadatakataloger, antalet resultat som ska visas per sida. Standardvärdet är 10.
- *Logga felsökningsmeddelanden*: alternativ för att aktivera loggning av felsökningsmeddelanden i *Panel för loggmeddelanden*.

Fel i katalogservern

I vissa fall fungerar katalogen i en webbläsare, men inte i MetaSearch. Detta kan bero på katalogserverns konfiguration / inställning. Katalogserverleverantörer bör se till att webbadresserna är konsekventa och uppdaterade i sin konfiguration (detta är vanligt i HTTP -> HTTPS-omdirigeringsscenarier). Se pycsw FAQ-artikeln för en djupare förklaring av problemet och lösningen. Även om FAQ-artikeln är pycsw-specifik kan den också tillämpas generellt på andra katalogservrar.

25.2.4 Tillägg för offline-redigering

Vid datainsamling är det vanligt att man arbetar med en bärbar dator eller en mobiltelefon offline ute på fältet. När man återvänder till nätverket måste ändringarna synkroniseras med huvuddatakällan (t.ex. en PostGIS-databas). Om flera personer arbetar samtidigt med samma dataset är det svårt att sammanfoga ändringarna för hand, även om personerna inte ändrar samma funktioner.

Tillägget :sup:`Offline Editing` automatiserar synkroniseringen genom att kopiera innehållet i datakällan till en SpatiaLite- eller GeoPackage-databas och lagra offline-redigeringarna i särskilda tabeller. Efter att ha anslutits till nätverket igen är det möjligt att tillämpa offline-redigeringarna på masterdatasetet.

Så här använder du tillägget:

1. Öppna ett projekt med några vektorlager (t.ex. från en Esri Shapefile-, PostGIS- eller WFS-T-datakälla).
2. Förutsatt att du redan har aktiverat tillägget (se *Kärn- och externa tillägg*) gå till *Database* ► *Offline Editing* ► *Convert to offline project*. Dialogrutan för eponym öppnas.
3. Välj *Storage type* (lagringstyp). Den kan vara av databastyp *GeoPackage* eller *SpatiaLite*.
4. Använd knappen *Browse* för att ange platsen för den databas där *Offline data* ska lagras. Det kan vara en befintlig fil eller en som ska skapas.
5. I avsnittet *Välj fjärrlager* markerar du de lager som du vill spara. Innehållet i lagren sparas i databastabeller.

   Observera

   Eftersom måldatabasformaten inte har inbyggt stöd för listor omvandlar tillägget för offline-redigering listfälten {string, number} till strängfält där värdena separeras med kommatecken. Detta möjliggör läsning och redigering av innehållet i dessa fält när du är offline.

   Om du vill hantera både fältet från det ursprungliga lagret och offline-lagret kan du förlita dig på uttrycksfunktionerna *try()* och *array*, t.ex:

   ```
   try(array_contains("field",1),array_contains(string_to_array("field"),1))
   ```

6. Du kan markera kryssrutan *Synkronisera endast valda funktioner om ett urval finns* så att du bara kan spara och arbeta med en delmängd. Det kan vara ovärderligt när det gäller stora lager.

   Det är allt!

7. Spara ditt projekt och ta med det ut på fältet.
8. Redigera lagren offline.
9. När du är ansluten igen laddar du upp ändringarna med *Database* ► *Offline Editing* ► *Synchronize*.

Observera

Lager som används offline är markerade med ikonen i panelen *Layers*.

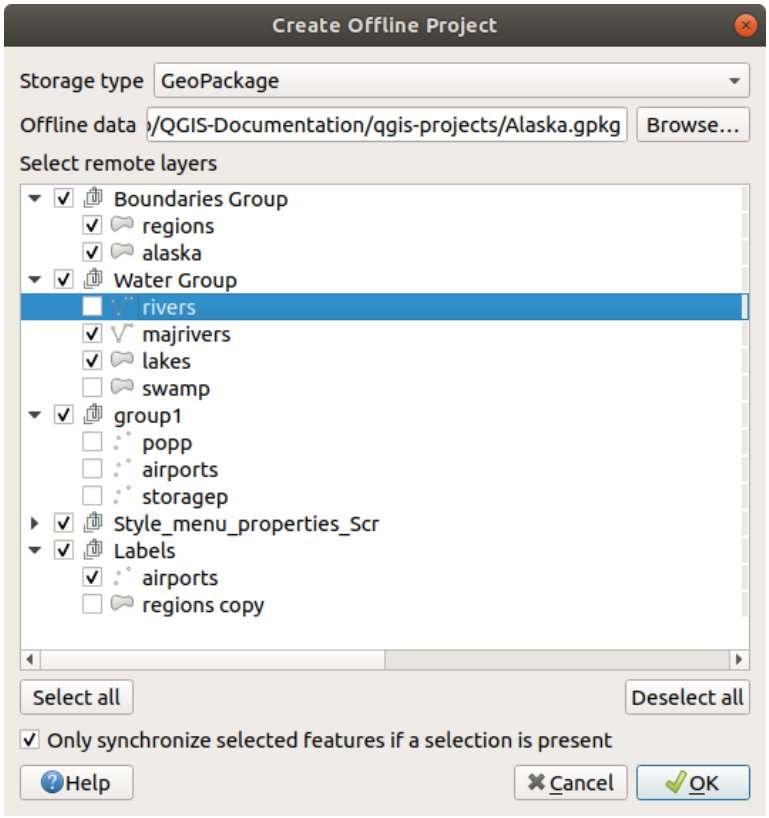


Fig. 25.13: Skapa ett offlineprojekt

25.2.5 Tillägget Topologikontroll

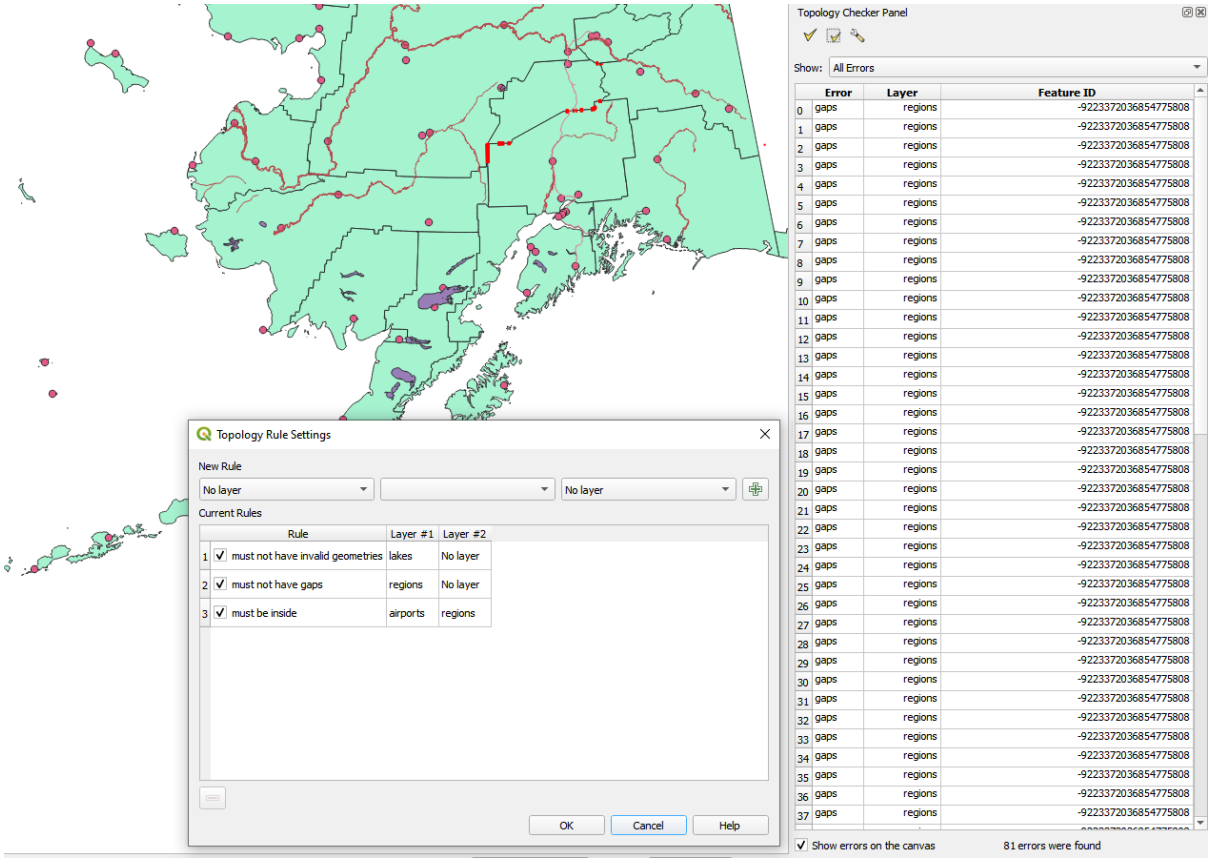


Fig. 25.14: Tillägget Topologikontroll

Topologi beskriver relationerna mellan punkter, linjer och polygoner som representerar funktionerna i en geografisk region. Med tillägget Topology Checker kan du titta på dina vektorfiler och kontrollera topologin med flera topologiregler. Dessa regler kontrollerar med spatiala relationer om dina funktioner ”är lika med”, ”innehåller”, ”täcker”, är ”täckta av”, ”korsar”, är ”disjunkta”, ”skär varandra”, ”överlappar”, ”rör vid” eller är ”inom” varandra. Det beror på dina individuella frågor vilka topologiregler du tillämpar på dina vektordata (t.ex. accepterar du normalt inte överskjutningar i linjelager, men om de visar återvändsgränder tar du inte bort dem från ditt vektorlager).

QGIS har en inbyggd funktion för topologisk redigering, som är utmärkt för att skapa nya funktioner utan fel. Men befintliga datafel och fel som orsakas av användaren är svåra att hitta. Detta tillägg hjälper dig att hitta sådana fel genom en lista med regler.

För att aktivera tillägget *Topology checker*:

1. Gå till *Tillägg*-menyn
2. Öppna *Hantera och installera tillägg* och välj *Installed*
3. Aktivera *Topologicheckare*
4. Stäng dialogrutan *Tilläggshanterare*. En post med namnet *Topologikontroll* läggs till i menyn *Vektor*.

När du har aktiverat *Topology checker* öppnar du den och väljer *Configure* för att skapa dina topologiregler.

För **punktlager** finns följande regler tillgängliga:

- *Måste täckas av*: Här kan du välja ett vektorlager från ditt projekt. Punkter som inte täcks av det givna vektorlagret visas i fältet ”Error”.

- *Måste täckas av ändpunkterna på*: Här kan du välja ett linjelager från ditt projekt.
- *Måste vara inuti*: Här kan du välja ett polygonlager från ditt projekt. Punkterna måste ligga inuti en polygon. Annars skriver QGIS ett ”Error” för punkten.
- *Måste inte ha dubbletter*: När en punkt representeras två eller flera gånger kommer den att visas i fältet ”Error”.
- *Måste inte ha ogiltiga geometrier*: Kontrollerar om geometrierna är giltiga.
- *Måste inte ha flerdelade geometrier*: Alla flerdelade punkter skrivs in i fältet ”Error”.

För **linjelager** finns följande regler tillgängliga:

- *Slutpunkter måste täckas av*: Här kan du välja ett punktlager från ditt projekt.
- *Får inte ha danglar*: Detta kommer att visa överskjutningarna i linjelagret.
- *Måste inte ha dubbletter*: När en linjefunktion finns representerad två eller flera gånger kommer den att visas i fältet ”Error”.
- *Måste inte ha ogiltiga geometrier*: Kontrollerar om geometrierna är giltiga.
- *Måste inte ha geometrier med flera delar*: Ibland är en geometri i själva verket en samling enkla (endelade) geometrier. En sådan geometri kallas flerdelad geometri. Om den bara innehåller en typ av enkel geometri kallar vi den multi-point, multi-linestring eller multi-polygon. Alla flerdelade linjer skrivs in i fältet ”Error”.
- *Måste inte ha pseudos*: En linjegeometris ändpunkt ska vara kopplad till ändpunkterna för två andra geometrier. Om ändpunkten är ansluten till endast en annan geometris ändpunkt kallas ändpunkten för en pseudonod.

På **polygonlager** finns följande regler tillgängliga:

- *Måste innehålla*: Polygonskiktet måste innehålla minst en punktgeometri från det andra lagret.
- *Får inte ha duplikat*: Polygoner från samma lager får inte ha identiska geometrier. När en polygonfunktion representeras två eller flera gånger kommer den att visas i fältet ”Error”.
- *Måste inte ha luckor*: Angränsande polygoner bör inte ha några luckor mellan sig. Administrativa gränser kan nämnas som ett exempel (polygoner i amerikanska delstater har inga luckor mellan sig…).
- *Måste inte ha ogiltiga geometrier*: Kontrollerar om geometrierna är giltiga. Några av de regler som definierar en giltig geometri är:
  - Polygonringar måste stängas.
  - Ringar som definierar hål bör vara inuti ringar som definierar yttre gränser.
  - Ringar får inte korsa varandra (de får varken röra vid eller korsa varandra).
  - Ringarna får inte röra vid andra ringar, utom vid en punkt.
- *Måste inte ha flerdelade geometrier*: Ibland är en geometri i själva verket en samling enkla (endelade) geometrier. En sådan geometri kallas flerdelad geometri. Om den bara innehåller en typ av enkel geometri kallar vi den multi-point, multi-linestring eller multi-polygon. Till exempel kan ett land som består av flera öar representeras som en multipolygon.
- *Måste inte överlappa varandra*: Angränsande polygoner får inte dela gemensam yta.
- *Måste inte överlappa med*: Angränsande polygoner från ett lager får inte dela gemensam yta med polygoner från ett annat lager.

När du skapar en *Ny regel* klickar du på Add rule för att lägga till den i *Current rules*. Du kan aktivera eller inaktivera enskilda regler genom att klicka på kryssrutan. Om du högerklickar på en regel får du följande alternativ:

- *Välj alla* reglerna
- *Aktivera* eller *Avaktivera* de valda reglerna
- *Toggle aktivering* av valda regler
- *Avlägsna* valda regler. Detta kan också göras med knappen Delete selected rules.

Tryck på *OK* och välj sedan från panelen *Topology checker*:

- Validate All: tillämpar de aktiva reglerna på alla funktioner i det eller de berörda lagren
- eller Validate Extent: tillämpar de aktiva reglerna på funktionerna i det eller de berörda lagren, inom den aktuella kartbilden. Knappen hålls intryckt och resultaten uppdateras i takt med att kartans omfattning ändras.

Fel kommer att visas i resultattabellen som innehåller typ av fel, lager och funktions-ID. Använd menyn *Filtrera fel efter regel* för att filtrera felen till en specifik feltyp.

Markera *Visa fel på kartbilden* för att visa felets plats på kartbilden. Om du klickar på en rad i tabellen zoomas kartbilden till den berörda funktionen, där du kan använda *QGIS digitaliseringsverktyg* för att åtgärda felet.

Nedan finns en lista över Core-tillägg som tillhandahålls med QGIS. De är inte nödvändigtvis aktiverade som standard.

Ikon	Tillägg	Beskrivning	Manuell referens
	DB Manager	Hantera dina databaser inom QGIS	*DB Manager Plugin*
	Geometri-kontroll	Kontrollera och reparera fel i vektorgeometrier	*Tillägget Geometrikontroll*
	GRASS 7	GRASS-funktionalitet	*GRASS GIS-integration*
	GRASS GIS-leverantör	GRASS GIS Bearbetningsfunktionalitet	*GRASS GIS-integration*
	MetaSearch katalogklient	Interagera med katalogtjänster för metadata (CSW)	*MetaSearch katalogklient*
	Offline-redigering	Offline-redigering och synkronisering med databas	*Tillägg för offline-redigering*
	Bearbetas	Ramverk för bearbetning av spatiala data	*Ramverk för bearbetning av QGIS*
	Topologikontroll	Hitta topologiska fel i vektorlager	*Tillägget Topologikontroll*

Observera

För att använda Core Tillägg GRASS 7, GRASS GIS provider, måste de konfigureras. Information finns *här*.

25.3 QGIS Python-konsol

Som du kommer att se senare i detta kapitel har QGIS utformats med en tilläggsarkitektur. Tillägg kan skrivas i Python, ett mycket känt språk i den geospatiala världen.

QGIS har ett Python API (se PyQGIS Developer Cookbook för några kodexempel) för att låta användaren interagera med dess objekt (lager, funktion eller gränssnitt). QGIS har också en Python-konsol.

QGIS Python Console är ett interaktivt skal för exekvering av Python-kommandon. Den har också en Python-filredigerare som gör att du kan redigera och spara dina Python-skript. Både konsolen och redigeraren är baserade på PyQScintilla2-paketet. För att öppna konsolen går du till *Tillägg* ► *Python-konsol* (`Ctrl+Alt+P`).

25.3.1 Den interaktiva konsolen

Konsolen är en Python-tolk som gör att du kan köra Python-kommandon. Moduler från QGIS (analysis, core, gui, server, processing, 3d) och Qt (QtCore, QtGui, QtNetwork, QtWidgets, QtXml) samt Pythons moduler math, os, re och sys är redan importerade och kan användas direkt.

Den interaktiva konsolen består av ett verktygsfält, en inmatningsyta och en utmatningsyta.

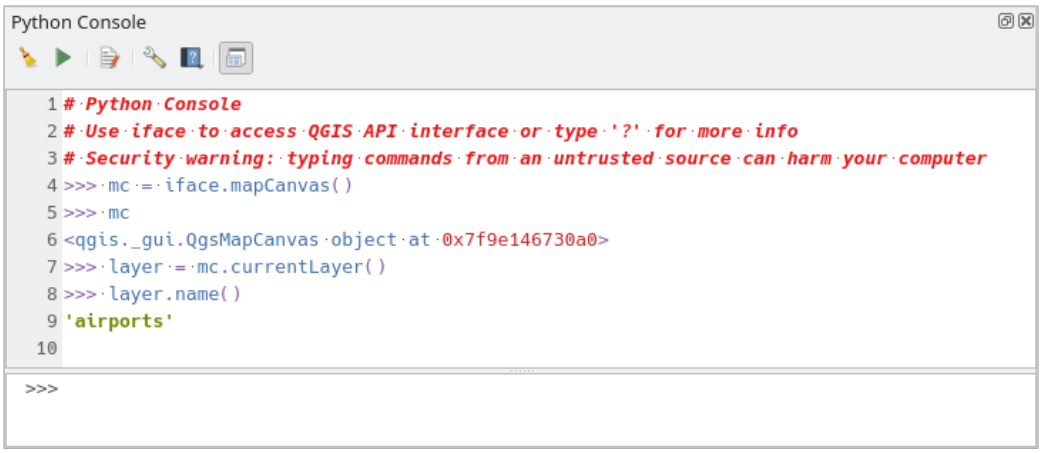


Fig. 25.15: Python-konsolen

Verktygsrad

I verktygsfältet finns följande verktyg:

- Clear Console för att rensa utmatningsområdet;
- Kör kommando tillgängligt i inmatningsområdet: samma sak som att trycka på `Enter`;
- Visa redigerare: växlar synlighet för *Kodredigeraren*;
- Alternativ…: öppnar en dialogruta för att konfigurera *konsolegenskaper*;
- Hjälp… innehåller en meny för att komma åt olika dokument:
  - *Python Console Help* (den aktuella sidan)
  - PyQGIS API-dokumentation
  - PyQGIS Cookbook
- Dock Code Editor för att docka eller undocka panelen i QGIS-gränssnittet

Inmatningsområde

De viktigaste funktionerna i konsolens inmatningsområde är

- Kodkomplettering, syntaxmarkering och anropstips för följande API:er:
  - Python
  - PyQGIS
  - PyQt5
  - QScintilla2
  - osgeo-gdal-ogr

- `Ctrl+Alt+Space` för att visa listan för automatisk komplettering om den är aktiverad i *Python-inställningar*;
- Kör kodsnuttar från inmatningsområdet genom att skriva och trycka på `Enter` eller *Run Command*;
- Exekvera kodavsnitt från utmatningsområdet med hjälp av *Enter Selected* från snabbmenyn eller genom att trycka på `Ctrl+E`;
- Bläddra igenom kommandohistoriken från inmatningsområdet med hjälp av piltangenterna `Upp` och `Ned` och kör det kommando du vill ha;
- `Ctrl+Shift+Space` för att visa kommandohistoriken: dubbelklicka på en rad för att utföra kommandot. Dialogrutan *Kommandohistorik* kan också nås från kontextmenyn i inmatningsområdet;
- Spara och rensa kommandohistoriken. Historiken sparas i filen `console_history.txt` under den aktiva mappen *användarprofil*;
- Skriv in följande specialkommandon:
  - `?` för att visa en hjälp av Python-konsolen
  - `_api` för att öppna QGIS C++ API dokumentation eller `_api(object)` för en specifik objektdokumentation (i QGIS C++ API eller Qt API dokumentation)
  - `_pyqgis` för att öppna QGIS Python API dokumentation eller `_pyqgis(object)` för en specifik objektdokumentation (i QGIS Python API eller Qt API dokumentation)
  - `_cookbook` to open PyQGIS Cookbook.
  - `!` följt av ett kommando för att köra Shell-kommandon från Python-konsolen. Konsolen startar en underprocess och vidarebefordrar dess utdata till Python Console Output. Medan subprocessen körs växlar Python Console Input till STDIN-läge och vidarebefordrar inmatade tecken till barnprocessen. Detta gör det möjligt att skicka en bekräftelse när barnprogrammet ber om det. När konsolen är i STDIN-läge kan man trycka på `Ctrl+C` för att döda underprocessen. Det är också möjligt att påverka resultatet av ett kommando till en variabel med syntaxen `var = !cmd`

```
>>> !echo QGIS Rocks!
QGIS Rocks

>>> !gdalinfo --version
GDAL 3.6.2, released 2023/01/02

>>> !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']
```

Tips

Återanvända utförda kommandon från utmatningspanelen

Du kan köra kodsnuttar från utdatapanelen genom att markera text och trycka på `Ctrl+E`. Det spelar ingen roll om den markerade texten innehåller tolkens prompt (`>>>`, `...`).

25.3.2 Kodredigeraren

Använd knappen Show Editor för att aktivera redigeringswidgeten. Den gör det möjligt att redigera och spara Python-filer och erbjuder avancerade funktioner för att hantera din kod (kommentera och avkommentera kod, kontrollera syntax, dela koden via GitHub och mycket mer). Huvudfunktionerna är:

- Kodkomplettering, syntaxmarkering och anropstips för följande API:er:
  - Python
  - PyQGIS
  - PyQt5
  - QScintilla2
  - osgeo-gdal-ogr
- `Ctrl+Space` för att se den automatiska kompletteringslistan.
- Dela kodsnuttar via *GitHub*.
- `Ctrl+4` Syntaxkontroll.
- Sökfält (öppna det med standardgenvägen i skrivbordsmiljön, vanligtvis `Ctrl+F`):
  - Använd standardgenvägen i skrivbordsmiljön för att hitta nästa/förra (`Ctrl+G` och `Shift+Ctrl+G`);
  - Hitta automatiskt den första matchningen när du skriver i sökrutan;
  - Ställ in initial söksträng till val när du öppnar sökningen;
  - Tryck på `Esc` för att stänga sökfältet.
- Objektinspektör: en klass- och funktionsbläddrare;
- Gå till en objektdefinition med ett musklick (från Object inspector);
- Kör kodavsnitt med kommandot *Run Selected* i den kontextuella menyn;
- Kör hela skriptet med kommandot *Run Script* (detta skapar en bytekompilerad fil med tillägget `.pyc`).

Observera

Om du kör ett skript helt eller delvis från *Code Editor* visas resultatet i konsolens utmatningsområde.

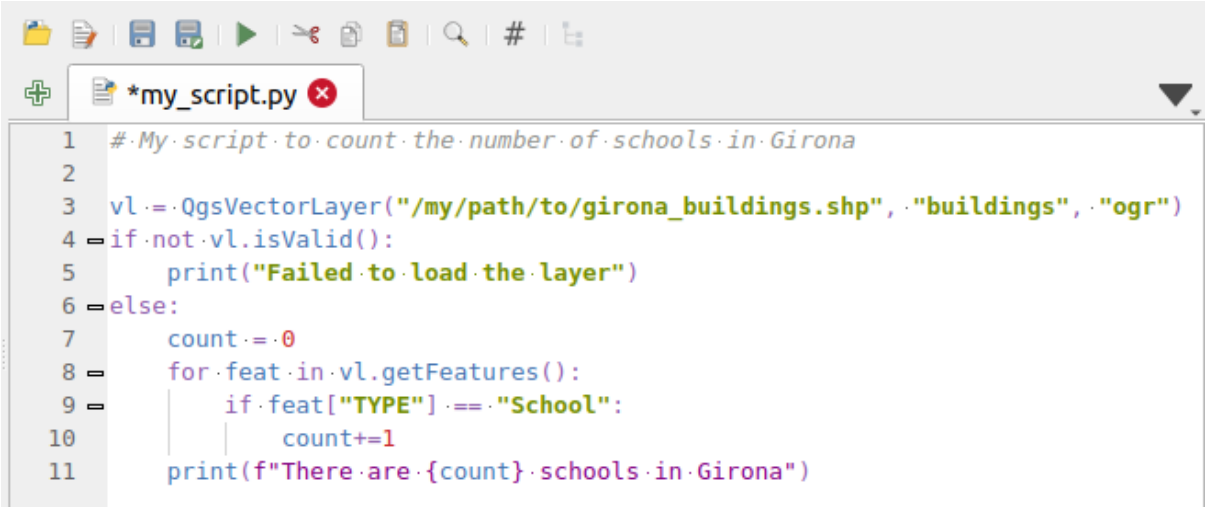


Fig. 25.16: Python Console-editorn

Tips

Spara på alternativen

För att spara tillståndet för konsolens widgets måste du stänga Python-konsolen från knappen Stäng. Detta gör att du kan spara geometrin så att den återställs vid nästa start.

KAPITEL 26

Bilagor

26.1 Bilaga C: QGIS-filformat

26.1.1 QGS/QGZ - QGIS-projektets filformat

Formatet **QGS** är ett XML-format för lagring av QGIS-projekt. Formatet **QGZ** är ett komprimerat (zip) arkiv som innehåller en QGS-fil och en QGD-fil. Filen **QGD** är den associerade sqlite-databasen för qgis-projektet som innehåller hjälpdata för projektet. Om det inte finns några hjälpdata kommer QGD-filen att vara tom.

En QGIS-fil innehåller allt som behövs för att lagra ett QGIS-projekt, inklusive:

- projektets titel
- projekt CRS
- lagerträdet
- inställningar för snäppning
- relationer
- kartans omfattning
- projektmodeller
- legend
- mapview-dockor (2D och 3D)
- lagren med länkar till de underliggande dataseten (datakällorna) och andra lageregenskaper, t.ex. utsträckning, SRS, sammanfogningar, stilar, rendering, blandningsläge, opacitet m.m.
- projektegenskaper

Figurerna nedan visar toppnivåtaggarna i en QGS-fil och den utökade taggen `ProjectLayers`.

Fig. 26.1: Taggarna på högsta nivå i en QGS-fil

```
−<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>
```


Fig. 26.2: Den expanderade ProjectLayers-taggen på högsta nivån för en QGS-fil

26.1.2 QLR - QGIS-skiktdefinitionsfilen

En skiktdefinitionsfil (QLR) är en XML-fil som innehåller en pekare till lagrets datakälla samt QGIS-stilinformation för lagret.

Användningsfallet för den här filen är enkelt: Att ha en enda fil för att öppna en datakälla och ta in all relaterad stilinformation. Med QLR-filer kan du också maskera den underliggande datakällan i en fil som är lätt att öppna.

Ett exempel på QLR-användning är för att öppna MS SQL-lager. I stället för att behöva gå till MS SQL-anslutningsdialogen, ansluta, välja, ladda och slutligen formatera, kan du helt enkelt lägga till en .qlr-fil som pekar på rätt MS SQL-lager med alla nödvändiga formateringar inkluderade.

I framtiden kan en .qlr-fil innehålla en referens till mer än ett lager.

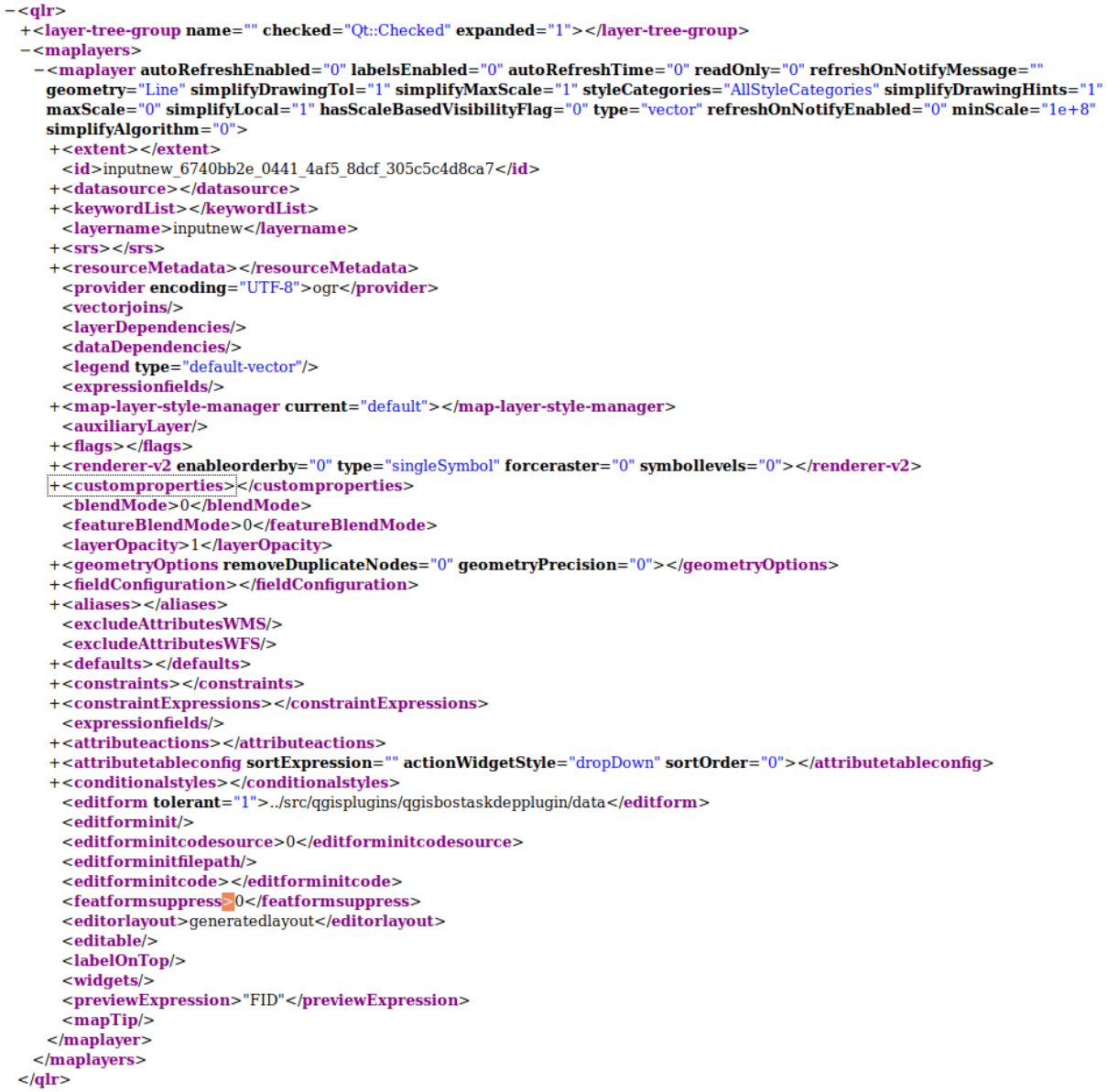


Fig. 26.3: De översta nivåtaggarna för en QLR-fil

26.1.3 QML - QGIS Style File Format

QML är ett XML-format för lagring av lagerstyling.

En QML-fil innehåller all information som QGIS kan hantera för rendering av objektgeometrier, inklusive symboldefinitioner, storlekar och rotationer, märkning, opacitet och blandningsläge med mera.

Bilden nedan visar toppnivåtaggarna i en QML-fil (med endast `renderer_v2` och dess `symbol` tagg expanderade).

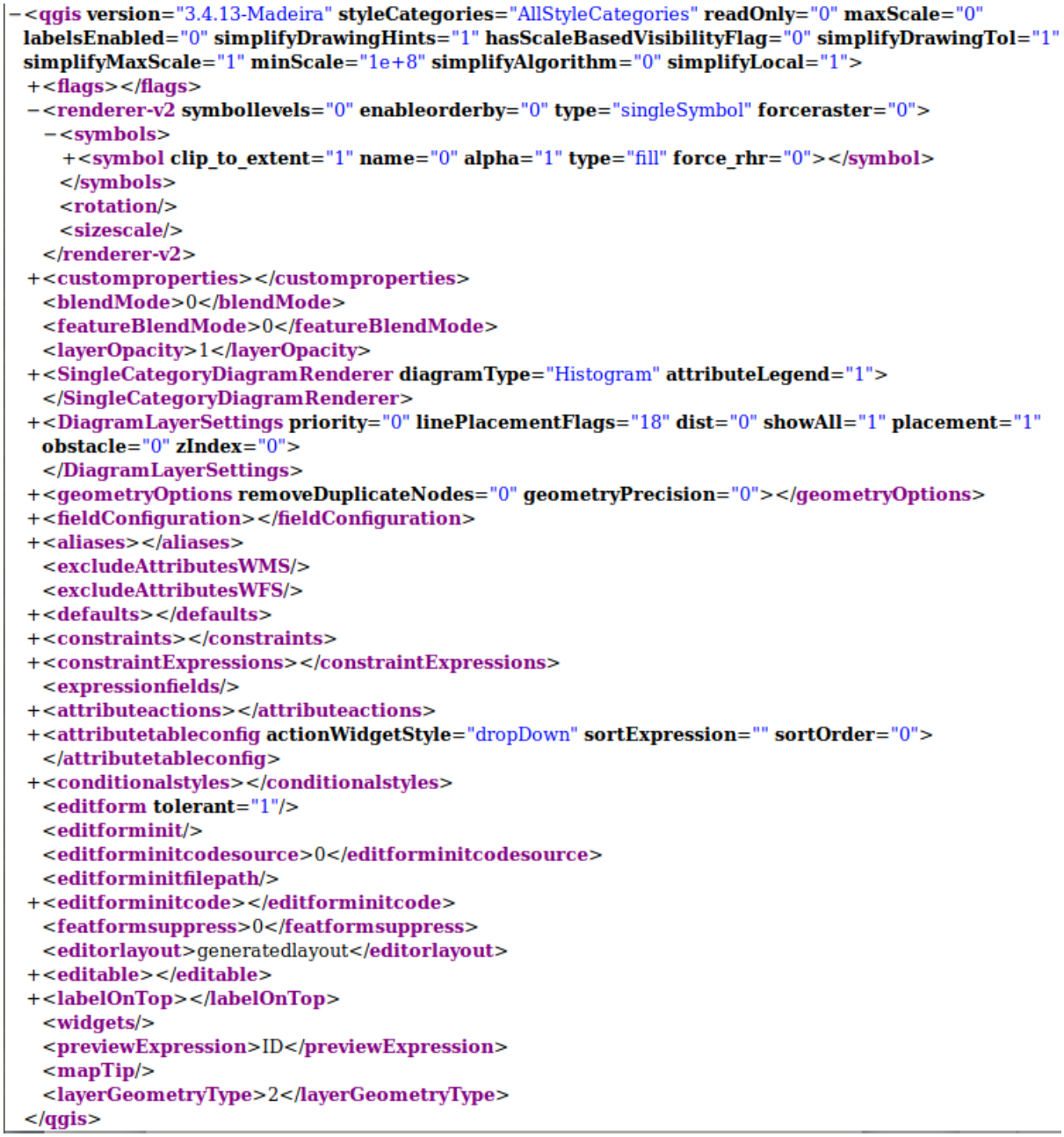


Fig. 26.4: Toppnivåtaggarna för en QML-fil (endast taggen renderer_v2 med dess symboltagg expanderas)

26.2 Bilaga D: Syntax för QGIS R-skript

Bidrag från Matteo Ghetta - finansierat av Scuola Superiore Sant'Anna

Att skriva R-skript i Processing är lite knepigt på grund av den speciella syntaxen.

Ett Processing R-skript börjar med att definiera dess **Inputs** och **Outputs**, som var och en föregås av dubbla hashtecken (##).

Före inmatningarna kan den grupp som algoritmen ska placeras i anges. Om gruppen redan finns kommer algoritmen att läggas till i den, om inte kommer gruppen att skapas. I exemplet nedan är gruppens namn *Min grupp*:

```
##Min grupp=grupp
```

26.2.1 Inmatningar

Alla indata och parametrar måste specificeras. Det finns flera olika typer av indata:

- vektor: `##Lager = vektor`
- vektorfält: `##F = fältlager` (där *lager* är namnet på ett ingångsvektorskikt som fältet tillhör)
- raster: `##r = raster`
- tabell: `##t = tabell`
- nummer: `##Num = nummer`
- sträng: `##Str = sträng`
- boolean: `##Bol = boolean`
- element i en rullgardinsmeny. Objekten måste separeras med semikolon `;`: `##type=valpunkt;linjer;punkt+linjer`

26.2.2 Utgångar

På samma sätt som för inmatningarna måste varje utmatning definieras i början av skriptet:

- vektor: `##output= utmatningsvektor`
- raster: `##output= utmatning raster`
- tabell: `##output= utmatningstabell`
- plots: `##output_plots_to_html` (##showplots i tidigare versioner)
- Om du vill visa R-utdata i *Result Viewer* sätter du > framför det kommando vars utdata du vill visa.

26.2.3 Syntaxsammanfattning för QGIS R-skript

Det finns ett antal olika typer av parametrar för in- och utmatning.

Typer av inmatningsparametrar

Parameter	Exempel på syntax	Återlämnande av objekt
vektor	Lager = vektor	sf-objekt (eller SpatialDataFrame-objekt, om ##load_vector_using_rgdal anges)
vektorpunkt	Lager = vektorpunkt	sf-objekt (eller SpatialDataFrame-objekt, om ##load_vector_using_rgdal anges)
vektorlinje	Lager = vektorlinje	sf-objekt (eller SpatialDataFrame-objekt, om ##load_vector_using_rgdal anges)
vektorpolygon	Lager = vektorpolygon	sf-objekt (eller SpatialPolygonsDataFrame-objekt, om ##load_vector_using_rgdal används)
multipel vektor	Lager = flera vektorer	sf-objekt (eller SpatialDataFrame-objekt om ##load_vector_using_rgdal har angetts)
tabell	Lager = tabell	dataframe-konvertering från csv, standardobjekt för funktionen `read.csv`
fält	Fält = fältlager	namnet på det valda fältet, t.ex. `"Area"`
raster	Lager = raster	RasterBrick-objekt, standardobjekt för `raster`-paketet
flera raster	Lager = flera raster	RasterBrick-objekt, standardobjekt i `raster`-paketet
nummer	N = antal	heltal eller flytande tal vald
sträng	S = sträng	sträng tillagd i rutan
långsträng	LS = långsträng	sträng som läggs till i lådan, kan vara längre än den normala strängen
urval	S = urval första;andra;tredje	sträng för det valda objektet som valts i rullgardinsmenyn
crs	C = crs	sträng av den resulterande CRS som valts, i formatet: `"EPSG:4326"`
omfattning	E = omfattning	Extent-objekt i paketet `raster`, du kan extrahera värden som `E@xmin`
poäng	P = punkt	när du klickar på kartan har du koordinaterna för punkten
fil	F = fil	sökvägen till den valda filen, t.ex. ”/home/matteo/file.txt”
folder	F = mapp	sökvägen till den valda mappen, t.ex. ”/home/matteo/Downloads”

En parameter kan vara **OPTIONAL**, vilket innebär att den kan ignoreras.

För att ange att en inmatning är valfri lägger du till strängen `optional` **före** inmatningen, t.ex:

```
##Layer = vector
##Field1 = Field Layer
##Field2 = optional Field Layer
```

Typer av utgångsparametrar

Parameter	Exempel på syntax
vektor	Utgång = utgångsvektor
raster	Output = utmatning av raster
tabell	Output = utmatningstabell
fil	Output = utdatafil

Observera

Du kan spara plottar som `png` från *Processing Result Viewer*, eller så kan du välja att spara plotten direkt från algoritmens gränssnitt.

Skriptets kropp

Skriptet följer R-syntaxen och panelen **Log** kan hjälpa dig om det är något fel på ditt skript.

Kom ihåg att du måste ladda alla ytterligare bibliotek i skriptet:

```
library(sp)
```

26.2.4 Exempel

Exempel med vektorutgång

Låt oss ta en algoritm från online-samlingen som skapar slumpmässiga punkter från omfattningen av ett inmatningslager:

```
##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)
```

Förklaring (per rad i skriptet):

1. `Punktmönsteranalys` är gruppen för algoritmen
2. `Lager` är det ingående **vektor**-lagret
3. `Size` är en **numerisk** parameter med ett standardvärde på 10
4. `Output` är det **vektor**-lager som kommer att skapas av algoritmen
5. `library(sp)`` laddar **sp**-biblioteket
6. `spatpoly = as(Layer, "Spatial")` översätt till ett sp-objekt
7. Anropa funktionen `spsample` i biblioteket `sp` och kör den med de indata som definierats ovan (`Layer` och `Size`)
8. Skapa ett *SpatialPointsDataFrame*-objekt med hjälp av funktionen `SpatialPointsDataFrame`
9. Skapa utmatningsvektorlagret med hjälp av funktionen `st_as_sf`

Så där ja! Kör bara algoritmen med ett vektorlager som du har i QGIS Legend, välj antalet slumpmässiga punkter. Det resulterande lagret kommer att läggas till på din karta.

Exempel med rasterutmatning

Följande skript utför grundläggande vanlig kriging för att skapa en rasterkarta med interpolerade värden från ett angivet fält i indatapunktens vektorlager med hjälp av funktionen `autoKrige` i R-paketet `automap`. Först beräknas krigingmodellen och sedan skapas ett raster. Rastret skapas med funktionen `raster` i R-paketet raster:

```
##Basic statistics=group
##Layer=vector point
##Field=Field Layer
##Output=output raster
##load_vector_using_rgdal
require("automap")
require("sp")
```

(continues on next page)

(fortsättning från föregående sida)

```
require("raster")
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
```

Genom att använda `##load_vector_using_rgdal` kommer ingångsvektorlagret att göras tillgängligt som ett `SpatialDataFrame`-objekt, så vi undviker att behöva översätta det från ett `sf`-objekt.

Exempel med tabellutdata

Låt oss redigera algoritmen `Summary Statistics` så att utdata blir en tabellfil (csv).

Skriptets huvuddel är följande:

```
##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
```

Den tredje raden anger **Vector Field** i indata och den fjärde raden talar om för algoritmen att utdata ska vara en tabell.

Den sista raden tar `Stat`-objektet som skapats i skriptet och konverterar det till en `csv`-tabell.

Exempel med konsolutmatning

Vi kan använda det tidigare exemplet och istället för att skapa en tabell skriva ut resultatet i **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:"))
```

(continues on next page)

(fortsättning från föregående sida)

```
colnames(Summary_statistics)<-c(Field)
>Summary_statistics
```

Manuset är exakt detsamma som det ovan, med undantag för två ändringar:

1. ingen utdata specificerad (den fjärde raden har tagits bort)
2. den sista raden börjar med >, vilket säger till Processing att göra objektet tillgängligt via resultatvisaren

Exempel med diagram

För att skapa plottar måste du använda parametern `##output_plots_to_html` som i följande skript:

```
##Basic statistics=group
##Layer=vector
##Field=Field Layer
##output_plots_to_html
####output_plots_to_html
qqnorm(Layer[[Field]])
qqline(Layer[[Field]])
```

Skriptet använder ett fält (`Field`) i ett vektorlager (`Layer`) som indata och skapar en *QQ Plot* (för att testa normaliteten i fördelningen).

Plotten läggs automatiskt till i Processing *Result Viewer*.

26.3 Bilaga E: Nätverksanslutningar för QGIS-applikationen

Detta är en lista över de förkonfigurerade/automatiserade nätverksanslutningar som QGIS gör. Vissa av anslutningarna är användarinitierade eftersom de kräver en åtgärd från användaren innan de sker, medan andra sker automatiskt.

Namn	Syfte	UX-läge	Server	Information som skickas till servern	Information som lagras på servern
qgis.org					
Hjälp med Python API	Browser PyQGIS-dokumentation	Användarinit	https://qgis.org/pyqgis/%1/search.html?q=%2	IP, QGIS-version, operativsystem	IP i serverlogg
version.c					
Kontroll av ny version	Meddela användaren om tillgängligheten av nya QGIS-versioner	Automatiskt	https://version.qgis.org	IP, QGIS-version, operativsystem	IP i serverlogg
feed.qgis					
QGIS-flöde	Hämta QGIS-nyheter från flödet	Automatiskt	https://feed.qgis.org	IP, QGIS-version, språkkod, tidsstämpel för senaste nedladdning, OS	IP i serverlogg; QGIS-version, operativsystem och IP aggregeras och används för att samla in viss statistik
plugins.c					
Sök efter tilläggsup	Meddela användaren om tilläggsuppdaterin	Användarinit (konfigurerba	https://plugins.qgis.org	IP, QGIS-version, operativsystem	IP i serverlogg
Lista över tillägg	Hämta listan över tillägg	Användarinit	https://plugins.qgis.org	IP, QGIS-version, operativsystem	IP i serverlogg
Installatic av tillägg	Ladda ner och installera ett tilläggspaket	Användarinit	https://plugins.qgis.org	IP, QGIS-version, operativsystem	Öka nedladdningsräknaren för tillägget med en
Stilar	Lista stilar som användaren bidragit med	Användarinit	https://hub.qgis.org/styles	IP, QGIS-version, operativsystem	Öka nedladdningsräknaren med en
3:e part					
Terrängd	Ta fram en DEM för 3D-vyer	Användarinit	https://s3.amazonaws.com/elevation-tiles-prod/terrarium/{z}/{x}/{	IP, QGIS-version, operativsystem	se Amazon TOS
Geokoda för Google-karta	Geokodningstjäns	Användarinit	https://maps.googleapis.com/maps/api/geocode/json	IP, QGIS-version, operativsystem	Se Google Maps API TOS
Nominati Geokoda	Geokodningstjäns	Användarinit	https://nominatim.qgis.org/ui/search.html	IP, QGIS-version, operativsystem	
Geodetisl rutnät	Lägg till ett nytt PROJ-rutnät	Användarinit	https://cdn.proj.org	IP, PROJ version	Åtkomstloggar raderas permanent efter en dag

KAPITEL 27

Litteratur- och webbreferenser

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.