QGIS Desktop 3.44 User Guide

QGIS Project

3 月 15, 2026

Contents

1 入门指南 **3**
- 1.1 安装 QGIS . . . 3
  - 1.1.1 二进制文件安装 . . . 3
  - 1.1.2 从源代码安装 . . . 3
  - 1.1.3 安装到外媒介 . . . 4
  - 1.1.4 下载示例数据 . . . 4
- 1.2 启动和关闭 QGIS . . . 4
- 1.3 案例：加载栅格图层和矢量图层 . . . 5

2 使用工程文件 **13**
- 2.1 QGIS 工程简介 . . . 13
- 2.2 处理损坏的文件路径 . . . 15
- 2.3 生成输出 . . . 16

3 QGIS 图形用户界面 **17**
- 3.1 菜单栏 . . . 18
  - 3.1.1 工程 . . . 18
  - 3.1.2 编辑 . . . 19
  - 3.1.3 视图 . . . 21
  - 3.1.4 图层 . . . 23
  - 3.1.5 设置 . . . 25
  - 3.1.6 插件 . . . 26
  - 3.1.7 矢量 . . . 26
  - 3.1.8 栅格 . . . 28
  - 3.1.9 数据库 . . . 29
  - 3.1.10 Web . . . 30
  - 3.1.11 网孔 . . . 30
  - 3.1.12 数据处理 . . . 30
  - 3.1.13 帮助 . . . 31
  - 3.1.14 QGIS . . . 31
- 3.2 面板和工具栏 . . . 31
  - 3.2.1 工具栏 . . . 31
  - 3.2.2 面板 . . . 32
- 3.3 状态栏 . . . 33
  - 3.3.1 搜索栏 . . . 33
  - 3.3.2 汇报行为 . . . 34
  - 3.3.3 控制地图画布 . . . 34
  - 3.3.4 实时信息 . . . 35

4 浏览器面板 **37**
- 4.1 从浏览器打开或运行资源 . . . 39

4.2 浏览器面板顶级条目 . . . 39
4.2.1 收藏夹 . . . 39
4.2.2 空间书签 . . . 40
4.2.3 工程主目录 . . . 40
4.2.4 驱动器和文件系统 . . . 40
4.2.5 数据类型 . . . 41
4.2.6 数据库条目 . . . 41
4.2.7 瓦片和 Web 服务 . . . 45
4.2.8 工程和支持的文件类型 . . . 46

5 QGIS 配置 **47**
5.1 选项 . . . 47
5.1.1 通用设置 . . . 48
5.1.2 系统设置 . . . 49
5.1.3 User Profiles Settings . . . 51
5.1.4 CRS and Transforms Settings . . . 51
5.1.5 Data Sources settings . . . 55
5.1.6 渲染设置 . . . 59
5.1.7 画布和图例设置 . . . 62
5.1.8 地图工具设置 . . . 64
5.1.9 Elevation settings . . . 68
5.1.10 3D settings . . . 68
5.1.11 Colors settings . . . 69
5.1.12 Fonts Settings . . . 70
5.1.13 Layouts settings . . . 71
5.1.14 Variables settings . . . 71
5.1.15 Authentication settings . . . 72
5.1.16 Network settings . . . 73
5.1.17 GPS settings . . . 75
5.1.18 Locator settings . . . 77
5.1.19 Acceleration settings . . . 79
5.1.20 IDE settings . . . 79
5.1.21 Processing settings . . . 83
5.1.22 Advanced settings . . . 83
5.2 使用用户配置文件 . . . 84
5.2.1 The concept . . . 84
5.2.2 Setting user profile . . . 85
5.3 工程属性 . . . 86
5.3.1 通用属性 . . . 86
5.3.2 元数据属性 . . . 89
5.3.3 View Settings . . . 89
5.3.4 CRS 属性 . . . 89
5.3.5 变换属性 . . . 90
5.3.6 Styles Properties . . . 90
5.3.7 Colors properties . . . 91
5.3.8 数据源属性 . . . 92
5.3.9 Relations Properties . . . 94
5.3.10 Variables Properties . . . 95
5.3.11 Macros Properties . . . 95
5.3.12 QGIS Server Properties . . . 96
5.3.13 Temporal Properties . . . 97
5.3.14 Elevation Properties . . . 97
5.3.15 Sensors Properties . . . 100
5.4 Customization . . . 101
5.5 Keyboard shortcuts . . . 103
5.6 Running QGIS with advanced settings . . . 104
5.6.1 Command line and environment variables . . . 104
5.6.2 Deploying QGIS within an organization . . . 110

6 使用投影 **113**
6.1 投影支持概述 . . . 113
6.2 图层坐标参照系 . . . 113
6.3 投影坐标参照系 . . . 115
6.4 坐标参照系选择器 . . . 116
6.5 自定义坐标参照系 . . . 117
6.5.1 在 QGIS 中集成 NTv2-变换 . . . 118
6.6 基准变换 . . . 119

7 地图可视化 **121**
7.1 2D 地图视图 . . . 121
7.1.1 探索地图视图 . . . 122
7.1.2 Controlling map rendering . . . 123
7.1.3 地图画布上基于时间的控件 . . . 124
7.1.4 The Elevation Controller panel . . . 126
7.1.5 Bookmarking extents on the map . . . 126
7.1.6 Decorating the map . . . 128
7.1.7 Annotation Tools . . . 136
7.1.8 Measuring . . . 141
7.1.9 设置其他地图视图 . . . 144
7.1.10 导出地图视图 . . . 145
7.2 3D 地图视图 . . . 147
7.2.1 场景配置 . . . 149
7.2.2 导航选项 . . . 156
7.2.3 创建动画 . . . 157
7.2.4 3D 矢量图层 . . . 157
7.3 Elevation Profile View . . . 157
7.3.1 The interface . . . 158
7.3.2 Creating an elevation profile . . . 160
7.3.3 Interacting with the profile Views . . . 161
7.3.4 Exporting from the elevation profile . . . 162

8 常用工具 **163**
8.1 上下文帮助 . . . 163
8.2 面板 . . . 163
8.2.1 图层面板 . . . 163
8.2.2 图层样式面板 . . . 170
8.2.3 图层顺序面板 . . . 173
8.2.4 概览图面板 . . . 173
8.2.5 日志消息面板 . . . 174
8.2.6 撤消/重做面板 . . . 174
8.2.7 统计摘要面板 . . . 174
8.2.8 调试/开发工具面板 . . . 175
8.3 从外部工程嵌入图层 . . . 179
8.4 与要素交互 . . . 180
8.4.1 选择要素 . . . 180
8.4.2 识别要素 . . . 183
8.5 保存和共享图层属性 . . . 187
8.5.1 管理自定义样式 . . . 187
8.5.2 在文件或数据库中存储样式 . . . 189
8.5.3 图层定义文件 . . . 191
8.6 编写您的数据 . . . 191
8.6.1 元数据 . . . 191
8.6.2 图层注解 . . . 192
8.7 将值存储在变量中 . . . 193
8.8 常用控件 . . . 195
8.8.1 颜色选择器 . . . 195
8.8.2 符号控件 . . . 199

8.8.3 远程或嵌入式文件选择器 . . . 199
8.8.4 可见比例选择器 . . . 200
8.8.5 空间范围选择器 . . . 200
8.8.6 Font Selector . . . 201
8.8.7 Unit Selector . . . 202
8.8.8 Number Formatting . . . 203
8.8.9 Blending Modes . . . 204
8.8.10 Data defined override setup . . . 206
8.8.11 认证 . . . 209

9 使用表达式提升 213
9.1 表达式 . . . 213
9.1.1 表达式字符串构建器 . . . 213
9.1.2 Function Editor . . . 218
9.2 函数列表 . . . 220
9.2.1 聚合函数 . . . 220
9.2.2 Array Functions . . . 231
9.2.3 Color Functions . . . 243
9.2.4 Conditional Functions . . . 252
9.2.5 Conversions Functions . . . 255
9.2.6 CRS Functions . . . 260
9.2.7 Custom Functions . . . 261
9.2.8 Date and Time Functions . . . 261
9.2.9 Fields and Values . . . 272
9.2.10 Files and Paths Functions . . . 272
9.2.11 Form Functions . . . 275
9.2.12 Fuzzy Matching Functions . . . 275
9.2.13 General Functions . . . 277
9.2.14 Geometry Functions . . . 279
9.2.15 Layout Functions . . . 353
9.2.16 Map Layers . . . 354
9.2.17 Maps Functions . . . 356
9.2.18 Mathematical Functions . . . 360
9.2.19 Meshes Functions . . . 368
9.2.20 Operators . . . 370
9.2.21 Processing Functions . . . 379
9.2.22 Rasters Functions . . . 379
9.2.23 Record and Attributes Functions . . . 381
9.2.24 Relations . . . 390
9.2.25 Sensors Functions . . . 390
9.2.26 String Functions . . . 390
9.2.27 User Expressions . . . 401
9.2.28 Variables . . . 401
9.2.29 Recent Functions . . . 405

10 样式库 407
10.1 样式管理器 . . . 407
10.1.1 样式管理器对话框 . . . 407
10.1.2 设置颜色渐变 . . . 412
10.1.3 创建图例图斑形状 . . . 414
10.2 符号选择器 . . . 416
10.2.1 符号图层树 . . . 417
10.2.2 配置符号 . . . 418
10.3 设置标注 . . . 435
10.3.1 格式化标注文本 . . . 437
10.3.2 配置与标注的交互 . . . 446
10.4 创建 3D 符号 . . . 455
10.4.1 点图层 . . . 456

10.4.2 线图层 . . . 457
10.4.3 多边形图层 . . . 458
10.4.4 晕渲纹理 . . . 460
10.4.5 应用示例 . . . 460

11 管理数据源 461
11.1 打开数据 . . . 461
11.1.1 浏览器面板 . . . 463
11.1.2 数据库管理器 . . . 466
11.1.3 基于提供程序加载工具 . . . 467
11.1.4 图层元数据搜索面板 . . . 487
11.1.5 QGIS 自定义格式 . . . 488
11.1.6 QLR - QGIS 图层定义文件 . . . 488
11.1.7 连接到 Web 服务 . . . 488
11.2 创建图层 . . . 496
11.2.1 新建矢量图层 . . . 497
11.2.2 从现有图层创建新图层 . . . 505
11.2.3 新建 DXF 文件 . . . 510
11.2.4 从剪贴板创建新图层 . . . 511
11.2.5 Creating SQL Query Layers . . . 511
11.2.6 创建虚拟图层 . . . 515
11.3 Georeferencer . . . 517
11.3.1 Georeferencing raster layer . . . 518
11.3.2 Georeferencing vector layer . . . 524
11.3.3 Show and adapt layer properties . . . 525
11.3.4 Configure the georeferencer . . . 525
11.4 探索数据格式与字段 . . . 526
11.4.1 栅格数据 . . . 526
11.4.2 矢量数据 . . . 526
11.4.3 图层横渡 180° 经度 . . . 534

12 使用矢量数据 537
12.1 矢量属性对话框 . . . 537
12.1.1 信息属性 . . . 538
12.1.2 源属性 . . . 538
12.1.3 Symbology Properties . . . 542
12.1.4 Labels Properties . . . 565
12.1.5 Diagrams Properties . . . 577
12.1.6 Masks Properties . . . 582
12.1.7 3D View Properties . . . 583
12.1.8 Fields Properties . . . 585
12.1.9 Attributes Form Properties . . . 586
12.1.10 Joins Properties . . . 598
12.1.11 Auxiliary Storage Properties . . . 598
12.1.12 Actions Properties . . . 606
12.1.13 Display Properties . . . 611
12.1.14 Rendering Properties . . . 614
12.1.15 Temporal Properties . . . 616
12.1.16 Variables Properties . . . 617
12.1.17 Elevation Properties . . . 617
12.1.18 元数据属性 . . . 619
12.1.19 Dependencies Properties . . . 619
12.1.20 Legend Properties . . . 619
12.1.21 QGIS Server Properties . . . 620
12.1.22 Digitizing Properties . . . 622
12.2 属性表操作 . . . 624
12.2.1 前言：空间与非空间属性表 . . . 624
12.2.2 属性表界面介绍 . . . 625

12.2.3 Interacting with features in an attribute table . . . 630
12.2.4 Editing attribute values . . . 634
12.2.5 Exploring features attributes through the Identify Tool . . . 638
12.3 编辑 . . . 640
12.3.1 设置捕捉容差和搜索半径 . . . 641
12.3.2 捕捉和数字化选项 . . . 641
12.3.3 拓扑编辑 . . . 643
12.3.4 数字化现有图层 . . . 645
12.3.5 Advanced digitizing . . . 656
12.3.6 Shape digitizing . . . 666
12.3.7 The Advanced Digitizing panel . . . 668
12.3.8 The Processing in-place layer modifier . . . 677
12.4 Connecting and Editing Data Across Layers . . . 679
12.4.1 Joining features between two layers . . . 679
12.4.2 Setting relations between multiple layers . . . 681
12.4.3 Storing and fetching an external resource . . . 695

13 使用栅格数据 **699**
13.1 Raster Properties Dialog . . . 699
13.1.1 信息属性 . . . 700
13.1.2 源属性 . . . 700
13.1.3 Symbology Properties . . . 701
13.1.4 Transparency Properties . . . 711
13.1.5 Labels Properties . . . 712
13.1.6 Histogram Properties . . . 714
13.1.7 Rendering Properties . . . 715
13.1.8 Temporal Properties . . . 716
13.1.9 Pyramids Properties . . . 717
13.1.10 Elevation Properties . . . 718
13.1.11 元数据属性 . . . 719
13.1.12 Legend Properties . . . 719
13.1.13 Display Properties . . . 720
13.1.14 QGIS Server Properties . . . 721
13.1.15 Attribute Tables . . . 722
13.1.16 Identify raster cells . . . 727
13.2 栅格分析 . . . 727
13.2.1 Raster Calculator . . . 727

14 Working with Mesh Data **731**
14.1 What's a mesh? . . . 731
14.2 Supported formats . . . 733
14.3 Mesh Dataset Properties . . . 734
14.3.1 信息属性 . . . 735
14.3.2 源属性 . . . 736
14.3.3 Symbology Properties . . . 737
14.3.4 Labels Properties . . . 742
14.3.5 3D View Properties . . . 744
14.3.6 Rendering Properties . . . 746
14.3.7 Temporal Properties . . . 747
14.3.8 Elevation Properties . . . 747
14.3.9 元数据属性 . . . 749
14.4 Exploring mesh datasets . . . 749
14.5 Editing a mesh layer . . . 750
14.5.1 Overview of the mesh digitizing tools . . . 751
14.5.2 Exploring the Z value assignment logic . . . 751
14.5.3 Selecting mesh elements . . . 753
14.5.4 Modifying mesh elements . . . 754
14.5.5 Reindexing meshes . . . 758

14.6 Mesh Calculator . . . 758

15 使用矢量瓦片 761
15.1 什么是矢量瓦片? . . . 761
15.2 支持的格式 . . . 762
15.3 矢量瓦片数据集属性 . . . 763
15.3.1 信息属性 . . . 763
15.3.2 源属性 . . . 763
15.3.3 Symbology and Labels Properties . . . 764
15.3.4 Rendering Properties . . . 765
15.3.5 元数据属性 . . . 766
15.4 QGIS Server Properties . . . 766

16 Working with Point Clouds 767
16.1 Introduction to Point Clouds . . . 767
16.2 Point Clouds Properties . . . 768
16.2.1 信息属性 . . . 768
16.2.2 源属性 . . . 769
16.2.3 Symbology Properties . . . 772
16.2.4 3D View Properties . . . 780
16.2.5 Rendering Properties . . . 782
16.2.6 Elevation Properties . . . 782
16.2.7 元数据属性 . . . 784
16.2.8 Statistics Properties . . . 784
16.3 Virtual point cloud . . . 785

17 Working with 3D Tiles 787
17.1 What are 3D Tiles? . . . 787
17.1.1 State of 3D Tiles Support . . . 788
17.2 3D Tiles Properties . . . 789
17.2.1 信息属性 . . . 789
17.2.2 源属性 . . . 789
17.2.3 Symbology Properties . . . 789
17.2.4 3D View Properties . . . 790
17.2.5 Rendering Properties . . . 791
17.2.6 Elevation Properties . . . 791
17.2.7 元数据属性 . . . 791

18 Laying out the maps 793
18.1 Overview of the Print Layout . . . 793
18.1.1 Sample Session for beginners . . . 793
18.1.2 The Layout Manager . . . 794
18.1.3 Menus, tools and panels of the print layout . . . 795
18.2 布局项 . . . 808
18.2.1 Layout Items Common Options . . . 808
18.2.2 The Map Item . . . 814
18.2.3 The 3D Map Item . . . 824
18.2.4 The Label Item . . . 825
18.2.5 图例项 . . . 830
18.2.6 The Scale Bar Item . . . 838
18.2.7 The Table Items . . . 842
18.2.8 The Marker, Picture and North Arrow Items . . . 851
18.2.9 The Elevation Profile Item . . . 855
18.2.10 The HTML Frame Item . . . 857
18.2.11 图形项 . . . 859
18.3 Creating an Output . . . 863
18.3.1 Export settings . . . 863
18.3.2 Export as Image . . . 864
18.3.3 Export as SVG . . . 865

18.3.4 Export as PDF . . . 866
18.3.5 Generate an Atlas . . . 868
18.4 Creating a Report . . . 874
18.4.1 What is it? . . . 874
18.4.2 Get started . . . 874
18.4.3 Layout Report Workspace . . . 875
18.4.4 Export settings . . . 890

19 Working with OGC / ISO protocols **891**
19.1 WMS/WMTS Client . . . 892
19.1.1 Overview of WMS Support . . . 892
19.1.2 Overview of WMTS Support . . . 892
19.1.3 Selecting WMS/WMTS Servers . . . 893
19.1.4 Loading WMS/WMTS Layers . . . 895
19.1.5 图块集 . . . 897
19.1.6 Using the Identify Tool . . . 897
19.1.7 Viewing Properties . . . 898
19.1.8 Show WMS legend graphic in table of contents and layout . . . 902
19.2 WCS Client . . . 902
19.3 WFS and WFS-T Client . . . 904
19.4 SensorThings . . . 907
19.4.1 Setting connection . . . 907
19.4.2 Loading SensorThings data . . . 909
19.4.3 Working with a vector layer from SensorThings . . . 910
19.5 STAC (SpatioTemporal Asset Catalogs) . . . 910
19.5.1 Setting connection . . . 911
19.5.2 Browsing STAC Catalogs . . . 912
19.5.3 Filtering and Searching STAC Items . . . 913
19.6 ArcGIS REST Server . . . 915

20 使用 GPS 数据 **917**
20.1 Introducing GNSS/GPS Data . . . 917
20.1.1 What is GPS? . . . 917
20.1.2 Transferring or loading GPS data . . . 918
20.2 Live GPS tracking . . . 920
20.2.1 GPS toolbar . . . 920
20.2.2 GPS Information Panel . . . 922
20.2.3 Connect to a Bluetooth GPS for live tracking . . . 924
20.2.4 Examples of GPS devices connection in QGIS . . . 924

21 认证系统 **927**
21.1 Authentication System Overview . . . 927
21.1.1 Authentication database . . . 928
21.1.2 Custom authentication databases . . . 928
21.1.3 Master password . . . 929
21.1.4 Authentication Configurations . . . 930
21.1.5 Authentication Methods . . . 933
21.1.6 Master Password and Auth Config Utilities . . . 940
21.1.7 Using authentication configurations . . . 941
21.1.8 Python bindings . . . 941
21.2 User Authentication Workflows . . . 942
21.2.1 HTTP(S) authentication . . . 942
21.2.2 Database authentication . . . 943
21.2.3 PKI authentication . . . 944
21.2.4 Handling bad layers . . . 951
21.2.5 Changing authentication config ID . . . 951
21.2.6 QGIS Server support . . . 952
21.2.7 SSL server exceptions . . . 953

22 GRASS GIS Integration **957**
22.1 Demo dataset . . . 957
22.2 Loading GRASS raster and vector layers . . . 957
22.3 Importing data into a GRASS LOCATION via drag and drop . . . 958
22.4 Managing GRASS data in QGIS Browser . . . 958
22.5 GRASS Options . . . 958
22.6 Starting the GRASS plugin . . . 958
22.7 Opening GRASS mapset . . . 959
22.8 GRASS LOCATION and MAPSET . . . 959
22.9 Importing data into a GRASS LOCATION . . . 959
22.9.1 Creating a new GRASS LOCATION . . . 960
22.9.2 Adding a new MAPSET . . . 961
22.10 The GRASS vector data model . . . 962
22.11 Creating a new GRASS vector layer . . . 962
22.12 Digitizing and editing a GRASS vector layer . . . 963
22.13 The GRASS region tool . . . 965
22.14 The GRASS Toolbox . . . 965
22.14.1 Working with GRASS modules . . . 966
22.14.2 GRASS module examples . . . 969
22.14.3 Customizing the GRASS Toolbox . . . 971

23 QGIS 处理架构 **973**
23.1 简介 . . . 973
23.2 Configuring the Processing Framework . . . 976
23.2.1 General . . . 976
23.2.2 Menus . . . 978
23.2.3 Models and Scripts . . . 978
23.2.4 Providers . . . 978
23.3 The Toolbox . . . 979
23.3.1 The algorithm dialog . . . 981
23.3.2 Data objects generated by algorithms . . . 989
23.4 The history manager . . . 990
23.4.1 The processing history . . . 990
23.4.2 The processing log . . . 991
23.5 The model designer . . . 991
23.5.1 The model designer interface . . . 992
23.5.2 Creating a model . . . 995
23.5.3 Saving and loading models . . . 1004
23.5.4 Editing a model . . . 1004
23.6 The batch processing interface . . . 1006
23.6.1 简介 . . . 1006
23.6.2 The parameters table . . . 1007
23.6.3 Filling the parameters table . . . 1008
23.6.4 Executing the batch process . . . 1009
23.7 从控制台使用处理算法 . . . 1009
23.7.1 从 Python 控制台调用算法 . . . 1010
23.7.2 创建脚本并从工具箱运行 . . . 1014
23.7.3 执行前与执行后脚本钩子 . . . 1016
23.8 Using processing from the command line . . . 1017
23.9 Writing new Processing algorithms as Python scripts . . . 1019
23.9.1 Extending QgsProcessingAlgorithm . . . 1019
23.9.2 The @alg decorator . . . 1023
23.9.3 Handing algorithm output . . . 1025
23.9.4 Communicating with the user . . . 1025
23.9.5 为脚本编写文档 . . . 1025
23.9.6 Flags . . . 1025
23.9.7 Best practices for writing script algorithms . . . 1026
23.10 处理算法的输入与输出 . . . 1026

23.10.1 输入类型 . . . 1026
23.10.2 Output types . . . 1028
23.11 Configuring external applications . . . 1029
23.11.1 A note for Windows users . . . 1029
23.11.2 A note on file formats . . . 1029
23.11.3 A note on vector layer selections . . . 1029
23.11.4 Using third-party Providers . . . 1030

24 数据处理提供程序和算法 1041
24.1 QGIS algorithm provider . . . 1041
24.1.1 3D Tiles . . . 1041
24.1.2 Cartography . . . 1043
24.1.3 Check Geometry . . . 1062
24.1.4 数据库 . . . 1108
24.1.5 File tools . . . 1115
24.1.6 Fix Geometry . . . 1118
24.1.7 GPS . . . 1144
24.1.8 Interpolation . . . 1147
24.1.9 Layer tools . . . 1159
24.1.10 网孔 . . . 1162
24.1.11 Metadata tools . . . 1174
24.1.12 Modeler tools . . . 1179
24.1.13 Network analysis . . . 1189
24.1.14 Plots . . . 1202
24.1.15 Point Cloud Conversion . . . 1215
24.1.16 Point Cloud Data Management . . . 1221
24.1.17 Point Cloud Extraction . . . 1234
24.1.18 栅格分析 . . . 1239
24.1.19 Raster Creation . . . 1301
24.1.20 栅格地形分析 . . . 1317
24.1.21 栅格工具 . . . 1330
24.1.22 Vector analysis . . . 1341
24.1.23 Vector coverage . . . 1367
24.1.24 Vector creation . . . 1370
24.1.25 Vector general . . . 1399
24.1.26 矢量几何图形 . . . 1438
24.1.27 Vector overlay . . . 1552
24.1.28 Vector selection . . . 1572
24.1.29 Vector table . . . 1589
24.1.30 Vector Tiles . . . 1606
24.2 GDAL algorithm provider . . . 1609
24.2.1 栅格分析 . . . 1609
24.2.2 Raster conversion . . . 1639
24.2.3 Raster extraction . . . 1647
24.2.4 Raster miscellaneous . . . 1654
24.2.5 Raster projections . . . 1673
24.2.6 Vector conversion . . . 1678
24.2.7 Vector geoprocessing . . . 1685
24.2.8 Vector miscellaneous . . . 1693

25 插件 1705
25.1 QGIS 插件 . . . 1705
25.1.1 Core and External plugins . . . 1705
25.1.2 The Plugins Dialog . . . 1706
25.2 使用 QGIS 核心插件 . . . 1709
25.2.1 DB Manager Plugin . . . 1709
25.2.2 Geometry Checker Plugin . . . 1714
25.2.3 MetaSearch Catalog Client . . . 1718

25.2.4 离线编辑 . . . 1725
25.2.5 Topology Checker Plugin . . . 1727
25.3 QGIS Python console . . . 1729
25.3.1 The Interactive Console . . . 1730
25.3.2 The Code Editor . . . 1732

26 附录 **1735**
26.1 附录 C：QGIS 文件格式 . . . 1735
26.1.1 QGS/QGZ - QGIS 工程文件格式 . . . 1735
26.1.2 QLR - QGIS 图层定义文件定义 . . . 1737
26.1.3 QML - QGIS 样式文件格式 . . . 1738
26.2 Appendix D: QGIS R script syntax . . . 1739
26.2.1 Inputs . . . 1740
26.2.2 输出 . . . 1740
26.2.3 Syntax Summary for QGIS R scripts . . . 1740
26.2.4 Examples . . . 1742
26.3 附录 E：QGIS 应用程序网络连接 . . . 1744

27 文献和网络参考 **1747**

CHAPTER 1

入门指南

本章简要介绍了 QGIS 的安装、QGIS 示例数据的下载和运行第一个简单的栅格数据和矢量数据可视化的案例。

1.1 安装 QGIS

QGIS 项目根据您使用的平台提供了不同的 QGIS 安装方法。

1.1.1 二进制文件安装

标准安装程序适用于 MS Windows 和 macOS，二进制软件包 (rpm 和 deb) 或软件资源库适用于更多版本的 GUN/Linux 操作系统。

有关您操作系统的更多信息和说明，请查看 https://qgis.org/download/ 。

1.1.2 从源代码安装

如果您需要从源代码构建 QGIS，请参阅安装说明。它们与 QGIS 源代码一起分散在一个名为 `INSTALL` 的文件中，您还可以在 https://github.com/qgis/QGIS/blob/release-3_44/INSTALL.md 在线进行查找。

如果您想构建特定版，而不是开发中的版本，您需要将上面提到的连接中的 `master` 替换为发行分支 (通常为 `release-X_Y` 的形式)。(安装说明可能会有不同)

1.1.3 安装到外媒介

可以将 QGIS（包括所有的插件和设置）安装到一个闪存驱动器上。这是通过定义一个*--profiles-path* 选项来实现的，该选项覆盖默认的用户配置文件（*user profile*）路径，并强制设置（**QSettings** ）也使用这个目录。更多信息请查阅系统设置（ 系统设置 ）章节。

1.1.4 下载示例数据

本用户指南包含基于 QGIS 样本数据集 (也称为 阿拉斯加数据集) 的示例。从 https://github.com/qgis/QGIS-Sample-Data/archive/master.zip 下载示例数据，并将存档文件解压到系统上任何方便的位置。

阿拉斯加数据集包含了用于示例和用户手册中截图的所有 GIS 数据。它还包含了一个较小的 GRASS 数据库。QGIS 样本数据集的投影坐标系是单位为英尺的阿拉斯加阿尔伯斯等面积坐标系（Alaska Albers Equal Area）。该投影坐标系的 EPSG 号为 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]]
```

如果打算使用 QGIS 作为 GRASS 的图形前端，您可以在 GRASS GIS 官方网站找到一些示例 (例如斯皮尔菲什或南达科他州)，网址是：https://grass.osgeo.org/download/data/ 。

1.2 启动和关闭 QGIS

QGIS 可以像您电脑上的其他应用一样打开。这意味着你可以通过以下方式打开：

- 使用 应用菜单，开始菜单，或者 DOCK
- 在应用文件夹或者桌面快捷方式上双击
- 双击现存的 QGIS 工程文件 (扩展名为 `.qgz` 或者 `.qgs`)。注意，这也将打开此工程。
- 在命令提示符里输入 `qgis` （假设 QGIS 已经被添加到默认路径 PATH 里面，或者当前目录为安装目录）

关闭 QGIS，使用：

- 菜单选项 `工程` ► `退出` *QGIS*，或者用快捷键 `Ctrl+Q`
- *QGIS* ► `退出` *QGIS*，或者使用快捷键 `Cmd+Q`
- 或者使用应用程序主界面右上角的红十字。

1.3 案例：加载栅格图层和矢量图层

现在您已经安装 *QGIS*，也有了示例数据集，接下来将演示第一个示例会话。在此示例中将栅格图层和矢量图层进行可视化，我们将使用：

- `landcover` 栅格图层 (`qgis_sample_data/raster/landcover.img`)
- `lakes` 矢量图层 (`qgis_sample_data/gml/lakes.gml`)

其中，`qgis_sample_data` 代表数据集解压路径。

1. 启动 QGIS，参见启动和关闭 *QGIS* 。
2. 将使用的数据位于 `Albers Equal Area`，因此让我们相应地设置工程的 CRS：
   1. 单击 QGIS 界面右下角的 选择投影 按钮。工程属性对话框打开时 *CRS* 选项卡处于激活状态。
   2. Type `2964` in the *Filter* text area.
   3. 选择 CRS 名称为 `NAD27 / Alaska Albers` 的行。

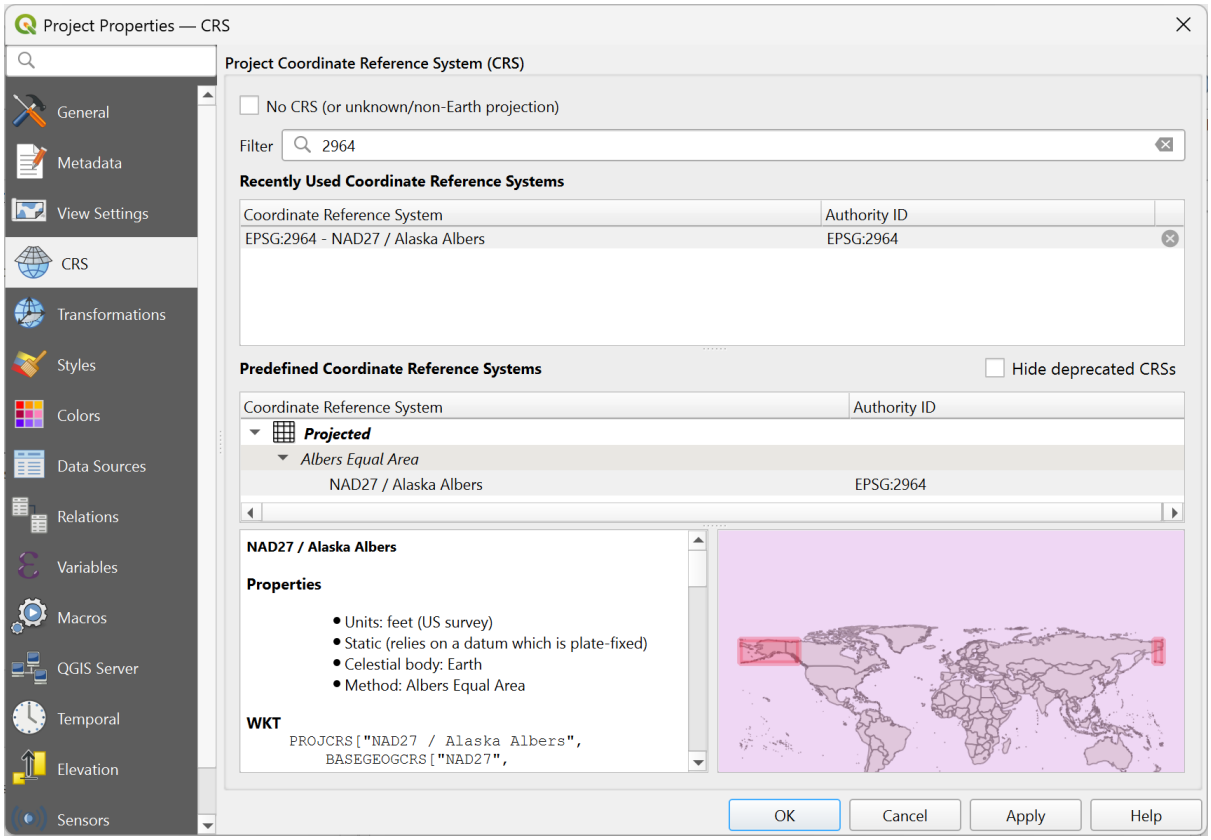


Fig. 1.1: 选择数据的坐标参照系

4. 按 确定按钮

> **Note**
>
> 您可以暂时忽略/关闭可能显示的” 相近变换” 信息。

3. 在 QGIS 中加载文件：
   1. 单击 打开数据源管理器 图标。数据源管理器应该在浏览器模式下打开。

2. 浏览到文件夹 `qgis_sample_data/raster/`

3. 选择 ERDAS IMG 文件 *landcover.img* 并双击。landcover 图层添加到画布，同时数据源管理器窗口依旧保持打开状态。

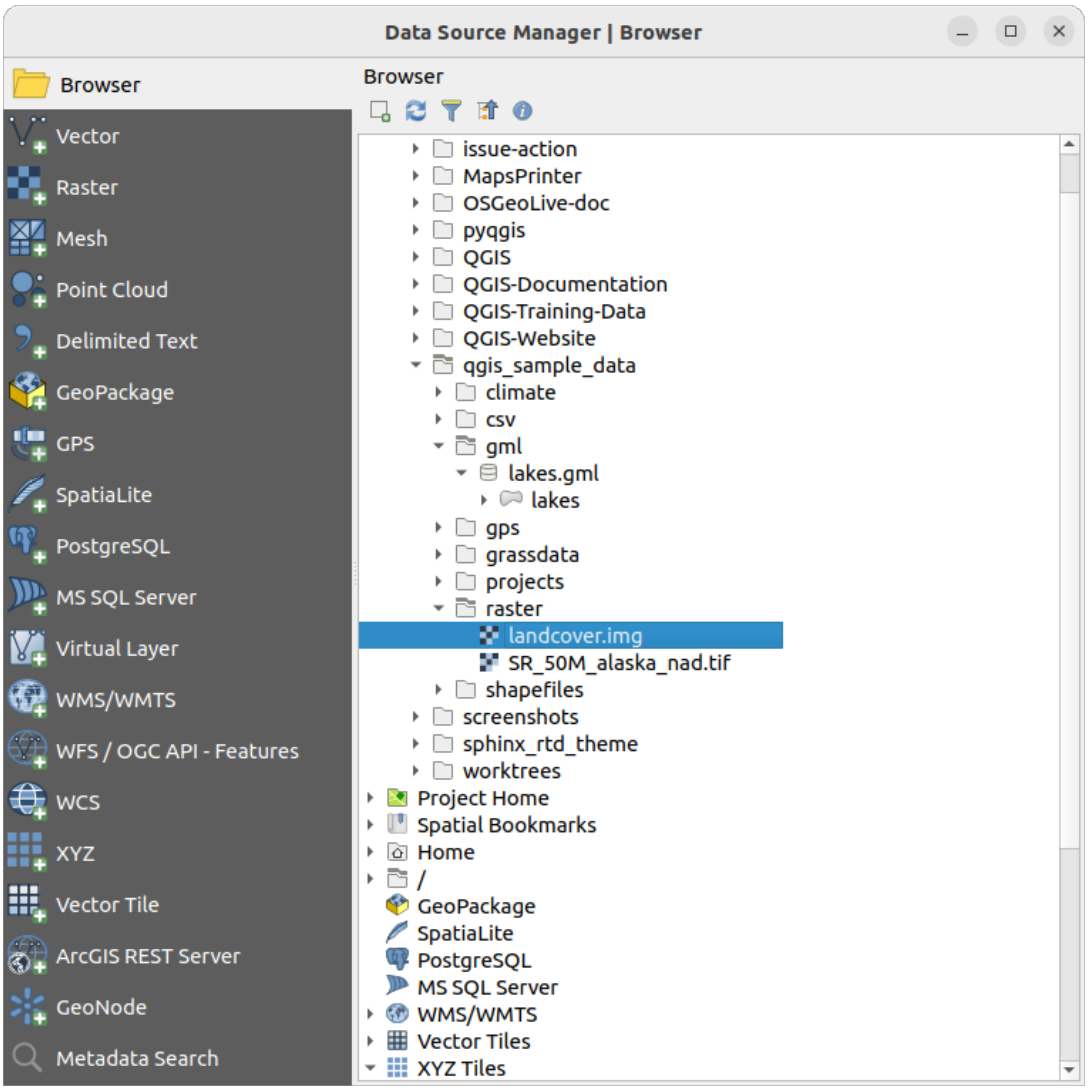


Fig. 1.2: 在 QGIS 中添加数据到新工程

4. 要加载 lakes 数据，请浏览至文件夹 `qgis_sample_data/gml/`，然后将 *lakes.gml* 文件拖放到 QGIS 主对话框上 (或者如上所述双击即可)。

5. 选择要添加的项对话框被打开并扫描文件。这是由于 `.gml` 文件格式能够一次储存多个图层。

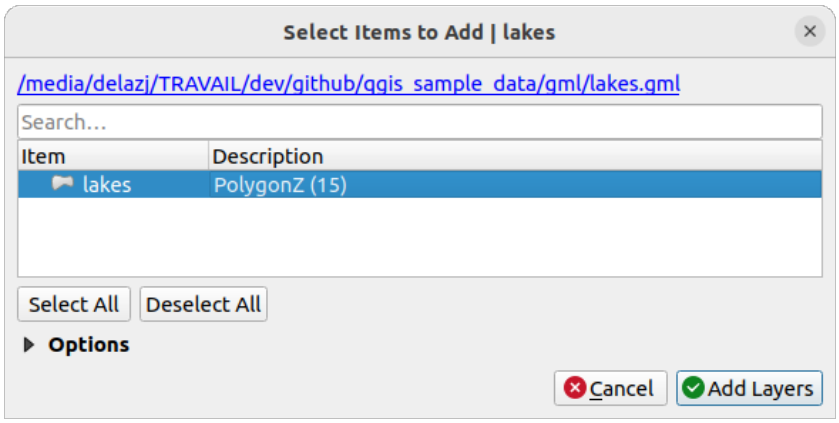


Fig. 1.3: 选择文件中的图层

6. 在此示例中，只有一个 *lakes* 图层。选择然后按 添加图层按钮。
7. 该图层被添加到 图层面板中

4. 关闭数据源管理窗口

在 图层面板，您可能注意到 *lakes* 图层旁边显示 图层未设置坐标参照系 。我们来调整一下。

1. 单击 图标，将打开 坐标参照系选择器对话框。
2. 如前所述，查找并选择 *NAD27 / Alaska Albers* CRS 条目。
3. 单击 确定

现在您的工程中有了一些随机颜色渲染的两个可用图层。让我们自定义 lakes 图层吧。

1. 在 导航工具栏上选择 放大 工具
2. 缩放到一些湖泊的区域
3. 在图例中双击 `lakes` 图层，打开 属性对话框
4. 改变湖泊颜色
    1. 单击 符号化选项卡
    2. 选择蓝色作为填充颜色

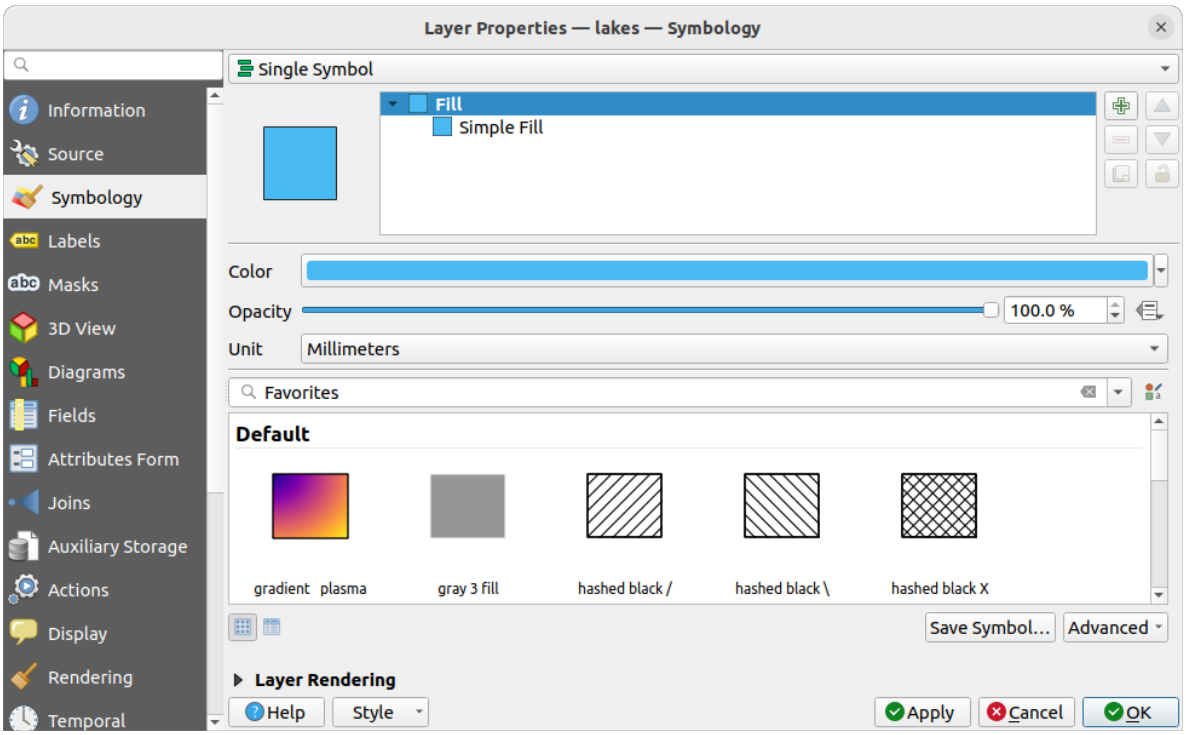


Fig. 1.4: 选择湖泊颜色

3. 按下 确定。现在地图画布中湖泊显示为蓝色了。

5. 显示湖泊名称:

   1. 重新打开 `lakes` layer 属性对话框
   2. 单击 标注选项卡
   3. 在下拉菜单中选择 单一标注来启用标注。
   4. 从 标注为列表中选择 NAMES 字段。

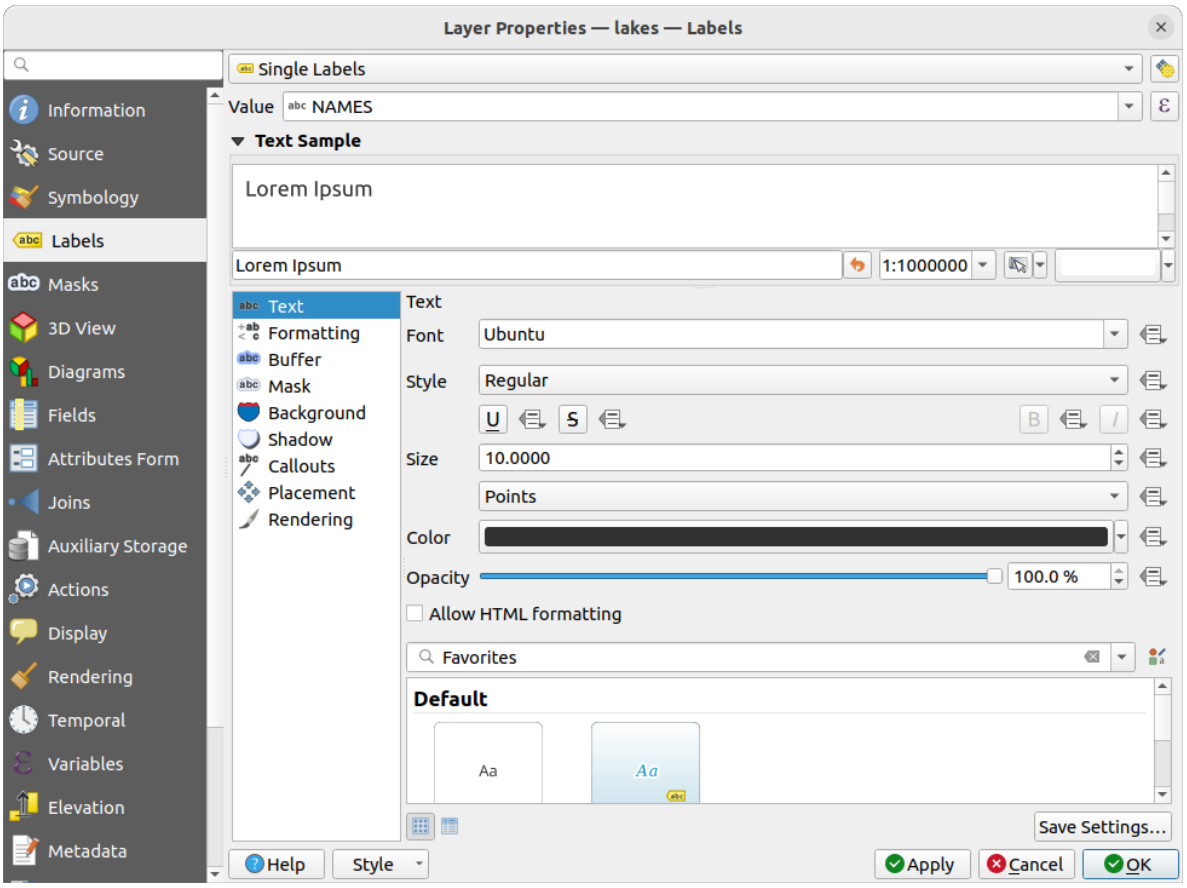


Fig. 1.5: 显示湖泊名称

5. 按下 *应用*。现在名称加载在边界上。

6. 你可以在标注周围添加白色轮廓已增加标注的辨识度：

   1. 单击左边列表中的 *描边*选项卡
   2. 勾选 *绘制文本描边*
   3. 描边大小选择 3
   4. 单击 *应用*
   5. 检查结果是否良好，如果需要可以调整轮廓大小值。
   6. 最后单击 *确定*来关闭 *图层属性*对话框并应用更改。

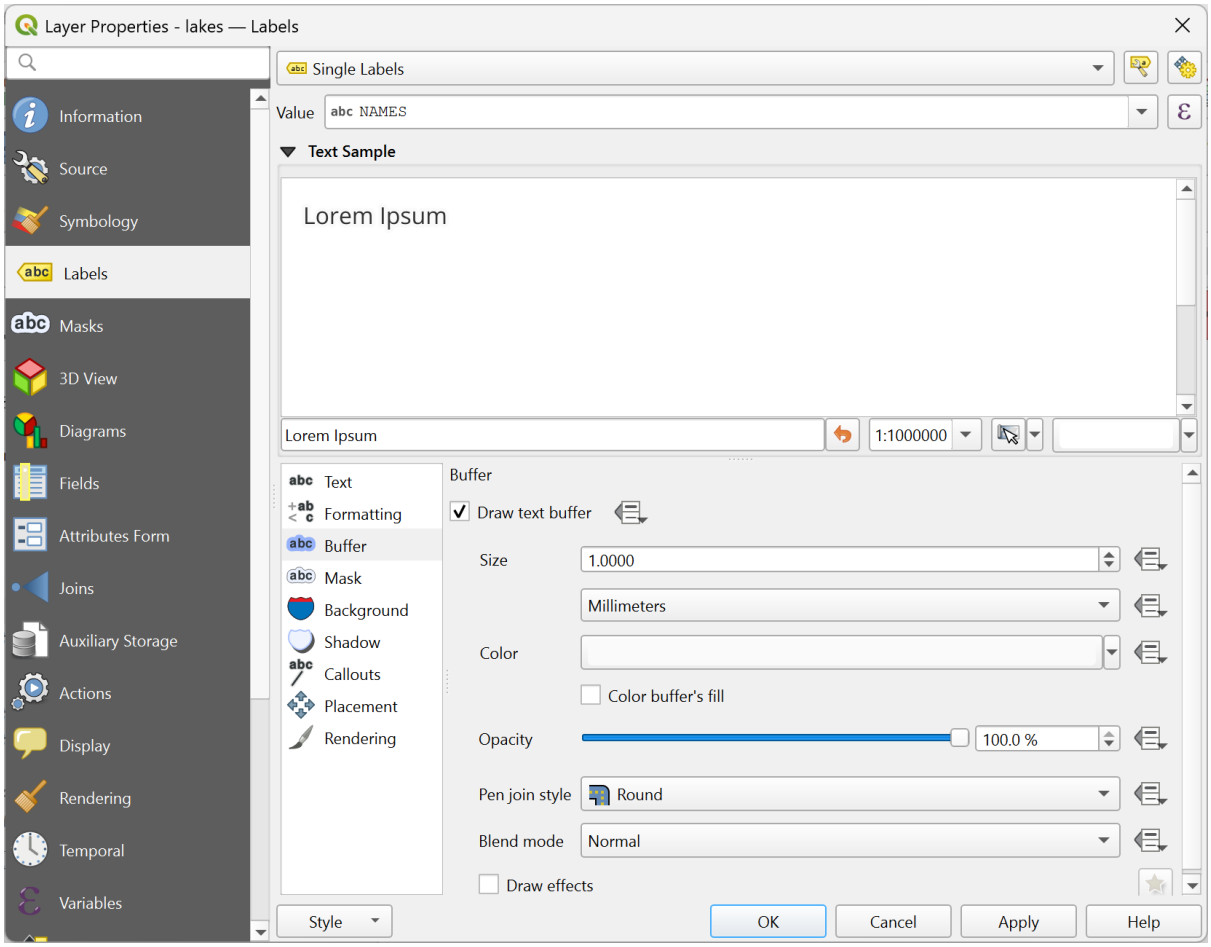


Fig. 1.6: 显示标注周围的描边

现在，让我们添加一些装饰以塑造地图并将其导出 QGIS：

1. 选择 视图 ► 装饰 ► 比例尺菜单
2. 在打开的对话框中，勾选 启用比例尺选项
3. 根据您的需要自定义对话框中的选项
4. 按 应用按钮
5. 同样，可以从装饰菜单中添加更多项（指北针、版权…）到自定义属性的地图画布中。
6. 单击 工程 ► 导入/导出 ► 导出为图片...
7. 在打开的对话框中按 保存按钮
8. 选择文件位置与格式，并再次按下 保存确认。
9. 按 工程 ► 保存... ，将您的修改保存到 `.qgz` 工程文件中。

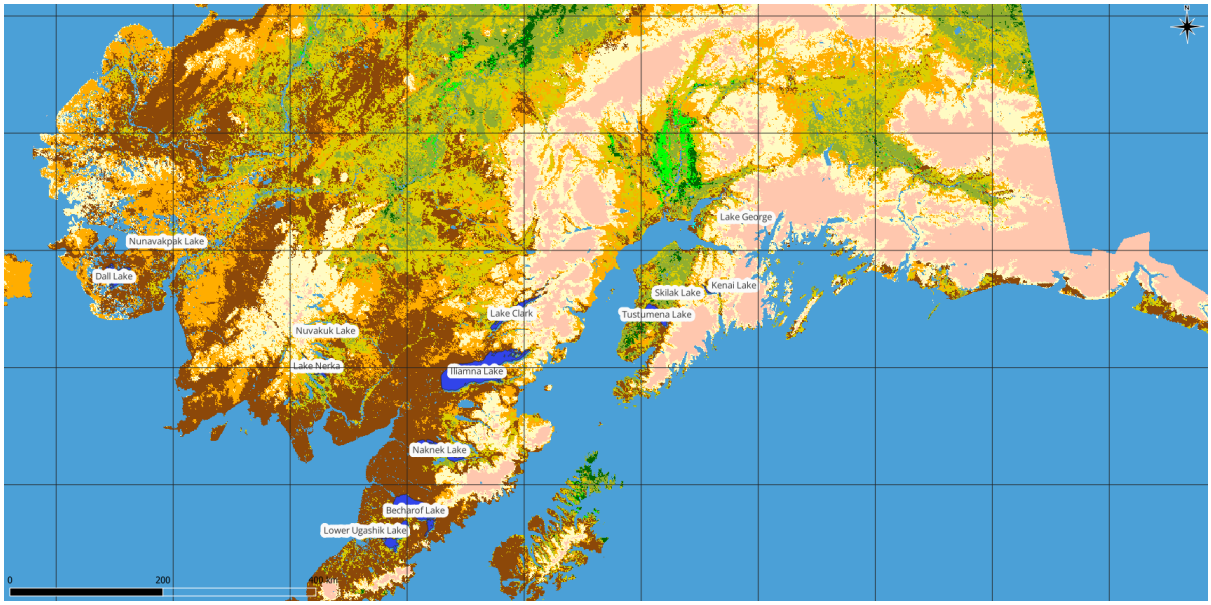


Fig. 1.7: 显示带装饰的导出地图

就这样！您可以看到在 QGIS 中可视化栅格和矢量图层、配置它们并以可以在其他软件中使用的图像格式生成地图是多么容易。让我们继续了解有关可用功能、特性和设置的更多信息，以及如何使用它们。

Note

跟随 培训手册，通过一步步练习来继续学习 QGIS。

CHAPTER 2

使用工程文件

2.1 QGIS 工程简介

QGIS 会话的状态称为工程，QGIS 一次只能处理一个工程。设置可以是指定的工程，也可以是新工程的应用类型默认设置 (请参阅选项)。QGIS 可以使用菜单选项 工程 ► 保存或 工程 ► 另存为... 将工作区的状态保存到*QGIS 工程文件* 。

Note

如果工程已被修改过，该 * 符号将出现在标题栏中，默认情况下，QGIS 将询问您是否要保存更改。此行为由 设置 ► 选项 ► 常规下的 需要时提示保存工程和数据源更改情况设置而进行控制的。

你可以通过 工程 ► 打开... 加载现有的项目，工程 ► 从模板新建或者 工程 ► 最近打开 ►，将现有工程加载进 QGIS 里。

从 浏览器面板，你可以 打开工程，提取符号…打开样式管理器将符号导出为 XML，将它们添加到默认样式，或导出为 PNG/SVG。您还可以访问 文件属性…对话框。可以展开工程文件以查看其图层。图层的上下文菜单提供与浏览器中其他地方相同的操作。

启动时，会显示 工程模板和 最近工程列表，包括截图、名称和文件路径（最多十个工程）。最近工程列表便于快速访问近期使用过的工程。双击某一条目即可打开对应的工程或工程模板；右键单击条目可选择 固定到列表、打开目录... 或 从列表中移除。如果您在 最近工程列表中看到某个工程，却无法在文件管理器中找到它，可使用 打开目录... 选项来帮助定位那些可能已被移动、重命名或丢失的工程。您也可以选择 清空列表，以从 最近工程列表中移除所有项目。如果列表中包含已固定的工程，执行 清空列表操作后会弹出一个消息框，询问是否同时移除这些已固定的工程。此外，您还可以通过添加图层来自动创建一个新工程，此时上述列表将自动消失，让位于地图画布。

如果您想清理会话并重新开始，请转到 工程 ► 新建。如果在现有工程打开或上次保存后进行了更改，将提示您保存该工程。

打开新工程时，标题栏将显示 未命名的工程，直到您将其保存。

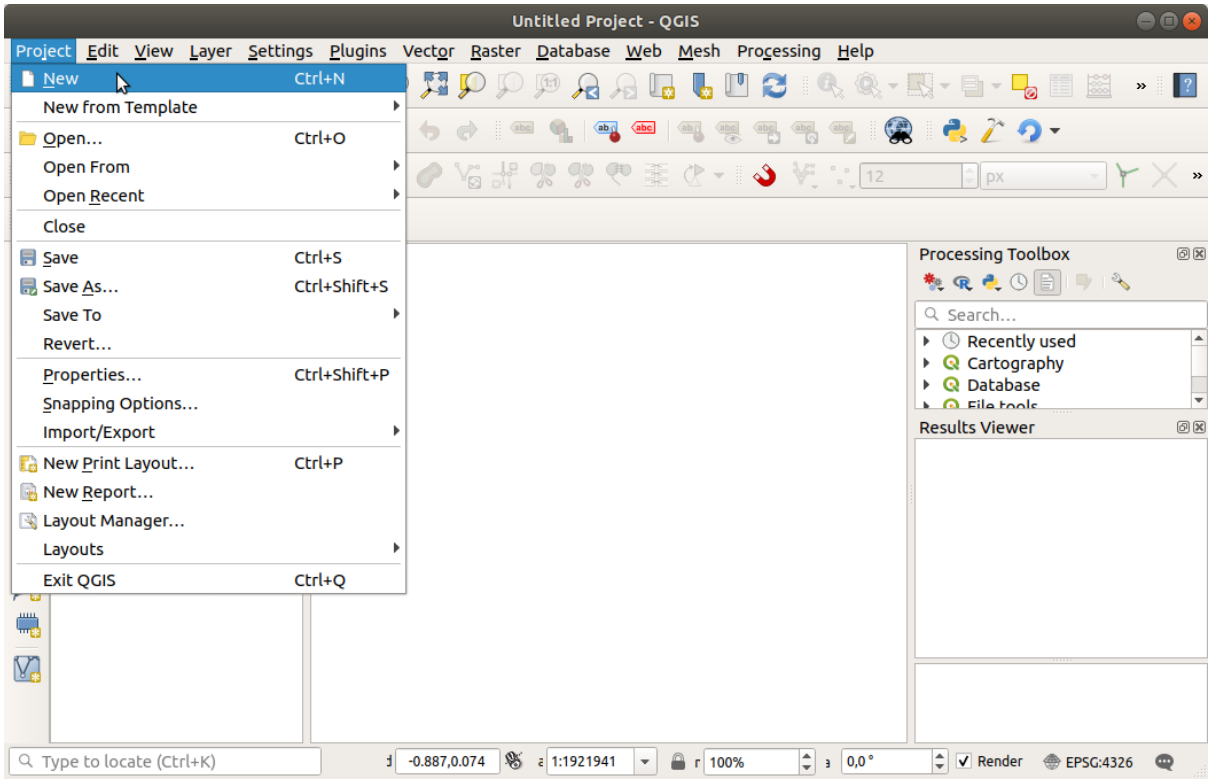


Fig. 2.1: 在 QGIS 中启动新工程

保存在工程文件中的信息包括：

- 添加图层
- 可以查询哪些图层
- 图层属性，包括符号化和样式
- 图层注解
- 2D 和 3D 地图视图
- 每个地图视图的投影
- 每个地图的上次查看范围
- 打印布局
- 使用设置打印布局元素
- 打印布局地图集设置
- 数字化设置
- 表关系
- 工程宏命令
- 工程默认样式
- 插件设置
- 工程属性中 OWS 设置选项卡的 QGIS Server 设置
- 数据库管理器中存储的查询

工程文件以 XML 格式保存 (参见*QGS/QGZ - QGIS 工程文件格式*) 。这意味着，如果您知道自己在做什么，就可以在 QGIS 之外编辑文件。工程文件格式已更新多次，来自较旧 QGIS 版本的工程文件可能不再正常工作。

> **Note**
>
> 默认情况下，QGIS 将警告您版本差异。此行为在 *设置* ► *选项* 的 *通用* 选项卡中进行控制 (*当打开旧版 QGIS 格式的工程文件时提出警告*) 。

每当您在 QGIS 中保存 `.qgs` 工程文件时，都会在与工程文件相同的目录中创建备份文件，扩展名为 `.qgs~` 。

QGIS 工程的扩展名是 `.qgs` ，但当从 QGIS 保存时，默认使用带有 `.qgz` 扩展名的压缩格式保存。`.qgs` 与用于*辅助数据* 相关的 SQLite 数据库 (`.qgd`) 一起嵌入到 `.qgz` 文件中 (zip 归档文件)，您可以通过解压 `.qgz` 文件来获取上述文件。

> **Note**
>
> *Auxiliary Storage Properties* 机制使压缩工程特别有用，因为它嵌入了辅助数据。

还可以使用以下工程菜单项将工程保存/加载到 PostgreSQL、地理包或 Oracle 数据库或从中加载工程：

- *工程* ► *打开从*
- *工程* ► *保存到*

这两个菜单项都有子菜单，其中列出了特别的工程存储实现方式 (PostgreSQL、地理包和 Oracle)。单击将打开对话框，选择地理包连接和工程、PostgreSQL 连接、架构和工程或 Oracle 连接、所有者和工程。

存储在地理包、PostgreSQL 或 Oracle 中的工程也可以通过 QGIS 浏览器面板加载，方法是双击或将它们拖到地图画布。

2.2 处理损坏的文件路径

在打开工程时，由于服务器/数据库不可用、重命名或移动了文件等原因，QGIS 可能无法访问某些数据源。随即 QGIS 打开 *处理不可用图层*对话框，检测并列出未找到的图层，您可以：

- 双击 *数据源*字段，校正每个图层的路径，然后单击 *应用更改*；
- 选择某一行，单击 *浏览*按钮找到相关图层所在数据源的正确位置，然后单击 *应用更改*；
- 单击 *自动查找浏览文件夹*，并尝试自动修复所有或选中的断开路径。请注意，浏览可能需要一些时间。然后单击 *应用更改*。
- 忽略该信息，单击 *保留不可用图层*按钮打开带有损坏路径的工程。然后，图层将显示在 *图层*面板，在 *图层*面板相关图层名称旁边显示 无效图层! 图标，单击该图标打开修复数据源对话框进行修复，在修复路径之前不会显示任何数据。或在图层上下文菜单中 *修复数据源...* 。

  使用 *修复数据源...* 工具，一旦图层路径被修复，QGIS 会扫描所有其他损坏的路径，并尝试自动修复那些具有相同损坏文件路径的路径。
- 从工程 *移除不可用图层*。

使用*--skipbadlayers* 选项从命令行启动 QGIS 可以帮助您在启动时跳过 *处理不可用图层*对话框。

2.3 生成输出

有几种方法可以从 QGIS 会话中生成输出。我们已经在*QGIS 工程简介* 讨论了保存为工程文件，生成输出文件的其他方法包括：

- 创建图像: 工程 ► 导入/导出 ► 导出地图为图像... 以自定义比例、分辨率和大小将地图画布渲染输出为图像格式 (PNG, JPG, TIFF...) 。可以在导出的图像中包含配准信息，只需启用 追加配准信息 (嵌入或者借助于坐标信息文件) 。更多信息请参阅导出地图视图 。
- 导出为 PDF 文件: 工程 ► 导入/导出 ► 导出地图为 *PDF* 文件... 以自定义比例、分辨率和一些高级设置 (简化、地理配准等) 将地图画布渲染输出为 PDF。更多详细信息请参见导出地图视图 。
- 导出为 DXF 文件: 工程 ► 导入/导出 ► 导出工程为 *DXF* 文件... 打开一个对话框，可以在其中定义要导出到 DXF 的' 符号模式', ' 符号比例' 和矢量图层。通过' 符号模式'，可以高保真地导出原始 QGIS 符号系统中的符号 (请参见新建 *DXF* 文件) 。
- 设计地图: 工程 ► 新建打印布局... 打开一个对话框，可以在其中布局和打印当前地图画布 (请参见*Laying out the maps* 一节)。

CHAPTER 3

QGIS 图形用户界面

QGIS 图形用户界面 (GUI) 如下图所示 (黄色圆圈中的数字 1 到 5 表示 QGIS 图形用户界面的重要元素，将在下面讨论)。

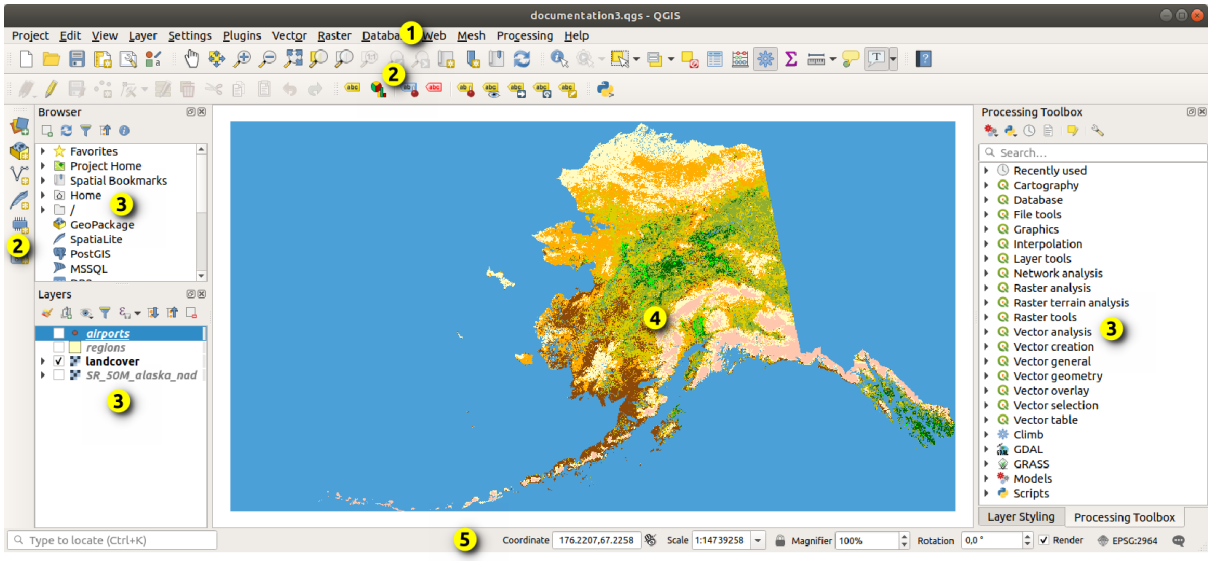


Fig. 3.1: 带阿拉斯加样本数据的 QGIS 图形用户界面

> **Note**
>
> 根据操作系统和窗口管理器的不同，窗口装饰（标题栏等）可能会有所不同。

QGIS 主界面 (Fig. 3.1) 由五个组件类型构成：

1. 菜单栏
2. 工具栏
3. 面板
4. 地图视图

5. 状态栏

向下滚动，并查看这些内容的详细说明。

3.1 菜单栏

菜单栏提供使用标准分层菜单访问 QGIS 功能的功能。菜单及其选项、相关图标和键盘快捷键如下所述。可以重新配置键盘快捷键 (设置 ► 键盘快捷键)。

大多数菜单选项都有相应的工具，反之亦然。但是，菜单的组织方式与工具栏不完全相同。下表显示了工具栏中菜单选项的位置。插件可以为菜单添加新的选项。有关工具和工具栏的详细信息，请参阅工具栏。

Note

QGIS 是一个跨平台的应用程序。工具通常在所有平台上都可以使用，但根据操作系统的不同，它们可能被放置在不同的菜单中。下面的列表显示了最常见的位置，包括已知的变量。

3.1.1 工程

工程菜单提供了工程文件 的访问和退出等选项，提供如下工具：

- 新建工程文件或使用其他工程文件作为模板 (请参阅工程文件选项 了解模板配置)
- guilabel: 打开... 来自文件、地理包、PostgreSQL 或 Oracle 数据库的工程
- 关闭工程或将其恢复到上次保存的状态
- 保存工程为 `.qgs` 或 `.qgz` 文件格式，作为文件或保存到地理包、PostgreSQL 或 Oracle 数据库中
- 将地图画布导出为不同格式，或使用打印布局 进行更复杂的输出
- 为几何图形编辑设置工程属性和捕捉选项。

Table 3.1: 工程菜单项

菜单选项	快捷键	工具栏	参考
新建	Ctrl+N	工程	*QGIS* 工程简介
由模板新建 ►			*QGIS* 工程简介
打开...	Ctrl+O	工程	*QGIS* 工程简介
打开 ►			*QGIS* 工程简介
► 地理包...			*QGIS* 工程简介
► *PostgreSQL*...			*QGIS* 工程简介
► *Oracle*...			*QGIS* 工程简介
最近打开的文件 ►	Alt+J + R		*QGIS* 工程简介
► *Clear List*			*QGIS* 工程简介
关闭			*QGIS* 工程简介
保存	Ctrl+S	工程	*QGIS* 工程简介
另存为...	Ctrl+Shift+S	工程	*QGIS* 工程简介
保存到 ►			*QGIS* 工程简介
► 模板...			*QGIS* 工程简介
► 地理包...			*QGIS* 工程简介
► *PostgreSQL*...			*QGIS* 工程简介
恢复...			
工程属性...	Ctrl+Shift+P		工程属性

continues on next page

Table 3.1 – continued from previous page

菜单选项	快捷键	工具栏	参考
捕捉选项...			捕捉和数字化选项
导入/导出 ►			
► 导出地图为图像...			导出地图视图
► 导出地图为 *PDF* 文件...			导出地图视图
► 导出工程为 *DXF* 文件...			新建 *DXF* 文件
► 从 *DWG/DXF* 文件导入图层...			导入 *DXF* 或 *DWG* 文件
新建打印布局...	Ctrl+P	工程	*Laying out the maps*
新建报告...			*Creating a Report*
布局管理器...		工程	*Laying out the maps*
布局 ►			*Laying out the maps*
模型 ►			*The model designer*
退出 *QGIS*	Ctrl+Q		

在 macOS 系统中，退出 *QGIS* 命令对应于 *QGIS* ► 退出 *QGIS* (Cmd+Q)。

3.1.2 编辑

编辑菜单提供了编辑图层属性或几何图形所需的大多数本机工具。要启用 编辑菜单选项，需要通过单击 切换编辑 (详情请参阅编辑) 切换到编辑模式。

Table 3.2: 编辑菜单项

菜单选项	快捷键	工具栏	参考
撤销	Ctrl+Z	数字化	*Undo and Redo*
重做	Ctrl+Shift+Z	数字化	*Undo and Redo*
剪切要素	Ctrl+X	数字化	*Cutting, Copying and Pasting Features*
复制要素	Ctrl+C	数字化	*Cutting, Copying and Pasting Features*
粘贴要素	Ctrl+V	数字化	*Cutting, Copying and Pasting Features*
粘贴要素为 ►			属性表操作
► 新建矢量图层...			属性表操作
► 临时图层...	Ctrl+Alt+V		属性表操作
删除所选要素		数字化	*Deleting Selected Features*
选择 ►			选择要素
► 选择要素		选择	选择要素
► 按多边形选择要素		选择	选择要素
► 自由手绘选择要素		选择	选择要素
► 按半径选择要素		选择	选择要素
► 按值选择要素...	F3	选择	选择要素

continues on next page

Table 3.2 – continued from previous page

菜单选项	快捷键	工具栏	参考
► 按表达式选择要素...	Ctrl+F3	选择	选择要素
► 在所有图层中取消对要素的选择	Ctrl+Alt+A	选择	选择要素
► 从当前活动图层中取消选择要素	Ctrl+Shift+A	选择	选择要素
► 重新选择要素			选择要素
► 选择全部要素	Ctrl+A	选择	选择要素
► 反选要素		选择	选择要素
添加记录	Ctrl+.	数字化	
添加点要素	Ctrl+.	数字化	*Adding Features*
添加线要素	Ctrl+.	数字化	*Adding Features*
添加多边形要素	Ctrl+.	数字化	*Adding Features*
添加注记 ►			*Annotation Tools*
► 表单注记		注记	*Annotation Tools*
► *HTML* 注记		注记	*Annotation Tools*
编辑属性 ►			
► 修改所选要素的属性		数字化	*Editing attribute values*
► 合并所选要素的属性		高级数字化	*Merge attributes of selected features*
编辑几何图形 ►			
► 移动要素		高级数字化	*Move Feature(s)*
► 复制并移动要素		高级数字化	*Move Feature(s)*
► 旋转要素		高级数字化	*Rotate Feature(s)*
► 缩放要素		高级数字化	*Scale Feature*
► 简化要素		高级数字化	*Simplify Feature*
► 添加环		高级数字化	*Add Ring*
► 添加部件		高级数字化	*Add Part*
► 填充环		高级数字化	*Fill Ring*
► 删除环		高级数字化	*Delete Ring*
► 删除部件		高级数字化	*Delete Part*
► 重塑要素		高级数字化	*Reshape Features*

continues on next page

Table 3.2 – continued from previous page

菜单选项	快捷键	工具栏	参考
► 偏移曲线		高级数字化	*Offset Curves*
► 分割要素		高级数字化	*Split Features*
► 分割部件		高级数字化	*Split parts*
► 合并所选要素		高级数字化	*Merge selected features*
► 顶点工具 (所有图层)		数字化	顶点工具
► 顶点工具 (当前图层)		数字化	顶点工具
► 反转线方向		高级数字化	*Reverse Line*
► 裁剪/扩展要素		高级数字化	*Trim/Extend Feature*
旋转点符号		高级数字化	*Rotate Point Symbols*
偏移点符号		高级数字化	*Offset Point Symbols*

依赖于所选图层几何图形类型 (即点、折线或多边形) 的工具会相应被激活：

Table 3.3: 基于” 移动要素” 几何图形的图标

菜单选项	点	折线	多边形
移动要素			
复制并移动要素			

3.1.3 视图

地图在地图视图中渲染。您可以使用 视图工具与这些视图进行交互。例如，您可以：

- 在主地图画布旁边新建 2D 或 3D 地图视图
- 缩放或平移 到任意位置
- 查询显示要素的属性或几何图形
- 使用预览模式、注记或装饰改善地图视图
- 访问所有面板或工具栏

该菜单还允许您使用以下操作重新组织 QGIS 界面：

- 切换全屏模式: 占用整个屏幕，同时隐藏标题栏
- 切换面板可见性: 显示或隐藏启用的面板 - 数字化要素 (获得最大画布可见区域) 以及使用 QGIS 主画布进行 (投影/录制) 演示时非常有用
- *Toggle Map Only*: hides panels, toolbars, menus and status bar and only shows the map canvas. Combined with the full screen option, it makes your screen display only the map.

Table 3.4: 视图菜单项

菜单选项	快捷键	工具栏	参考
新建地图视图	Ctrl+M	地图浏览	*2D 地图视图*
3D 地图视图 ►			*3D 地图视图*
► 新建 *3D* 地图视图	Ctrl+Alt+M	地图浏览	*3D 地图视图*
► 管理 *3D* 地图视图			*3D 地图视图*
平移地图		地图浏览	*探索地图视图*
平移地图到所选内容		地图浏览	*探索地图视图*
放大	Ctrl+Alt++	地图浏览	*探索地图视图*
缩小	Ctrl+Alt+-	地图浏览	*探索地图视图*
Data Filtering ►			*2D 地图视图*
► *Temporal Controller*		地图浏览	*时态控制面板*
► *Elevation Controller*			*The Elevation Controller panel*
识别要素	Ctrl+Shift+I	属性	*识别要素*
测量 ►		属性	*Measuring*
► 测量线	Ctrl+Shift+M	属性	*Measuring*
► 测量面积	Ctrl+Shift+J	属性	*Measuring*
► 测量方位角		属性	*Measuring*
► 测量角度		属性	*Measuring*
统计汇总		属性	*统计摘要面板*
Elevation Profile			*Elevation Profile View*
全图显示	Ctrl+Shift+F	地图浏览	*探索地图视图*
缩放到选中的区域	Ctrl+J	地图浏览	*探索地图视图*
缩放到图层		地图浏览	*探索地图视图*
缩放到原始分辨率 *(100%)*		地图浏览	*探索地图视图*
回到上一视图		地图浏览	*探索地图视图*
下一视图		地图浏览	*探索地图视图*
装饰 ►	Alt+V + D		*Decorating the map*
► 网格...			*Grid*
► 比例尺...			*Scale Bar*
► 图像...			*Image Decoration*
► 指北针...			*North Arrow*
► 标题标注...			*Title Label*
► 版权标注...			*Copyright Label*
► 布局范围...			*Layout Extents*
预览模式 ►			
► 正常			
► 模拟单色			
► 模拟全色盲 *(灰度)*			
► 模拟红色盲 *(无红色)*			

continues on next page

Table 3.4 – continued from previous page

菜单选项	快捷键	工具栏	参考
► 模拟绿色盲 (无绿色)			
► 模拟蓝色盲 (无蓝色)			
显示地图提示		属性	*Display Properties*
新建空间书签...	Ctrl+B	地图浏览	*Bookmarking extents on the map*
显示空间书签	Ctrl+Shift+B	地图浏览	*Bookmarking extents on the map*
显示空间书签管理器			*Bookmarking extents on the map*
刷新	F5	地图浏览	
图层可见性 ►			图层面板
► 显示所有图层	Ctrl+Shift+U		图层面板
► 隐藏所有图层	Ctrl+Shift+H		图层面板
► 显示选中的图层			图层面板
► 隐藏选中的图层			图层面板
► 切换选中图层			图层面板
► 独立切换选中图层			图层面板
► 隐藏未选中的图层			图层面板
面板 ►			面板和工具栏
工具栏 ►			面板和工具栏
切换全屏模式	F11		
切换面板可见性	Ctrl+Tab		
只切换地图	Ctrl+Shift+Tab		

在 Linux KDE 系统中, 面板 ►, 工具栏 ► 和 切换全屏模式在 设置菜单中。

3.1.4 图层

图层菜单提供了大量工具用于创建 新数据源, 添加 它们到工程中或保存修改。使用相同的数据源，您还可以:

- 创建副本用于生成副本的图层，可以在其中修改名称、样式 (符号、标注等)、连接等，副本使用与原始副本相同的数据源。
- 复制和 粘贴图层或组从一个工程到另一个工程作为一个新实例，其属性可以独立修改。至于 * 副本 *，图层仍然基于相同的数据源。
- 或 嵌入图层和组... 来自另一个工程，作为只读副本您无法进行修改 (请参阅从外部工程嵌入图层)

The 图层菜单还包含用于配置、复制或粘贴图层属性的工具 (样式、比例、CRS...)。

Table 3.5: 图层菜单项

菜单选项	快捷键	工具栏	参考
数据源管理器	Ctrl+L	数据源管理器	打开数据
创建图层 ►			新建矢量图层
► 新建地理包图层...	Ctrl+Shift+N	数据源管理器	新建地理包图层

continues on next page

Table 3.5 – continued from previous page

菜单选项	快捷键	工具栏	参考
► 新建 *Shapefile* 图层...		数据源管理器	新建形状文件图层
► 新建 *SpatiaLite* 图层...		数据源管理器	新建 *SpatiaLite* 图层
► 新建临时图层...		数据源管理器	新建临时图层
► 新建网孔图层...		数据源管理器	新建网孔图层
► *New GPX Layer...*		数据源管理器	新建 *GPX* 图层
► 新建虚拟图层...		数据源管理器	创建虚拟图层
添加图层 ►			打开数据
► 添加矢量图层.....	Ctrl+Shift+V	图层管理	从文件加载图层
► 添加栅格图层...	Ctrl+Shift+R	图层管理	从文件加载图层
► 添加网孔图层...		图层管理	加载网孔图层
► 添加分隔文本图层...	Ctrl+Shift+T	图层管理	导入分隔文本文件
► *Add PostgreSQL Layer...*	Ctrl+Shift+D	图层管理	数据库相关工具
► 添加 *SpatiaLite* 图层...	Ctrl+Shift+L	图层管理	*SpatiaLite* 图层
► 添加 *MS SQL Server* 图层...		图层管理	数据库相关工具
► 添加 *Oracle* 空间图层...		图层管理	数据库相关工具
► 添加 *SAP HANA* 空间图层...		图层管理	数据库相关工具
► 添加/编辑虚拟图层...		图层管理	创建虚拟图层
► 添加 *WMS/WMTS* 图层...	Ctrl+Shift+W	图层管理	*Loading WMS/WMTS Layers*
► 添加 *XYZ* 图层...			使用 *XYZ* 瓦片服务
► 添加 *WCS* 图层...		图层管理	*WCS Client*
► *Add WFS / OGC API - Features Layer...*		图层管理	*WFS and WFS-T Client*
► 添加 *ArcGIS REST Server* 图层...		图层管理	使用 *ArcGIS REST* 服务器
► 添加矢量瓦片图层...			使用矢量瓦片服务
► 添加点云图层...			*Working with Point Clouds*
► 添加 *GPX* 图层...			*Introducing GNSS/GPS Data*
嵌入图层和组...			从外部工程嵌入图层
从图层定义文件添加...			图层定义文件
配准工具...			*Georeferencer*
复制样式			保存和共享图层属性
粘贴样式			保存和共享图层属性
复制图层			
粘贴图层/组			
打开属性表	F6	属性	属性表操作
过滤属性表 ►			属性表操作
► 打开属性表 (所选要素)	Shift+F6	属性	属性表操作

continues on next page

Table 3.5 – continued from previous page

菜单选项	快捷键	工具栏	参考
► 打开属性表 (可见要素)	Ctrl+F6	属性	属性表操作
► 打开属性表 (编辑和新建要素)		属性	属性表操作
切换编辑模式		数字化	数字化现有图层
保存图层编辑		数字化	*Saving Edited Layers*
当前编辑 ►		数字化	*Saving Edited Layers*
► 保存所选图层		数字化	*Saving Edited Layers*
► 恢复所选图层		数字化	*Saving Edited Layers*
► 取消所选图层		数字化	*Saving Edited Layers*
► 保存所有图层		数字化	*Saving Edited Layers*
► 恢复所有图层		数字化	*Saving Edited Layers*
► 取消所有图层		数字化	*Saving Edited Layers*
另存为...			从现有图层创建新图层
另存为图层定义文件...			图层定义文件
移除图层/组	Ctrl+D		
创建图层副本			
设置图层可见比例			可见比例选择器
设置图层 *CRS*	Ctrl+Shift+C		图层坐标参照系
将图层的 *CRS* 设置为工程的 *CRS*			投影坐标参照系
图层属性...			• *vector* • *raster* • *mesh* • *point cloud* • *vector tiles*
过滤...	Ctrl+F		查询构建器
标注			*Labels Properties*
在鹰眼图中显示			概览图面板
在鹰眼图中显示所有数据			概览图面板
在鹰眼图中隐藏所有数据			概览图面板

3.1.5 设置

Table 3.6: 设置菜单项

菜单选项	参考
用户配置 ►	使用用户配置文件
► 默认	使用用户配置文件
► 打开当前配置文件夹	使用用户配置文件
► 新建配置...	使用用户配置文件
样式管理器...	样式管理器
自定义投影...	自定义坐标参照系
键盘快捷键...	*Keyboard shortcuts*
界面自定义...	*Customization*
选项...	选项

在 Linux KDE 系统中，您在 设置菜单中会找到更多工具，例如：面板 ►, 工具栏 ► 和 切换全屏模式。

3.1.6 插件

Table 3.7: 插件菜单项

菜单选项	快捷键	工具栏	参考
管理并安装插件...			*The Plugins Dialog*
Python Console	Ctrl+Alt+P	插件	*QGIS Python console*

首次启动 QGIS 时，并非所有核心插件都加载。

3.1.7 矢量

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

> **Note**
>
> The list of tools and their menu can be modified/extended with any Processing algorithms (read *Configuring the Processing Framework*) or by some third-party *plugins*.

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

Table 3.8: 矢量菜单项

菜单选项	快捷键	工具栏	参考
检查几何图形...			*Geometry Checker Plugin*
拓扑结构检查器		矢量	*Topology Checker Plugin*
地理处理工具 ►	Alt+O + G		
► 缓冲区...			*Buffer*
► 裁剪...			*Clip*
► 凸包...			*Convex hull*
► 差集...			*Difference*
► 融合...			融合（*dissolve*）
► 相交...			*Intersection*
► 交集取反...			*Symmetrical difference*
► 联合...			*Union*
► 清除选中的多边形...			*Eliminate selected polygons*
几何图形工具 ►	Alt+O + E		
► 质心...			*Centroids*
► 集合几何图形...			*Collect geometries*

continues on next page

Table 3.8 – continued from previous page

菜单选项	快捷键	工具栏	参考
► 提取顶点...			*Extract vertices*
► 多部件转单部件...			*Multipart to singleparts*
► 多边形转线...			*Polygons to lines*
► 简化...			*Simplify*
► 检查有效性...			*Check validity*
► 德洛内三角网...			*Delaunay triangulation*
► 按数量增密...			*Densify by count*
► 添加几何图形属性...			*添加几何图形属性*
► 线转多边形...			*Lines to polygons*
► 泰森多边形...			*Voronoi polygons*
分析工具 ►	Alt+O + A		
► 线相交...			*Line intersections*
► 平均坐标...			*Mean coordinate(s)*
► 字段基本统计...			*Basic statistics for fields*
► 统计点在多边形中的数量...			*Count points in polygon*
► 距离矩阵...			*Distance matrix*
► 列出唯一值...			*List unique values*
► 最近邻分析...			*Nearest neighbour analysis*
► 计算线总长...			*Sum line lengths*
数据管理工具 ►	Alt+O + D		
► 合并矢量图层...			*Merge vector layers*
► 重投影图层...			*Reproject layer*
► 创建空间索引...			*Create spatial index*
► 按位置连接属性...			*Join attributes by location*
► 分割矢量图层...			*Split vector layer*
研究工具 ►	Alt+O + R		
► 创建网格...			*Create grid*
► 提取图层范围...			*Extract layer extent*
► 范围内随机生成点...			*Random points in extent*
► 多边形内的随机点...			*Random points in polygons*
► 线上的随机点...			*Random points on lines*
► 按位置选择...			*Select by location*
► *Select Within Distance...*			*Select within distance*
► 图层边界内生成随机点...			*Random points in layer bounds*
► 多边形内部的随机点...			*Random points inside polygons*
► 随机选择...			*Random selection*

continues on next page

Table 3.8 – continued from previous page

菜单选项	快捷键	工具栏	参考
► 子集内随机选择...			*Random selection within subsets*
► 生成规则点...			*Regular points*

3.1.8 栅格

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

Note

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

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

Table 3.9: 栅格菜单项

菜单选项	快捷键	工具栏	参考
栅格计算器...			*Raster Calculator*
Align Rasters...			*Align rasters*
分析 ►			
► 坡向...			*坡向*
► 填充无数据区域...			*Fill NoData*
► 网格化 (滑动平均法)...			*Grid (Moving average)*
► 网格化 (数据度规法)...			*Grid (Data metrics)*
► 网格化 (距离倒数乘方法)...			*Grid (Inverse distance to a power)*
► 网格化 (最近邻算法)...			*Grid (IDW with nearest neighbor searching)*
► 边框转换...			*Near black*
► 山体阴影...			*山体阴影*
► 邻近分析 (生成栅格距离图)...			*Proximity (raster distance)*
► 粗糙度...			*Roughness*
► 滤除碎斑...			*Sieve*
► 坡度...			*坡度*
► 地形位置指数 (TPI)...			*Topographic Position Index (TPI)*
► 地形耐用指数 (TRI)...			*Terrain Ruggedness Index (TRI)*
投影 ►			
► 指定投影...			*Assign projection*
► 提取投影...			*Extract projection*

continues on next page

Table 3.9 – continued from previous page

菜单选项	快捷键	工具栏	参考
► 变形 (重投影)...			*Warp (reproject)*
杂项 ►			
► 构建虚拟栅格...			*Build virtual raster*
► 栅格信息...			*Raster information*
► 合并...			*Merge*
► 构建鹰眼图 (金字塔)...			*Build overviews (pyramids)*
► 瓦片索引...			*Tile index*
提取 ►			
► 按范围裁剪栅格...			*Clip raster by extent*
► 按掩膜图层裁剪栅格...			*Clip raster by mask layer*
► 等高线...			*Contour*
转换 ►			
► *PCT 转 RGB*...			*PCT to RGB*
► 栅格矢量化 (栅格转矢量)...			*Polygonize (raster to vector)*
► 矢量栅格化 (矢量转栅格)...			*Rasterize (vector to raster)*
► *RGB 转 PCT*...			*RGB to PCT*
► 转换 (格式转换)...			*Translate (convert format)*

3.1.9 数据库

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

Table 3.10: 数据库菜单项

菜单选项	快捷键	工具栏	参考
离线编辑...	Alt+D + O		离线编辑
► 转换为离线工程...		*Database*	离线编辑
► 同步		*Database*	离线编辑
Query History...		*Database*	*Creating SQL Query Layers*
数据库管理器...		*Database*	*DB Manager Plugin*

首次启动 QGIS 时，并非所有核心插件都加载。

3.1.10 Web

如果启用了所有核心插件，以下就是 *Web* 菜单包含项。如果没有启用 Web 插件，则不会有 *Web* 菜单。

Table 3.11: Web 菜单项

菜单选项	快捷键	工具栏	参考
MetaSearch ►	Alt+W + M		*MetaSearch Catalog Client*
► 元搜索		*Web*	*MetaSearch Catalog Client*
► 帮助			*MetaSearch Catalog Client*

首次启动 QGIS 时，并非所有核心插件都加载。

3.1.11 网孔

网孔菜单提供了操作网孔图层 所需的工具。第三方插件可以将条目添加到此菜单。

Table 3.12: 网孔菜单项

菜单选项	快捷键	工具栏	参考
网孔计算器...			*Mesh Calculator*
重新索引面和顶点			*Reindexing meshes*

3.1.12 数据处理

Table 3.13: 数据处理菜单项

菜单选项	快捷键	工具栏	参考
工具箱	Ctrl+Alt+T		*The Toolbox*
Model Designer...	Ctrl+Alt+G		*The model designer*
历史...	Ctrl+Alt+H		*The history manager*
结果查看器	Ctrl+Alt+R		*Configuring external applications*
原地编辑要素			*The Processing in-place layer modifier*

首次启动 QGIS 时，并非所有核心插件都加载。

3.1.13 帮助

Table 3.14: 帮助菜单项

菜单选项	快捷键	工具栏	参考
帮助	F1	帮助	
API 文档			
插件 ►			
报告问题			
需要商业支持?			
QGIS 主页	Ctrl+H		
检查 *QGIS* 版本			
关于			
QGIS 维护人员			

3.1.14 QGIS

此菜单仅在 macOS 下可用，并包含一些与操作系统相关的命令。

Table 3.15: QGIS 菜单项

菜单选项	快捷键	参考
关于 *QGIS*		
首选项	Cmd+,	选项
隐藏 *QGIS*	Cmd+H	
显示全部		
隐藏其他	Shift+Cmd+H	
退出 *QGIS*	Cmd+Q	

对于其他平台 首选项对应于 设置 ► 选项, 关于 *QGIS* 对应于 帮助 ► 关于和 退出 *QGIS* 对应于:menuselection: 工程 --> 退出 *QGIS* 。

3.2 面板和工具栏

从 视图菜单 (或 设置)，您可以打开和关闭 QGIS 控件 (面板 ►) 和工具栏 (工具栏 ►)。要 (取消) 激活其中任何一项，请右键单击菜单栏或工具栏，选择所需项。面板和工具栏可以在 QGIS 界面中移动和放置在任何你喜欢的地方。该列表还可以通过激活核心或外部插件 进行扩展。

3.2.1 工具栏

工具栏提供了对菜单中大多数功能的访问，以及与地图交互的其他工具。每个工具栏项都有可用的弹出帮助。将鼠标悬停在所选项上，将显示工具用途的简短描述。

可用工具栏包括:

Table 3.16: QGIS 工具栏

工具栏名称	工具主要参考资料
高级数字化	*Advanced digitizing*
注记	*Annotation Tools*
属性	属性表操作, 常用工具
数据源管理器	管理数据源
Database	*DB Manager Plugin*
数字化	数字化现有图层
GPS	*Live GPS tracking*
GRASS	*GRASS GIS Integration*
帮助	
标注	*The Label Toolbar*
图层管理	打开数据
地图浏览	探索地图视图
网孔数字化	*Editing a mesh layer*
插件	插件
工程	使用工程文件, *Laying out the maps*, 样式库
处理算法	*Configuring the Processing Framework*
栅格	插件
选择	选择要素
形状数字化	*Shape digitizing*
捕捉	设置捕捉容差和搜索半径
矢量	插件
Web	插件, *MetaSearch Catalog Client*

Note

第三方插件可以使用自己的工具或提供其自己的工具栏扩展默认工具栏。

Tip

恢复工具栏

如果不小心隐藏了工具栏，您可以使用 视图 ► 工具栏 ►(或 设置 ► 工具栏 ►) 将其恢复。如果由于某种原因，工具栏 (或任何其他控件) 完全从界面中消失，您可以在恢复初始 *GUI* 中找到恢复它的提示。

3.2.2 面板

QGIS 提供了许多面板。面板是特殊控件，可以与之交互的 (选择选项、复选框、填充值…) 执行更复杂的任务。

以下是 QGIS 提供的默认面板列表:

Table 3.17: QGIS 面板

面板名称	快捷键	参考
高级数字化	Ctrl+4	*The Advanced Digitizing panel*
浏览器	Ctrl+2	浏览器面板
浏览器 (2)		浏览器面板
调试/开发工具	F12	调试/开发工具面板
高程剖面		*Elevation Profile View*
几何图形验证		*Digitizing Properties*
GPS 信息	Ctrl+0	*Live GPS tracking*
GRASS 工具		*GRASS GIS Integration*
图层顺序	Ctrl+9	图层顺序面板
图层样式	Ctrl+3	图层样式面板
图层	Ctrl+1	图层面板
日志信息		日志消息面板
鹰眼图	Ctrl+8	概览图面板
工具箱		*The Toolbox*
结果查看器		*The Toolbox*
捕捉和数字化选项		设置捕捉容差和搜索半径
空间书签管理器	Ctrl+7	*Bookmarking extents on the map*
统计	Ctrl+6	统计摘要面板
时态控制		时态控制面板
瓦片比例		图块集
撤消/重做	Ctrl+5	撤消/重做面板
顶点编辑器		*The Vertex Editor Panel*

3.3 状态栏

状态栏为您提供有关地图视图和已处理或可用操作的一般信息，并提供管理地图视图的工具。

3.3.1 搜索栏

在状态栏的左侧，搜索栏是一个快速搜索控件，可帮助您查找并运行 QGIS 中的任何功能或选项：

1. 在文本控件中单击以激活搜索栏，或按 `Ctrl+K`。
2. 键入与您要查找的工程相关联的文本 (名称、标签、关键字等)。默认情况下，将启用搜索过滤的返回结果，但您可以通过在文本前面加上搜索过滤 前缀，将搜索限制在指定范围内，即键入 `l cad` 将仅返回名称包含 `cad` 的图层。

   在访问搜索控件时，还可以双击显示的菜单来选择过滤。
3. 根据项类型，单击搜索结果执行相应的操作。

> **Tip**
>
> **将查找限制为活动图层的特定字段**
>
> 默认情况下，使用” 活动图层要素” 过滤 (`f`) 的搜索将遍历图层的整个属性表。您可以使用 `@` 前缀将搜索限制到特定字段。例如 `f @name sal` 或 `@name sal` 只返回其”name” 属性包含’sal’ 的要素。书写时，文本自动补全处于活动状态，建议使用 `Tab` 键。
>
> 可以从图层 字段选项卡对查询的字段进行更高级的控制。有关详细信息，请参阅*Fields Properties* 。

搜索是使用线程处理的，因此即使安装了慢速搜索过滤器，其结果也始终尽快可用。它们也会在遇到过滤时立即出现，这意味着，例如文件搜索过滤会在扫描文件树时逐个显示结果。这确保了用户界面始终

响应，即使存在非常慢的搜索过滤 (例如，使用在线服务的过滤)。

Note

Nominatim 定位器工具的行为可能与 OpenStreetMap Nominatim 使用规则 不同 (无自动补全搜索、获取结果延迟等)。

Tip

快速访问搜索的配置

单击状态栏上搜索控件内的 图标，以显示可使用的过滤列表，以及打开 设置 ► 选项... 的 搜索标签页的 配置条目

3.3.2 汇报行为

在搜索栏旁边的区域中，将在需要时显示已执行操作的摘要 (例如选择图层中的要素、移除图层、平移距离和方向) 或悬停在工具的详细描述 (并非所有工具)。

如果需要执行较长时间的操作，例如在栅格图层中收集统计信息、执行处理算法或在地图视图中渲染多个图层，状态栏中将显示进度条。

3.3.3 控制地图画布

坐标选项显示鼠标的当前位置，在地图视图中移动时跟随鼠标。可以在 工程 ► 工程属性... ► 常规选项卡中设置单位 (和精度)。单击文本框右侧的小按钮，可以在坐标选项和 范围选项之间切换，该选项以地图单位显示地图视图当前左下角和右上角的坐标。

在显示坐标旁边，您会看到 比例显示，它显示地图视图的比例。有一个比例选择器，允许您在预定义和自定义比例 之间进行选择。

在显示比例的右侧，按下 按钮锁定比例，以使用放大镜放大或缩小画布。放大镜允许在不更改地图比例的情况下放大地图，从而更轻松准确调整标注和符号的位置。放大级别以百分比表示。如果 放大镜的级别为 100%，当前地图不会被放大，即以相对于显示器分辨率 (DPI) 的标准比例渲染。默认放大倍率值可以在 设置 ► 选项 ► 渲染 ► 渲染行为中定义，这对于高分辨率屏幕放大小符号非常有帮助。此外，设置中的 设置 ► 选项 ► 画布和图例 ► *DPI* 可以方便控制 QGIS 是遵循每个显示器的物理 DPI 还是使用整体系统逻辑 DPI。

在放大镜工具右侧，可以为地图视图定义当前顺时针方向旋转角度，单位为度。

在状态栏右侧， 渲染复选框可用于暂停地图视图渲染 (请参阅章节*Controlling map rendering*) 。

在 渲染功能的右侧，可以找到显示当前工程 CRS 的 *EPSG:* 代码按钮。单击此按钮将打开 工程属性对话框，并允许重投影地图视图或调整任何其他工程属性。

Tip

计算地图画布的正确比例

启动 QGIS 时，默认的 CRS 为 `WGS 84 (EPSG 4326)`，单位为度。这意味着 QGIS 将按照指定的度数解释图层中的任何坐标。要获得正确的比例值，可以在 工程 ► 工程属性... 下的 常规选项卡中手动更改此设置 (例如为米)，或者您可以使用上面看到的 EPSG: 代码 图标。在后一种情况下，单位设置为工程投影指定的单位 (例如 `+units=us-ft`)。

请注意，启动时的 CRS 选项可以在 设置 ► 选项 ► *CRS* 处理中设置。

3.3.4 实时信息

点击 信息 按钮打开 日志信息面板，其中包含有关底层进程的信息 (QGIS 启动、插件加载、处理工具等)

CHAPTER 4

浏览器面板

- 从浏览器打开或运行资源
- 浏览器面板顶级条目
  - 收藏夹
  - 空间书签
  - 工程主目录
  - 驱动器和文件系统
  - 数据类型
  - 数据库条目
  - 瓦片和 *Web* 服务
  - 工程和支持的文件类型

QGIS 浏览器面板是浏览、搜索、检查、复制和加载 QGIS 资源的绝佳工具。只有 QGIS 知道如何处理的资源才会显示在浏览器中。

使用浏览器面板，您可以搜索、检查和添加数据，如浏览器面板 所述。此外，浏览器面板还支持拖放许多 QGIS 资源，例如工程文件、Python 脚本、处理脚本和处理模型。

Python 脚本、处理脚本和处理模型也可以在外部编辑器和图形建模器中打开进行编辑。

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

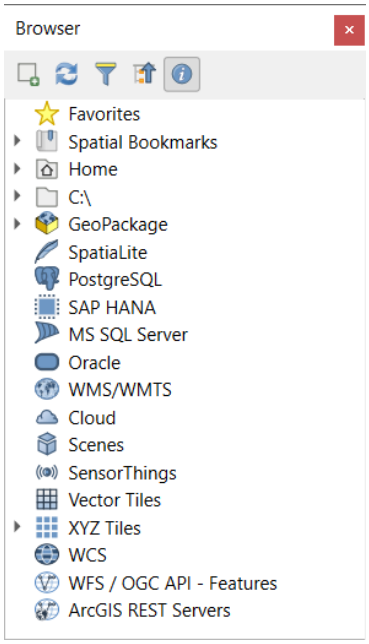


Fig. 4.1: 浏览器面板

浏览器面板 (Fig. 4.1) 构成一个可扩展的层次结构，其中包含一些固定的顶级条目，用于组织浏览器处理的资源。通过单击条目名称左侧的 展开节点条目，单击 可折叠分支，全部折叠 按钮可折叠所有顶级条目。

在 设置 ► 界面自定义中可以禁用资源。例如，如果您不想在浏览器中显示 Python 脚本，可以取消勾选 浏览器 ► *py* 条目；如果想在浏览器中排除您的主文件夹，可以取消勾选 浏览器 ► *special:Home* 条目。

过滤器 (过滤浏览器) 可用于根据条目名称 (层次结构中的目录条目和节点条目) 进行搜索。使用过滤器文本字段旁边的 选项 下拉菜单，可以

- 切换 区分大小写搜索
- 将 过滤模式语法设置为
  - 正常
  - 通配符
  - 正则表达式

属性控件显示有关条目或资源的实用信息，可以使用 启用/禁用属性控件 按钮启用或禁用。启用后，它将在浏览器面板的底部打开，如 Fig. 4.2 所示。

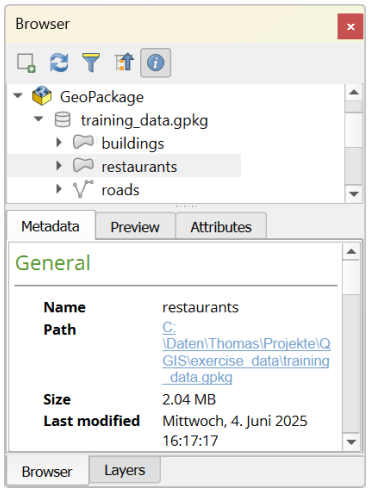


Fig. 4.2: 属性控件

通过在 *视图* ► *面板* 激活 *浏览器 (2)* 面板，可以打开第二个浏览器面板。当在位于浏览器层次结构的不同分支中的资源之间复制图层时，使用两个浏览器面板非常有用。

4.1 从浏览器打开或运行资源

在浏览器面板可以完成很多工作

- 通过双击、拖动到地图画布或单击 添加选中的图层 按钮 (选择图层后)，将矢量、栅格和网孔图层添加到地图
- 双击或拖动到地图画布，运行 Python 脚本 (包括处理算法)
- 双击或拖动到地图画布，运行模型
- 使用上下文菜单从 QGIS 工程文件中 *提取符号...*
- 使用默认应用程序打开文件 (在上下文菜单中的 *打开 <file type> 的外部编辑器...*)。例如：HTML 文件、电子表格、图像、PDF、文本文件...
- 复制条目
- 重命名和删除 (多个) 图层 (上下文菜单: *管理器* ►)
- 打开文件资源管理器窗口，直接选择文件 *在文件资源管理器中显示*

以下列出了在顶级条目下排序的不同资源组的特定于资源的操作。

4.2 浏览器面板顶级条目

4.2.1 收藏夹

经常使用的文件系统位置可以标记为收藏夹，您标记的内容将显示在此处。

除了 *主目录* 中描述的操作外，上下文菜单还允许您 *重命名收藏条目...* 和 *移除收藏条目*。

4.2.2 空间书签

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

Table 4.1: available actions for spatial bookmarks

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

4.2.3 工程主目录

工程文件保存后可用，工程主目录条目即为一个文件夹，其中包含可在当前工程使用的数据和其他内容(脚本、模型、文本等)，并显示在浏览器面板中，允许您快速访问工程的数据和其他文件。

默认为工程文件夹，但可以通过 工程 ► 工程属性... ► 通用 ► 工程主目录选项进行更改，或者右键单击浏览器面板的 工程主目录项，然后选择 设置工程主目录... 。在 QGIS 工程未与数据集一起存储在组织的'工程' 根文件夹中的情况下，自定义该文件夹特别有用。

4.2.4 驱动器和文件系统

浏览器面板的下一项取决于所使用的操作系统，并涉及其文件系统的顶级条目。

主要是:

- 主文件夹，指向当前用户的主文件夹
- 在基于 Unix 的计算机上，是根 / 文件夹
- 连接的驱动器、本地驱动器或网络驱动器。根据操作系统的不同，直接通过 (如 `C:\`, `D:\`) 或 `/Volumes` 条目列出。

从这些每个文件夹或驱动器的上下文菜单中，您可以:

- 刷新内容
- 创建 新建 ► 该子项是 目录、地理包或 ESRI 形状文件格式数据集
- 隐藏目录 (从浏览器中隐藏)
- 设置颜色: 自定义文件夹图标的颜色，有助于复杂文件夹结构时的快速浏览器导航
- 启用 扫描:
  - 监控更改: 允许手动控制是否应该监控和自动更新特定目录。此设置适用于所选目录和所有子目录。这意味着，如果您知道没有问题，可以手动选择监控网络驱动器，或者手动选择不监控由于其他原因不希望监控的大型目录。默认不会自动监控远程或网络驱动器。
  - 快速扫描此目录
- 在文件管理器中打开目录 (打开目录...)
- 在终端窗口中打开目录 (在终端打开...)
- 检查 属性... 或父 目录属性...

4.2.5 数据类型

Table 4.2: Available actions for plain layer files

Level and Context menu	图层的类型 Raster (*Reference*)	Vector (*Reference*)	Point clouds (*Reference*)	Mesh (*Reference*)
File				
删除文件 *<dataset name>*	☑	☑		
在文件资源管理器中显示	☑	☑	☑	☑
管理	☑	☑	☑	☑
File Properties…	☑	☑	☑	☑
Open <file type> Externally…	☑			
Layer				
添加图层到工程	☑	☑	☑	☑
导出图层	☑	☑		
Layer Properties…	☑	☑	☑	☑
Open with Data Source Manager…	☑	☑		
Execute SQL…		☑		

4.2.6 数据库条目

根据所使用的操作系统和安装的驱动器，您可能有权访问要在 QGIS 中使用的不同数据库类型。下面列出了数据集树每个级别的上下文菜单的不同条目。

Table 4.3: Available actions for database entries

Level and Context menu	数据库的类型 GeoPackage ([1])	SpatiaLite	PostgreSQL	SAP HANA	MS SQL Server	Oracle
Top menu						
创建与现有数据库的 新建连接…	☑	☑	☑	☑	☑	☑
创建数据库…	☑	☑				
创建数据库和图层…	☑					
保存连接…详细信息到文件			☑	☑	☑	
加载连接…			☑	☑	☑	
Connection / Database						
刷新连接			☑	☑	☑	☑
编辑连接…设置			☑	☑	☑	☑
删除连接			☑	☑	☑	☑
移除连接	☑	☑	☑	☑	☑	☑

continues on next page

Table 4.3 – continued from previous page

Import Vector Layer…	☑	☑	☑	☑	☑
New Field Domain ► New Range Domain	☑				
New Field Domain ► New Coded Values Domain	☑				
New Field Domain ► New Glob Domain	☑				
删除 <*database_name*>	☑	☑			
压缩数据库 (*Vacuum*)	☑				
Create a 新建架构…			☑	☑	☑
创建 新建表格…	☑	☑	☑	☑	
执行 *SQL*…查询	☑	☑	☑	☑	
Schema					
刷新架构			☑	☑	☑
架构运算 ► 重命名架构…			☑	☑	☑
架构运算 ► 删除架构…			☑	☑	☑
创建 新建表格…			☑	☑	
执行 *SQL*…查询			☑	☑	
Table / Layer					
表运算 ► 重命名表格…			☑	☑	☑
表运算 ► 快速清空表格…			☑		☑
执行 *SQL*…查询	☑	☑	☑		
导出图层 ► 到文件…	☑	☑	☑	☑	☑
管理 ► 重命名图层 <*layer_name*>…	☑	☑			☑
管理 ► 删除图层 <*layer_name*>…	☑	☑	☑	☑	☑
管理 ► 删除所选图层	☑	☑	☑	☑	☑
管理 ► 添加图层到工程	☑	☑	☑	☑	☑
管理 ► 添加选中的图层到工程	☑	☑	☑	☑	☑
Manage ► Move to Another Schema			☑		☑
打开 图层属性…对话框	☑	☑	☑	☑	☑
打开 文件属性…对话框	☑				
Open with Data Source Manager	☑	☑			
Fields					
添加新字段…	☑	☑	☑	☑	
Field					
设置字段域	☑				
Rename Field ([2])	☑		☑		☑

continues on next page

Table 4.3 – continued from previous page

设置别名…	☑			
设置注释…	☑		☑	
删除字段…	☑	☑	☑	☑

Note

当矢量层加载到 QGIS 中时，将自动检测具有 *Field Domains* 的字段（例如在 geoppackage 或 ESRI File Geodatabase 中定义的字段）。这些域是数据库级别的约束，这意味着它们由数据库本身强制执行，并应用于不同的应用程序，而不仅仅是 QGIS。

Importing vector layers into databases using Browser panel

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

You can start the import in two ways:

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

Note

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

[1] 不同的条目也可以用于某些 GDAL 支持的 矢量文件格式，如 ESRI 文件地理数据库、FlatGeobuf、GeoParquet、NetCDF 等可以并存时。

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

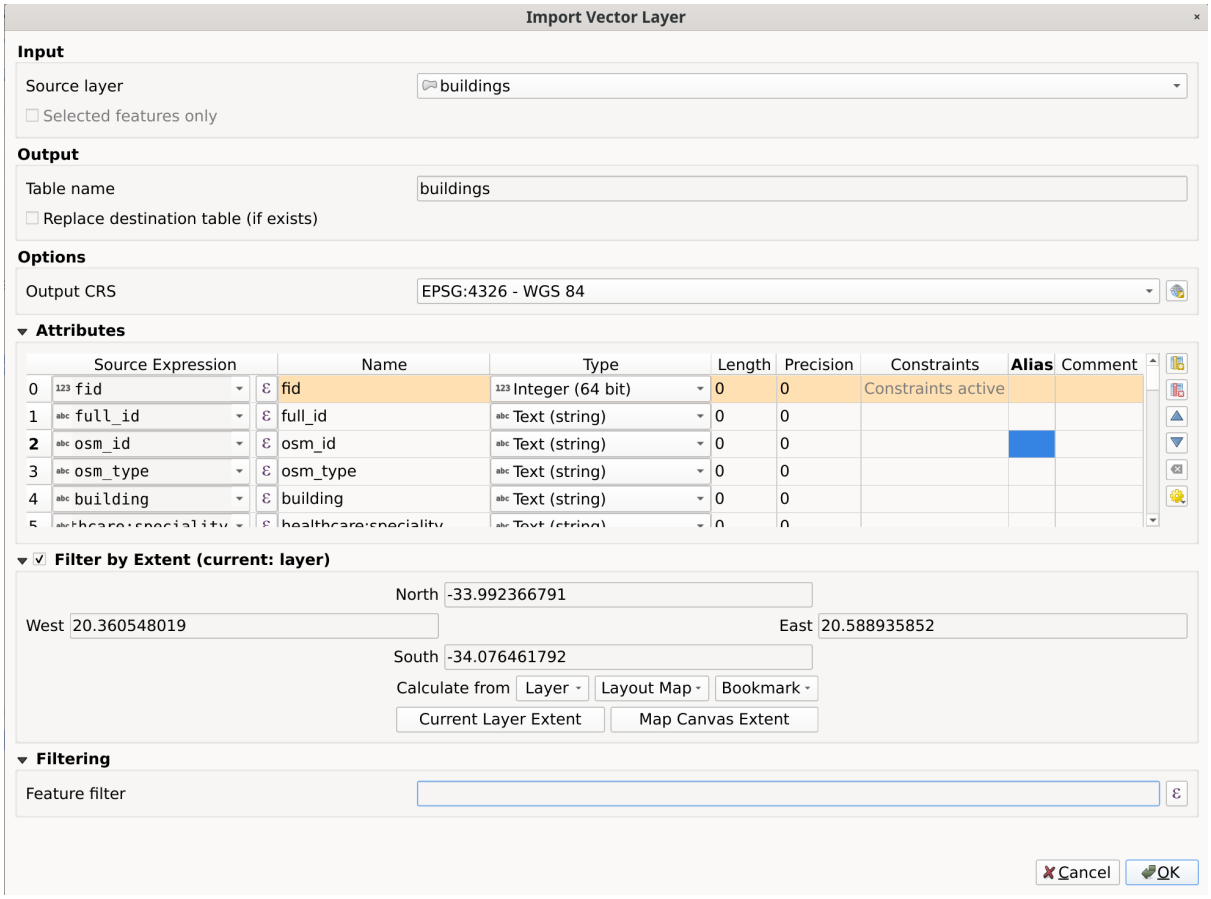


Fig. 4.3: Import Vector Layer dialog

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

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

   You can use buttons on the right side to:

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

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

4.2.7 瓦片和 Web 服务

Table 4.4: Available actions for web services (1)

Level and Context menu	服务类型 WMS / WMTS (参考)	XYZ Tiles (参考)	WCS (参考)	WFS / OGC API - Features (参考)	ArcGIS REST Servers (参考)
Top menu					
创建 新建连接…	☑	☑	☑	☑	☑
保存连接…详细信息到文件	☑	☑	☑	☑	☑
加载连接…	☑	☑	☑	☑	☑
Connection					
刷新连接	☑	☑	☑	☑	☑
编辑连接…设置	☑	☑	☑	☑	☑
删除连接	☑	☑	☑	☑	☑
移除连接	☑	☑	☑	☑	☑
在 Web 浏览器 显示服务信息					☑
Table / Layer					
导出图层 ► 到文件…	☑	☑	☑	☑	☑
添加图层到工程	☑	☑	☑	☑	☑
Add Filtered Layer to Project…					☑
打开 图层属性…对话框	☑	☑	☑	☑	☑
在 Web 浏览器 显示服务信息					☑

Table 4.5: Available actions for web services (2)

Level and Context menu	服务类型 Vector Tiles (参考)	Scene (参考)	Cloud (参考)	Sensor-Things (参考)	STAC (*Reference*)
Top menu					
创建 新建连接…		☑	☑	☑	☑
创建 新建通用连接…	☑				
创建 新建 *ArcGIS* 矢量瓦片服务器连接…	☑				
Save Connections… details to a file	☑	☑	☑	☑	☑
加载连接…	☑	☑	☑	☑	☑
Connection					
编辑连接…设置	☑	☑	☑	☑	☑
删除连接	☑	☑	☑	☑	
Delete Connection…	☑	☑	☑	☑	☑
Refresh					☑
Details…					☑
Table / Layer					
导出图层 ► 到文件…				☑	
添加图层到工程	☑	☑		☑	
打开 图层属性…对话框	☑	☑		☑	
Refresh					☑
Download assets…					☑
Details…					☑

4.2.8 工程和支持的文件类型

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

文件类型	上下文菜单
QGIS 工程文件 (.qgz, .qgs)	打开工程
	提取符号… (打开样式管理器，允许您将符号导出为 XML 文件，添加到默认样式或者导出到 PNG/SVG 格式)
	文件属性…
QGIS 图层定义 (.qlr)	导出图层 ► 到文件…
	添加图层到工程
	图层属性…
处理模型 (.model3)	*Run Model*…
	编辑模型…
打印 Composer 模板 (.qpt)	由模板新建布局
Python 脚本 (.py)	运行脚本…
	使用外部编辑器打开

CHAPTER 5

QGIS 配置

QGIS 是高度可配置的。通过 设置菜单，提供了以下不同工具：

- 样式管理器...: 创建和管理符号，样式和颜色渐变.
- 自定义投影...: 创建您自己的坐标参照系.
- 键盘快捷键...: 定义您自己的一组键盘快捷键. 此外，工程属性 可以在每个 QGIS 会话期间替换它们 (可在 工程菜单下访问).
- 界面自定义...: 配置应用程序界面，隐藏您可能不需要的对话框或工具.
- 选项...: 设置全局选项 以应用于软件的不同领域. 这些首选项保存在当前用户配置 设置中，并在使用此配置文件打开新工程时默认应用.

5.1 选项

可以使用 选项对话框选择 QGIS 的一些基本选项。选择菜单选项 设置 ► 选项，您可以根据需要修改选项，某些更改可能需要重新启动 QGIS 才能生效。

下面介绍了可以自定义选项的选项卡。

> **Note**
>
> **插件可以在选项对话框中嵌入它们的设置**
>
> 虽然以下只展示了核心设置，但请注意，通过*installed plugins* 在标准选项对话框中插件可展现自己的选项，并以此扩展列表。这避免了每个插件都有自己的配置对话框，并为它们提供额外的菜单项...

5.1.1 通用设置

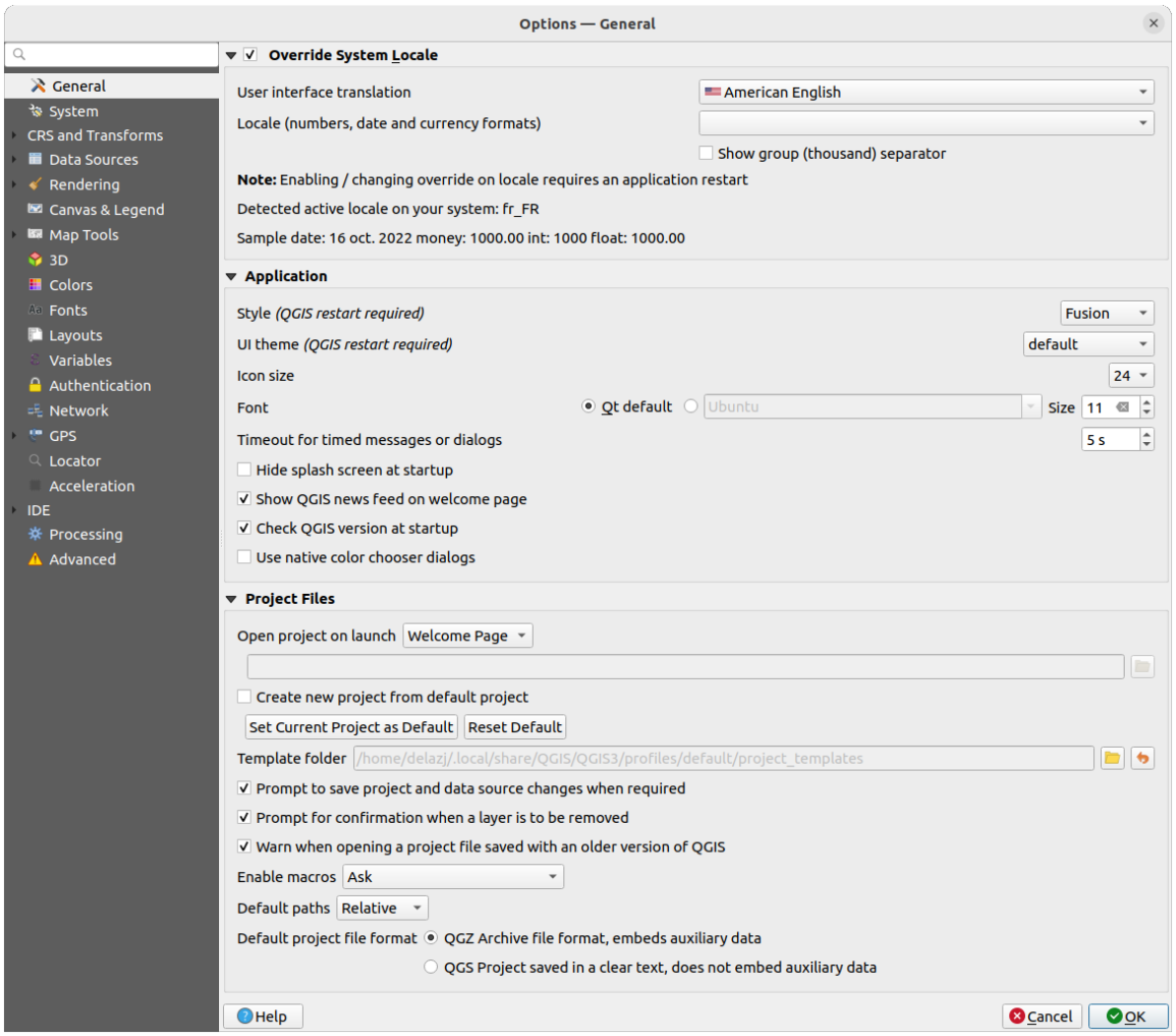


Fig. 5.1: 通用设置

本地化语言环境

默认情况下，QGIS 依赖于您的操作系统配置来设置语言和控制数值。启用该组允许您自定义自己的行为习惯。

- 从 *用户界面语言* 选择要应用于 GUI 的语言
- 在 *本地化 (数字、日期和货币格式)* 选择输入和呈现日期和数值的语言系统
- *显示数字分组 (千位) 分隔符*

框架底部会显示选中设置的摘要以及这些设置的作用。

应用软件

- 选择 *样式 (重启 QGIS 生效)*，即窗口控件在对话框中的外观和位置。可能的值取决于您的操作系统。
- 定义 *用户界面主题 (重启 QGIS 生效)* . 可以是'default', 'Night Mapping' 或'Blend of Gray'
- 定义 *图标大小*
- 定义 *字体和 字号*. 字体可以是 *Qt 默认*或用户从操作系统安装的字体进行选择

- 更改 限时信息或对话框超时
- 启动时隐藏初始界面
- 在欢迎页面显示 *QGIS* 动态资讯: 在欢迎页面上显示精选的 QGIS 动态消息，让您直接了解项目新闻 (用户/开发人员会议日期和摘要、社区调查、发布公告、各种提示...)
- 打开 *QGIS* 软件时检查更新如果发布了新版本，将在启动软件时检查并通知您
- 使用操作系统颜色选择对话框 (参见颜色选择器)

工程文件

- 启动时打开工程为
  - ' 欢迎页面' (默认): 可以显示用户配置 的动态资讯、工程模板和最近工程 (带有缩略图)。默认情况下不打开任何工程。
  - ' 新建': 基于默认模板打开一个新工程
  - ' 最近': 重新打开上次保存的工程
  - ' 特定': 打开一个特定的工程，使用 ... 按钮定义默认情况下要使用的工程文件。
- 由默认工程创建新工程. 您可以选择 设置当前工程为默认工程或 重置为默认. 可以浏览文件并定义目录，并在其中找到用户定义的工程模板。如果您先激活 由默认工程创建新工程, 然后将工程保存在工程模板文件夹中，这将被添加到 工程 ► 由模板新建列表中。
- 需要时提示保存工程和数据源更改情况避免丢失所做的更改。
- 移除图层时提示确认
- 当打开旧版 *QGIS* 格式的工程文件时提出警告. 您始终可以打开使用较旧版本 QGIS 创建的工程，但一旦保存该工程，尝试使用较低版本打开可能会失败，这是因为该版本没有相对于高版本可用的功能。
- *Enable project's embedded Python code* . This option handles execution of macros that are written to perform an action on project events, as well as custom Python functions to be used as expressions. You can choose between 'Never', 'Ask', 'For this session only' and 'Always (not recommended)'.
- *Default paths*: defines whether paths to files and layers used in new projects are stored as 'Absolute' or 'Relative' to the project file. This setting can be overwritten at the project level.
- 默认工程文件格式
  - *QGZ*: 存档文件格式，嵌入辅助数据 (请参见辅助数据)
  - *QGS*: 工程文件以明码文本保存，未嵌入辅助数据: 辅助数据存储在单独的 `.qgd` 文件，工程文件则在同一目录存储不同后缀名的同名文件。

5.1.2 系统设置

SVG 路径

添加或移除 可伸缩矢量图形 *(SVG)* 符号的搜索路径. 这些 SVG 文件可用于符号化、标注要素或装饰地图构图。

Also read 远程或嵌入式文件选择器 for different ways to refer to svg files in a QGIS path.

插件路径

添加或移除 附加 *C++* 插件库的搜索路径.

文档路径

Add or Remove *Documentation Path(s)* to use for QGIS help. By default, a link to the official online User Manual corresponding to the version being used is added (i.e., `https://docs.qgis.org/$qgis_short_version/$qgis_locale/docs/user_manual/`). You can however add other links and prioritize them from top to bottom: each time you click on a *Help* button in a dialog, the topmost link is checked and if no corresponding page is found, the next one is tried, and so on.

Note

文档仅针对 QGIS 长期发行版 (LTR) 进行版本控制和翻译，这意味着如果您运行的是常规版本 (如 QGIS 3.0)，帮助按钮默认打开下一版 LTR 手册页面 (即 3.4 LTR)，其中可能包含较新版本 (3.2 和 3.4) 的功能描述。如果没有可用的 LTR 文档，则使用较新和开发版本功能描述的 测试文档。

设置

如果进行了任何自定义 设置，它可以帮助您 重置用户界面为默认设置 *(重启 QGIS 生效)* 。

环境

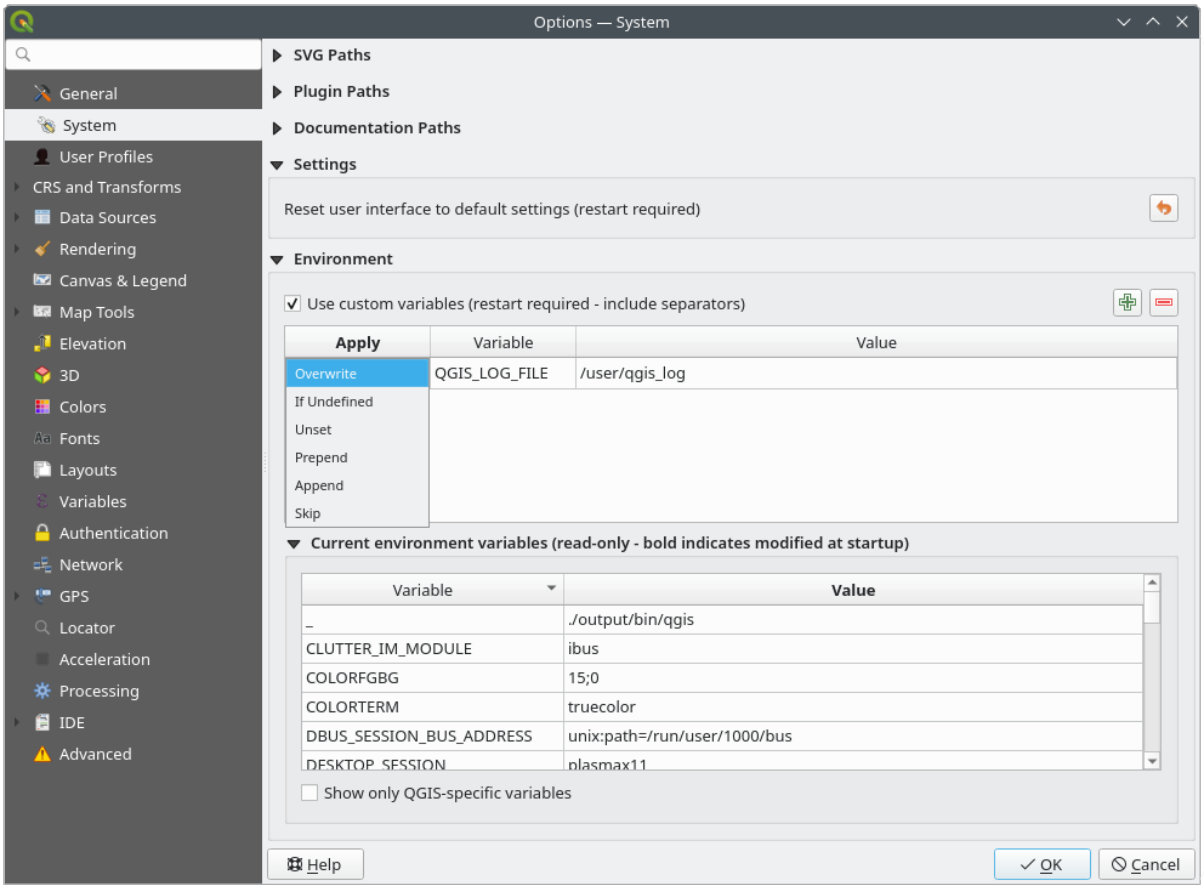


Fig. 5.2: System environment variables

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

Check *Use custom variables (restart required - include separators)* and you can Add and Remove environment variables. For each new item, you can configure a *Variable* name, its *Value* and the *Apply* method to use, among which:

- *Overwrite*: replace any preexisting value of the variable

- *If undefined*: use this value for the variable if not already defined at a higher level (e.g., OS or application levels)
- *Unset*: remove the variable from the environment (the *Value* parameter is not used)
- *Prepend*: prepend the value to the preexisting value of the variable
- *Append*: append the value to the preexisting value of the variable
- *Skip*: the item is kept in the list for future reference but unused

Already defined environment variables are displayed in *Current environment variables*, and it's possible to filter them by activating *Show only QGIS-specific variables*.

5.1.3 User Profiles Settings

> **Note**
>
> For more information on how to manage user profiles, please read the dedicated section at 使用用户配置文件.

5.1.4 CRS and Transforms Settings

> **Note**
>
> 有关 QGIS 如何处理图层投影的更多信息，请阅读使用投影 专区。

CRS Handling

In the *CRS Handling* tab you can configure which CRS will be used for a new project or layer.

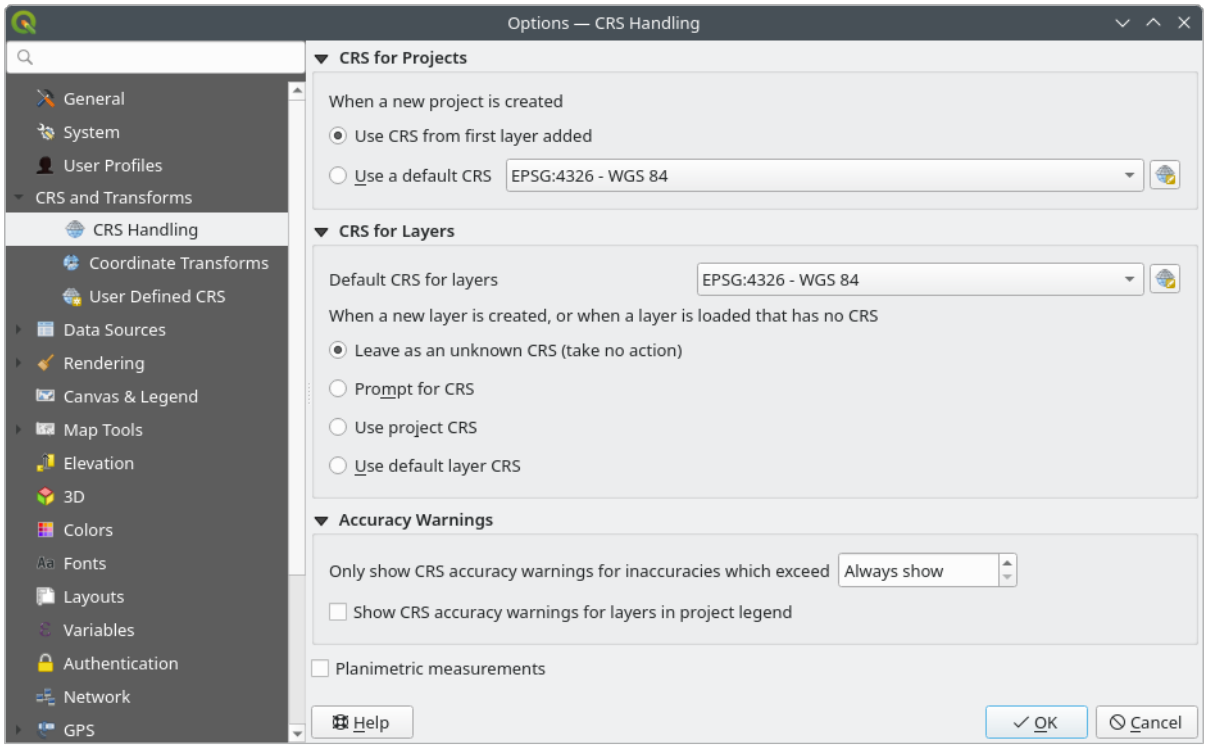


Fig. 5.3: CRS 设置

CRS for Projects

有个选项可以自动设置新建工程的 CRS：

- 使用添加的第一个图层的 *CRS*: 工程的 CRS 将设置为加载到其中的第一个图层的 CRS
- 使用默认 *CRS*: 预选 CRS 将默认应用于新建工程，并在向工程添加图层时保持不变。

该选项将保存供后续 QGIS 会话使用，工程的坐标参照系仍然可以从 工程 ► 工程属性... ► *CRS* 选项卡进行重新定义。

CRS for Layers

默认图层 *CRS*: 选择创建图层时要使用的默认 CRS

还可以定义创建新图层或加载不含 CRS 的图层时要执行的操作。

- 保留未知 *CRS (无动作)*
- 提示选择 *CRS*
- 使用工程 *CRS*
- 使用默认图层 *CRS*

Accuracy Warnings

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

Show CRS accuracy warning for layers in project legend: If checked, any layer with a CRS with accuracy issues (i.e., a dynamic crs with no coordinate epoch available, or a CRS based on a datum ensemble with inherent inaccuracy exceeding the user-set limit) will have the warning icon in the *Layers* panel reflecting that it is a low-accuracy layer.

This is designed for use in engineering, BIM, asset management, and other fields where inaccuracies of meter/submeter level are potentially very dangerous or expensive!

Planimetric measurements: sets the default for the *planimetric measurements* property for newly created projects.

Coordinate Transforms

The *Coordinate Transforms* tab helps you set coordinate transformations and operations to apply when loading a layer to a project or reprojecting a layer.

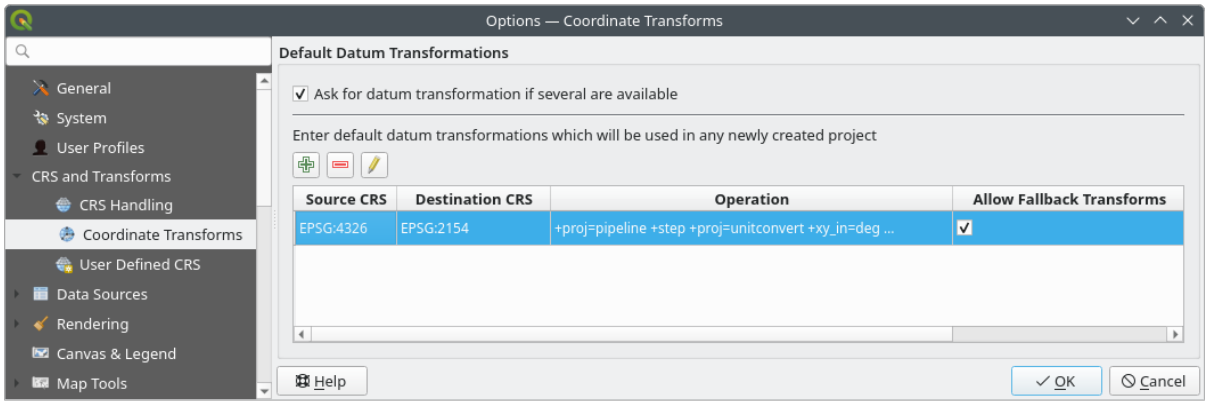


Fig. 5.4: Transformations settings

Default Datum Transformations

Here you can control whether reprojecting layers to another CRS should be:

- 使用 QGIS 默认变换设置自动处理；
- 和/或更多由您通过自定义首选项控制，例如：
  - 存在多个时询问基准面变换
  - 默认情况下要应用的预定义基准变换列表。更多信息，请参阅基准变换 。

You can Add, Remove or Edit transformations, which will be used in any newly created project.

User Defined CRS

The *User Defined CRS* tab helps you to define a custom CRS which must conform to a WKT or Proj string format.

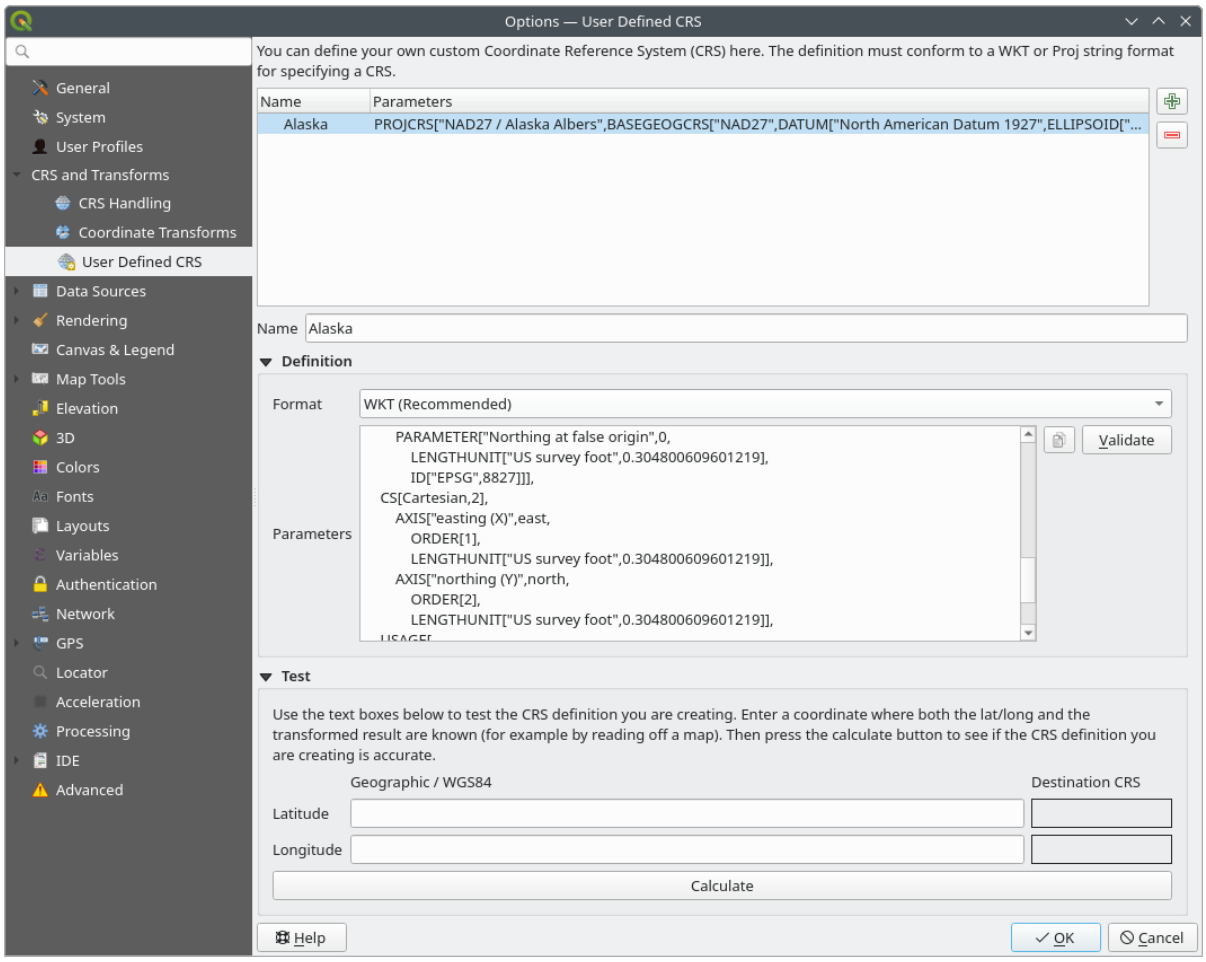


Fig. 5.5: User Defined CRS

Set a *Name* and use Add new CRS. If you want to delete an existing one you can use Remove CRS.

Definition

- **格式**
  - WKT (Recommended)
  - Proj String (Legacy - Not Recommended)
- ***Parameters***
  - Copy parameters from an existing CRS.
  - *Validate* tests if your expression is correct.

Test

Here you can test your created CRS definition by Latitude and Longitude. Use a known coordinate to control if your definition is accurate.

5.1.5 Data Sources settings

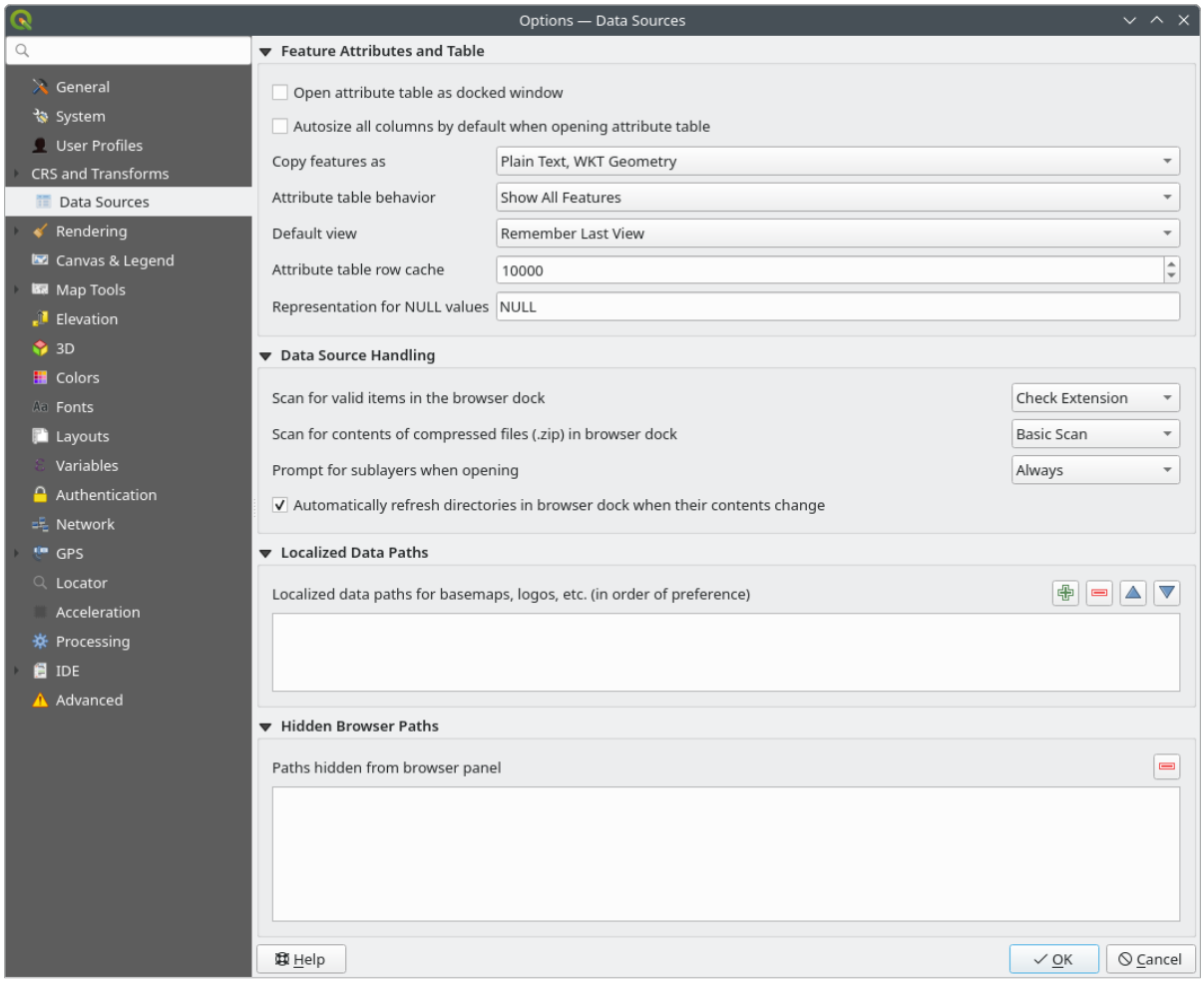


Fig. 5.6: Data Sources settings

要素属性和表格

- *Open attribute table as docked window*
- *Autosize all columns by default when opening attribute table*
- *Copy features as*: defines the format to use for data when pasting features in other applications.
  - *Plain text, no geometry*: attributes are copied as text but the geometry is skipped
  - *Plain text, WKT geometry*: attributes are copied as text and the geometry is returned in WKT
  - *Plain text, WKB geometry*: attributes are copied as text and the geometry is returned in WKB (hex format)
  - *GeoJSON*: attributes and geometry are formatted as GeoJSON data
- 属性表行为 : 默认打开属性表时设置过滤选项。有三种可能: ' 显示所有要素', ' 显示选中的要素' 和' 显示地图上可见的要素' 。
- 默认视图: 定义每次打开属性表的视图模式。它可以是' 记住最后视图', ' 表格视图' 或' 表单视图' 。
- 属性表记录缓存 1,00 . 该行缓存可以保存最后加载的 N 个属性记录，以便更快地处理属性表。关闭属性表时，缓存将被删除。
- 空值表示为. 在此，您可以为包含空值 (NULL) 的数据字段定义一个值。

> **Tip**
>
> **改进大数据属性表的打开速度**
>
> 处理有大量记录的图层时，打开属性表的速度可能会很慢，因为对话框会索取图层中的所有行。将 *属性表行为* 设置为 **显示地图可见要素** 将使 QGIS 在打开表时仅请求当前地图画布中的要素，从而允许快速加载数据。
>
> 请注意，此属性表实例中的数据将始终关联到打开时使用的画布范围，这意味着选择 **显示所有要素** 将不会显示新要素。但是，您可以通过更改画布范围并在属性表选择 **显示地图可见要素** 选项来更新显示的要素集。

数据源处理

- *在浏览器停靠区扫描有效项* . 您可以在' 检查扩展名' 和' 检查文件内容' 之间进行选择。
- *在浏览器停靠区扫描压缩文件 (.zip) 的内容* 定义在查询此类文件时浏览器面板底部的控件信息的详细程度，' 否',' 基本扫描' 和' 深度扫描' 是可能的选项。
- *Prompt for sublayers when opening*. Some rasters support sublayers --- they are called subdatasets in GDAL. An example is netCDF files --- if there are many netCDF variables, GDAL sees every variable as a subdataset. The option allows you to control how to deal with sublayers when a file with sublayers is opened. You have the following choices:
  - '始终'：始终询问 (如果存在子图层)
  - '需要时'：询问图层是否没有波段，但有子图层
  - '从不'：从不提示，不会加载任何内容
  - '全部加载'：从不提示，但加载所有子图层
- *Automatically refresh directories in browser dock when their contents change*: Allows you to manually opt-out of monitoring directories in the *Browser* panel by default (eg, to avoid potential slow down due to network latency).

Localized data paths

可以对任何类型的基于文件的数据源使用本地路径，它们是用于抽象数据源位置的路径列表。例如，如果 `C:\my_maps` 列在本地路径中，则 `C:\my_maps\my_country\ortho.tif` 作为数据源的图层将在工程中保存后使用 `localized:my_country\ortho.tif`.

路径按优先顺序列出，换句话说，QGIS 将首先在第一个路径中查找文件，然后在第二个路径中查找文件，以此类推。

Hidden browser paths

此控件列出了您在浏览器面板 中选择隐藏的所有文件夹，从列表中移除文件夹将使其在 浏览器面板中可用。

GDAL 设置

GDAL is a data exchange library for geospatial data that supports a large number of vector and raster formats. It provides drivers to read and (often) write data in these formats. The *GDAL* tab exposes the drivers for raster and vector formats with their capabilities.

GDAL raster and vector drivers

The *Raster Drivers* and *Vector Drivers* tabs allow you to define which GDAL driver is enabled to read and/or write files, as in some cases more than one GDAL driver is available.

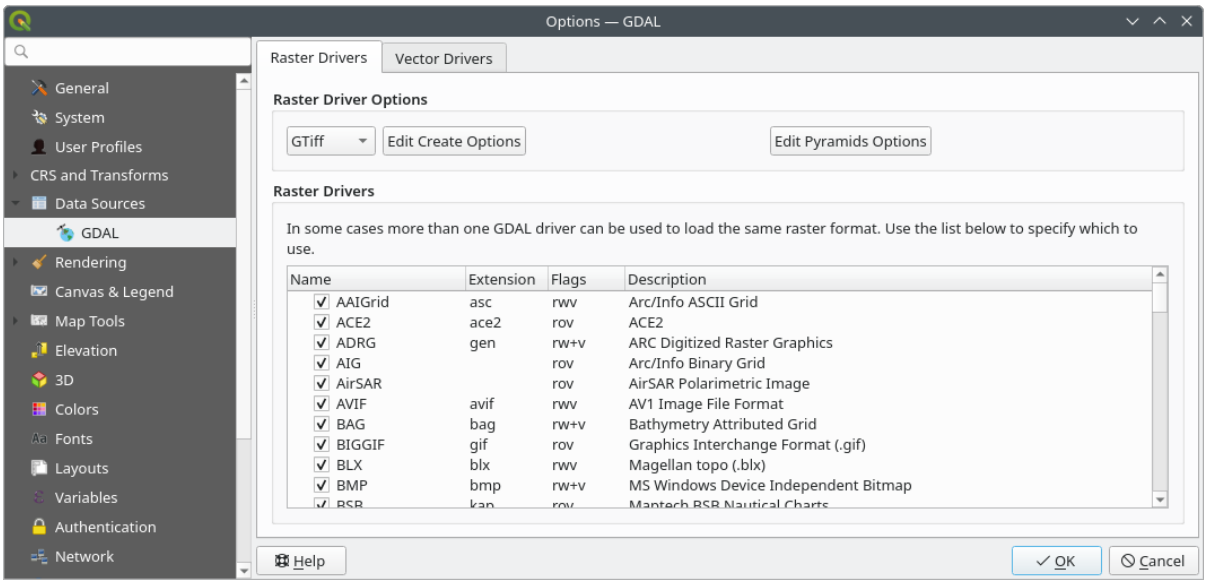


Fig. 5.7: GDAL Settings - Raster drivers

Tip

双击允许读写访问的栅格驱动 (`rw+(v)`) 打开*编辑创建选项* 对话框进行自定义编辑。

Raster driver options

此框架提供了自定义支持读写访问的栅格驱动程序行为的方法：

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

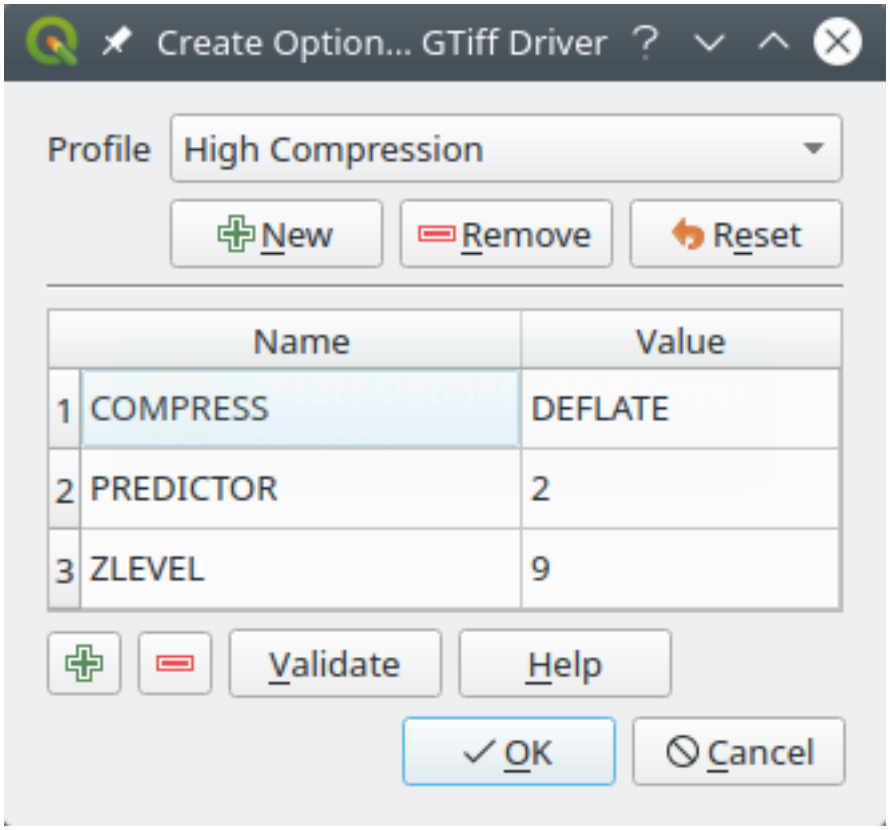


Fig. 5.8: 创建选项配置文件示例 (适用于 GeoTiff)

对话框的上半部分列出了当前配置文件，并允许您添加新配置文件或删除其中任何一个。如果您更改了配置文件的默认参数，也可以将其重置为默认参数。一些驱动程序 (例如 GeoTiff) 有一些可以使用的配置文件样本。

在对话框底部：

- 按钮允许您添加行以填充参数名称和值
- 按钮删除所选参数
- 单击 验证按钮，检查为给定格式输入的创建选项是否有效
- Use the *Help* button to find the parameters to use, or refer to the GDAL raster drivers documentation.

- 编辑金字塔选项

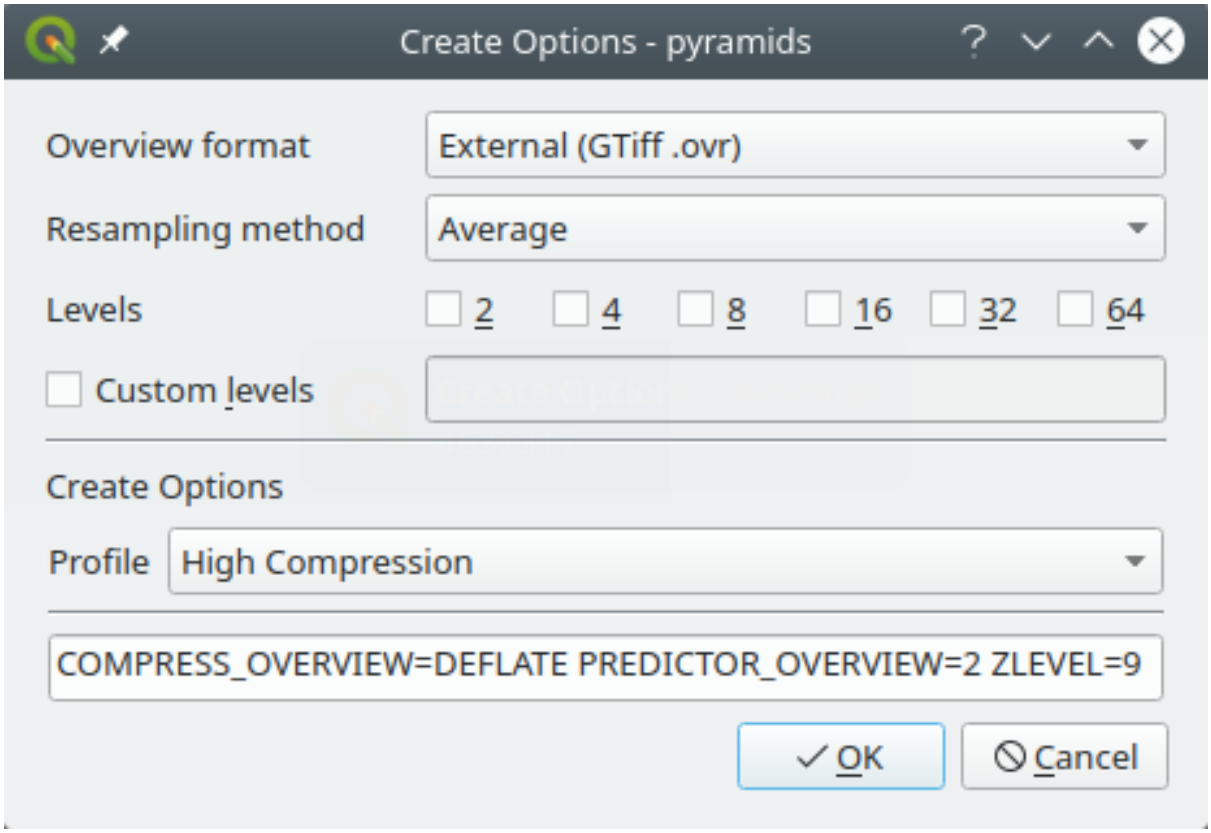


Fig. 5.9: Sample of pyramids profile

5.1.6 渲染设置

The *Rendering* tab provides settings for controlling layers rendering in the map canvas.

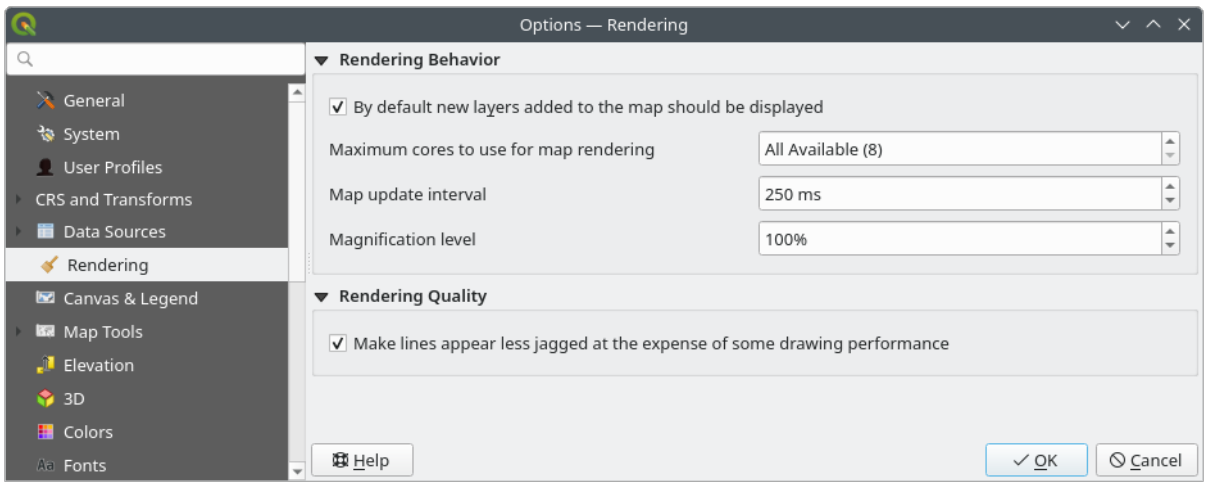


Fig. 5.10: Rendering settings

Rendering Behavior

- 默认显示地图中新添加的图层: 在加载多个图层时，取消勾选此选项会很有用，这可以避免在画布中渲染每个新图层和减慢处理过程

- Set the *Maximum cores to use for map rendering*
- The map canvas renders in the background onto a separate image and at each *Map update interval* (defaults to 250 ms), the content from this (off-screen) image will be taken to update the visible screen representation. However, if rendering finishes faster than this duration, it will be shown instantaneously.
- *Magnification level* (see the *magnifier*)

Rendering Quality

- ☑ 牺牲绘制效率使线的显示更光滑

Vector rendering settings

The *Vector* tab contains specific settings for rendering vector layers.

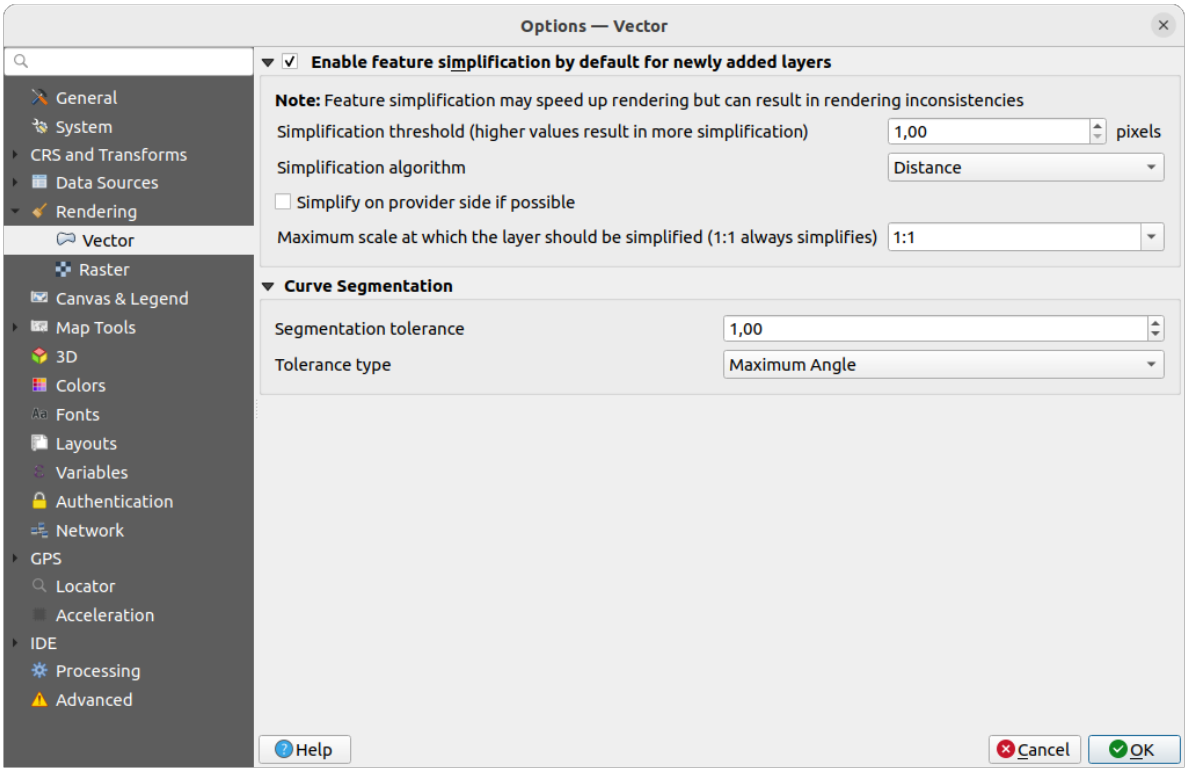


Fig. 5.11: Vector rendering settings

- ☑ *Enable Feature Simplification by Default for Newly Added Layers*: you simplify features' geometry (fewer nodes) and as a result, they display more quickly. Be aware that this can cause rendering inconsistencies. Available settings are:
  - *Simplification threshold (higher values result in more simplification)*
  - *Simplification algorithm*: This option performs a local "on-the-fly" simplification on feature's and speeds up geometry rendering. It doesn't change the geometry fetched from the data providers. This is important when you have expressions that use the feature geometry (e.g., calculation of area) - it ensures that these calculations are done on the original geometry, not on the simplified one. For this purpose, QGIS provides three algorithms: 'Distance' (default), 'SnapToGrid' and 'Visvalingam'.
  - ☐ *Simplify on provider side if possible*: the geometries are simplified by the provider (PostgreSQL, Oracle...) and unlike the local-side simplification, geometry-based calculations may be affected
  - *Maximum scale at which the layer should be simplified (1:1 always simplifies)*

> **Note**
>
> 除了全局设置外，还可以从其 *图层属性* ► *渲染* 菜单中为任何特定图层设置要素简化

- *Curve Segmentation*
  - *分割容差*: 此设置控制圆弧的渲染方式。**较小的**最大角度 (两个连续顶点与曲线中心之间的角度，以度为单位) 或最大差值 (两个顶点的线段与曲线之间的距离，单位为地图单位)，渲染时将使用 **更多直线**段。
  - *容差类型*: 可以是 *最大角度*，也可以是近似值和曲线之间的 *最大差值*。

Raster rendering settings

The *Raster* tab contains specific settings for rendering raster layers.

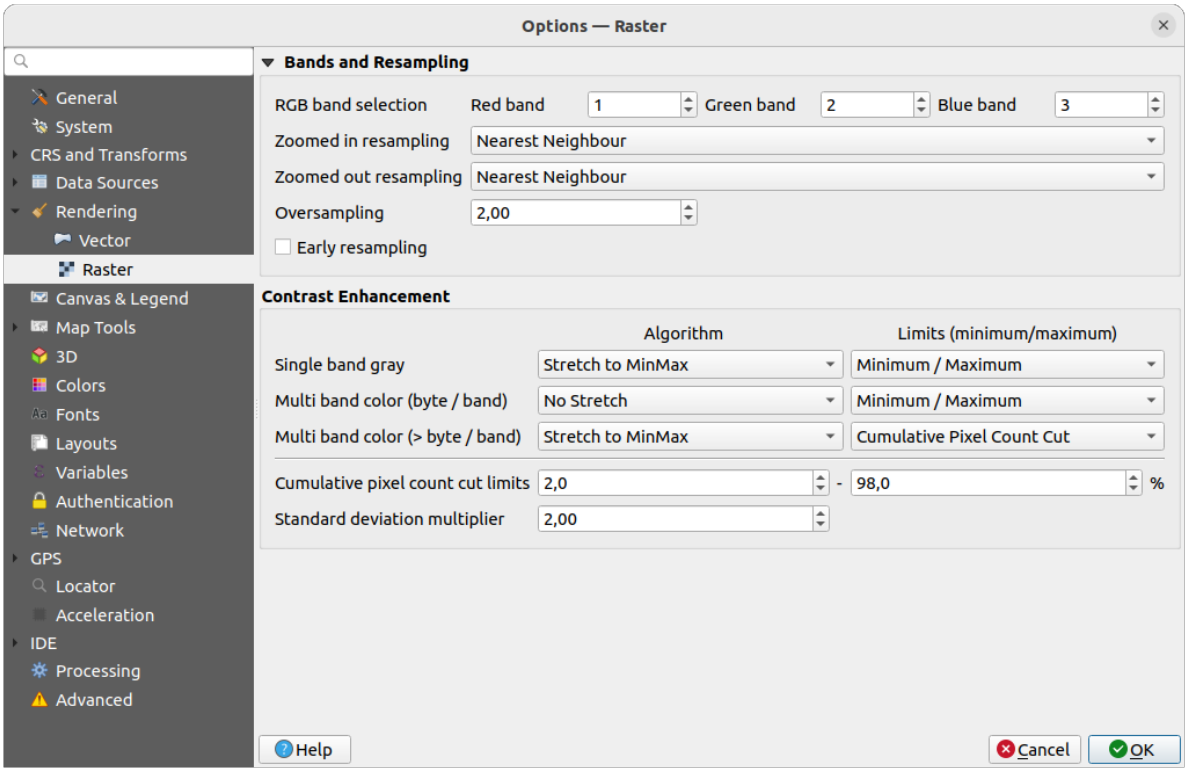


Fig. 5.12: Raster rendering settings

Under *Bands and Resampling*:

- 通过 *RGB* 波段选择, 您可以定义红色、绿色和蓝色波段的编号。
- The *Zoomed in resampling* and the *Zoomed out resampling* methods can be defined. For *Zoomed in resampling* you can choose between three resampling methods: 'Nearest neighbour', 'Bilinear (2x2 kernel)' and 'Cubic (4x4 kernel)'. For *Zoomed out resampling* you can choose between 'Nearest Neighbour' and 'Average'. You can also set the *Oversampling* value (between 0.0 and 99.99 - a large value means more work for QGIS - the default value is 2.0).
- *Early resampling*: allows to calculate the raster rendering at the provider level where the resolution of the source is known, and ensures a better zoom in rendering with QGIS custom styling. Really convenient for tile rasters loaded using an *interpretation method*. The option can also be set at the layer level (*Symbology* properties)

Contrast Enhancement options can be applied to *Single band gray*, *Multi band color (byte/band)* or *Multi band color (>byte/band)*. For each, you can set:

- 要使用的 算法, 其值可以是' 不拉伸', ' 拉伸至极小极大', ' 拉伸并裁剪至极小极大' 或' 裁剪到极小极大'
- 要应用的 限制 (最小/最大值), 其值包括' 累计像素总数裁剪', ' 最小值/最大值', ' 平均 +/- 标准差'。

The *Contrast Enhancement* options also include:

- 累计像素总数裁剪限制
- 标准差乘数

5.1.7 画布和图例设置

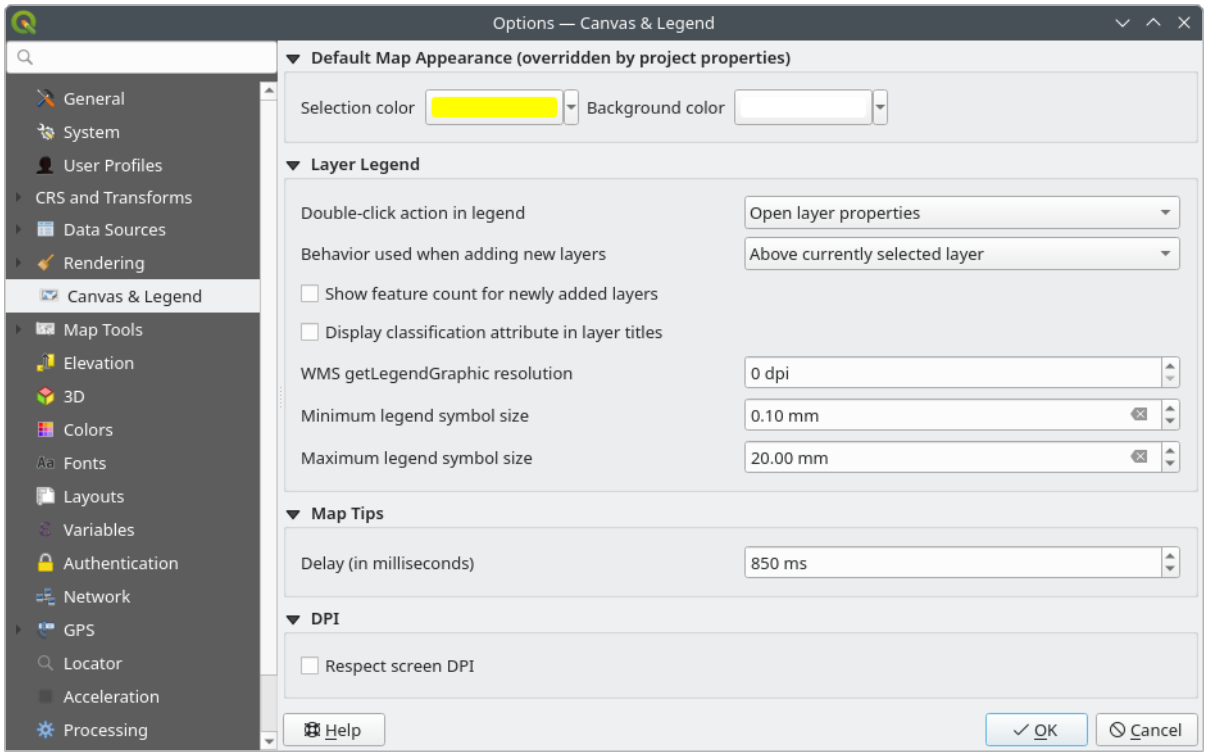


Fig. 5.13: Canvas & Legend settings

这些属性允许您设置:

- **默认地图外观 (由工程属性覆盖)**: 包括 选中要素颜色和 背景颜色。
- **图层图例**交互操作:
  - *Double click action in legend*: whether a double-click on a layer should either *Open layer properties* (default), *Open attribute table* or *Open layer styling dock*.
  - *Behavior used when adding new layers*: determines where layers are placed in the *Layers* panel when loaded into QGIS. It can be:
    - *Above currently selected layer*
    - *Always on top of the layer tree*
    - *Optimal index within current layer tree group*: Unlike the other options that sort the new layers among them and place them as a stack at the desired location, this option extents the sorting logic to the whole layer tree (or the active group) and inserts new layers in an "optimal" fashion by insuring that

point layers sit on top of point layers, followed by line layers on top of lines, followed by polygon layers, etc.

  - *Show feature count for newly added layers*: displays in the *Layers* panel the number of features next to the layer name. Feature count of classes, if any, is as well displayed. You can right-click on a layer to turn on/off its feature count.
  - *Display classification attribute names* in the Layers panel, e.g., when applying a categorized or rule-based renderer (see *Symbology Properties* for more information).
  - *WMS* 获取图例图形分辨率
  - 控制 图层面板符号大小显示的 最小图例符号尺寸和 最大图例符号尺寸
- 以毫秒为单位的图层地图提示 显示 延迟
- Whether QGIS should *Respect screen DPI*: If enabled, QGIS will attempt to display the canvas with physically accurate scale on screen, depending on the monitor's physical DPI. Symbology with specified display size will also be rendered accurately, e.g., a 10mm symbol will show as 10mm on screen. However, label font sizes on canvas may differ from those in QGIS' UI or other applications. If this setting is turned off, QGIS will use the operating system's logical DPI, which will be consistent with other applications on the system. However, canvas scale and symbology size may be physically inaccurate on screen. In particular, on high-dpi screens, symbology is likely to appear too small.

  For best experience, it is recommended to enable *Respect screen DPI*, especially when using multiple or different monitors and preparing visually high-quality maps. Disabling *Respect screen DPI* will generate output that may be more suitable for mapping intended for on-screen use only, especially where font sizes should match other applications.

Note

Rendering in layouts is not affected by the *Respect screen DPI* setting; it always respects the specified DPI for the target output device. Also note that this setting uses the physical screen DPI as reported by the operating system, which may not be accurate for all displays.

5.1.8 地图工具设置

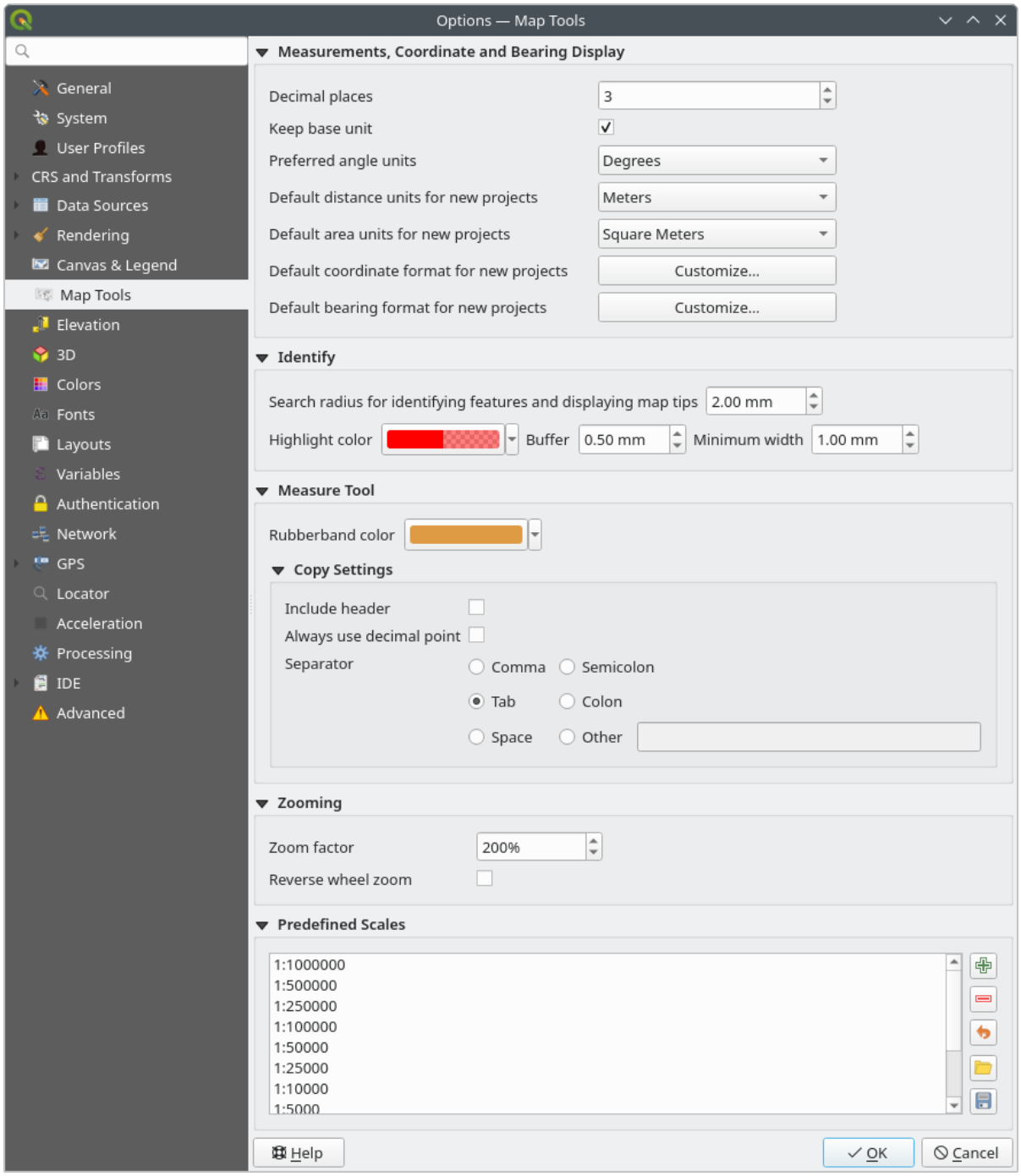


Fig. 5.14: Map tools settings

Measurements, Coordinate and Bearing Display

This section provides ways to configure default units parameters:

- 定义 小数位数
- 保持基本单位不自动转换大数目单位 (例如，米到千米)
- 首选角度单位: 选项有’ 度’, ’ 弧度’, ’ 百分度’, ’ 角分’, ’ 角秒’, ’ 转数/圈数 (Turns/revolutions)’, 毫弧度 (SI 定义) 或密位 (北约军用定义)

- *Default distance units for new projects*: options are 'Meters', 'Kilometers', 'Feet', 'Yards', 'Miles', 'Nautical Miles', 'Centimeters', 'Millimeters', 'Inches', 'Degrees' or 'Map Units' (automatically matches the project's CRS units)
- *Preferred area units for new projects*: options are 'Square meters', 'Square kilometers', 'Square feet', 'Square yards', 'Square miles', 'Hectares', 'Acres', 'Square nautical miles', 'Square centimeters', 'Square millimeters', 'Square inches', 'Square degrees' or 'Map Units' (automatically matches the project's CRS units)
- *Default coordinate format for new projects*, as displayed in the *Coordinates* box on QGIS status bar and in the *Derived* section of the Identify features tool's results
- *Default bearing format for new projects*, as displayed in the status bar for the map canvas panning direction and by the Measure bearing tool.

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

Identify tool

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

- 识别要素和显示地图提示的搜索半径是一个容差距离，只要在该容差范围内单击，识别工具就会描述结果。
- 突出显示颜色允许您选择突出显示所识别要素的颜色。
- 轮廓确定要从识别突出显示的轮廓渲染的缓冲距离。
- 最小宽度决定突出显示对象轮廓的宽度。

测量工具

- 定义测量工具的 橡皮筋颜色

Measure Tool Copy Settings

These settings allow you to control the behavior of the *Copy* button in the Measure Line window. You can choose to *Include header* to keep column names, and you can also select *Separator* of your choice. You can also choose *Always use decimal point* to keep your coordinates copied to the clipboard with a dot as a decimal separator, even if your language settings in QGIS options are set up to use a comma. In this case, if you don't have the *Always use decimal point* option checked, you will be unable to select comma as a field separator.

平移和缩放

- 为缩放工具或鼠标滚轮定义 缩放因子
- *Reverse wheel zoom* allows you to adjust mouse wheel scrolling direction.

预定义比例

Here, you find a list of predefined scales to display by default in the scale-related drop-down widgets, such as the status bar *Scale*, the visibility scales selector or secondary 2D map view settings,... With the and buttons you can add or remove your personal scales. You can also import or export scales from/to a `.XML` file. Note that you still have the possibility to remove your changes and reset to the predefined list.

From the project properties dialog, you can also set your own list of scales, overriding this global one in the widgets.

数字化设置

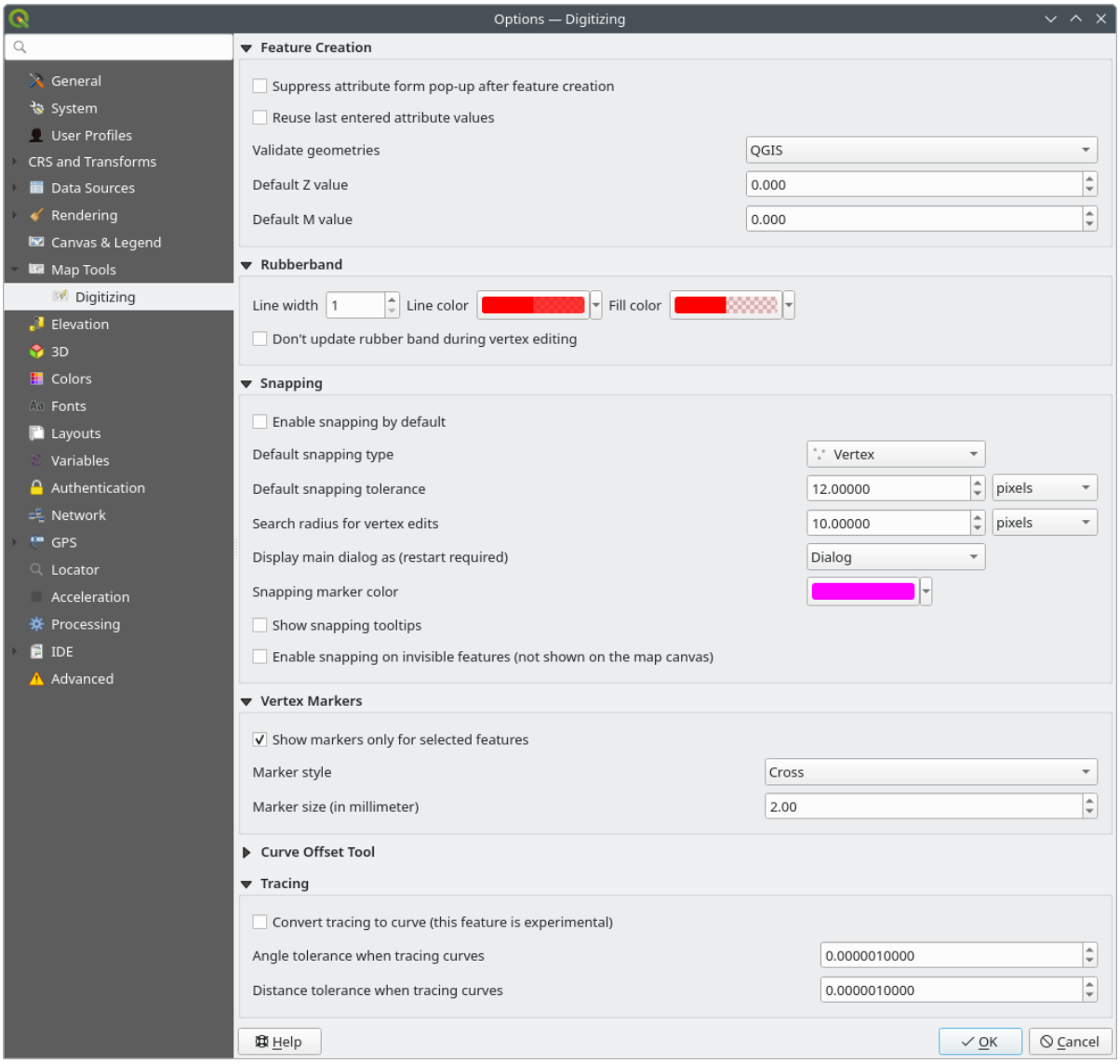


Fig. 5.15: 数字化设置

此选项卡可帮助您在编辑矢量图层 (属性和几何图形) 时配置常规设置。

创建要素

- 要素创建后不要弹出属性表单: 可以在每个图层属性对话框中重写此选项。
- *Reuse last entered attribute values*: remember the last used value of every attribute and use it as default for the next feature being digitized. Works per layer. This behavior can also be controlled on a per-field basis (see *Configure the field behavior*).
- *Validate geometries*. Editing complex lines and polygons with many nodes can result in very slow rendering. This is because the default validation procedures in QGIS can take a lot of time. To speed up rendering, it is possible to select GEOS geometry validation or to switch it off. GEOS geometry validation is much faster, but the disadvantage is that only the first geometry problem will be reported.

  请注意，根据选择，几何图形错误报告可能会有所不同 (请参阅*Types of error messages and their meanings*)

- 默认 Z 值用于创建新的 3D 要素。

橡皮筋

- 定义橡皮筋的 线宽, 线颜色和 填充颜色.
- 编辑顶点时不更新橡皮筋.

捕捉

- 默认启用捕捉打开工程时激活捕捉
- Define *Default snap mode* (’Vertex’, ’Segment’, ’Centroid’, ’Middle of segments’, ’Line endpoints’, ’Area’)
- 以地图单位或像素为单位定义 默认捕捉容差
- 以地图单位或像素为单位定义 顶点编辑的搜索半径
- 主对话框显示为 *(重启 QGIS 生效)*: 设置高级捕捉对话框应显示为” 对话框” 还是” 停靠”。
- 捕捉标记颜色
- 显示捕捉工具提示例如您要捕捉其要素图层的名称，当多个要素重叠时很有帮助。
- 对不可见的要素 *(未在地图画布中显示)* 也启用捕捉

顶点标记

- 仅为选中的要素显示标记
- 定义顶点的 标记样式 (’ 十字’ (默认), ’ 半透明圆圈’ 或’ 无’)
- 定义顶点的 标记尺寸 *(毫米)*

曲线偏移工具

The next options refer to the Offset Curve tool in *Advanced digitizing*. Through the various settings, it is possible to influence the shape of the line offset.

- *Join style*: specifies whether ’Round’, ’Miter’ or ’Bevel’ joins should be used when offsetting corners
- *Quadrant segments*: controls the number of line segments to use to approximate a quarter circle when creating rounded offsets
- *Miter limit*: Sets the maximum distance from the offset geometry to use when creating a mitered join as a factor of the offset distance (only applicable for miter join style)
- *End cap style*: controls how line endings are handled; they can be ’Round’, ’Flat’ or ’Square’.

追踪

通过激活 将追踪转换为曲线您可以在数字化时创建曲线段。请记住，您的数据提供程序必须支持此功能。

5.1.9 Elevation settings

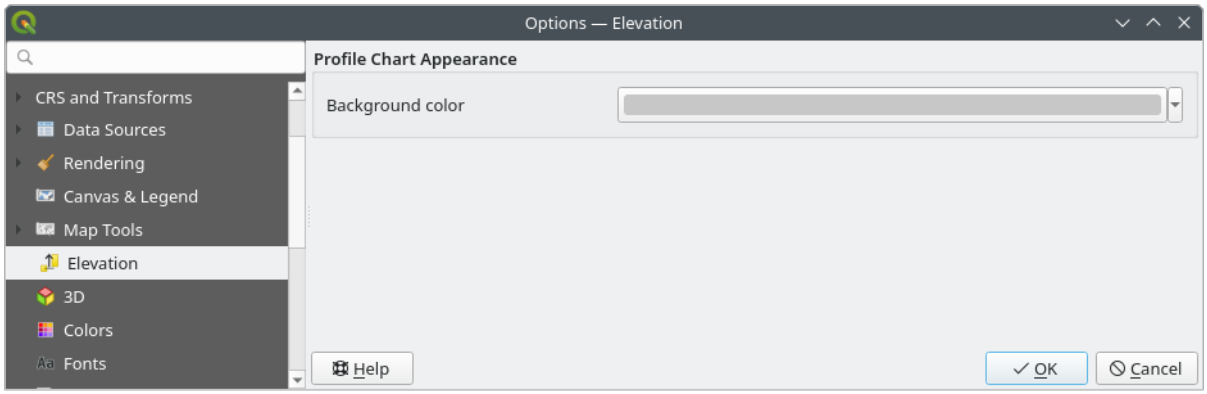


Fig. 5.16: Elevation settings

In *Elevation* menu, you can set a specific color to use as the *Background color* for *elevation profiles*. This can make the chart more readable for certain datasets, e.g., point clouds with RGB coloring, where the default background color is too similar to point colors to be easily discernable. If unchanged, the elevation profiles will continue to display using the standard system background color.

5.1.10 3D settings

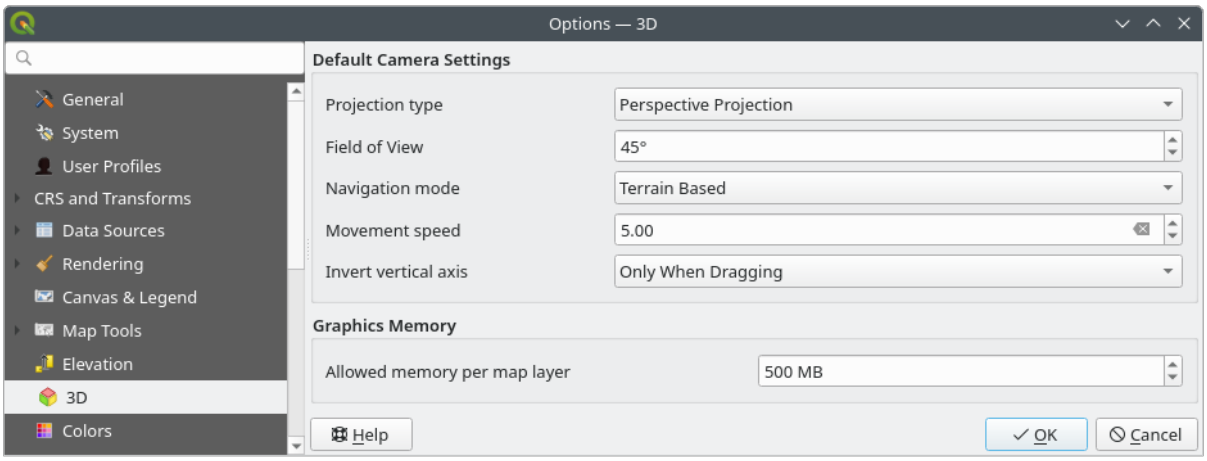


Fig. 5.17: 3D settings

The *3D* menu helps you configure some default settings to use for any *3D Map view*. These can refer to:

- *Default Camera Settings*:
  - *Projection type*: allowing to view the 3D scene in a:
    - *Perspective projection* (default): Parallel lines appear to meet in the distance. Objects appear to shrink the farther they are from the camera.
    - or an *Orthogonal projection*: Parallel lines appear parallel. Objects appear the same size regardless of distance.
  - Camera's *Field of view*: only relevant in perspective projection, specifies the current vertical field of view in degrees and determines how much of the scene is visible to the camera. Default value is 45°.

  - *Navigation mode*: provides different means to interact with the 3D scene. Available modes are:
    * *Terrain based*: the camera follows around a fixed position on the surface of the terrain as the scene is navigated.
    * *Walk mode (first person)*

    Depending on the selected mode, *navigation commands* differ.
  - *Movement speed*
  - *Invert vertical axis*: Controls whether vertical axis movements should be inverted from their normal behaviour. Only affects movement in the *Walk mode*. It can be set to:
    * *Never*
    * *Only when dragging*: causes the vertical motion to inverted only when performing a click-and-drag camera rotation
    * and *Always*: causes the motions to be inverted when both click-and-dragging and when the camera movement is locked to the cursor (via a ~ key press)
- Under *Graphics memory*, the *Allowed memory per layer* option lets you set the GPU memory limit configuration on each layer. This is useful for users utilizing large 3D scenes which exhaust the available GPU memory resources. When a limit is hit, a warning is also displayed, which should assist in troubleshooting large scenes.

5.1.11 Colors settings

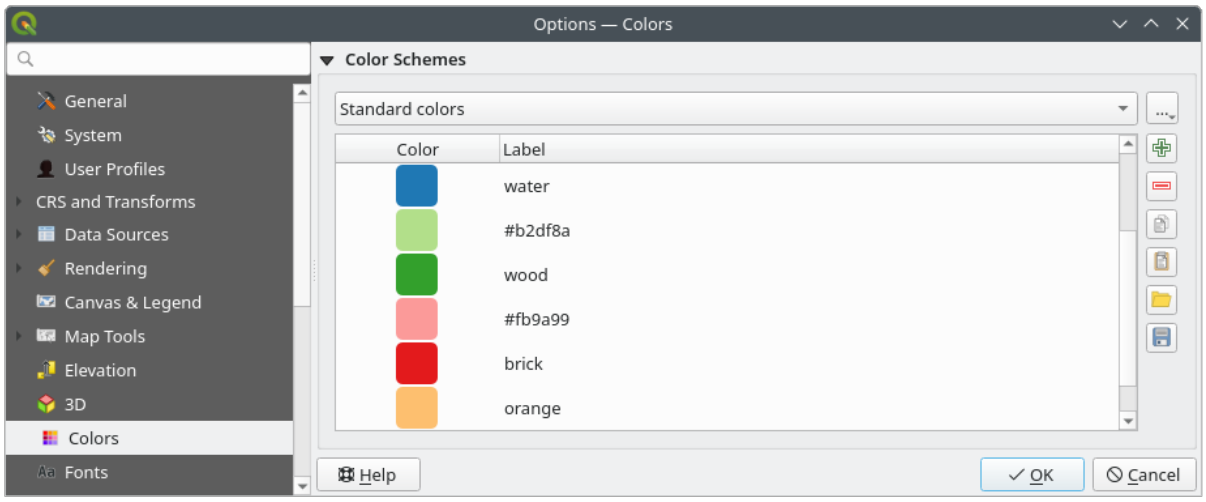


Fig. 5.18: Colors settings

此菜单允许您在颜色选择控件 中创建或更新整个应用程序使用的颜色调色板。您可以选择：

- 最近使用过的颜色显示最近使用的颜色
- 标准颜色, 默认调色板的颜色
- 工程颜色, 特定于当前工程的一组颜色 (更多信息，请参阅*Styles Properties*)
- 新建图层颜色, 将新图层添加到 QGIS 时默认使用的一组颜色
- 或自定义调色板，您可以使用调色板组合框旁边的 ... 按钮创建或导入调色板。

默认情况下，不能移除 最近使用过的颜色, 标准颜色和 工程颜色调色板，并将其设置为显示在颜色按钮下拉列表中。使用 在颜色按钮中显示选项，还可以将自定义调色板添加到此控件中。

对于任何调色板，都可以使用框架旁边的工具组管理颜色列表，即：

- 添加或 移除颜色
- 复制或 粘贴颜色
- 导入或 导出颜色集从/到 `.gpl` 文件。

双击列表中的颜色，在选择颜色 对话框中对其进行调整或替换，也可以通过双击 标注列对其进行重命名。

5.1.12 Fonts Settings

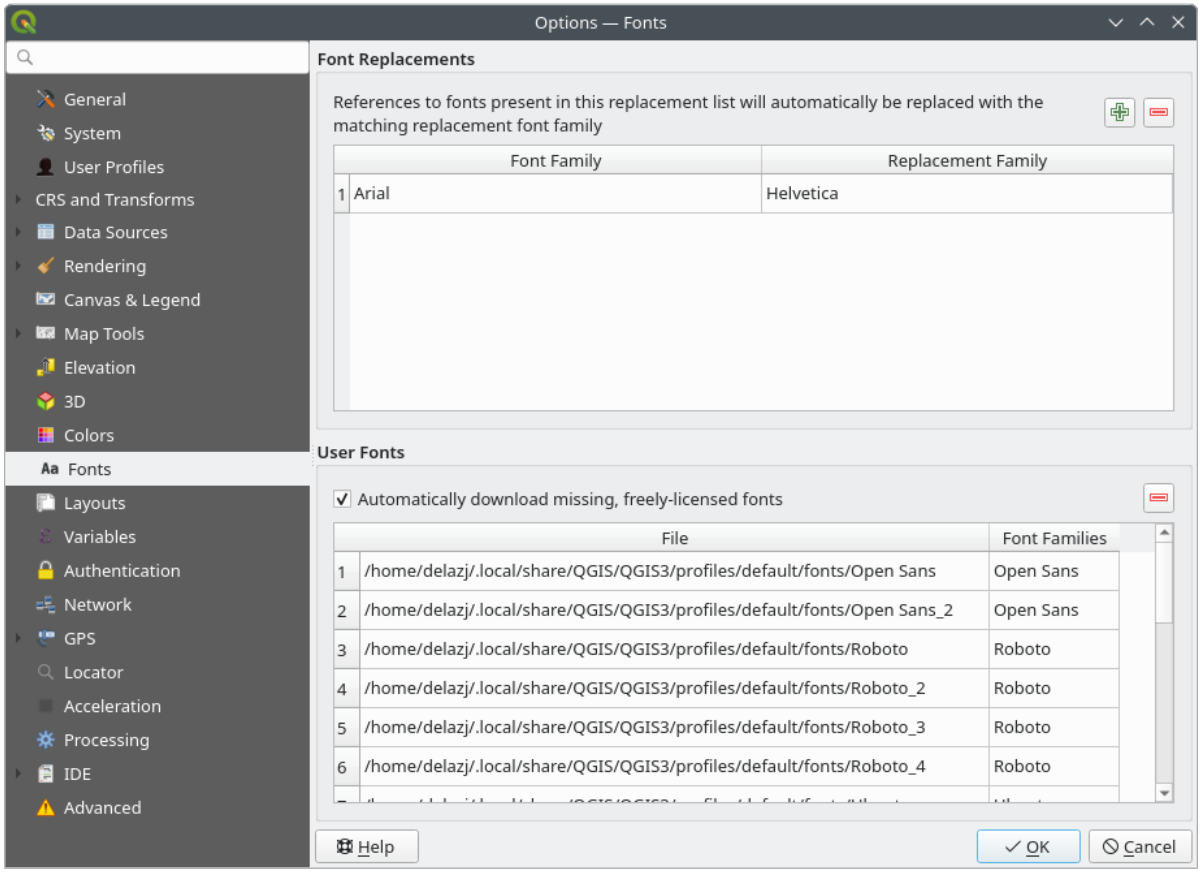


Fig. 5.19: Fonts settings

The *Fonts* tab provides support to manage fonts used across the projects:

- *Font Replacements*: gives ability to populate a list of automatic font replacements to apply when loading projects or styles, allowing better support for projects and styles to be used across different operating systems (e.g., replace ”Arial” with ”Helvetica”).
- *User Fonts*: Allows you to place TTF or OTF fonts in the `fonts` sub-folder of the *user profile*. These fonts can be automatically loaded at QGIS startup time. This provides a way to use fonts without requiring them to be installed on an operating system level, which is often blocked in enterprise environments. The panel lists all installed user fonts and allows you to manage (i.e., remove) previously installed user fonts.

  It is also possible to *Automatically download missing, freely-licensed fonts*: E.g., if you open a project or style, or try to load a vector tile layer that references fonts that aren’t currently available, then a hard-coded list of freely licensed fonts to download via URL is consulted to determine whether it’s possible to automatically download the font to the user profile font directory (with notification of the font license).

5.1.13 Layouts settings

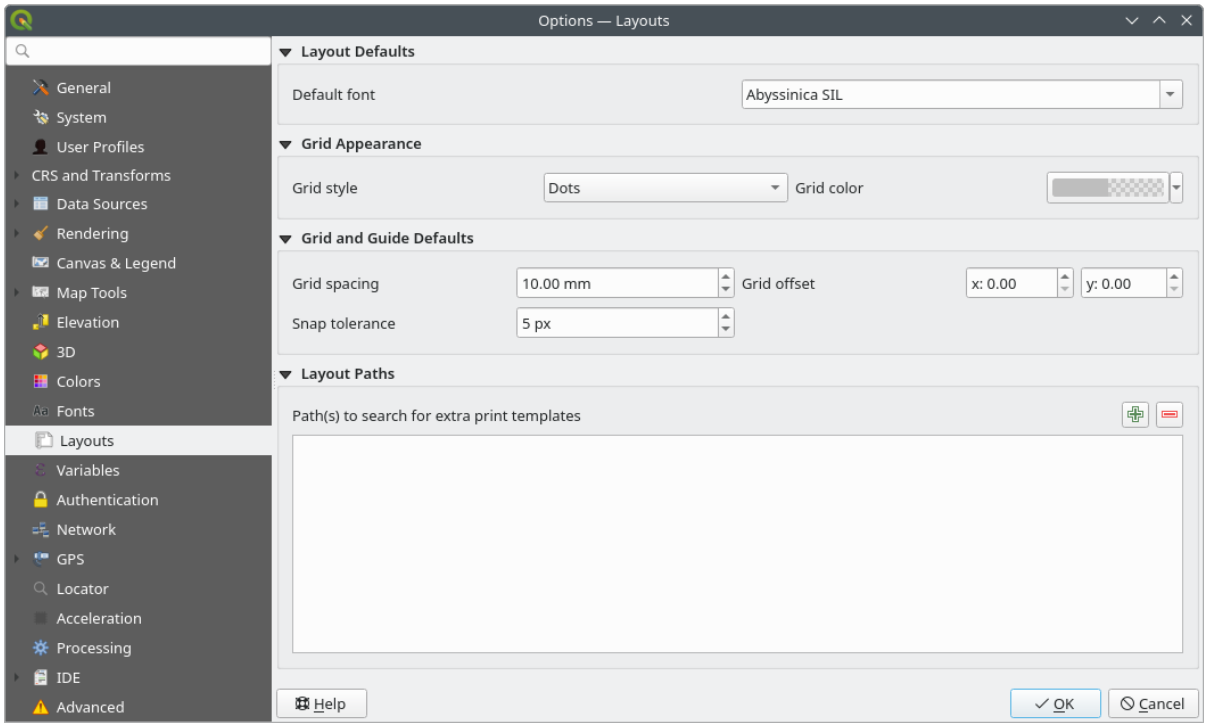


Fig. 5.20: Layouts settings

缺省布局

您可以定义在*打印布局* 中使用的 *默认字体*。

网格外观

- Define the *Grid style* ('Solid', 'Dots', 'Crosses')
- 定义 *网格颜色*

网格和参考线默认值

- Define the *Grid spacing*
- Define the *Grid offset* for X and Y
- Define the *Snap tolerance*

布局路径

- 定义 *附加打印模板的搜索路径*: 包含自定义布局模板的文件夹列表，可在创建新模板时使用。

5.1.14 Variables settings

变量选项卡列出了全局级别的所有可用变量。

它还允许用户管理全局级别的变量。单击 按钮添加新的自定义全局级别变量；同样，从列表中选择一个自定义全局级别变量，然后单击 按钮将其删除。

有关变量的更多信息，请参见*将值存储在变量中* 章节。

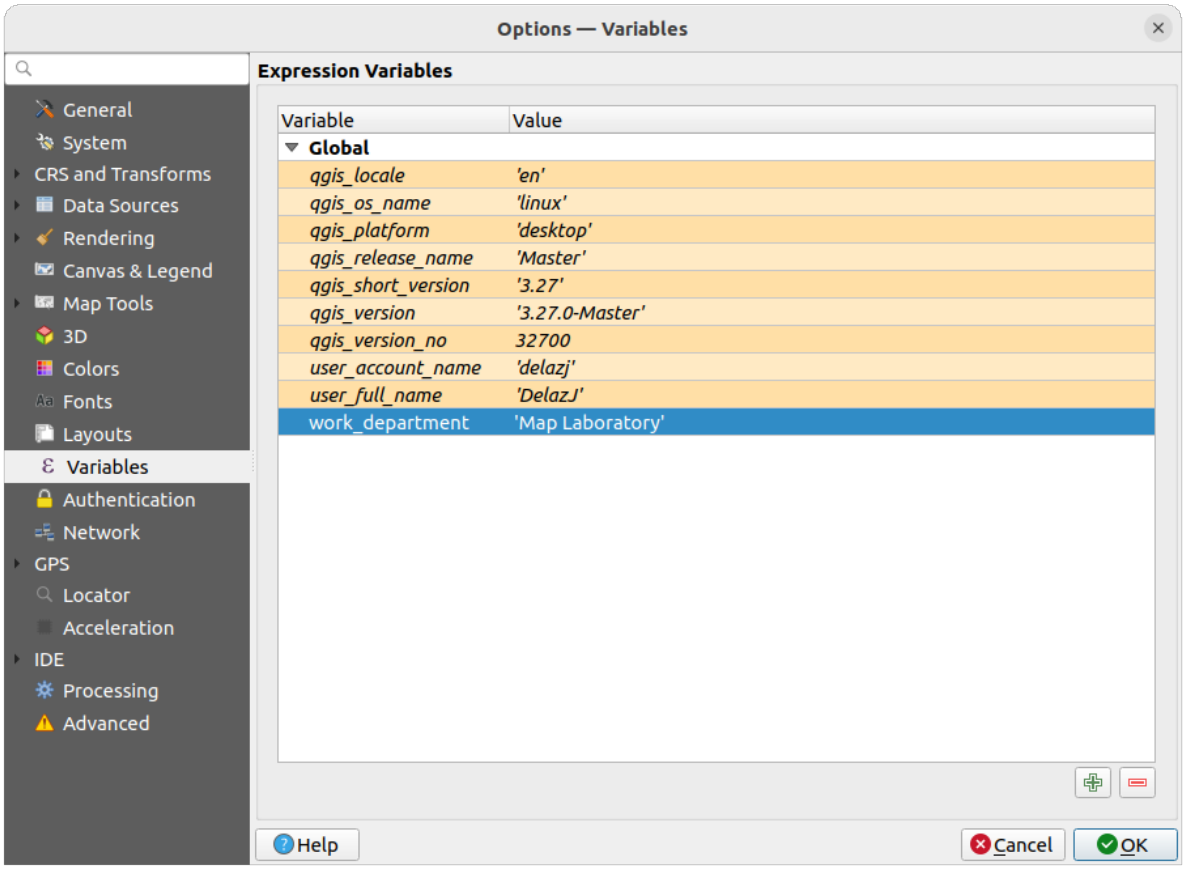


Fig. 5.21: Variables settings

5.1.15 Authentication settings

在 认证选项卡中，可以设置认证配置和管理 PKI 证书。更多信息，请参阅认证系统 。

To manage authentications, you can use the list of tools next to the frame, ie:

- Add new authentication configuration
- Remove selected authentication configuration
- Edit selected authentication configuration

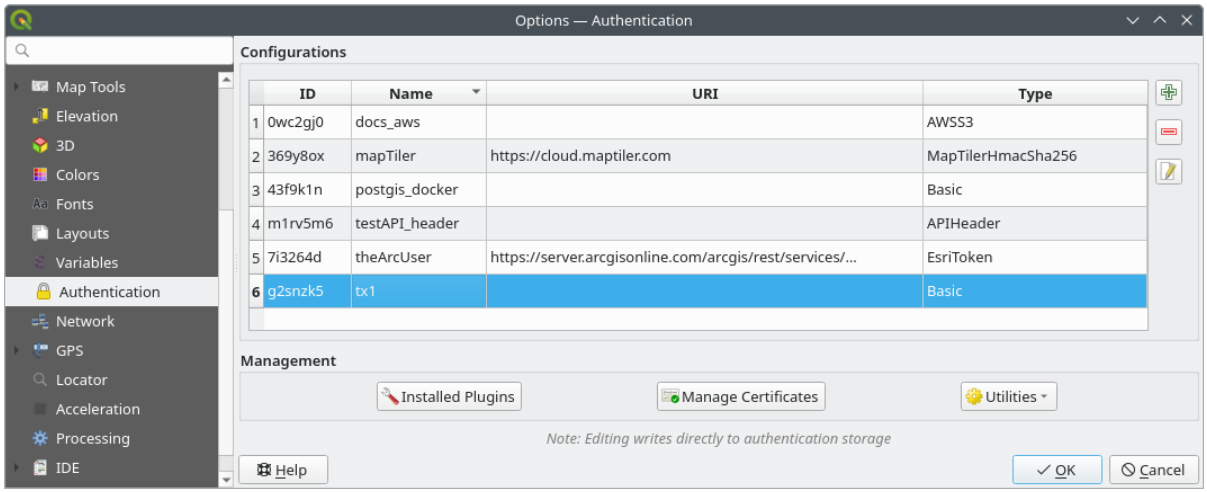


Fig. 5.22: Authentication settings

5.1.16 Network settings

通用

- 定义 网络请求超时 *(毫秒)* - 默认是 60000
- 定义 *WMS* 功能描述的默认有效期 *(小时)* - 默认是 24
- 定义 *WMS-C/WMTS* 瓦片的默认过期时间 *(小时)* - 默认是 24
- 定义 瓦片或要素请求遇到错误时的最大重试次数
- Define *User-Agent prefix* which defaults to `Mozilla/5.0`. This value will be prepended to both QGIS and OS version e.g., `Mozilla/5.0 QGIS/33801/Ubuntu 22.04.4 LTS` to shape the user-agent.

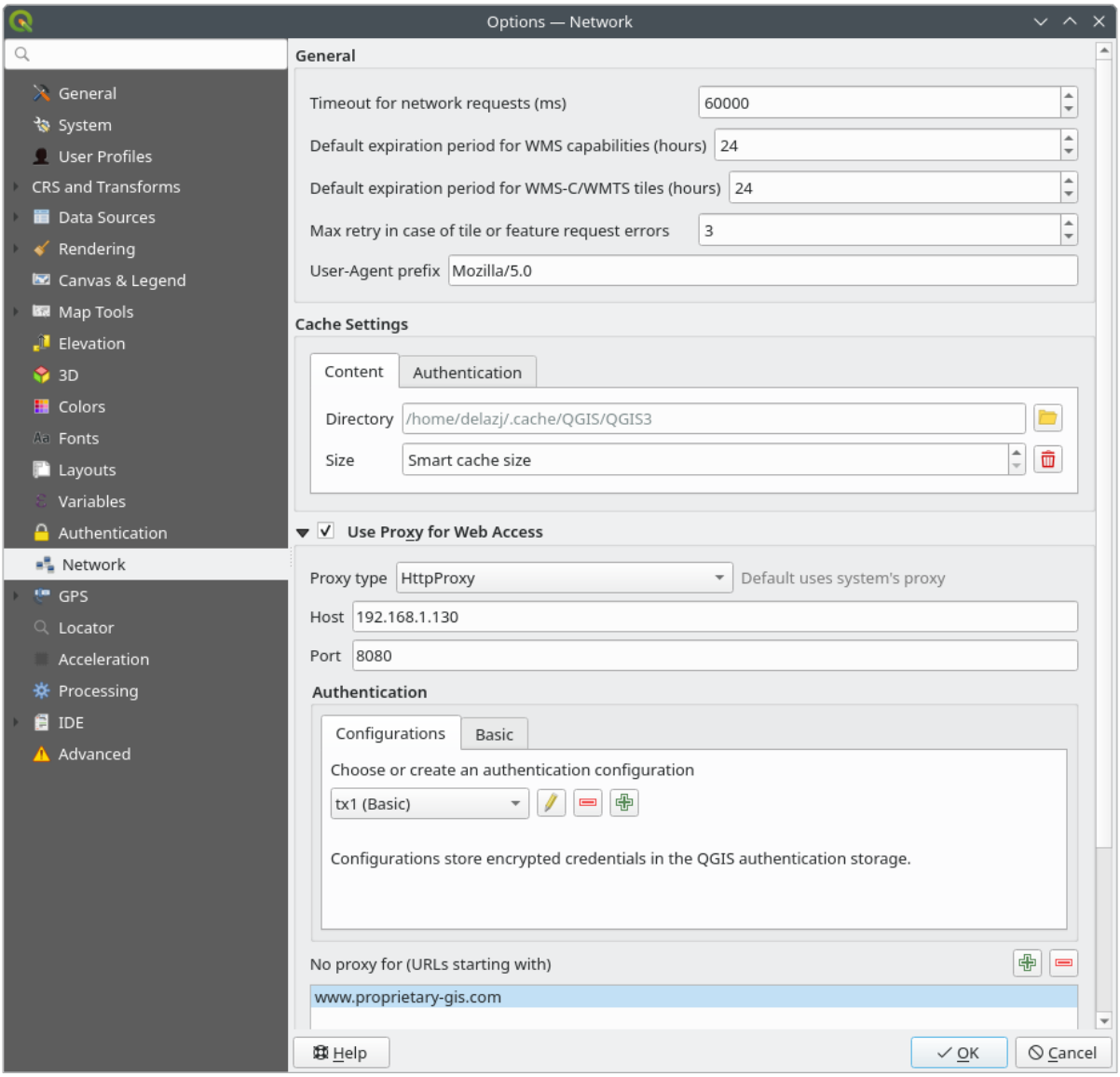


Fig. 5.23: Network and proxy settings

缓存设置

Defines the *Directory* and a *Size* for the cache specified in megabytes. You can clear the value to enable the *Smart cache size*, which sets the maximum cache size based on available disk space. Also offers tools to *automatically clear the connection authentication cache on SSL errors (recommended)*.

Web 访问代理

- *Web 访问使用代理*
- 根据需要设置 *代理类型* , 并定义" 主机" 和" 端口"。可用的代理类型包括:
  - *默认代理*: 代理根据系统的代理确定
  - *Socks5Proxy*: 任何类型连接的通用代理，支持 TCP、UDP、绑定到端口 (传入连接) 和认证。
  - *HttpProxy*: 使用"CONNECT" 命令实现，只支持传出 TCP 连接；支持认证。
  - *HttpCachingProxy*: 使用普通的 HTTP 命令实现，仅在 HTTP 请求的上下文中有用。
  - *FtpCachingProxy*: 使用 FTP 代理实现，仅在 FTP 请求的上下文中有用。

使用认证控件 设置代理的凭据。

排除部分 URL 可以添加到代理设置下方的文本框中 (请参阅 Fig. 5.23). 如果目标 URL 以此文本框中列出的字符串之一开头，则不会使用代理。

If you need more detailed information about the different proxy settings, please refer to the manual of the underlying QT library documentation at https://doc.qt.io/archives/qt-5.9/qnetworkproxy.html#ProxyType-enum

Tip

使用代理

使用代理有时会很棘手，通过对上述代理类型进行" 反复试验" 来检查它们在您的实例中是否成功，是很必要的。

5.1.17 GPS settings

The *GPS* dialog helps you configure GPS devices connections and properties in QGIS. It also provides settings for GPS tracking and data digitizing.

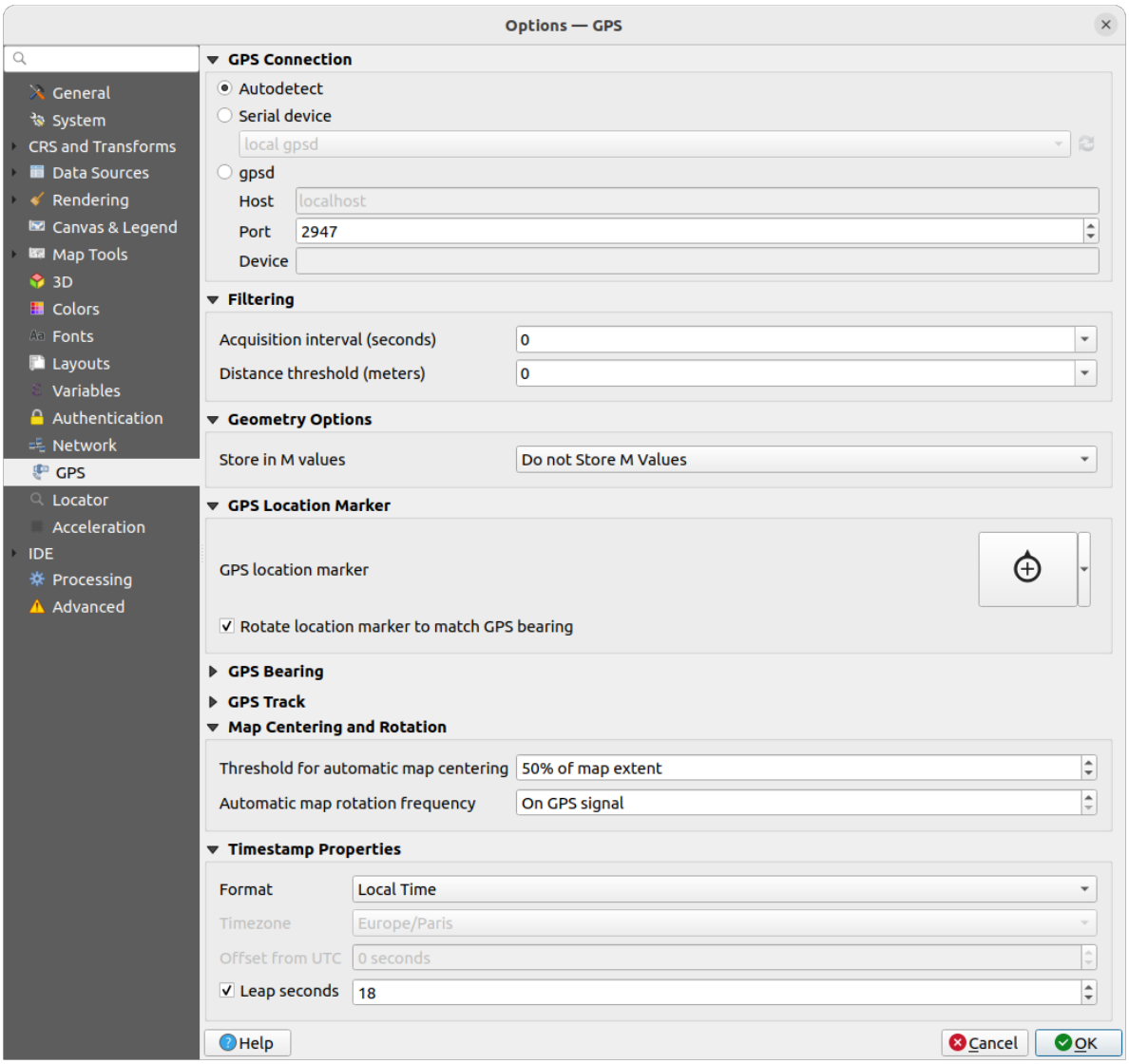


Fig. 5.24: GPS settings

您可以指定：

- *GPS Connection*: provides different means to connect the device to QGIS
  - *Autodetect*
  - *Serial device* (reload required if a new GPS Device is connected)
  - *gpsd* (selecting the *Host* and *Port* your GPS is connected to, and providing a *Device* name)

  In case of connection problems, you can try to switch from one to another.
- *Filtering*: You can set an *Acquisition interval (seconds)* and/or a *Distance threshold (meters)* parameters to keep the cursor still active when the receiver is in static conditions.
- Under *Geometry Options*, the *Store in M values* option allows to create geometries with M values from the inbuilt GPS logging tools. This applies to both features digitized from GPS logs and from the *Log to Geopackage/SpatiaLite* functionality... Options include storing timestamps (as ms since epoch), ground speed, altitudes, bearings, and accuracy components as m values.
- *GPS Location Marker*
  - Configure a symbol for the *GPS location marker*, indicating the current GPS position

  - *Rotate location marker to match GPS bearing*: whether the marker symbol should be rotated to match the GPS direction

- *GPS Bearing*:
  - configure a *Bearing line style* using QGIS line symbol properties
  - set whether to *Calculate Bearings from travel direction*: If checked, the bearing reported by the GPS device will be ignored and the bearing will instead be calculated by the angle between the previous two GPS locations.
- *GPS Track*: set symbol to use for the *Track line style*
- *Map Centering and Rotation*: defines when the map canvas is updated according to the user displacement on the field:
  - *Threshold for automatic map centering*: defines the minimal offset of the GPS position from the map canvas center to trigger an automatic *map recentering* if enabled.
  - *Automatic map rotation frequency*: defines how often the map rotation to match the GPS bearing could happen; it can be on a custom duration or *On GPS signal*.
- *Timestamp Properties* to configure how time values are displayed and stored in the data. Parameters include the *Format* which can be:
  - *Local time*
  - *UTC*
  - *UTC with offset*, to account for daylight savings offsets or other complex time zone issues
  - a specific *Time zone*

  Moreover, *Leap seconds* correction can be applied, by adding the seconds to GPS timestamp.

GPSBabel

GPSBabel converts waypoints, tracks, and routes between popular GPS receivers such as Garmin or Magellan and mapping programs like Google Earth or Basecamp. Literally hundreds of GPS receivers and programs are supported. QGIS relies on GPSBabel to interact with these devices and *manipulate their data*.

For details on how-to, please refer to *Loading to or from a device*.

5.1.18 Locator settings

The 搜索选项卡允许您配置搜索栏, 这是状态栏上的一个快速搜索控件，可帮助您在应用程序中执行搜索。它提供了一些默认过滤 (带有前缀) 供使用：

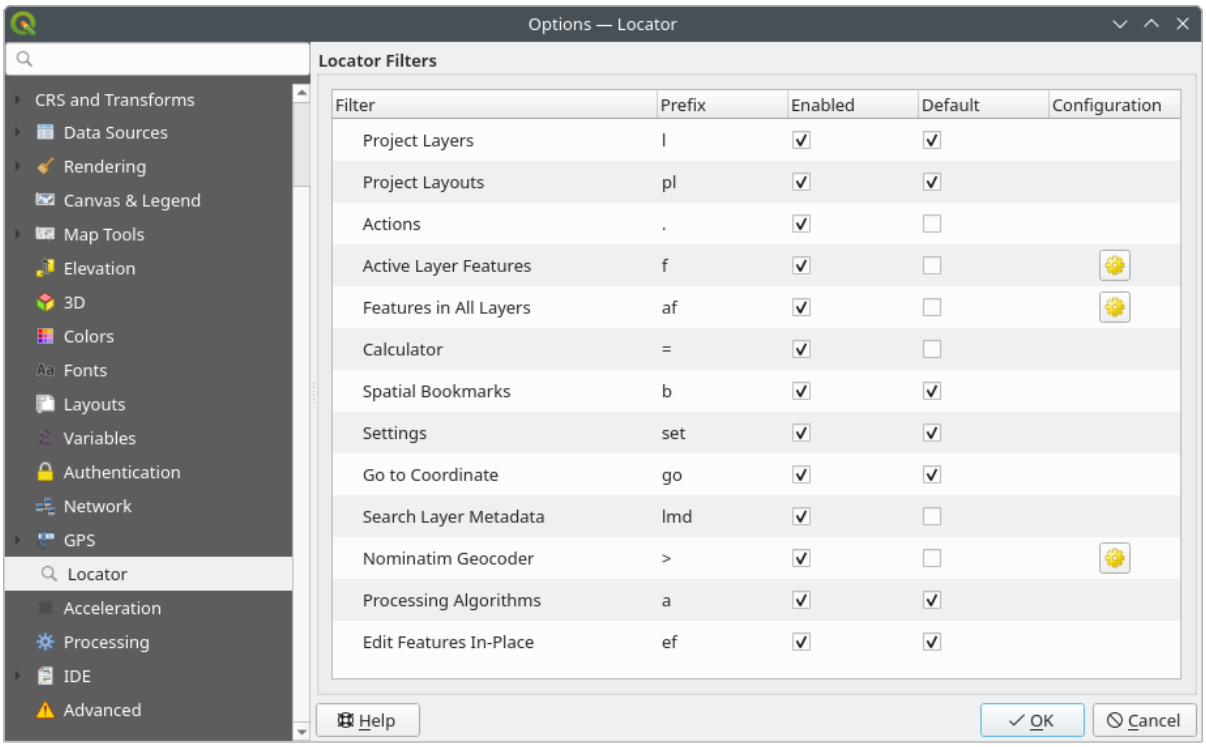


Fig. 5.25: Locator settings

- *Project Layers* (`l`): finds and selects a layer in the *Layers* panel.
- *Project Layouts* (`pl`): finds and opens a print layout.
- 动作 (`.`): 查找并执行 QGIS 动作；动作可以是 QGIS 中的任何工具或菜单、打开面板...
- *Active Layer Features* (`f`): searches for matching attributes in any field from the current active layer and zooms to the selected feature. Press to configure the maximum number of results.
- *Features in All Layers* (`af`): searches for matching attributes in the *display name* of each *searchable layers* and zooms to the selected feature. Press to configure the maximum number of results and the maximum number of results per layer.
- *Calculator* (`=`): allows evaluation of any QGIS expression and, if valid, gives an option to copy the result to the clipboard.
- *Spatial Bookmarks* (`b`): finds and zooms to the bookmark extent.
- *Settings* (`set`): browses and opens project and application-wide properties dialogs.
- *Go to Coordinate* (`go`): pans the map canvas to a location defined by a comma or space separated pair of x and y coordinates or a formatted URL (e.g., OpenStreetMap, Leaflet, OpenLayer, Google Maps, ...). The coordinate is expected in WGS 84 (`epsg:4326`) and/or map canvas CRS.
- *Nominatim Geocoder* (`>`): geocodes using the Nominatim geocoding service of the OpenStreetMap Foundation. Press to limit results to one or more countries using list of country codes. This helps narrow down the search to specific countries, avoiding ambiguous results from other locations. For example, entering `DE` or `DE,FR` restricts results to Germany or Germany and France, respectively.
- Processing Algorithms (`a`): searches and opens a Processing algorithm dialog.
- *Edit Selected Features* (`ef`): gives quick access and runs a compatible *modify-in-place* Processing algorithm on the active layer.

在对话框中，您可以：

- customize the filter *Prefix*, i.e., the keyword to use to trigger the filter
- 设置是否 *启用过滤*：可以在搜索中使用过滤，并在搜索栏菜单中提供快捷键
- 设置过滤是否为 *默认*: 不使用过滤的搜索只返回默认过滤类别的结果。
- 一些过滤提供了配置搜索结果数量的方法。

默认搜索过滤集可以通过插件进行扩展，例如用于 OSM Nominatim 搜索、直接数据库搜索、图层目录搜索等

5.1.19 Acceleration settings

OpenCL 加速设置。

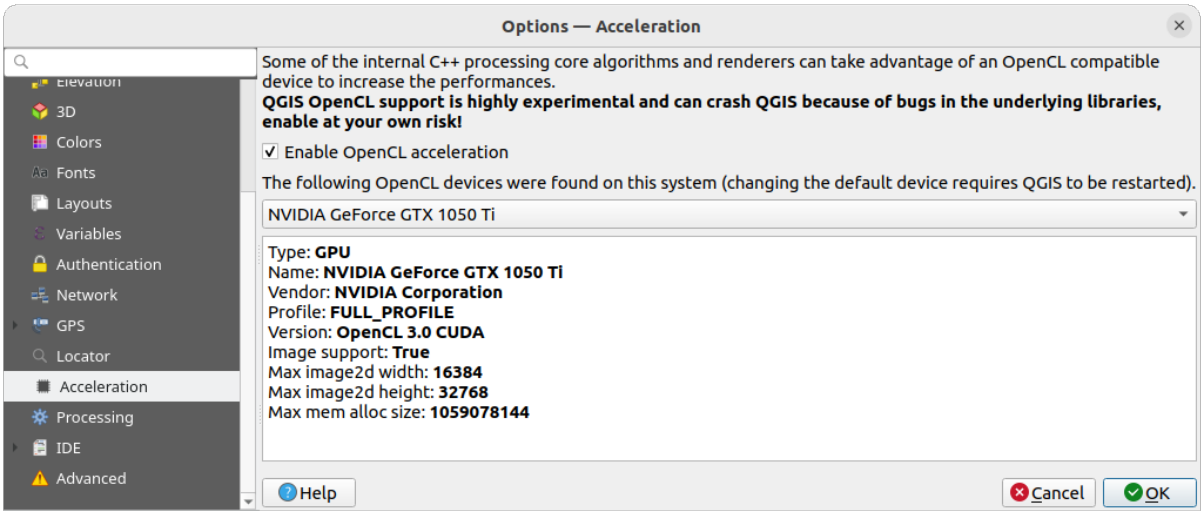


Fig. 5.26: Acceleration settings

根据您的硬件和软件情况，可能需要安装其他库来启用 OpenCL 加速。

5.1.20 IDE settings

Under *GitHub access token*, you can generate a personal token allowing you to share code snippets from within the Python code editor. More details on GitHub authentication

Code Editor settings

在 *代码编辑器*选项卡，可以控制代码编辑器控件 (Python 交互式控制台和编辑器、表达式控件和函数编辑器等) 的外观和行为。

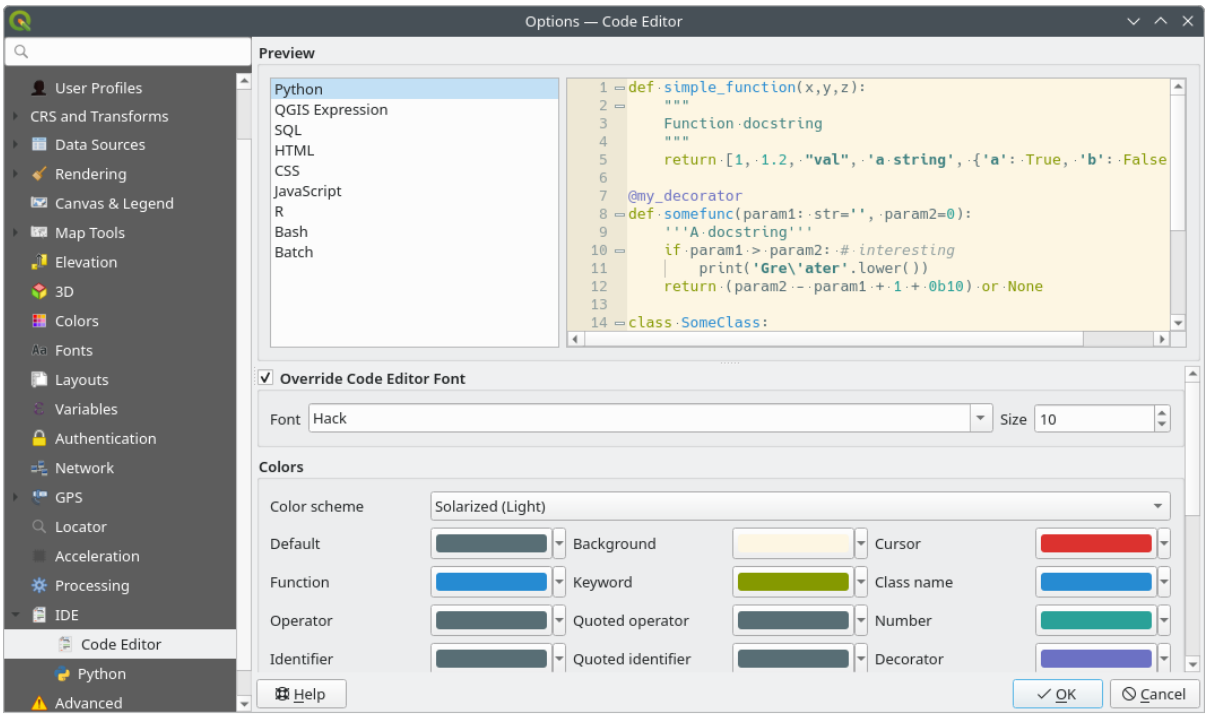


Fig. 5.27: Code Editor settings

在对话框顶部，控件以各种编码语言 (Python, QGIS 表达式, HTML, SQL, JavaScript) 提供当前设置的实时预览。调整设置的个性化方法如下：

- Check *Override Code Editor Font* to modify the default *Font* family and *Size*.
- 在 颜色组下，您可以：
  - 选择 颜色方案: 预定义方案的设置为 默认, 曝光暗和 曝光亮。一旦修改颜色，就会触发 自定义方案，并且可以通过选择预定义方案进行重置。
  - 在编写代码时更改每个元素的颜色, 例如用于注释、引号、函数、背景等的颜色。

Python settings

The *Python* settings help you manage and control the behavior of the Python editors (*interactive console*, *code editor*, *project macros*, *custom expressions*, ...). It can also be accessed using the Options... button from:

- *Python* 控制台工具栏
- *Python* 控制台控件的上下文菜单
- 以及代码编辑器的上下文菜单。

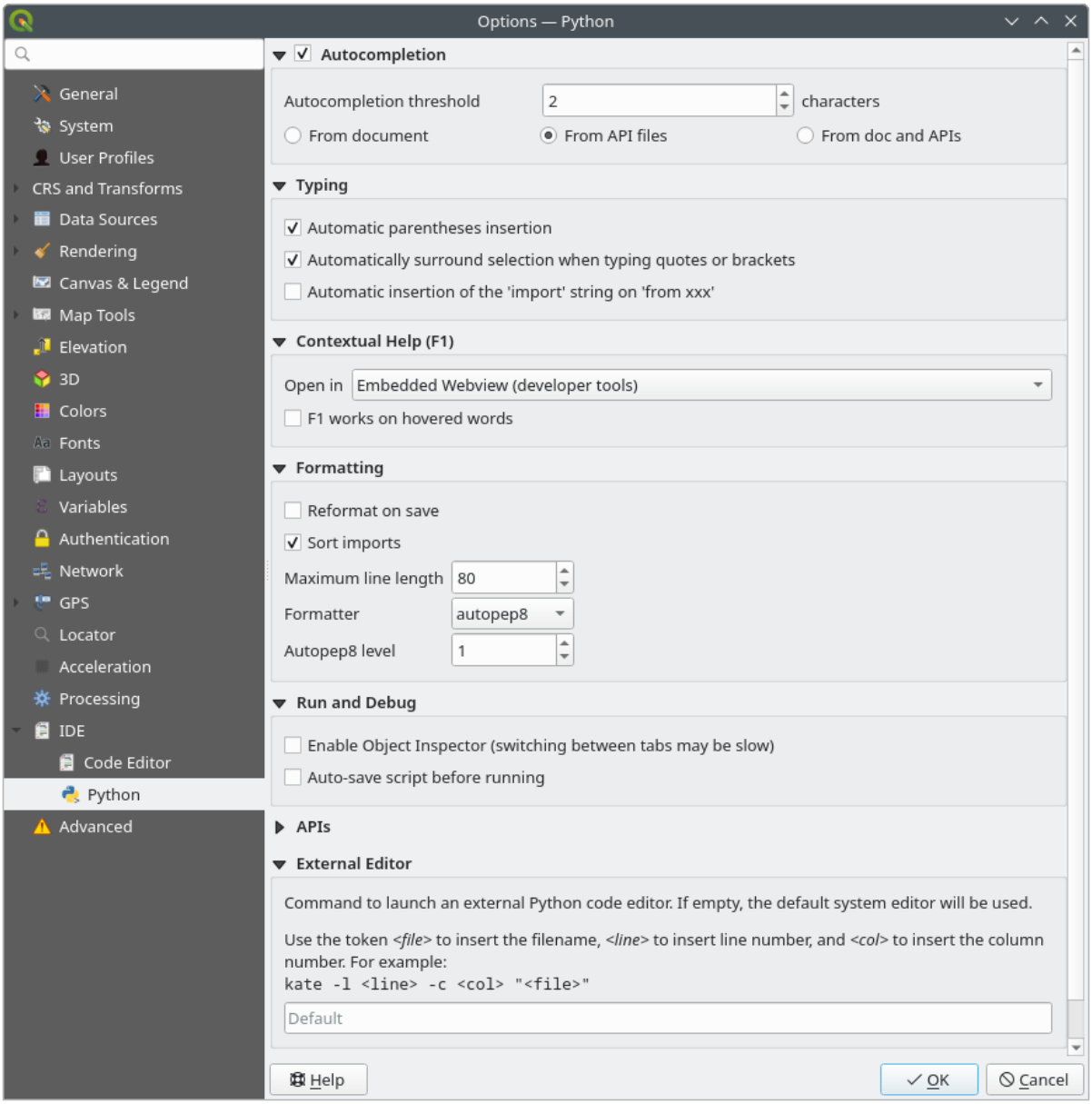


Fig. 5.28: Python settings

您可以指定：

- *Autocompletion*: Enables code completion. You can get autocompletion from the current document, the *installed API* files or both.
  - *Autocompletion threshold*: Sets the threshold for displaying the autocompletion list (in number of characters)
- 在 键盘输入下面
  - *Automatic parentheses insertion*: When no text is selected, if an opening character (parentheses, quotes, brackets, ...) is entered, inserts the matching closing character just after the cursor. Note that this behavior is disabled if the current cursor is inside a string or comment.
  - *Automatically surround selection when typing quotes or brackets*: When an opening character is entered, the selected text is enclosed in the opening/closing pair. Selection remains the same, so it is possible to quote a selected word and enclose it in parentheses just by typing " then (.

Special case for multiline selection with quotes and double quotes: selection is enclosed in triple single/double quotes.

  - ☐ *Automatic insertion of the 'import' string on 'from xxx'*: Enables insertion of 'import' when specifying imports

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

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

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

- under *Formatting*, you can add automated tools to reformat the code you are writing:
  - ☐ *Reformat on save*: formatting is applied just before saving the script
  - ☑ *Sort imports*: sorts 'import' statements using the isort library
  - *Maximum line length*: controls how the formatter will wrap the lines, and controls the editor ruler
  - *Formatter* - supported tools are *autopep8* and *black*, with dedicated option:
    - *Autopep8 level* - more details at autopep8 advanced usage
    - ☐ *Normalize quotes*: replaces all single quotes with double quotes if possible
- 在 运行和调试下面
  - ☐ *Enable Object Inspector (switching between tabs may be slow)*
  - ☐ 运行之前自动保存的脚本: 执行时自动保存脚本。此操作将储存一个临时文件 (在临时系统目录中)，该文件将在运行后自动删除。

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

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

  You can also choose to add specific API files.

- ☐ 使用预备的 *API* 文件: 如果勾选，所选 `*.pap` 文件将用于代码完成。要生成预备的 API 文件，必须至少加载一个 `*.api` 文件，然后通过单击 编译 *API...* 按钮对其进行编译。

The *External Editor* group allows you to provide command line instructions to launch an external Python code editor, given a file name, a line and column number. If empty, the default system editor will be used.

5.1.21 Processing settings

The *Processing* tab provides you with general settings of tools and data providers that are used in the QGIS Processing framework. More information at *Configuring the Processing Framework*.

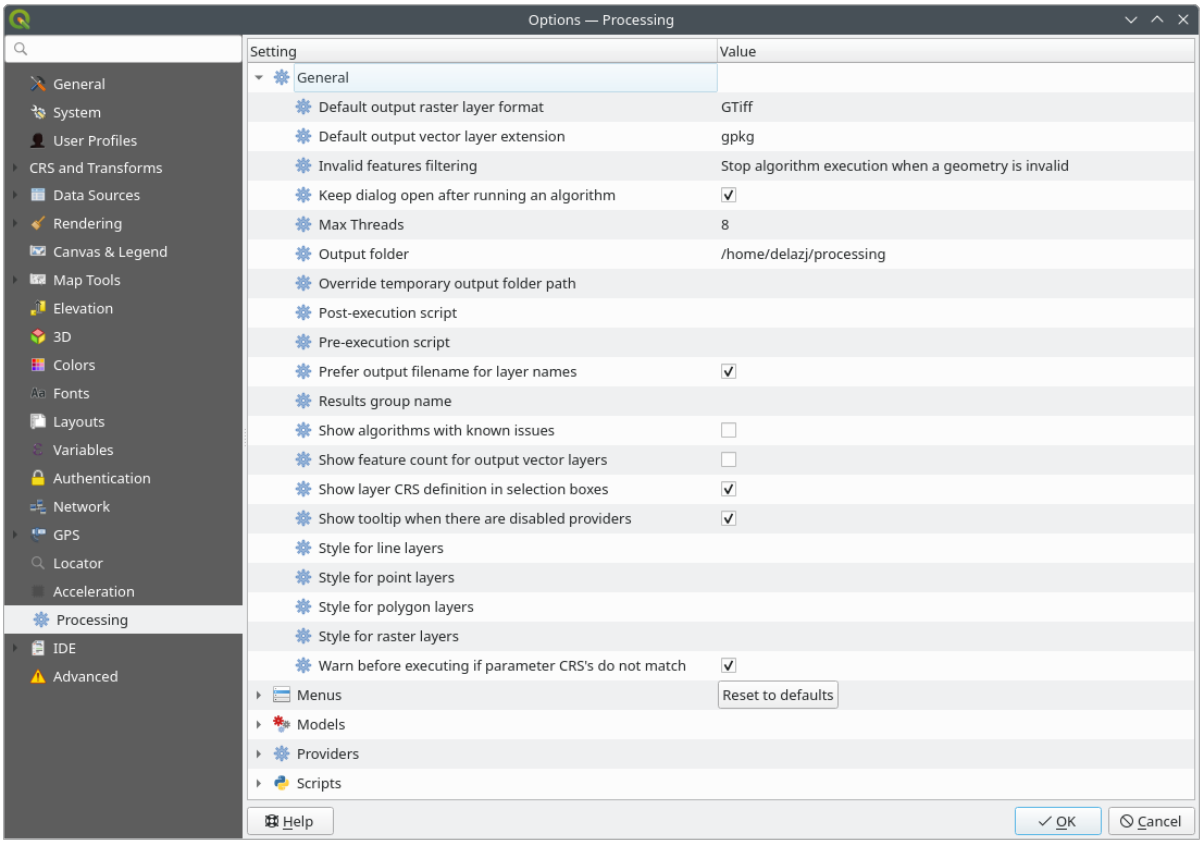


Fig. 5.29: Processing settings

5.1.22 Advanced settings

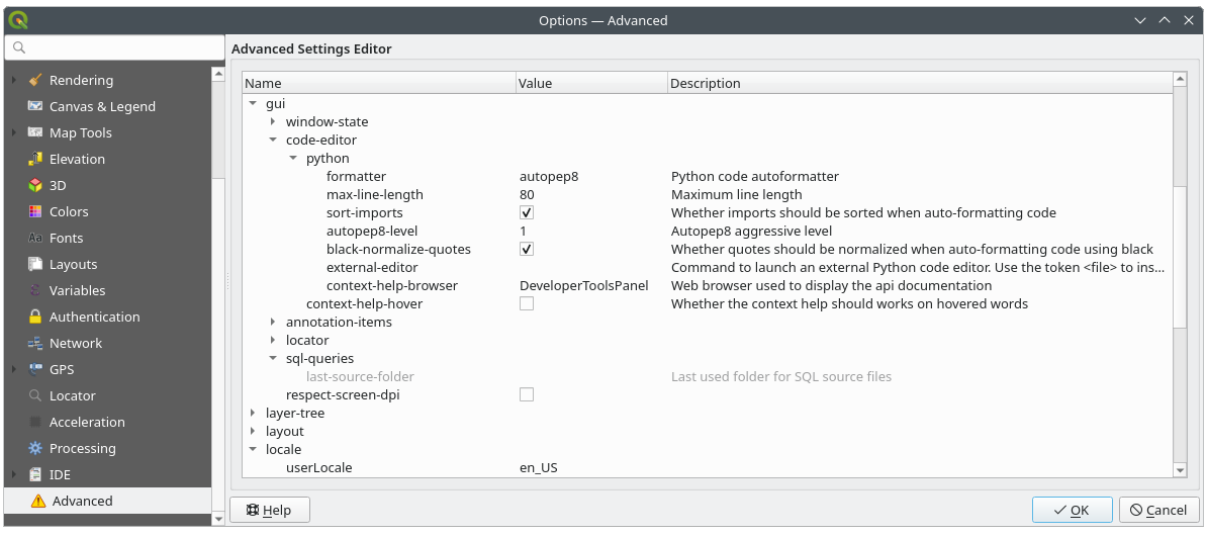


Fig. 5.30: Advanced settings

所有与 QGIS 相关的设置 (UI、工具、数据提供程序、处理配置、默认值和路径、插件选项、表达式、几何图形检查…) 都保存在当前用户配置 目录下的 QGIS/QGIS3.ini 文件中。通过将此文件复制到其他设备，可以共享配置。

在 QGIS 中, 高级选项卡提供了通过 高级设置编辑器管理这些设置的方法。在您保证要小心之后，这个控件将被一个包含所有现有设置的树所填充，您可以编辑它们的值。右键单击设置或组，可以将其删除 (要添加设置或组，必须编辑 QGIS3.ini 文件)。所做更改会自动保存在 QGIS3.ini 文件中。

Warning

避免盲目使用高级选项卡设置

鉴于更改会自动应用，因此修改此对话框中的项时一定要小心。在不知情的情况下进行更改可能会以各种方式破坏 QGIS 设置。

5.2 使用用户配置文件

5.2.1 The concept

设置 ► 用户配置菜单提供设置和访问用户配置文件的功能。用户配置文件是统一的应用程序配置，允许在单个文件夹中储存：

- 所有全局设置, 包括语言环境、投影、认证设置、调色板、快捷键等
- GUI 配置和定制
- 为基准变换而安装的网格文件和其他 Proj 辅助文件
- 已安装的插件 及其配置
- 工程模板和保存工程的历史记录及其图像预览
- 数据处理设置 、日志、脚本和模型。

默认情况下，QGIS 安装包含一个名为 default 的单用户配置文件夹。但您可以创建任意数量的用户配置文件夹：

1. Click the *Settings* ► *User Profiles* ► *New profile...* entry.
2. 系统将提示您提供配置文件名，并在 ~/<UserProfiles>/ 下创建同名文件夹，其中：
   - ~ represents the **HOME** directory, which on Windows is usually something like C:\Users\<username>.
   - 和 <UserProfiles> 表示主配置文件夹，即：
     - .local/share/QGIS/QGIS3/profiles/
     - %AppData%\Roaming\QGIS\QGIS3\profiles\
     - Library/Application Support/QGIS/QGIS3/profiles/

   可以使用 打开当前配置文件夹从 QGIS 打开用户配置文件夹。
3. 使用原始的配置启动 QGIS 新实例，之后您可以设置自定义配置。

如果您的 QGIS 设置中有多个配置文件，当前配置文件的名称将显示在应用程序标题栏的方括号之间。

由于每位用户配置文件都包含独立的设置、插件和历史记录，因此非常适合不同的工作流、演示、同一台计算机的用户或测试设置等。您可以通过 设置 ► 用户配置菜单，或从命令行 使用特定用户的配置文件运行 QGIS。

Tip

在新用户配置文件下运行 QGIS，以检查 Bug 的持久性

Bugs you may encounter with functions in QGIS can be related to leftovers in the current user profile. Running QGIS under another user profile can help you solve them or check the origin of the issue. The general advice is to launch QGIS under a new user profile, thus a cleaner configuration, and run the commands again.

If the bug prevents you to create a new user profile from within the *Settings* ► *User Profiles* menu, you can either:

- Rename in the file explorer, the "broken" user profile folder in the `QGIS3/profiles` folder and restart QGIS. A new `default` user profile will be created and executed.
- Start QGIS from the command line, using the new *profile name* argument:

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

5.2.2 Setting user profile

By default, QGIS opens a new session with the profile of the last closed session. This, among other settings, can be customized in *Settings* ► *Options* ► *User Profiles* tab:

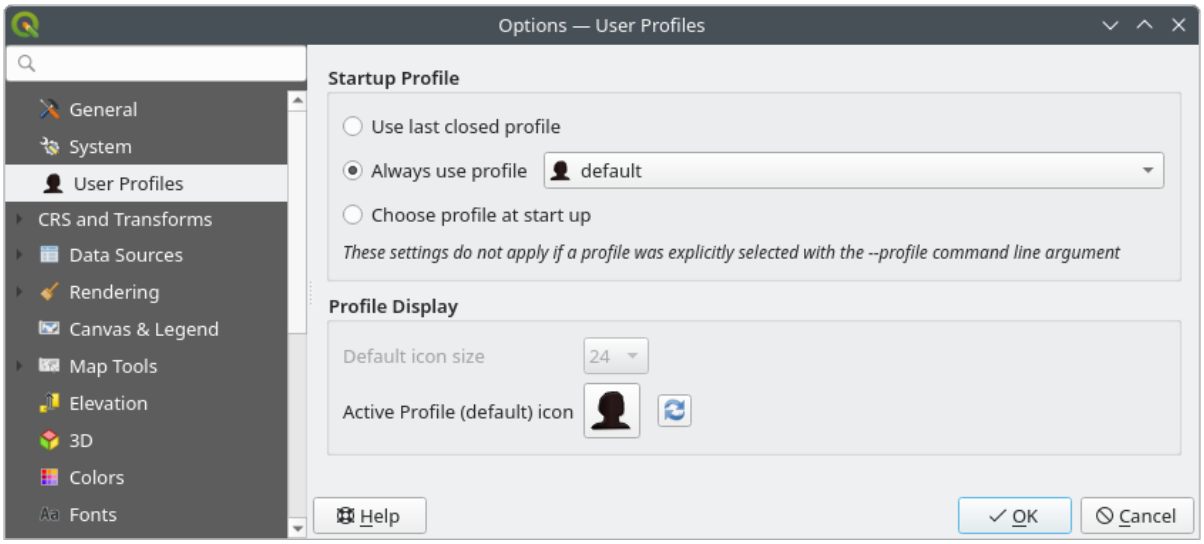


Fig. 5.31: User profiles settings

- *Startup profile*: indicates the user profile to use when starting a QGIS session. It can be:
  - *Use last closed profile*
  - a specific user profile to select from *Always use profile* drop-down menu
  - *Choose profile at start up*: Opens the *User Profile Selector* dialog listing available user profiles. Double-click an entry or select a profile and press *OK* to use that user profile for the starting session. It is also possible to *Add new profile* to the list: a profile is created from scratch and automatically run with the opening session.
- Under *Profile display*, you can set:
  - The size of icons to use when selecting a profile from the *User Profile Selector* dialog.
  - A specific icon to display next to the current profile in the *Settings* ► *User profiles* menu or the *User Profile Selector* dialog. Press Reset profile icon to remove custom modifications.

5.3 工程属性

在 工程 ► 工程属性下的工程属性窗口中，可以设置工程特定的选项。特定于工程的选项将重写上述 选项对话框中的等效选项。

5.3.1 通用属性

In the *General* tab, the *General settings* let you:

- 查看工程文件的位置
- set the folder for the project home (available in the *Project home* item of the *Browser* panel). The path can be relative to the folder of the project file (type it in) or absolute. The project home can be used for storing data and other content that is useful for the project. Convenient when dataset and project files are not stored at the same place. If not filled, the *Project home* defaults to the project file folder.
- 在工程文件路径旁边指定工程的标题
- 选择要素被选中时要使用的颜色
- 选择背景色：用于地图画布的颜色
- 设置工程中图层的路径应保存为绝对 (完整) 还是相对于工程文件位置。当图层和工程文件都可以移动或共享，或从不同平台的计算机访问工程时，您可能更喜欢相对路径。
- 选择在工程渲染为地图瓦片时避免伪影。请注意，勾选此选项可能会导致性能下降。
- *Remember attribute tables windows and docks between sessions*: If checked for a project, then any opened attribute tables will be saved into the project and immediately restored when loading that project. This can improve workflows when you constructed a project with a particular set of attribute table configurations for your requirements, and re-setting up these attribute tables is a hassle.

计算面积和距离是 GIS 中的常见需求。但是，这些值实际与基础投影设置有关。计量框架允许您控制这些参数，您可以选择：

- the *Ellipsoid*, on which distance, area and angle calculations are entirely based; it can be:
  - **None/Planimetric**: returned values are in this case cartesian measurements. This option can be set as default for new projects from the *Settings* ► *Options* ► *CRS Handling* menu
  - **自定义**: 您需要设置长半轴和短半轴的值。
  - 或预定义列表中的现有椭球体 (Clarke 1866, Clarke 1880 IGN, New International 1967, WGS 84...)。
- the *Units for distance measurements* for length and perimeter, and the *Units for area measurements*. These settings default to their corresponding *global options* and can be manually changed for the current project. They are used in:
  - 属性表字段更新栏
  - 字段计算器的计算
  - *Identify tool* derived length, perimeter and area values
  - *measure dialog*
  - *scale bar decoration*

> **Note**
>
> To ensure measurement units (for distance and area) always reflect the units of the current project's CRS, select *Map Units* for these options.

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

    Note

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

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

Note

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

The *Coordinate and Bearing display* allows you to customize the display of:

- the coordinates shown in the *Coordinates* box on QGIS status bar and in the *Derived* section of the Identify features tool's results
- the bearing value displayed in the status bar for the map canvas panning direction and by the Measure bearing tool.

Available parameters are:

- *Display coordinates using* either:
  - `Map Units`, based on the project CRS
  - `Map Geographic (degrees)`: based on the project CRS if it is of geographic type, otherwise uses its associated geographic CRS. This is helpful e.g., for non-earth celestial bodies.
  - or `Custom Projection Units`: allows to rely on any CRS you desire for coordinates display
- In the *Coordinate CRS* option, you can view or define the CRS to use depending on your display mode.
- *Coordinate format*: you can configure it as `Decimal Degrees`, `Degrees, Minutes` or `Degrees, Minutes, Seconds`, and whether it should display:

  - ☐ *Show directional suffix*
  - ☐ *Show leading zeros for minutes and seconds*
  - ☐ *Show leading zeros for degrees*
  - ☐ 显示尾随零

- *Coordinate precision*: the number of decimal places can be automatic (derived from the type of CRS) or set manually
- *Coordinate order*: you can opt to display the coordinates in the native order of the CRS (`Default`) or switch it to either `Easting, Northing (Longitude, Latitude)` or `Northing, Easting (Latitude, Longitude)` order
- *Bearing format* possible values are `0 to 180°, with E/W suffix`, `-180 to +180°` or `0 to 360°`. The number of *Decimal places* as well as whether to *Show trailing zeros* can be set.

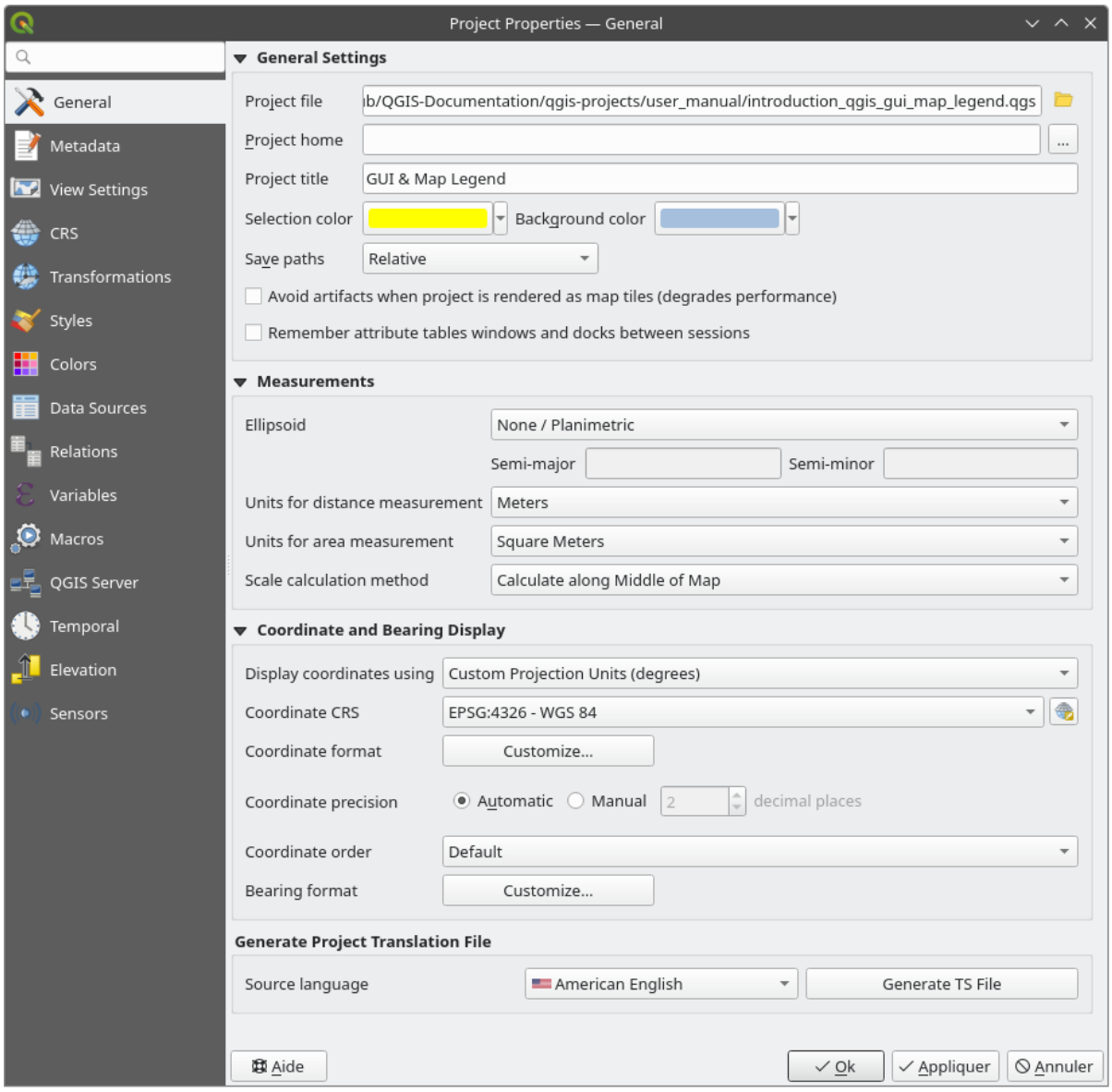


Fig. 5.32: 工程属性对话框的通用选项卡

5.3.2 元数据属性

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

5.3.3 View Settings

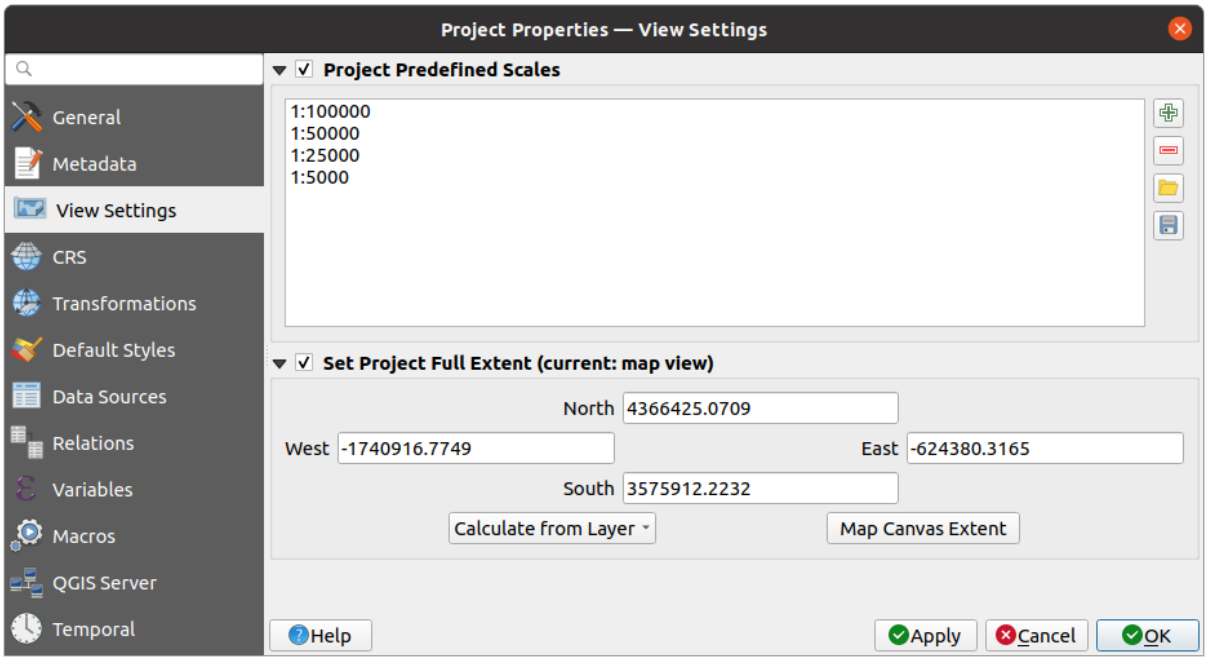


Fig. 5.33: View Settings tab of the Project Properties dialog

The *View Settings* tab provides means to control the project map canvas. You can:

- set *Project predefined scales*: the list of scales to display in scale-related drop-down widgets, such as the status bar *Scale*, the visibility scales selector or secondary 2D map view settings,... in replacement of the global *predefined scales*.
- *Set Project full Extent*: this extent will be used instead of the extent of all layers when zooming to full map extent ( ). It's useful when a project contains web layers/national layers/global layers yet the actual area of interest for the project is a smaller geographic area. The project full extent coordinates can be set with the *extent selector* widget.

5.3.4 CRS 属性

Note

有关 QGIS 如何处理工程投影的更多信息，请阅读使用投影 专用章节。

CRS 选项卡可帮助您设置要在此工程中使用的坐标参照系。它可以是：

- *无 CRS (或未知/非地球投影)*: 图层是根据其原始坐标绘制的
- 或者现有坐标参照系，可以是 *地理*, *投影* 或 *用户自定义*。添加到工程中的图层会动态转换到此 CRS，以便叠加它们，而不考虑其原始 CRS。

5.3.5 变换属性

变换选项卡通过配置要应用于当前工程的基准面变换首选项，帮助您控制图层重投影设置。照例，这些设置会覆盖任何相应的全局设置。更多信息，请参阅基准变换 。

5.3.6 Styles Properties

Under *Styles* tab, you can configure symbols and colors inherent to the project, allowing to safely share the project among different machines.

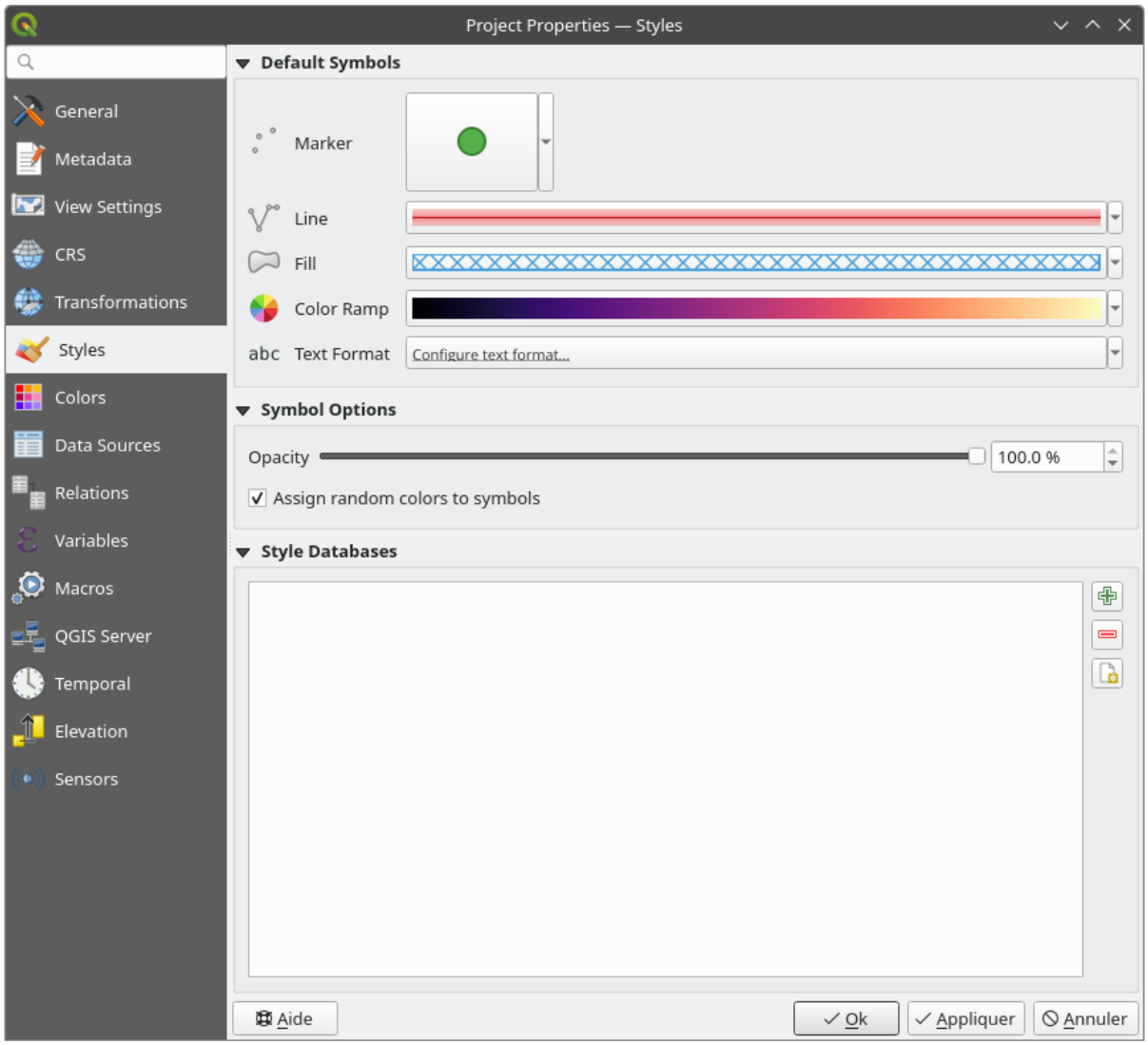


Fig. 5.34: Styles tab

Default symbols

The *Default Symbols* group lets you control how new layers will be drawn in the project when they do not have an existing `.qml` style defined. You can set *Marker*, *Line*, *Fill* to apply depending on the layer geometry type as well as default *Color Ramp* and *Text Format* (e.g., when enabling labeling). Any of these items can be reset using the *Clear* entry from the corresponding drop-down widget.

Symbol options

In the *Symbol options* group, you can:

- 将默认 不透明度应用于新建图层
- 给符号指派随机颜色，修改符号填充颜色，从而避免对所有图层进行相同的渲染。

Style Database

A style database in QGIS is a structured repository designed to store symbols, text formats, and other styling elements. It serves as a centralized location where you can organize and manage your symbology resources efficiently. You can create a dedicated style database for a specific client, housing symbols tailored to that client's need. This ensures a clean and organized approach, without cluttering the default style database. In multi-user environments, it's possible to store project-specific styles in a shared location. By linking a project to these styles, all users within the project gain access to common symbology, eliminating the need for manual import or updates to individual local style database. Storing symbols in a project file offers a practical solution, preventing users from overcrowding their global style database with project-specific symbology. This approach guarantees that other users working on the same project immediately have access to all the necessary symbology upon loading the project.

In the *Style Database* section you can choose to *Add* or *Remove* style database or you can *Create new style database*. When you add or remove a style database in this section, the changes will be automatically reflected in the *Symbology Properties*.

5.3.7 Colors properties

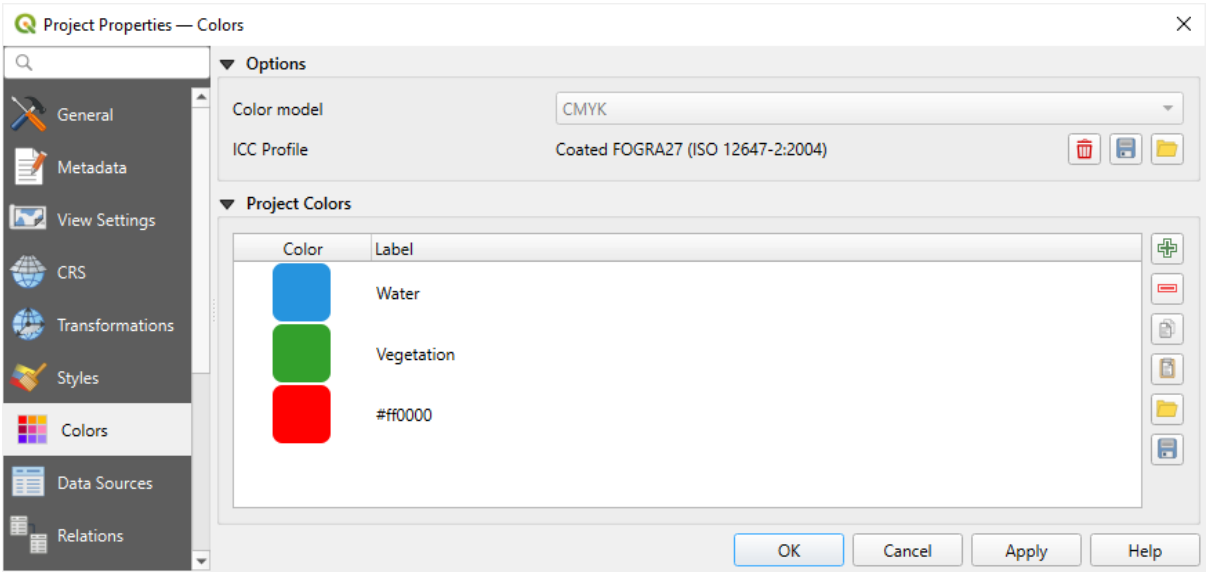


Fig. 5.35: Colors tab

In the *Colors* tab, you can select the *Color model*, i.e., whether selecting a color should by default be defined as `RGB` or `CMYK`. Any color defined in a different color model than the one specified here will be converted to this color model when exporting a layout.

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

Note

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

还有附加部件，您可以在其中为正在运行的工程定义指定的颜色。与全局颜色 一样，您可以：

- 添加或 移除颜色
- 复制或 粘贴颜色
- 导入或 导出颜色集从/到 `.gpl` 文件。

双击列表中的颜色，在选择颜色 对话框中对其进行调整或替换，也可以通过双击 `标注`列对其进行重命名。

这些颜色被认定为 `工程颜色`，并作为颜色控件 的一部分列出。

Tip

使用工程颜色快速分配和更新颜色控件

工程颜色可以使用其标注来引用，并且所使用的颜色控件与之绑定。这意味着您不必为许多属性设置相同的颜色而重复替换，为了避免繁琐的更新，您可以：

1. 将颜色定义为工程颜色
2. 单击要设置颜色属性旁边的数据定义重写控件
3. 将鼠标悬停在 `颜色` 菜单上并选择 `工程颜色`。然后，属性被指定表达式 `project_color('color_label')`，颜色控件反映该颜色。
4. 根据需要重复步骤 2 和 3
5. 每次更新工程颜色，所做的更改就会反映在使用工程颜色的任何地方。

5.3.8 数据源属性

In the *Data Sources* tab, you can:

- *Transaction mode*, defines how edits are sent to the data provider:
  - *Local Edit Buffer*: edits are buffered locally and sent to the provider when toggling layer editing mode or clicking *Save edits*.
  - *Automatic Transaction Groups*: on supported datasources (postgres and geopackage databases) the edit state of all tables that originate from the same database are synchronized and executed in a server side transaction. Also, instead of buffering edit changes locally, they are directly sent to a transaction in the database which gets committed when toggling layer editing mode or clicking *Save edits*.

  - *Buffered Transaction Groups*: all editable layers, disregarding from which provider, are toggled synchronously and all edits are saved in a local edit buffer. Saving changes is executed within a single transaction on all layers (per provider).

  Note that you can change this option only if no layer is being edited in the project.

- *Evaluate default values on provider side*: When adding new features in a PostgreSQL table, fields with default value constraint are evaluated and populated at the form opening, and not at the commit moment. This means that instead of an expression like `nextval('serial')`, the field in the *Add Feature* form will display expected value (e.g., `25`).
- *Remember editable layer status between sessions*: makes sure that all layers that are editable in a project will be remembered as such when saving the project, as well as making sure that those layers are immediately made editable whenever the project is restored.
- Configure the *Layers Capabilities*, i.e.:
  - Set (or disable) which layers are `identifiable`, i.e., will respond to the *identify tool*. By default, layers are set queryable.
  - Set whether a layer should appear as `read-only`, meaning that it can not be edited by the user, regardless of the data provider's capabilities. Although this is a weak protection, it remains a quick and handy configuration to avoid end-users modifying data when working with file-based layers.
  - Define which layers are `searchable`, i.e., could be queried using the *locator widget*. By default, layers are set searchable.
  - Define which layers are defined as `required`. Checked layers in this list are protected from inadvertent removal from the project.
  - Define which layers are `private`, i.e., hidden from the *Layers* panel. This is meant for accessory layers (basemap, join, lookups for value-relations, most probably aspatial layers, ...) that you still need in a project but you don't want them to pollute the legend tree and other layer selection tools. If set visible, they are still displayed in the map canvas and rendered in the print layout legend. Use the *Filter legend* ► *Show private layers* option in the *Layers* panel top toolbar to temporarily turned them on for any interaction.

  The *Layers Capabilities* table provides some convenient tools to:

  - Select multiple cells and press *Toggle Selection* to have them change their checkbox state;
  - *Show spatial layers only*, filtering out non-spatial layers from the layers list;
  - *Filter layers...* and quickly find a particular layer to configure.
- Under the *Advanced Settings* group, you can select *Trust project when data source has no metadata*: To speed up project loading by skipping data checks. Useful in QGIS Server context or in projects with huge database views/materialized views. The extent of layers will be read from the QGIS project file (instead of data sources) and when using the PostgreSQL provider the primary key unicity will not be checked for views and materialized views.

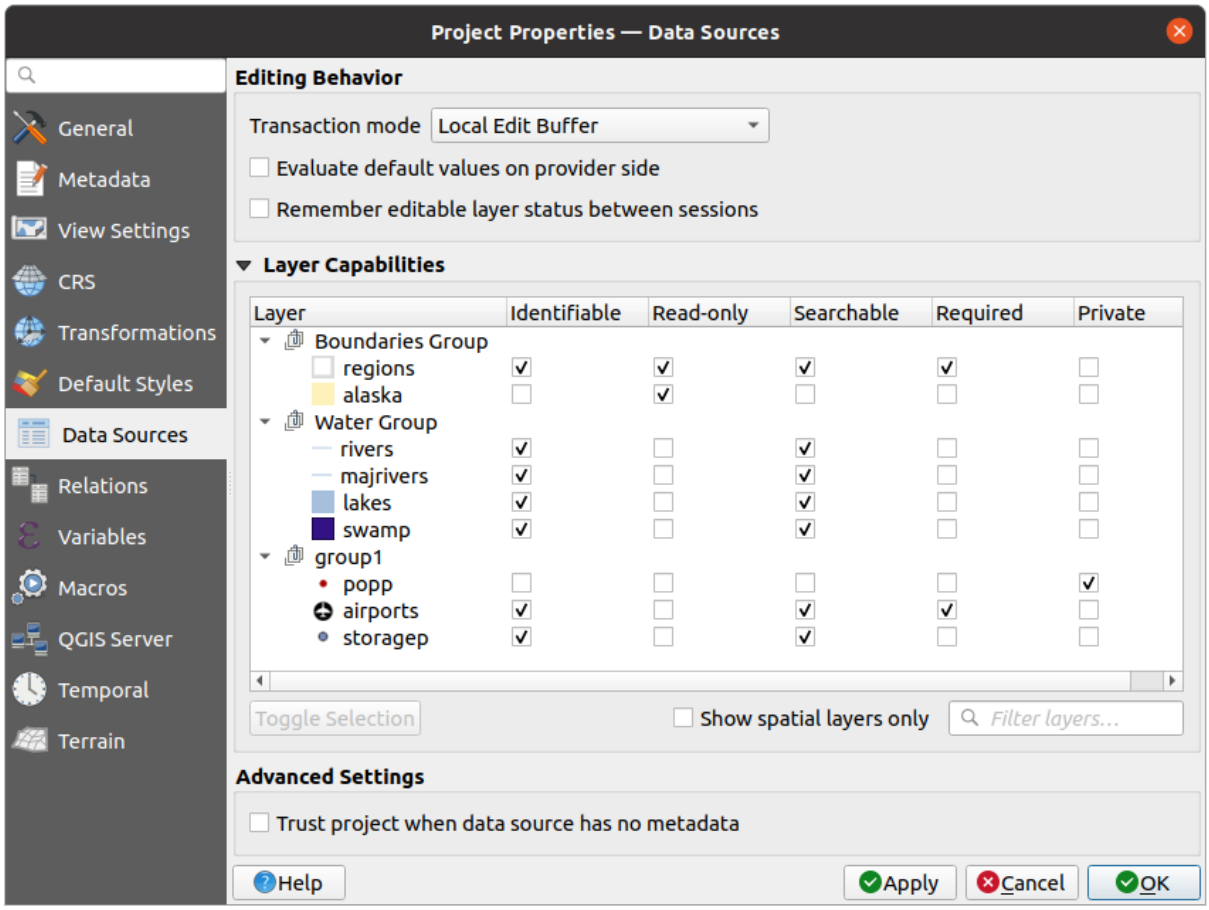


Fig. 5.36: Data Sources tab

5.3.9 Relations Properties

The *Relations* tab is used to define relations between layers. The relations can be of one to one, many to many or polymorphic type. They are defined in the project properties dialog. Once relations exist for a layer, a new user interface element in the form view (e.g., when identifying a feature and opening its form) will list the related entities. This provides a powerful way to express e.g., the inspection history on a length of pipeline or road segment. You can find out more about relations support in section *Setting relations between multiple layers*.

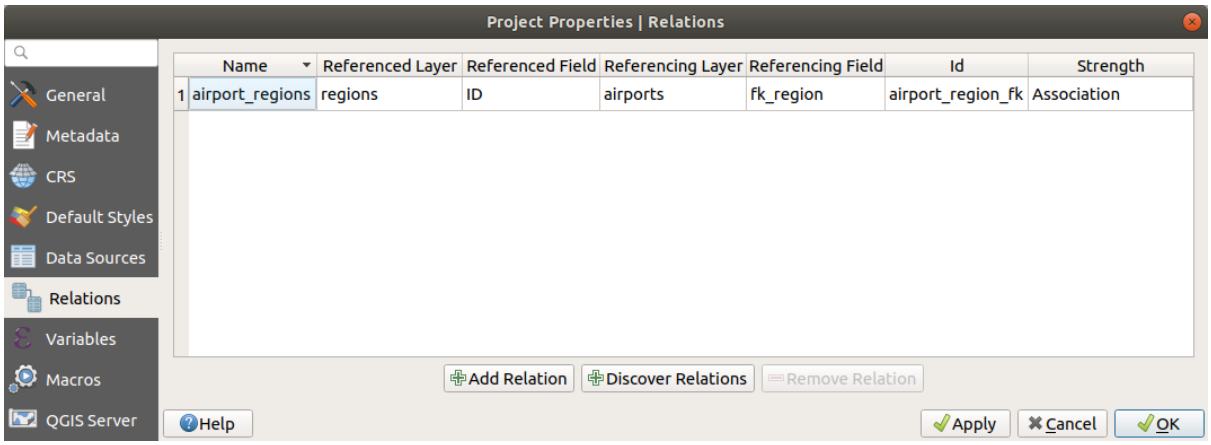


Fig. 5.37: Relations tab

5.3.10 Variables Properties

The *Variables* tab lists all the variables available at the project's level (which includes all global variables). Besides, it also allows the user to manage project-level variables. Click the button to add a new custom project-level variable. Likewise, select a custom project-level variable from the list and click the button to remove it. More information on variables usage in the General Tools 将值存储在变量中 section.

5.3.11 Macros Properties

The *Macros* tab is used to edit Python macros for projects. Currently, only three macros are available: `openProject()`, `saveProject()` and `closeProject()`.

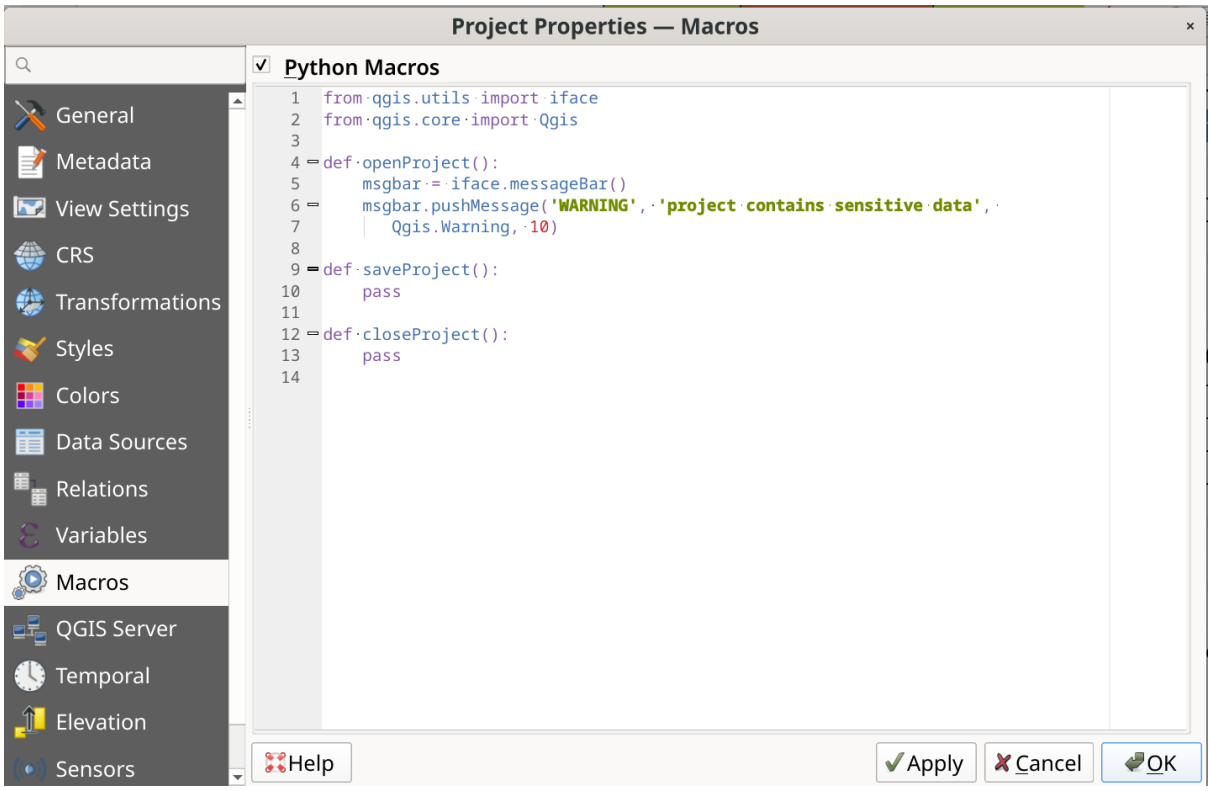


Fig. 5.38: Macro settings

5.3.12 QGIS Server Properties

The *QGIS Server* tab allows you to configure your project in order to publish it online. Here you can define information about the QGIS Server WMS and WFS capabilities, extent and CRS restrictions. More information available in section Creatingwmsfromproject and subsequent.

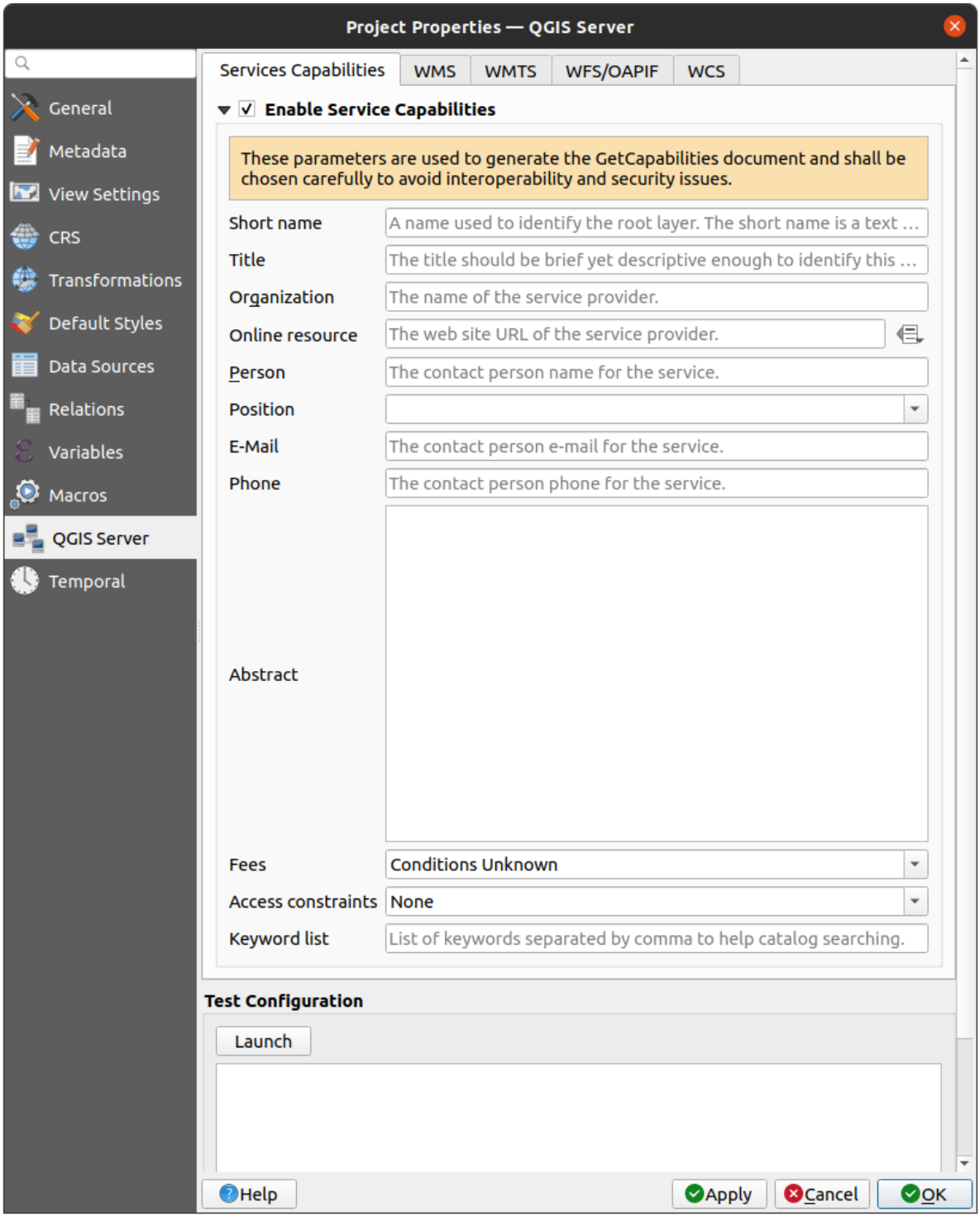


Fig. 5.39: QGIS Server settings

5.3.13 Temporal Properties

The *Temporal* tab is used to set the temporal range of your project, either by using manual *Start date* and *End date* inputs or by calculating it from the current project temporal layers. The project time range can then be used in the *Temporal controller panel* to manage the map canvas *temporal navigation*.

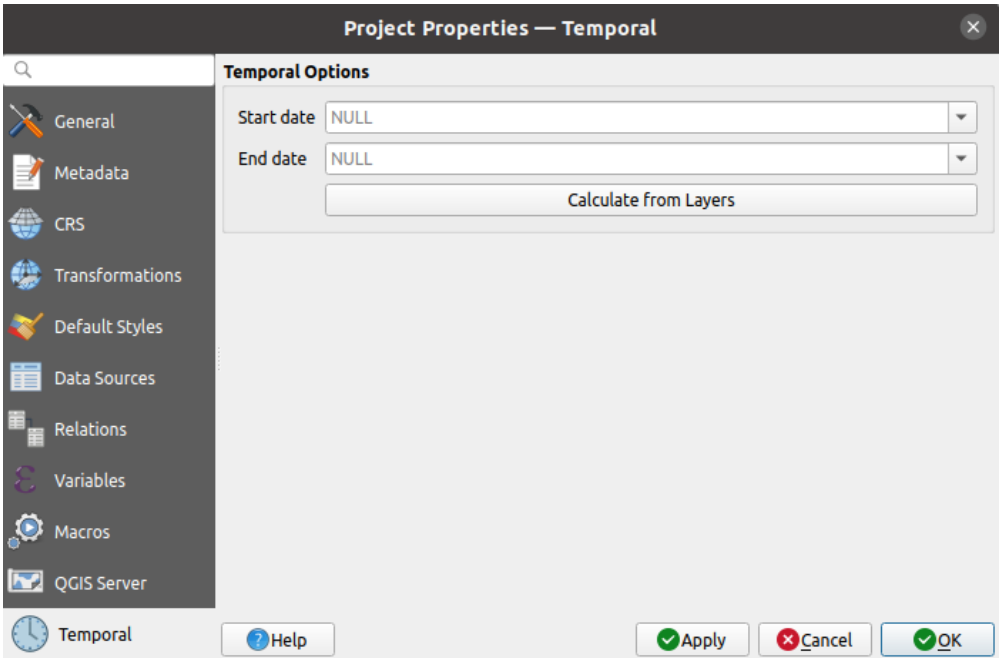


Fig. 5.40: Project Temporal tab

5.3.14 Elevation Properties

The *Elevation* tab helps you configure default settings for the terrain and elevation. When any new *3d map* is created in the project, the map will default to using the same terrain settings as are defined for the project. The project elevation settings will also be respected by the *elevation profile* tool.

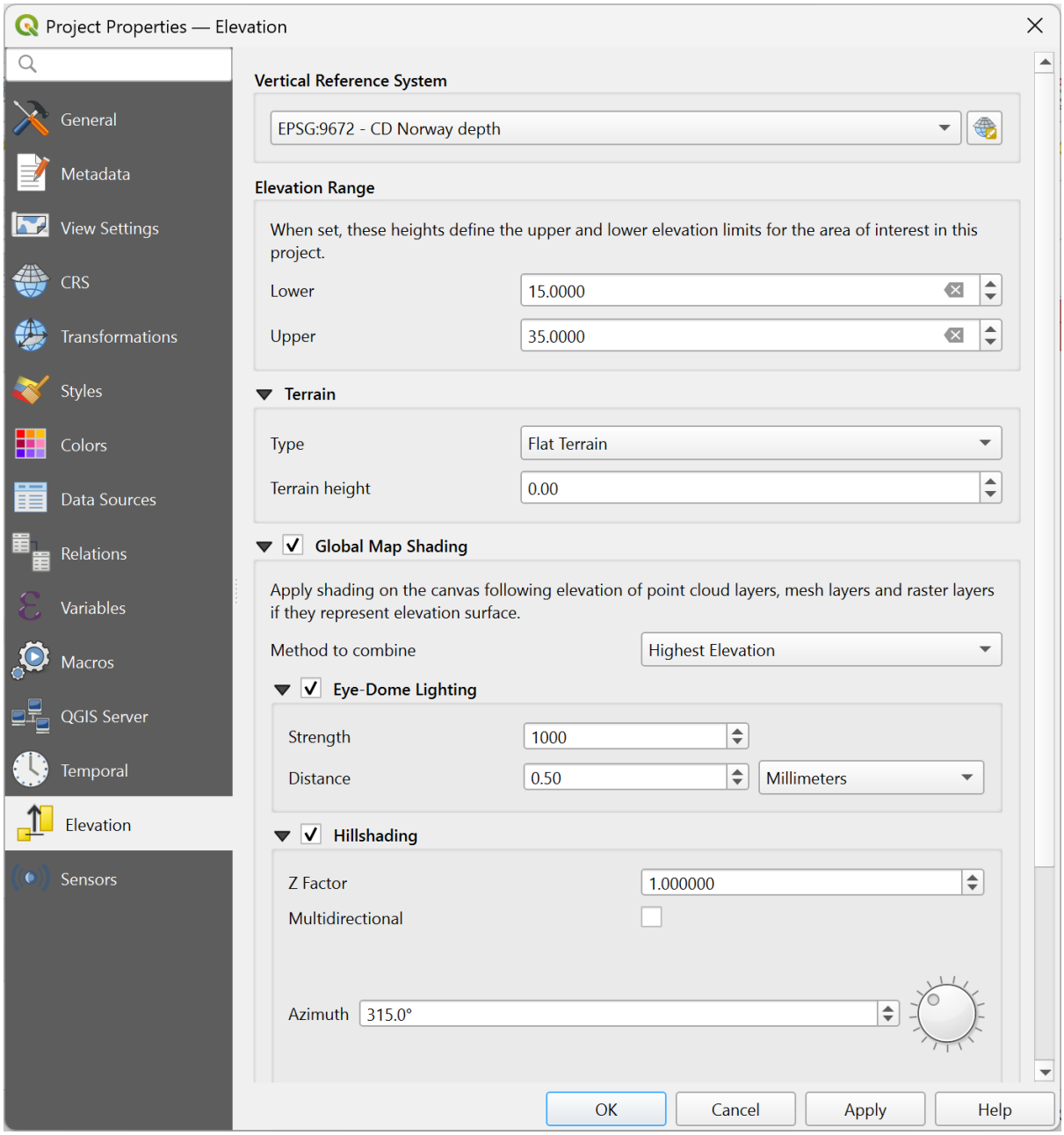


Fig. 5.41: Project Elevation tab

- *Vertical Reference System*: If the CRS of your project is compound one (including a Z dimension), then the vertical CRS used for the project will be the vertical component of the project CRS. If your project CRS is horizontal, then you can select a specific vertical CRS by clicking on the Select CRS.
- *Elevation Range*: helps you define the *Lower* and *Upper* limits for the *elevation controller* in the project.
- *Terrain* and elevation options are available for:
  - *Flat terrain* with *Terrain height* setting
  - *DEM (Raster Layer)*: with setting for defining the *Raster layer*, a *Vertical scale* factor to apply to band values and a vertical *Offset*
  - *Mesh*: with setting for defining the *Mesh layer*, a *Vertical scale* factor to apply to vertices Z value and a vertical *Offset*

  These settings can be overwritten from the 3D map *configuration dialog*.

- With *Global map shading* settings, you apply a global shading effect to the map, based on the elevation of all the active layers that have elevation properties enabled, including:
  - *raster layers*
  - *mesh layers*
  - *point cloud layers*

  Method to combine: the elevation value at any particular position for all the relevant layers are combined, and the considered elevation is chosen depending on one of these methods:
  - *Highest elevation* value
  - *Based on layer's order*: the elevation on the topmost layer in the layer tree is considered.

  Supported shading options are:
  - *Eye-dome lighting*: applies shading effects to the map canvas for a better depth rendering. Following parameters can be controlled:
    - *Strength*: increases the contrast, allowing for better depth perception
    - *Distance*: represents the distance of the used pixels off the center pixel and has the effect of making edges thicker.
  - *Hillshading*, shaping some reliefs on the map using shading (levels of gray):
    - *Z Factor*: Scaling factor for the elevation value
    - *Multidirectional*: Specify if multidirectional hillshading is to be used
    - *Azimuth*: The azimuth of the light source
    - *Altitude*: The elevation angle of the light source

> **Note**
>
> A shortcut to *Global map shading* properties is available through the *Layer Styling* panel.

5.3.15 Sensors Properties

The *Sensors* tab is used to configure sensors and toggle their connection status. When active, sensors will passively collect data in the background and make available their latest data to expressions and python scripts.

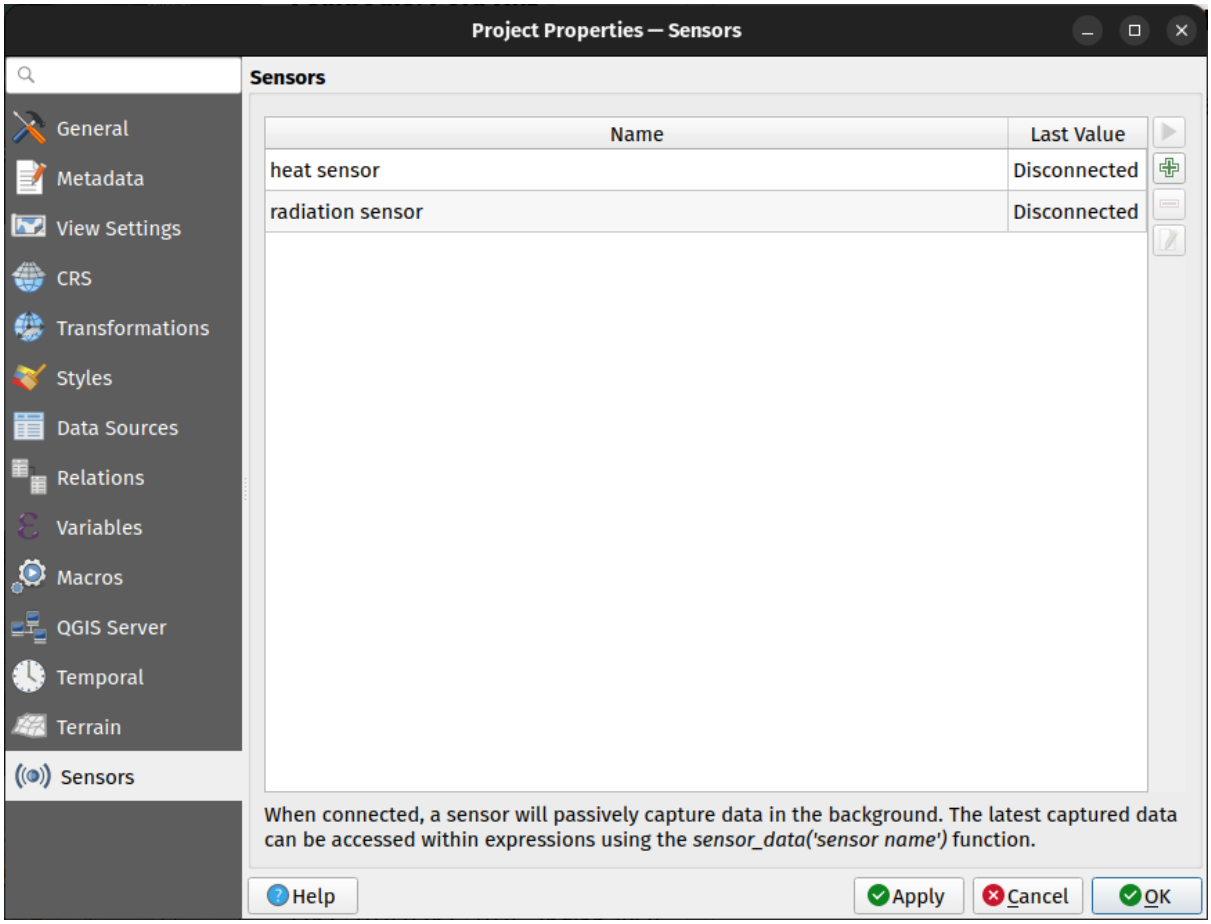


Fig. 5.42: Project Sensors tab

To add a new sensor, click on the button. A settings sub-panel will open and allow you to configure:

- the *Sensor name*: used to retrieve sensor values in expressions and python scripts;
- the *Sensor type*: TCP, UDP, serial port, etc.; and
- additional type-specific details (e.g., host name and port).

  For serial port sensors, comboboxes with the system's available serial ports and connection baud rate are available for their selection. A *Data frame delimiter* can be used to parse returned data. In case multiple data frames are found in the last chunk of data pushed by the serial port, all the data frames are returned. You can then use an expression to further split the returned sensor last data value into individual frames (e.g., using *string_to_array* function).

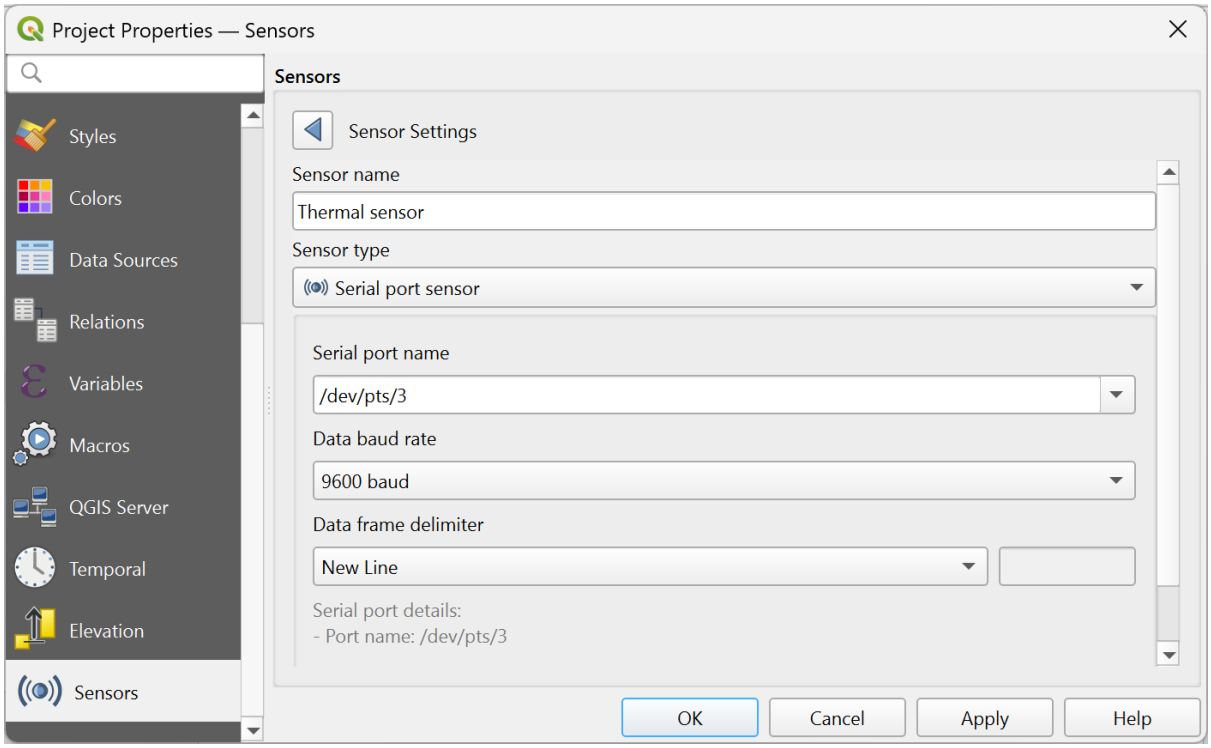


Fig. 5.43: Sensor Settings sub-panel

Once a sensor is configured, you can use the Start button to connect the sensor. Once active, the last collected data will be displayed in the sensors table's *Last value* column.

5.4 Customization

The *Customization* dialog lets you (de)activate almost every element in the QGIS user interface. This can be very useful if you want to provide your end-users with a 'light' version of QGIS, containing only the icons, menus or panels they need.

> **Note**
>
> Before your changes are applied, you need to restart QGIS.

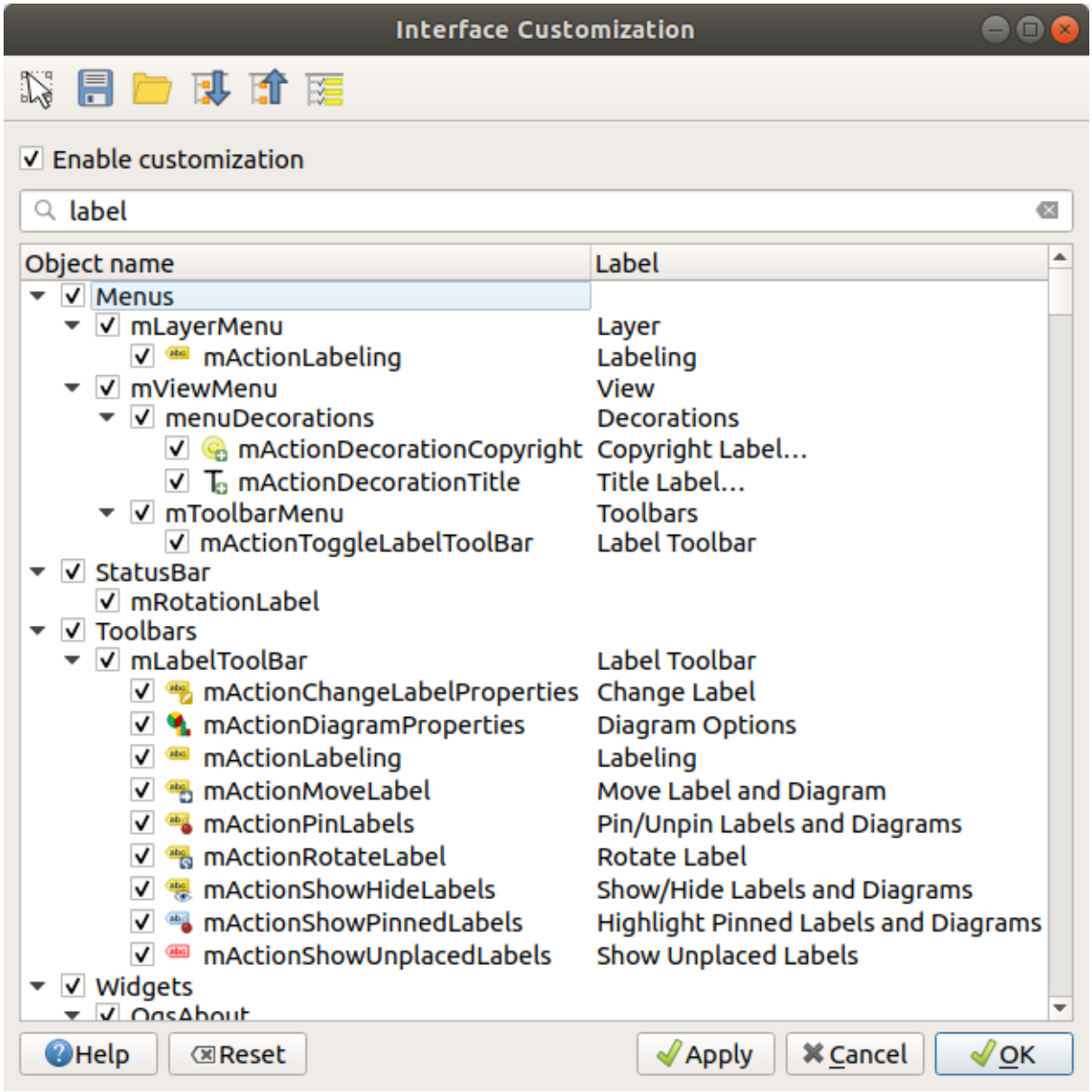


Fig. 5.44: The Customization dialog

Ticking the *Enable customization* checkbox is the first step on the way to QGIS customization. This enables the toolbar and the widget panel from which you can uncheck and thus disable some GUI items.

The configurable item can be:

- a **Menu** or some of its sub-menus from the 菜单栏
- a whole **Panel** (see 面板和工具栏)
- the **Status bar** described in 状态栏 or some of its items
- a **Toolbar**: the whole bar or some of its icons
- or any **widget** from any dialog in QGIS: label, button, combobox...

With Switch to catching widgets in main application, you can click on an item in QGIS interface that you want to be hidden and QGIS automatically unchecks the corresponding entry in the Customization dialog. You can also use the *Search* box to find items by their name or label.

Once you setup your configuration, click *Apply* or *OK* to validate your changes. This configuration becomes the one used by default by QGIS at the next startup.

The modifications can also be saved in a `.ini` file using Save To File button. This is a handy way to share a common QGIS interface among multiple users. Just click on Load from File from the destination computer in order to import the `.ini` file. You can also run *command line tools* and save various setups for different use cases as well.

Tip

Easily restore predefined QGIS

The initial QGIS GUI configuration can be restored by one of the methods below:

- unchecking *Enable customization* option in the Customization dialog or click the Check All button
- pressing the *Reset* button in the **Settings** frame under *Settings* ► *Options* menu, *System* tab
- launching QGIS at a command prompt with the following command line `qgis --nocustomization`
- setting to `false` the value of *UI* ► *Customization* ► *Enabled* variable under *Settings* ► *Options* menu, *Advanced* tab (see the *warning*).

In most cases, you need to restart QGIS in order to have the change applied.

5.5 Keyboard shortcuts

QGIS provides default keyboard shortcuts for many features. You can find them in section 菜单栏. Additionally, the menu option *Settings* ► *Keyboard Shortcuts...* allows you to change the default keyboard shortcuts and add new ones to QGIS features.

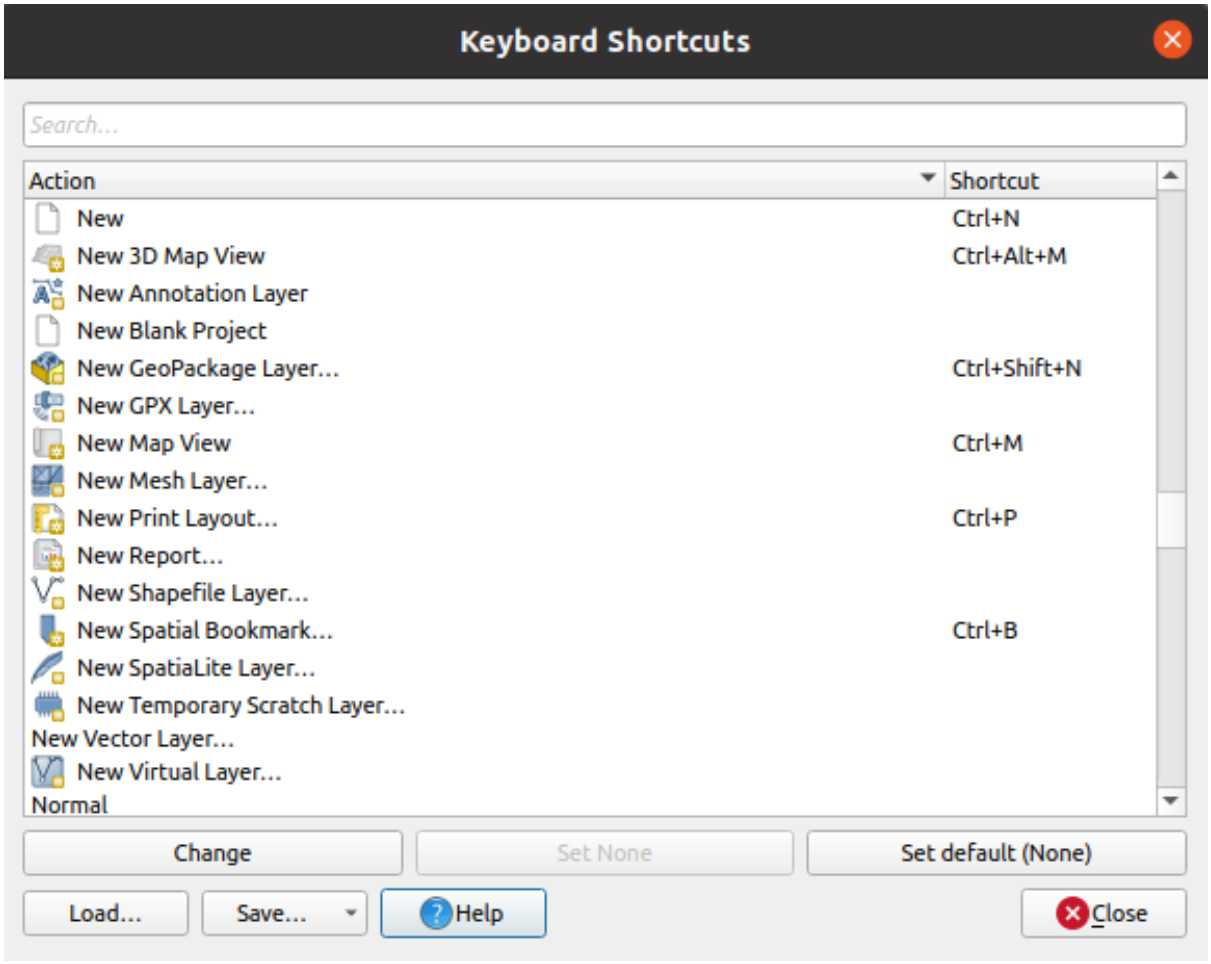


Fig. 5.45: Define shortcut options

Configuration is very simple. Use the search box at the top of the dialog to find a particular action, select it from the list and click on :

- *Change* and press the new combination you want to assign as new shortcut
- *Set None* to clear any assigned shortcut
- or *Set Default* to backup the shortcut to its original and default value.

Proceed as above for any other tools you wish to customize. Once you have finished your configuration, simply *Close* the dialog to have your changes applied. You can also *Save* the changes either as an `.XML` file with only the User Shortcuts or with all Shortcuts or as an `.PDF` file with all Shortcuts and *Load* them into another QGIS installation.

5.6 Running QGIS with advanced settings

5.6.1 Command line and environment variables

We've seen that *launching QGIS* is done as for any application on your OS. QGIS provides command line options for more advanced use cases (in some cases you can use an environment variable instead of the command line option). To get a list of the options, enter `qgis --help` on the command line, which returns:

```
QGIS is a user friendly Open Source Geographic Information System.
Usage: /usr/bin/qgis.bin [OPTION] [FILE]
  OPTION:
```

(continues on next page)

(continued from previous page)

```
        [-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
```

Tip

Example Using command line arguments

You can start QGIS by specifying one or more data files on the command line. For example, assuming you are in the `qgis_sample_data` directory, you could start QGIS with a vector layer and a raster file set to load on startup using the following command: `qgis ./raster/landcover.img ./gml/lakes.gml`

--version

This option returns QGIS version information.

--snapshot

This option allows you to create a snapshot in PNG format from the current view. This comes in handy when you have many projects and want to generate snapshots from your data, or when you need to create snapshots of the same project with updated data.

Currently, it generates a PNG file with 800x600 pixels. The size can be adjusted using the `--width` and `--height` arguments. The filename can be added after `--snapshot`. For example:

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

--width

This option returns the width of the snapshot to be emitted (used with `--snapshot`).

--height

This option returns the height of the snapshot to be emitted (used with `--snapshot`).

--lang

Based on your locale, QGIS selects the correct localization. If you would like to change your language, you can specify a language code. For example, `qgis --lang it` starts QGIS in Italian localization.

--project

Starting QGIS with an existing project file is also possible. Just add the command line option `--project` followed by your project name and QGIS will open with all layers in the given file loaded.

--extent

To start with a specific map extent use this option. You need to add the bounding box of your extent in the following order separated by a comma:

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

This option probably makes more sense when paired with the `--project` option to open a specific project at the desired extent.

--nologo

This option hides the splash screen when you start QGIS.

--noversioncheck

Skip searching for a new version of QGIS at startup.

--noplugins

If you have trouble at start-up with plugins, you can avoid loading them at start-up with this option. They will still be available from the Plugins Manager afterwards.

--nocustomization

Using this option, any existing *GUI customization* will not be applied at startup. This means that any hidden buttons, menu items, toolbars, and so on, will show up on QGIS start up. This is not a permanent change. The customization will be applied again if QGIS is launched without this option.

This option is useful for temporarily allowing access to tools that have been removed by customization.

--skipbadlayers

Using this option, you can avoid QGIS prompting the *Handle Unavailable Layers* dialog at startup. The project file is loaded, with missing layers kept as unavailable. More details on the topic at 处理损坏的文件路径.

--customizationfile

Using this option, you can define a UI customization file, that will be used at startup.

--globalsettingsfile

The equivalent environment variable is `QGIS_GLOBAL_SETTINGS_FILE`.

Using this option, you can specify the path for a Global Settings file (`.ini`), also known as the Default Settings. The settings in the specified file replace the original inline default ones, but the user profiles' settings will be set on top of those.

QGIS looks for the default global settings file in the following order and only the first found file will be used:

- path specified by the commandline parameter
- path defined by the environment variable
- the AppDataLocation folder, where persistent application data can be stored; it is managed by the user or system administrator and is not touched by installer and does not require any additional setup like passing commandline parameters or settings environment variable. Depending on the OS, it is:
  - `$HOME/.local/share/QGIS/QGIS3/`
  - `C:\Users\<username>\%AppData%\Roaming\QGIS\QGIS3\`
  - `$HOME/Library/Application Support/QGIS/QGIS3/`
- the installation directory, i.e., `your_QGIS_package_path/resources/qgis_global_settings.ini`.

Presently, there's no way to specify a file to write settings to; therefore, you can create a copy of an original settings file, rename, and adapt it.

Setting the `qgis_global_setting.ini` file path to a network shared folder, allows a system administrator to change global settings and defaults in several machines by only editing one file.

--authdbdirectory

This option is similar to `--globalsettingsfile`, but defines the path to the directory where the authentication database will be stored and loaded.

--code

This option can be used to run a given python file directly after QGIS has started.

For example, when you have a python file named `load_alaska.py` with following content:

```
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)
```

Assuming you are in the directory where the file `load_alaska.py` is located, you can start QGIS, load the raster file `landcover.img` and give the layer the name 'Alaska' using the following command:

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

--py-args

This option allows to pass arguments to the script provided via the *--code* parameter and more generally to each python execution. After `--py-args`, it is not possible to pass any other parameter than `--` to QGIS. All arguments after `--py-args` till `--` are passed over to the python interpreter and ignored by QGIS.

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

In the above code, `test.py` will have this content in `sys.argv`: ['/home/user/test.py', '--specialScriptArguments', 'a text arg', 'and another arg']. `layer1` and `layer2` will be normally handled by QGIS as layers to load.

--defaultui

On load, **permanently resets** the user interface (UI) to the default settings. This option will restore the panels and toolbars visibility, position, and size. Unless it's changed again, the default UI settings will be used in the following sessions.

Notice that this option doesn't have any effect on *GUI customization*. Items hidden by GUI customization (e.g., the status bar) will remain hidden even using the `--defaultui` option. See also the `--nocustomization` option.

--hide-browser

On load, hides the *Browser* panel from the user interface. The panel can be enabled by right-clicking a space in the toolbars or using the *View* ► *Panels* (*Settings* ► *Panels* in Linux KDE).

Unless it's enabled again, the Browser panel will remain hidden in the following sessions.

--dxf-*

These options can be used to export a QGIS project into a DXF file. Several options are available:

- *--dxf-export*: the DXF filename into which to export the layers;
- *--dxf-extent*: the extent of the final DXF file;
- *--dxf-symbology-mode*: several values can be used here: `none` (no symbology), `symbollayer` (Symbol layer symbology), `feature` (feature symbology);
- *--dxf-scale-denom*: the scale denominator of the symbology;
- *--dxf-encoding*: the file encoding;
- *--dxf-map-theme*: choose a *map theme* from the layer tree configuration.

--take-screenshots

Takes screenshots for the user documentation. Can be used together with `--screenshots-categories` to filter which categories/sections of the documentation screenshots should be created (see QgsAppScreenShots::Categories).

--profile

Loads QGIS using a specific profile from the user's profile folder. If the named profile doesn't exist, it will be created. This option takes precedence over *user profile startup setting*.

--profiles-path

With this option, you can choose a path to load and save the profiles (user settings). It creates profiles inside a `{path}\profiles` folder, which includes settings, installed plugins, processing models and scripts, and so on.

This option allows you to, for instance, carry all your plugins and settings in a flash drive, or, for example, share the settings between different computers using a file sharing service.

The equivalent environment variable is `QGIS_CUSTOM_CONFIG_PATH`.

--version-migration

If settings from an older version are found (*e.g.*, the `.qgis2` folder from QGIS 2.18), this option will import them into the default QGIS profile.

--openclprogramfolder

Using this option, you can specify an alternative path for your OpenCL programs. This is useful for developers while testing new versions of the programs without needing to replace the existing ones.

The equivalent environment variable is `QGIS_OPENCL_PROGRAM_FOLDER`.

--

This option allows to pass following arguments as files to open in QGIS. They can be supported rasters or vectors files, QGIS layer definition files or project files.

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

The above code will start QGIS, open the `project` file and load `layer1` in it.

5.6.2 Deploying QGIS within an organization

If you need to deploy QGIS within an organization with a custom configuration file, first you need to copy/paste the content of the default settings file located in `your_QGIS_package_path/resources/qgis_global_settings.ini`. This file already contains some default sections identified by a block starting with `[]`. We recommend that you keep these defaults values and add your own sections at the bottom of the file. If a section is duplicated in the file, QGIS will take the last one from top to bottom.

You can change `allowVersionCheck=false` to disable the QGIS version check.

If you do not want to display the migration window after a fresh install, you need the following section:

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

If you want to add a custom variable in the global scope:

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

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

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

Not all settings can be prepared from QGIS GUI, but to discover the possibilities of the settings `.INI` file, we suggest that you set the config you would like in QGIS Desktop and then search for it in your `.INI` file located in your profile using a text editor.

Finally, you need to set the environment variable `QGIS_GLOBAL_SETTINGS_FILE` to the path of your customized file.

In addition, you can also deploy files such as Python macros, color palettes, layout templates, project templates… either in the QGIS system directory or in the QGIS user profile.

- Layout templates must be deployed in the `composer_templates` directory.
- Project templates must be deployed in the `project_templates` directory.
- Custom Python macros must be deployed in the `python` directory.

CHAPTER 6

使用投影

坐标参照系统，或简称 CRS，是一种将数值坐标与地球表面上的位置相关联的方法。QGIS 支持大约 7,000 个标准坐标参照系统，每个都有不同的用例、优缺点！为您的 QGIS 项目和数据选择合适的坐标参照系统可能是一项复杂的任务，但幸运的是 QGIS 可以帮助您完成选择，并让使用不同的坐标参照系尽可能透明和准确。

6.1 投影支持概述

QGIS 支持大约 7,000 个已知的坐标参照系。这些标准坐标参照系基于欧洲石油调查组 (EPSG) 和法国国家地理研究所 (IGNF) 定义的标准，并通过底层的 “Proj” 投影库在 QGIS 中提供。通常，这些标准投影是通过使用 “管理机构: 代码” 组合来识别的，其中管理机构是组织名称，例如 “EPSG” 或 “IGNF”，代码是与特定 CRS 关联的唯一编号。例如，常见的 WGS 84 经纬度坐标参照系的标识符为 “EPSG:4326“，而网络地图标准坐标参照系为 “EPSG:3857“。

自定义的、用户创建的坐标参照系存储在用户坐标参照系数据库中。有关管理自定义坐标参照系统的信息，请参阅自定义坐标参照系 部分。

6.2 图层坐标参照系

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

每当将图层加载到 QGIS 中时，QGIS 都会尝试自动确定该图层的正确坐标参照系。在某些情况下可能无法成功，例如当提供了一个不包含坐标系信息的图层时。当 QGIS 无法自动确定图层的正确坐标参照系时，您可以手动配置 QGIS：

1. 打开 设置 ► 选项... ► *CRS*

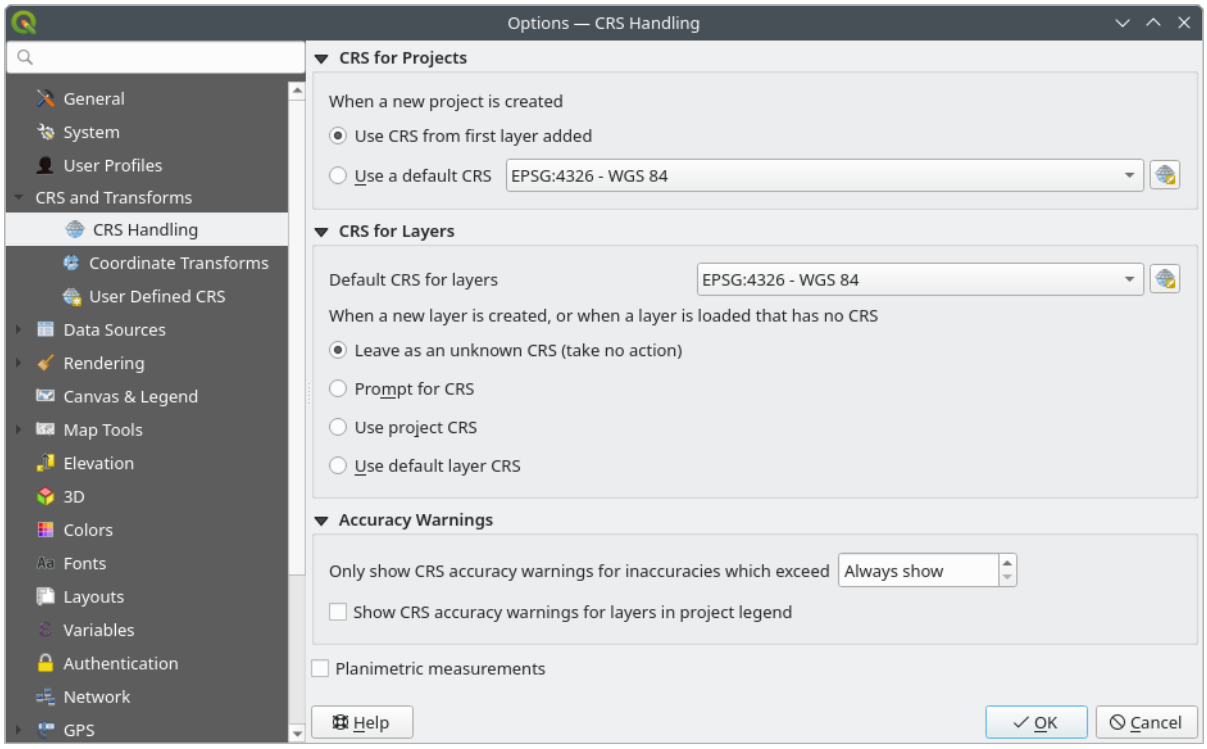


Fig. 6.1: QGIS 选项对话框中的 CRS 选项卡

2. 在:guilabel: 图层 *CRS* 选项组中，设置:guilabel:‘当创建新图层或加载没有 CRS 的图层时 ‘的操作。从以下选择其一：

   - 保留未知 *CRS (无动作)*：加载没有 CRS 的图层时，将不会提示选择 CRS，将 CRS 选择推迟到以后。一次加载很多图层时很方便。在 图层面板中这些图层旁边将会显示 图标。它们也将是无参照的，来自该图层的坐标被视为纯数字的、非地球的值，也就是说，当:ref:‘一个项目被设置为没有 CRS 时 <project_crs> ‘ ，所有图层的行为都是一样的。
   - 提示 *CRS* ： 它将提示您手动选择 CRS。选择正确的选择是至关重要的，因为一个错误的选择将把你的层放在地球表面的错误位置！有时，随附的元数据将描述层的正确 CRS，在其他情况下，您需要联系数据的原始作者以确定要使用的正确 CRS。
   - 使用工程 *CRS*
   - 使用默认图层 *CRS*, 就像设置上面的 默认 *CRS* 图层组合框

Tip

将相同的 CRS 分配给多个没有 CRS 或在一次操作中有错误 CRS 的层：

1. 在图层面板中选择图层
2. 使用快捷键 `Ctrl+Shift+C` ，或者右键单击选中图层以进入 CRS 选择选菜单 图层 *CRS* ► 设置图层 *CRS*
3. 找到并选择正确的 CRS
4. 点击 *OK* 现在检查图层属性对话框下的 源标签页的信息，图层的 CRS 已被正确地设置

请注意，在此设置中更改 CRS 并不会修改原始数据源，而只是改变了当前 QGIS 工程下 QGIS 对该图层原始坐标的解释。

6.3 投影坐标参照系

QGIS 中的每个项目都有一个相关的坐标参考系。项目 CRS 决定如何将数据从底层原始坐标投影到 QGIS 地图画布中呈现的平面地图。

QGIS 支持光栅和矢量数据的“实时动态”CRS 转换。这意味着，无论项目中特定映射层的底层 CRS 如何，它们总是会自动转换为为项目定义的公共 CRS。在幕后，QGIS 透明地将项目中包含的所有图层重新投影到项目的 CRS 中，这样它们就会在彼此相关的正确位置上呈现！

为您的 QGIS 项目选择适当的 CRS 非常重要。选择不合适的 CRS 可能会导致您的地图看起来扭曲，并且很难反映地貌的实际大小和位置。通常，在较小的地理区域内工作时，会有多个在特定国家或行政区内使用的标准 CRS。针对你正在绘制的区域，研究哪些 CRS 是合适或标准选择，并确保您的 QGIS 项目遵循这些标准，这一点非常重要。

默认情况下，QGIS 在创建每个新工程时都会使用一个全局默认投影。该默认 CRS 为“EPSG:4326“（也称为“WGS 84”），是一种基于经纬度的全球参考系统。您可以通过 设置 ► 选项... 中的 *CRS* 处理选项卡下的 当新工程创建时设置来更改此默认 CRS（参见 Fig. 6.1）。您可以选择在将第一个图层加载到新工程时自动将工程 CRS 设置为该图层的 CRS，也可以选择另一个 CRS 作为所有新建工程的默认 CRS。此选择将被保存，并在后续的 QGIS 会话中继续使用。

工程的 *CRS* 也可以通过:menuselection: 工程--> 工程属性... 的 *CRS* 选项卡来设置。也会显示在 QGIS 状态栏的最右下方。

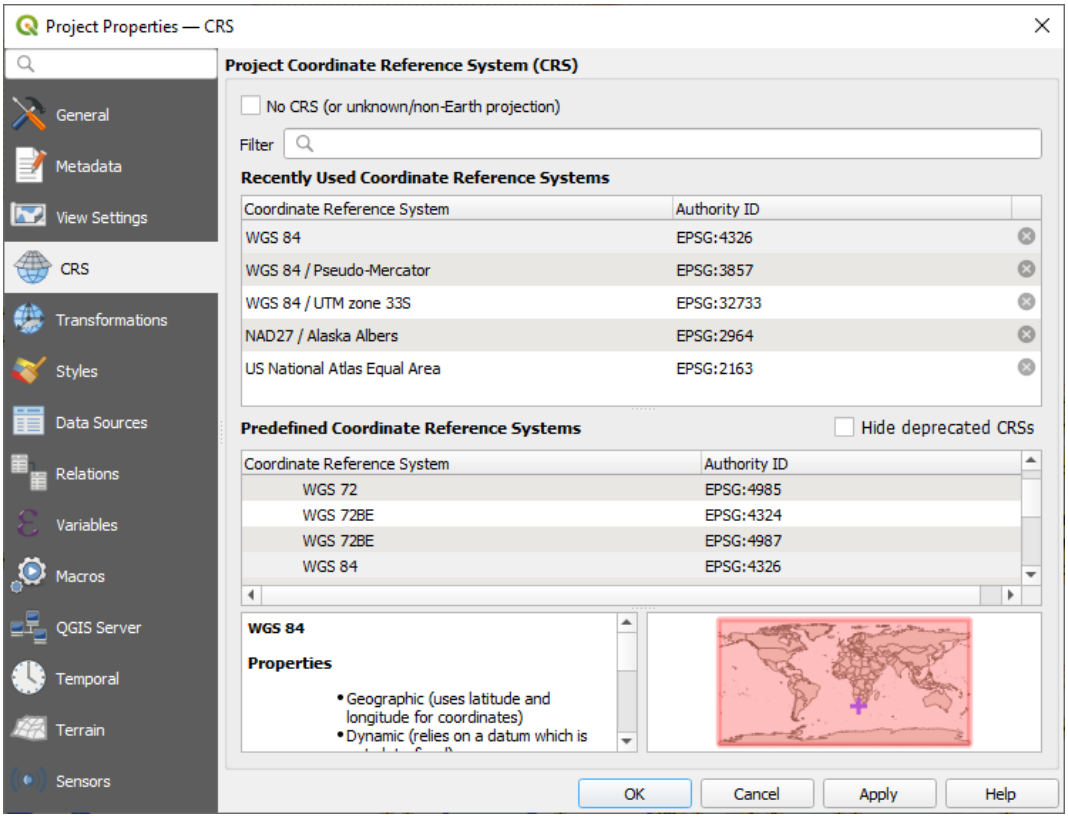


Fig. 6.2: 工程属性对话框

可用的选项包括：

- 无 *CRS (或未知/非地球投影)*：勾选此设置将禁用 QGIS 项目中的所有投影处理，导致所有图层和地图坐标被视为简单的二维笛卡尔坐标，与地球表面的位置无关。它可以用来猜测一个层的 CRS（基于其原始坐标），或者当 QGIS 用于非地球用途时，如角色扮演游戏地图，建筑地图或微观的东西。在这种情况下：
  - 在渲染层时不进行重投影：只是使用原始坐标绘制特征。

  - 椭球体被锁定并被迫“没有/平面” `None/Planimetric` 。
  - 距离、面积单位、坐标显示被锁定，强制为“未知单位”；所有的测量都以未知的地图单位进行，不可能进行转换。
- 或现有的坐标参考系统，可以是地理的、投影的或用户定义的。显示地球上 CRS 范围的预览，以帮助您选择合适的 CRS 范围。添加到项目中的图层被实时转换到这个 CRS，以便覆盖它们，而不管它们的原始 CRS 是什么。使用单位和椭球体设置是有意义的，你可以执行相应的计算。

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

Tip

从图层设置工程 CRS

你可以使用图层的 CRS 为工程分配一个 CRS：

1. 在 *图层面板*，右键单击要选择 CRS 的图层
2. 选择 *图层 CRS-> 将图层的 CRS 设为工程 CRS* 。

使用图层的 CRS 重新定义工程的 CRS，地图画布范围，坐标显示相应更新，工程中的所有图层都实时转换为新的工程 CRS。

6.4 坐标参照系选择器

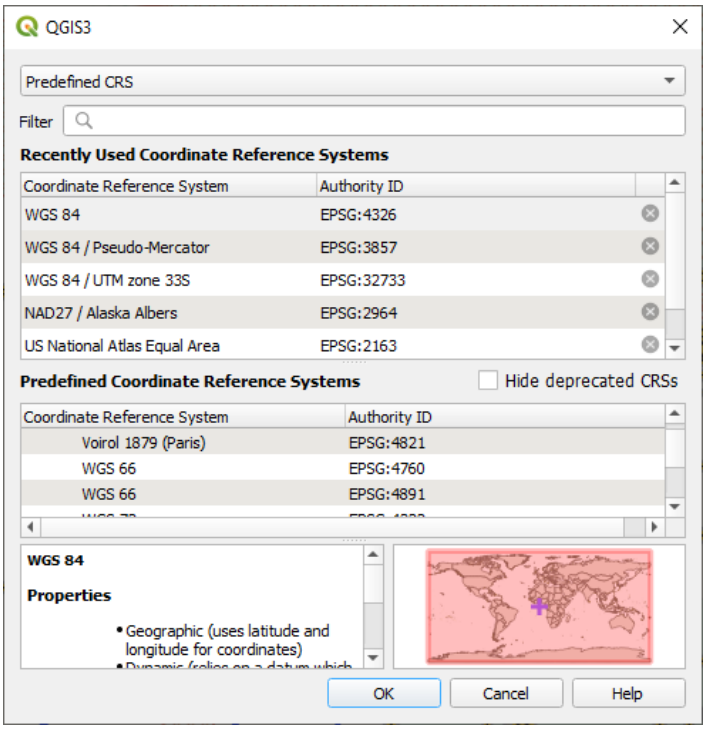


Fig. 6.3: CRS 选择器

此对话框可帮助您为工程或图层分配坐标参照系统，并提供一组投影数据库。对话框中的元素有：

- **过滤**: 如果你知道 EPSG 代码、标识符或坐标参照系的名称，您可以使用搜索功能来查找它。输入 EPSG 代码、标识符或名称。

- **最近使用的坐标参照系**：如果您在日常 GIS 工作中经常使用某些 CRS，这些 CRS 将显示在此列表中。点击其中一个项目，选择相关的 CRS。
- **预定义坐标参照系**：这是 QGIS 支持的所有 CRS 的列表，包括地理、投影和自定义坐标参考系统。请展开列表中适当的节点并选择 CRS，已预选活跃的 CRS。
- **PROJ 文本**：左下角是 PROJ 投影引擎使用的 CRS 字符串。此文本为只读，仅供参考。

如果你想从列表中删除最近使用的 CRS，可以通过以下几种方式完成：

- 按右侧的 从最近使用的 CRS 中移除 按钮。
- 选择要移除的 CRS 后，按 `Del` 键。
- 在用过的 CRS 上鼠标右键单击，并选择 从最近使用的 *CRS* 中移除选定的 *CRS* 来移除单个记录，或者 清除所有最近使用的 *CRS* 来移除所有曾用记录。

CRS 选择器还显示所选 CRS 有效使用的地理区域的粗略预览。许多 CRS 的设计仅用于小地理区域，您不应该在设计区域之外使用这些 CRS。每当从列表中选择 CRS 时，预览地图就会显示大致的使用区域（红色）。此外，此预览图还显示了当前主画布地图范围的指示器（紫色）。

6.5 自定义坐标参照系

如果 QGIS 不提供所需的坐标参照系，则可以定义自定义 CRS。要定义 CRS，从:menuselection: 设置菜单 中选择 自定义投影... . 自定义的 CRS 存储在 QGIS 用户数据库中。除了自定义 CRS 之外，该数据库还包含空间书签和其他自定义数据。

在 QGIS 中定义自定义 CRS 需要对 PROJ 投影库有很好的理解。首先，请参考美国地质调查局 Gerald I. Evenden 的 UNIX 环境的制图投影程序-用户手册。”Cartographic Projection Procedures for the UNIX Environment - A User’s Manual” by Gerald I. Evenden, U.S. Geological Survey Open-File Report 90-284, 1990 (可在 https://pubs.usgs.gov/of/1990/of90-284/ofr90-284.pdf 找到)。

本手册描述了 `proj` 和相关命令行实用程序的使用。与 `proj` 一起使用的制图参数在用户手册中有描述，与 QGIS 使用的参数相同。

用户自定义 *CRS*‘对话框只需要两个参数定义用户 CRS：

1. 描述性名称
2. 使用 PROJ 或 WKT 格式的制图参数

要创建一个新的 CRS：

1. 单击 添加新的 CRS 按钮
2. 输入描述性名称
3. 选择格式，它可能是 *Proj* 字符串或 *WKT*
4. 输入 CRS 参数。

> **Note**
>
> **首选以 WKT 格式存储 CRS 定义**
>
> 虽然都支持 `Proj String` 和 `WKT` 格式，但强烈建议以 WKT 格式存储投影定义。因此，如果可用的定义是 proj 格式，请选择该格式，输入参数，然后切换到 WKT 格式。QGIS 将把定义转换为稍后可以保存的 WKT 格式。

5. 点击 验证按钮以测试 CRS 定义是否为有效的投影定义。

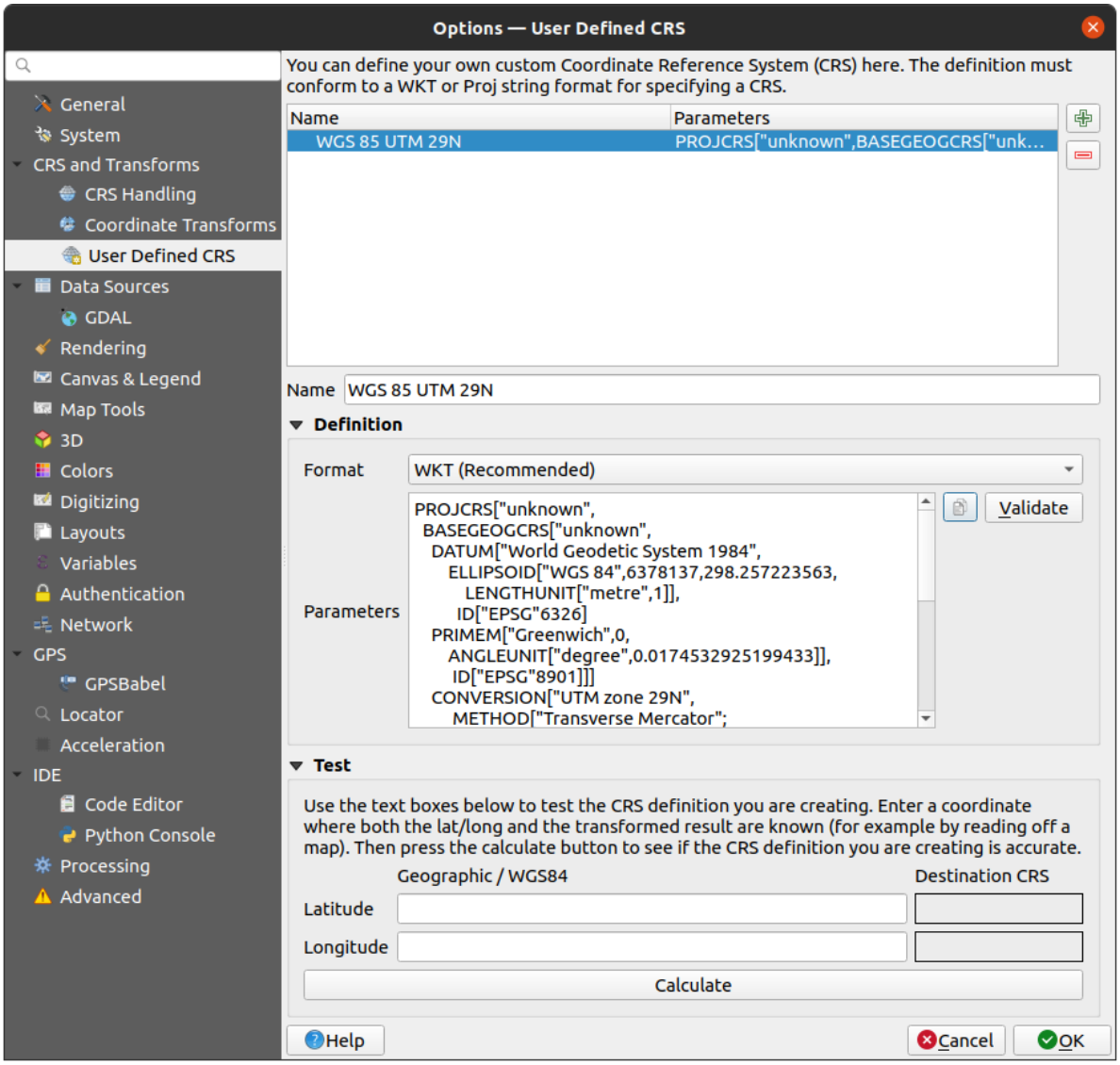


Fig. 6.4: 自定义 CRS 对话框界面

您可以测试 CRS 参数，看看它们是否给出相同的结果。为此，分别在 纬度 and 经度字段中输入已知的 WGS 84 纬度和经度值。单击 计算，并将结果与坐标参考系中的已知值进行比较。

6.5.1 在 QGIS 中集成 NTv2-变换

要在 QGIS 中集成 NTv2 变换文件，您还需要一个步骤：

1. 将 NTv2 文件（.gsb）放在 QGIS 使用的 CRS/Proj 文件夹中例如 Windows 的 `C:\OSGeo4W64\share\proj` 路径）
2. 添加 **nadgrids** (`+nadgrids=nameofthefile.gsb`) 到工程定义 参数字段 of the 自定义坐标参照系定义 (设置 ► *Custom Projections...*).

Fig. 6.5: 设置 NTv2 变换

6.6 基准变换

在 QGIS 中，默认启用了“实时动态”CRS 转换功能。这意味着，当你使用具有不同坐标系统的图层时，QGIS 会透明地将它们重新投影到项目 CRS。对于某些 CRS，存在多种可能的转换方式来重新投影到项目的 CRS！

默认情况下，QGIS 将尝试使用最准确的可用变换。然而，在某些情况下，这可能是不可能的，例如，每当需要额外的支持文件来使用变换时。每当有更精确的变换可用，但目前无法使用时，QGIS 将显示一条信息丰富的警告消息，建议您使用更精确的转换，以及如何在系统上启用它。通常，这需要下载一个变换支持文件的外部包，并将它们解压缩到 QGIS 用户配置 文件夹下的 `proj` 文件夹。

如果需要，QGIS 还可以在两个 CRS 之间进行多种可能的转换时提示您，并允许您做出明智的选择，选择最适合您的数据的转换。

此自定义设置位于 设置 ► 选项 ► 坐标转换选项卡中的 默认基准面变换组中：

- 勾选 存在多个时询问基准面变换：当某源 CRS 与目标 CRS 的组合存在多个适用的基准面转换时，系统将自动弹出对话框，提示用户选择在当前工程中使用哪一个基准面转换。如果在该对话框中选择转换时勾选了 设为默认复选框，则此选择将被记住，并自动应用于所有后续新建的 QGIS 工程。
- 或者定义一个适当的基准变换列表，以便在将图层加载到工程或重投影图层时作为默认值使用。

  使用 按钮打开:guilabel:‘选择基准面变换 ‘的对话框。然后:

  1. 指定图层的 源 *CRS* ，使用下拉菜单或 选择 CRS 控件。
  2. 以同样的方式提供 目标 *CRS* 。
  3. 表中将显示从源到目标的可用变换列表。单击一行显示应用的设置的详细信息，以及相应的变换使用的准确性和区域。

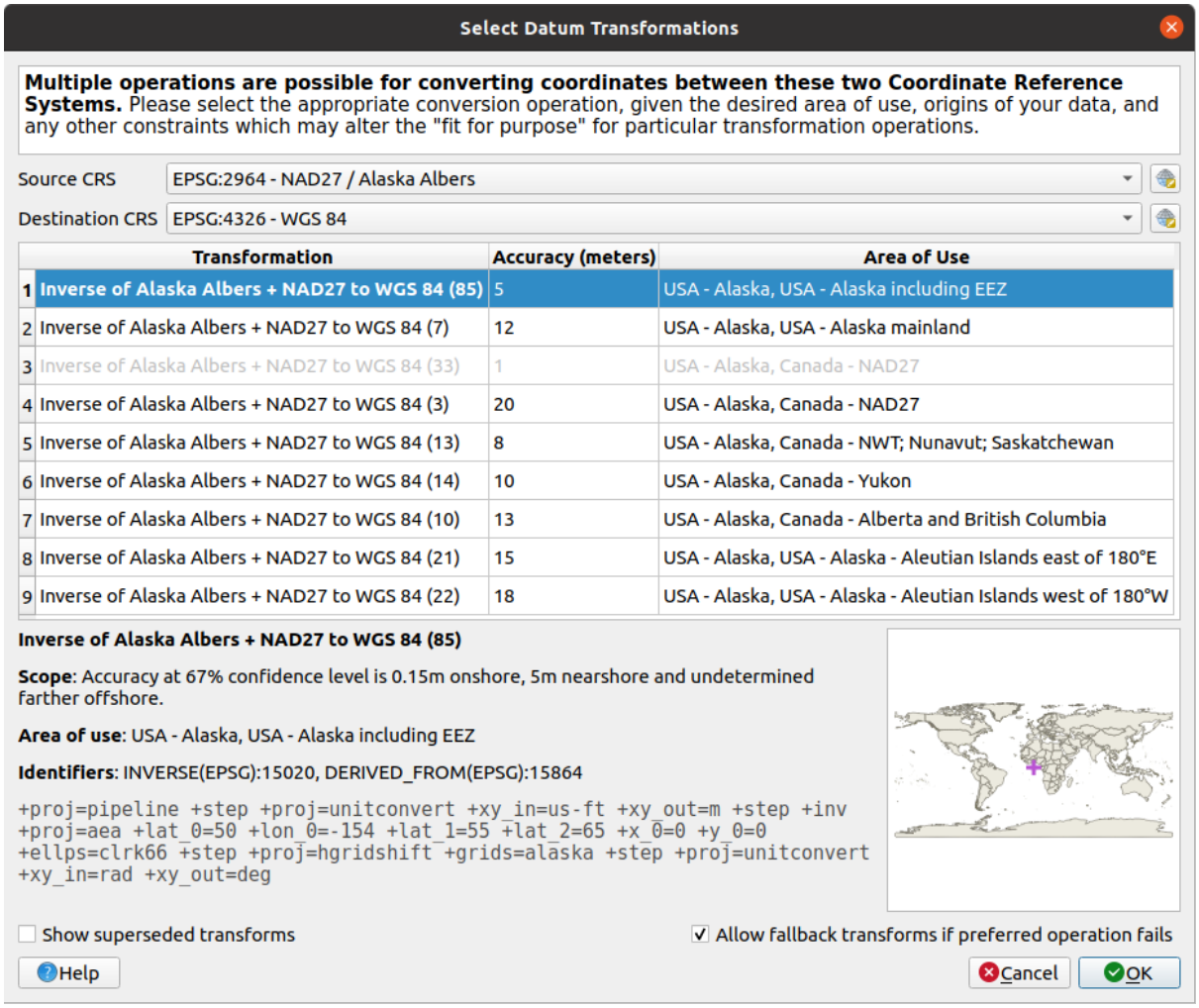


Fig. 6.6: 选择首选的默认基准面变换

在某些情况下，您的系统可能无法使用某些转换功能。此时，该转换仍会显示在此列表中（灰色状态），但需安装所需的转换支持包后才能选择。通常会提供一个按钮用于下载并安装相应的网格文件，该文件将存储在*user profile* 目录下的 `proj` 文件夹。

4. 找到你喜欢的变换，选择它
5. 设置右下角的勾选 如果首选操作失败，允许回退转换
6. 单击 确定。

   在 默认基准面变换下的表中添加一行，其中包含有 源 *CRS* 、目标 *CRS* 、为转换应用的:guilabel:`操作`和是否启用了 允许回退转换的信息。

从现在开始，QGIS 将自动使用所选的数据转换在这两个 CRS 之间进一步转换，直到您将它从列表中移除 ( ) 或更改条目 ( )。

在 设置 ► 选项 ► 变换选项卡中设置的基准面转换将被系统上创建的所有新 QGIS 工程继承。此外，特定工程还可以通过 工程 ► 工程属性... 中的 *Project properties*‘（工程属性）对话框的:*guilabel:‘CRS* 选项卡指定其专属的基准面转换。这些设置仅适用于当前工程。

CHAPTER 7

地图可视化

制作地图是 QGIS 的“主业”之一。地图视图是用于可视化项目中加载的空间层的画布，具有应用于它们的各种渲染设置。它们可以是*2D* 、*3D* 类型，或者沿着*profile line* 绘制，显示不同的比例或范围，或者由于地图主题*map themes* 而显示不同的加载层集。

7.1 2D 地图视图

2D 地图视图（也称为 ** 地图画布 **）位于窗口的中心，用于显示地图。QGIS 启动后默认打开一个以 2D 方式显示图层的地图视图（称为“主地图 “），靠近:guilabel:‘图层 ‘面板。该窗口显示了应用于已加载图层的渲染信息（符号、标注、可见性等）。

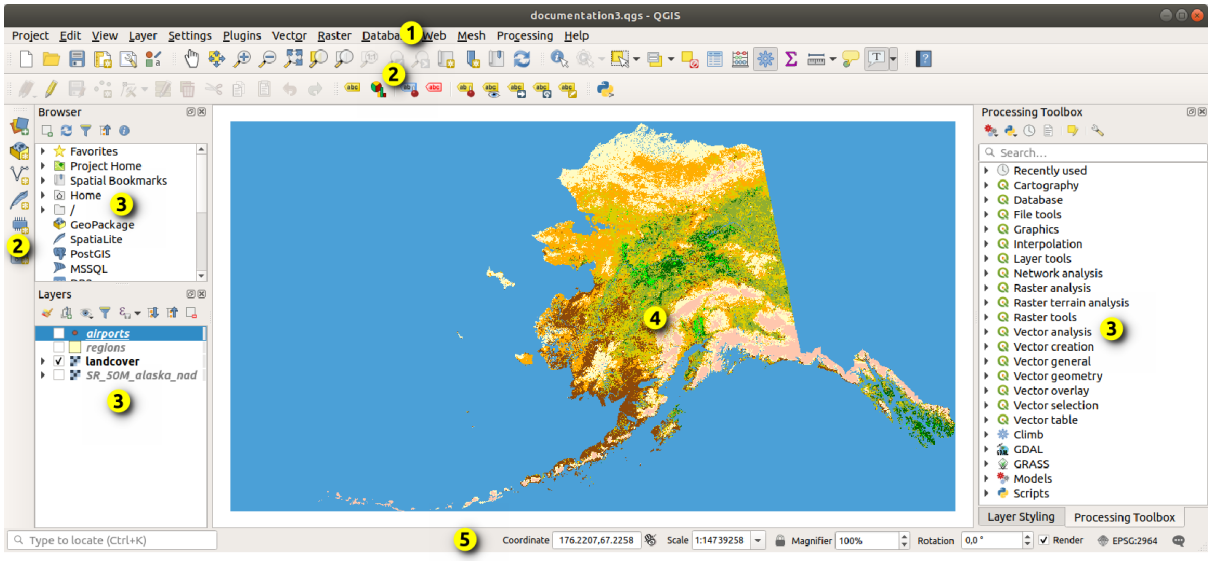


Fig. 7.1: QGIS 图形用户界面

7.1.1 探索地图视图

当您添加图层时 (例如，请参阅打开数据)，QGIS 自动查找其 CRS。如果默认情况下为工程设置了不同的 CRS (请参阅投影坐标参照系)，图层范围将" 动态" 转换为该 CRS，如果从空 QGIS 工程开始，地图视图将缩放到该范围。如果工程中已存在图层，则不会执行地图画布调整，因此只有位于当前地图画布范围内的要素可见。

点击地图视图后，您能够与地图视图进行交互，如平移或缩放地图至不同的区域。

Table 7.1: Map canvas navigation tools

工具	用法
Pan Map	• Single left click: the map is centered on the clicked point, at the same scale • Hold down the left mouse button and drag the map canvas.
Zoom In	• Single left click: the map is centered on the clicked point, while the scale gets doubled • Drag a rectangle on the map canvas with the left mouse button to zoom in to an area. • Hold the `Alt` key to switch to the Zoom Out tool.
Zoom Out	• Single left click: the map is centered on the clicked point, while the scale gets halved • Drag a rectangle on the map canvas with the left mouse button to zoom out from an area. • Hold the `Alt` key to switch to the Zoom In tool.
Pan Map to Selection	Pan the map to the selected features of all the selected layers in the *Layers* panel.
Zoom To Selection	Zoom to the selected features of all the selected layers in the *Layers* panel. *Also available in the layer contextual menu*
Zoom To Layer(s)	Zoom to the extent of all the selected layers in the *Layers* panel. *Also available in the layer contextual menu*
Zoom Full	Zoom to the extent of all the layers in the project or to the *project full extent*.
Zoom Last	Zoom the map to the previous extent in history.
Zoom Next	Zoom the map to the next extent in history.
Zoom to Native Resolution	Zoom the map to a level where one pixel of the active raster layer covers one screen pixel. *Also available in the layer contextual menu*
Mouse wheel	• Pan map: Hold and drag the mouse wheel. • Zoom: Roll the mouse wheel to zoom in or zoom out. With `Ctrl` key pressed while rolling the mouse wheel results in a finer zoom. • Press the back or forward button to browse the map canvas zoom history.

continues on next page

Table 7.1 – continued from previous page

工具	用法
Keyboard	• Pan map: Hold down the `Space` key and move the mouse. Press the arrow keys to pan up, down, left and right. • Zoom in: Press `PgUp` or `Ctrl++` • Zoom out: Press `PgDown` or `Ctrl+-` • Zoom to area: When certain map tools are active (Identify, Measure...), hold down `Shift` and drag a rectangle on the map to zoom to that area. Not compatible with active selection or edit tools.

右键单击地图，您应该能够在地图的 CRS、WGS84 或自定义 CRS 中 复制坐标单击点的坐标，然后可以将复制的信息粘贴到表达式、脚本、文本编辑器或电子表格中...

7.1.2 Controlling map rendering

By default, QGIS renders all visible layers whenever the map canvas is refreshed. The events that trigger a refresh of the map canvas include:

- changing the visibility of a layer
- modifying symbology of a visible layer
- adding a layer
- panning or zooming
- resizing the QGIS window

QGIS allows you to control the rendering process in a number of ways.

- at the *global level*
- per layer, using e.g. the *scale dependent rendering*
- or with dedicated tools in the GUI.

To stop the map drawing, press the `Esc` key. This will halt the refresh of the map canvas and leave the map partially drawn. It may however take a bit of time after pressing `Esc` for the map drawing to halt.

To suspend rendering, click the *Render* checkbox in the bottom-right corner of the status bar. When *Render* is unchecked, QGIS does not redraw the canvas in response to any of the usual triggers mentioned earlier. Examples of when you might want to suspend rendering include:

- adding many layers and symbolizing them prior to drawing
- adding one or more large layers and setting scale dependency before drawing
- adding one or more large layers and zooming to a specific view before drawing
- any combination of the above

Checking the *Render* checkbox enables rendering and causes an immediate refresh of the map canvas.

7.1.3 地图画布上基于时间的控件

QGIS 可以处理加载图层中的时态控制，即根据时间变化修改地图画布渲染。为此，您需要：

1. 设置动态时态属性的图层。QGIS 支持使用自定义设置对不同数据提供程序进行时态控制，主要是设置图层显示的时间范围：
   - 栅格图层: 控制是否显示图层。
   - *WMTS layers*: controls whether the data should be rendered based on a static time range or following a dynamic temporal range
   - 矢量图层: 根据与其属性关联的时间值过滤要素。
   - 网孔图层: 动态显示活动数据集组值。

   当为图层启用动态时态选项时，图层面板中的图层旁边会显示一个 图标，提醒您该图层是受时态控制的。单击图标更新时态设置。
2. 使用时态控制面板 启用地图画布的时态导航。面板已激活：
   - 使用位于 地图导航工具栏中的 时态控制面板 图标
   - 或是 视图 ► 面板 ► 时态控制面板菜单
   - or from the *View* ► *Data Filtering* ► *Temporal controller panel* menu

时态控制面板

时态控制面板具有以下几种状态:

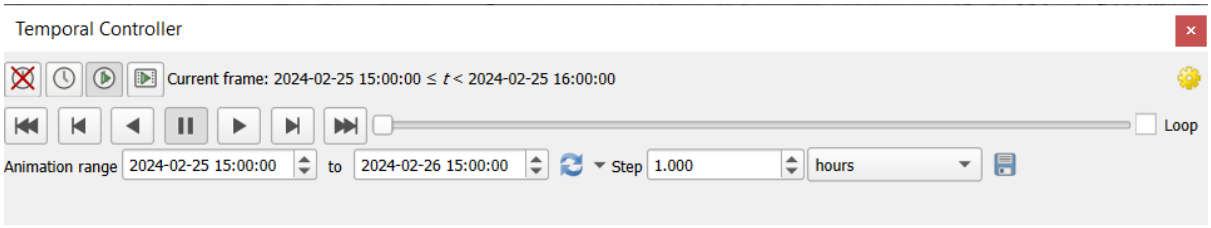


Fig. 7.2: 导航状态下的时态控制面板

- 关闭时态导航: 所有时态设置都被禁用，可见图层正常渲染
- 固定范围的时态导航: 设置时间范围，并且仅在地图上显示时态范围与该范围重叠的图层 (或要素)
- 动画时态导航: 设置时间范围，将其分为多个步幅，只有时态范围与每个帧重叠的图层 (或要素) 才会在地图上显示
- Animated movie: a total number of frames is set for the animation, with the map settings advancing at each step, displaying the progression frame by frame without any time-based filtering of data.
- 设置 用于动画的常规控制
  - 帧速率: 每秒显示的步幅数
  - *Cumulative range*: all animation frames will have the same start date-time but different end dates and times. This is useful if you wish to accumulate data in your temporal visualization instead of showing a ‘moving time window’ across your data.

Animating a temporal navigation

An animation is based on a varying set of visible layers at particular times within a time range. To create a temporal animation:

1. Toggle on the Animated temporal navigation, displaying the animation player widget
2. Enter the *Time range* to consider. Using the button, this can be defined as:
   - *Set to full range* of all the time enabled layers
   - *Set to preset project range* as defined in the *project properties*
   - *Set to single layer's range* taken from a time-enabled layer
3. Fill in the time *Step* to split the time range. Different units are supported, from `seconds` to `centuries`. A `source timestamps` option is also available as step: when selected, this causes the temporal navigation to step between all available time ranges from layers in the project. It's useful when a project contains layers with non-contiguous available times, such as a WMS-T service which provides images that are available at irregular dates. This option will allow you to only step between time ranges where the next available image is shown.
4. Click the button to preview the animation. QGIS will generate scenes using the layers rendering at the set times. Layers display depends on whether they overlap any individual time frame.

   The animation can also be previewed by moving the time slider. Checking the *Loop* checkbox will repeatedly run the animation while clicking stops a running animation. A full set of video player buttons is available.

   Horizontal scrolling using the mouse wheel (where supported) with the cursor on the map canvas will also allow you to navigate, or "scrub", the temporal navigation slider backwards and forwards.
5. Click the Export animation button if you want to generate a series of images representing the scene. They can be later combined in a video editor software:

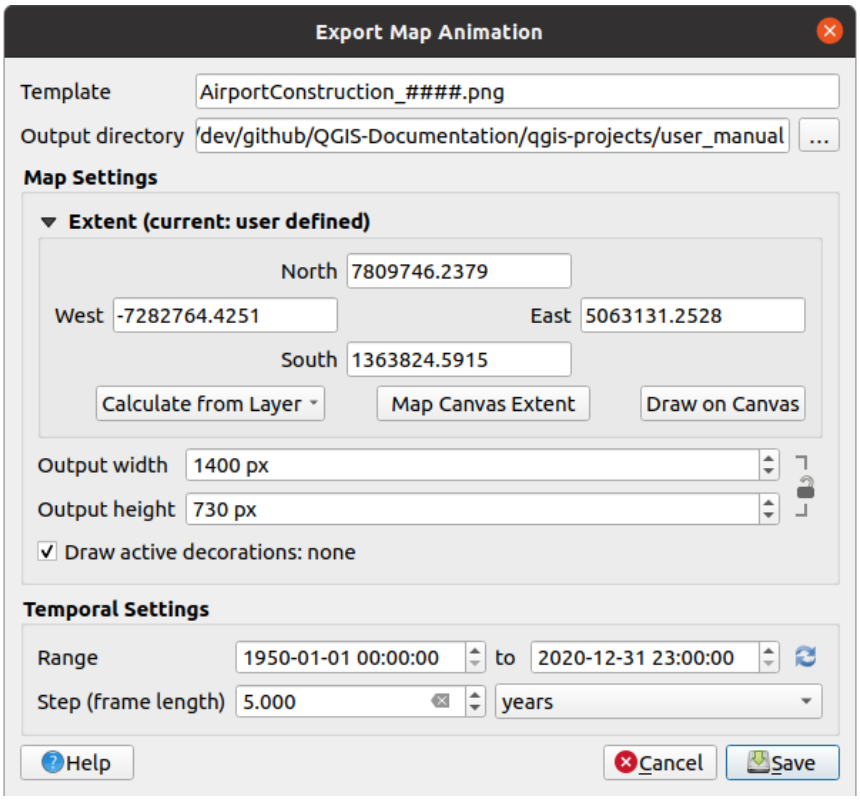


Fig. 7.3: Exporting map canvas animation scenes to images

- The filename *Template*: the `####` are replaced with frame sequence number
- The *Output directory*
- Under *Map settings*, you can:
  - redefine the *spatial extent* to use
  - control the *Resolution* of the image (*Output width* and *Output height*)
  - *Draw active decorations*: whether active *decorations* should be kept in the output
- Under *Temporal settings*, you can redefine:
  - the time *Range* for the animation
  - the *Step (frame length)* in the unit of your choice

7.1.4 The Elevation Controller panel

Elevation Controller allows you to handle elevation z-range data in 2D maps. This option currently supports point cloud layers and raster DEMs. Use the panel to visualize and interact with elevation data in the map canvas.

Activate the *Elevation Controller* panel in the map canvas by clicking on the *Data Filtering* in the *View* menu. The *Elevation Controller* appears as a range slider on the left side of the map canvas. At the top of the slider, there is a Settings menu where you can:

- Set the *Fixed Range Size* value to lock the elevation range to a specific size. This setting is saved within the project and retained across sessions, ensuring consistency for your elevation filter. Use the *Invert Direction* option to reverse the elevation filter slider, so that it moves from high values at the bottom to low values at the top.
- *Set Elevation Range* to open the project elevation tab, where you can set dedicated lower and upper bounds for the elevation range
- *Disable Elevation Filter* to remove the elevation filter from the map canvas

7.1.5 Bookmarking extents on the map

Spatial Bookmarks allow you to ”bookmark” a geographic location and return to it later. By default, bookmarks are saved in the user’s profile (as *User Bookmarks*), meaning that they are available from any project the user opens. They can also be saved for a single project (named *Project Bookmarks*) and stored within the project file, which can be helpful if the project is to be shared with other users.

Creating a Bookmark

To create a bookmark:

1. Zoom and pan to the area of interest.
2. Select the menu option *View* ► *New Spatial Bookmark...*, press `Ctrl+B` or right-click the *Spatial Bookmarks* entry in the *Browser* panel and select *New Spatial Bookmark*. The *Bookmark Editor* dialog opens.

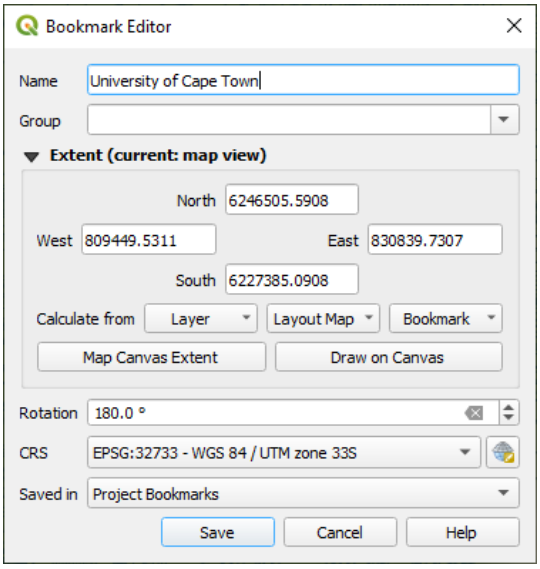


Fig. 7.4: The Bookmark Editor Dialog

3. Enter a descriptive name for the bookmark
4. Enter or select a group name in which to store related bookmarks
5. Select the extent of the area you wish to save, using the *extent selector* widget
6. Change the map *Rotation*
7. Indicate the *CRS* to use for the extent
8. Select whether the bookmark will be *Saved in User Bookmarks* or *Project Bookmarks* (by default, this drop-down list is set to *User Bookmarks*)
9. Press *Save* to add the bookmark to the list

Note that you can have multiple bookmarks with the same name.

Working with Bookmarks

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

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

You can perform the following tasks:

Table 7.2: Managing bookmark actions

Task	Spatial Bookmark Manager	Browser
Zoom to a Bookmark	Double-click on it, or select the bookmark and press the *Zoom to bookmark* button.	Double-click on it, drag and drop it to the map canvas, or right-click the bookmark and select *Zoom to Bookmark*.
Delete a bookmark	Select the bookmark and click the *Delete bookmark* button. Confirm your choice.	Right-click the bookmark and select *Delete Spatial Bookmark*. Confirm your choice.

continues on next page

Table 7.2 – continued from previous page

Task	Spatial Bookmark Manager	Browser
Export bookmarks to XML	Click the *Import/Export Bookmarks* button and select *Export*. All the bookmarks (user or project) are saved in an xml file.	Select one or more folders (user or project) or subfolders (groups), then right-click and select *Export Spatial Bookmarks...*. The selected bookmark subset is saved.
Import bookmarks from XML	Click the *Import/Export Bookmarks* button and select *Import*. All bookmarks in the XML file are imported as user bookmarks.	Right-click the *Spatial Bookmarks* entry or one of its folders (user or project) or subfolders (groups) to determine where to import the bookmarks, then select *Import Spatial Bookmarks*. If performed on the *Spatial Bookmarks* entry, the bookmarks are added to *User Bookmarks*.
Edit bookmark	You can change a bookmark by changing the values in the table. You can edit the name, the group, the extent and if it is stored in the project or not.	Right-click the desired bookmark and select *Edit Spatial Bookmark...*. The *Bookmark Editor* will open, allowing you to redefine every aspect of the bookmark as if you were creating it for the first time. You can also drag and drop the bookmark between folders (user and project) and subfolders (groups).

You can manage bookmark actions by right-click on the desired bookmark in the *Spatial Bookmarks Manager*. You can also zoom to bookmarks by typing the bookmark name in the *locator*.

7.1.6 Decorating the map

Decorations include Grid, Title Label, Copyright Label, Image, North Arrow, Scale Bar and Layout Extents. They are used to 'decorate' the map by adding cartographic elements.

Grid

Grid allows you to add a coordinate grid and coordinate annotations to the map canvas.

1. Select menu option *View* ► *Decorations* ► *Grid...* to open the dialog.

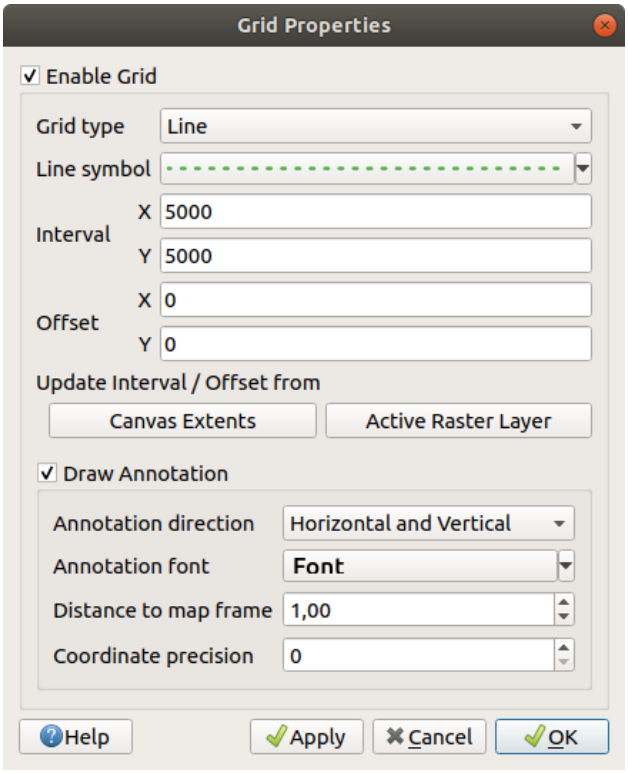


Fig. 7.5: The Grid Dialog

2. Tick *Enable grid* and set grid definitions according to the layers loaded in the map canvas:
   - The *Grid type*: it can be *Line* or *Marker*
   - The associated *Line symbol* or *marker symbol* used to represent the grid marks
   - The *Interval X* and *Interval Y* between the grid marks, in map units
   - An *Offset X* and *Offset Y* distance of the grid marks from the bottom left corner of the map canvas, in map units
   - The interval and offset parameters can be set based on the:
     - *Canvas Extents*: generates a grid with an interval that is approximatively 1/5 of the canvas width
     - *Active Raster Layer* resolution
3. Tick *Draw annotations* to display the coordinates of the grid marks and set:
   - The *Annotation direction*, ie how the labels would be placed relative to their grid line. It can be:
     - *Horizontal* or *Vertical* for all the labels
     - *Horizontal and Vertical*, ie each label is parallel to the grid mark it refers to
     - *Boundary direction*, ie each label follows the canvas boundary, and is perpendicular to the grid mark it refers to
   - The *Annotation font* (text formatting, buffer, shadow...) using the *font selector widget*
   - The *Distance to map frame*, margin between annotations and map canvas limits. Convenient when *exporting the map canvas* eg to an image format or PDF, and avoid annotations to be on the ”paper” limits.
   - The *Coordinate precision*
4. Click *Apply* to verify that it looks as expected or *OK* if you're satisfied.

Title Label

Title Label allows you to decorate your map with a **Title**.

To add a Title Label decoration:

1. Select menu option *View* ► *Decorations* ► *Title Label...* to open the dialog.

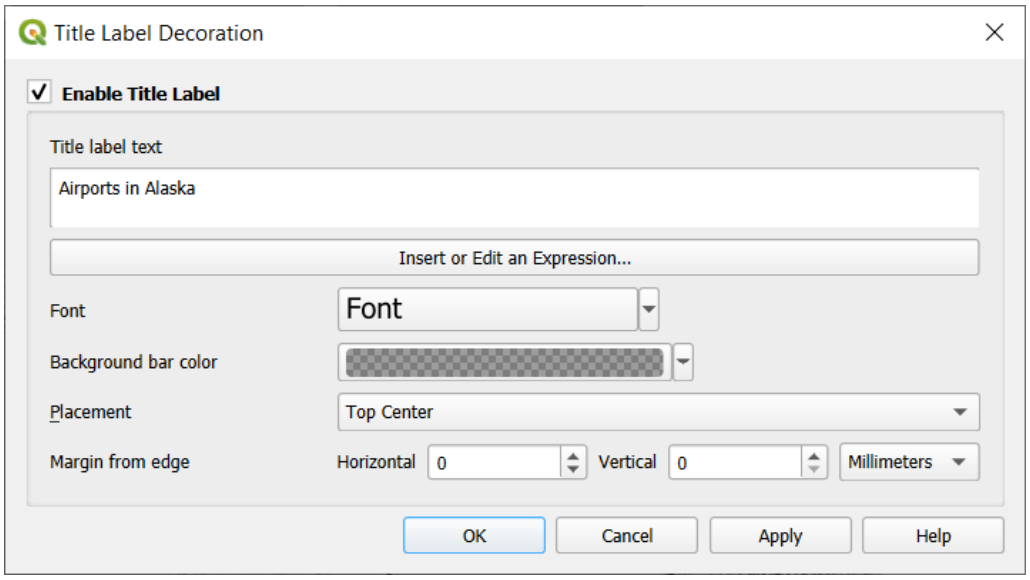


Fig. 7.6: The Title Decoration Dialog

2. Make sure *Enable Title Label* is checked
3. Enter the title text you want to place on the map. You can make it dynamic using the *Insert or Edit an Expression...* button.
4. Choose the *Font* for the label using the *font selector widget* with full access to QGIS *text formatting* options. Quickly set the font color and opacity by clicking the black arrow to the right of the font combo box.
5. Select the *color* to apply to the title's *Background bar color*.
6. Choose the *Placement* of the label in the canvas: options are *Top left*, *Top Center* (default), *Top Right*, *Bottom left*, *Bottom Center* and *Bottom Right*.
7. Refine the placement of the item by setting a horizontal and/or vertical *Margin from Edge*. These values can be in **Millimeters** or **Pixels** or set as a **Percentage** of the width or height of the map canvas.
8. Click *Apply* to verify that it looks as expected or *OK* if you're satisfied.

Copyright Label

Copyright Label can be used to decorate your map with a **Copyright** label.

To add this decoration:

1. Select menu option *View* ► *Decorations* ► *Copyright Label...* to open the dialog.

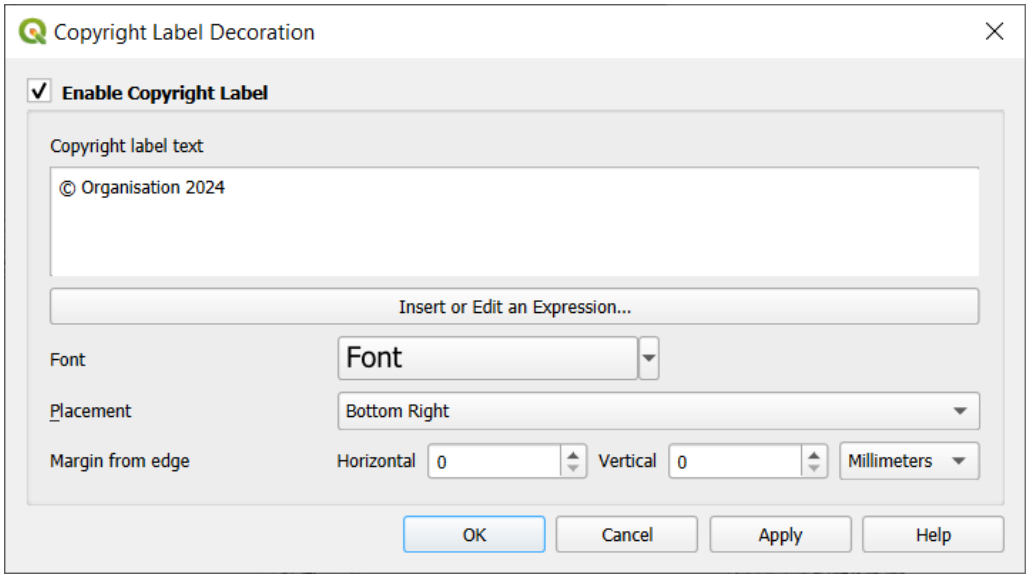


Fig. 7.7: The Copyright Decoration Dialog

2. Make sure *Enable Copyright Label* is checked
3. Enter the copyright text you want to place on the map. You can make it dynamic using the *Insert or Edit an Expression...* button.
4. Choose the *Font* for the label using the *font selector widget* with full access to QGIS *text formatting* options. Quickly set the font color and opacity by clicking the black arrow to the right of the font combo box.
5. Choose the *Placement* of the label in the canvas: options are *Top left*, *Top Center*, *Top Right*, *Bottom left*, *Bottom Center*, and *Bottom Right* (default for Copyright decoration)
6. Refine the placement of the item by setting a horizontal and/or vertical *Margin from Edge*. These values can be in **Millimeters** or **Pixels** or set as a **Percentage** of the width or height of the map canvas.
7. Click *Apply* to verify that it looks as expected or *OK* if you're satisfied.

Image Decoration

Image allows you to add an image (logo, legend, ..) on the map canvas.

To add an image:

1. Select menu option *View* ► *Decorations* ► *Image...* to open the dialog.

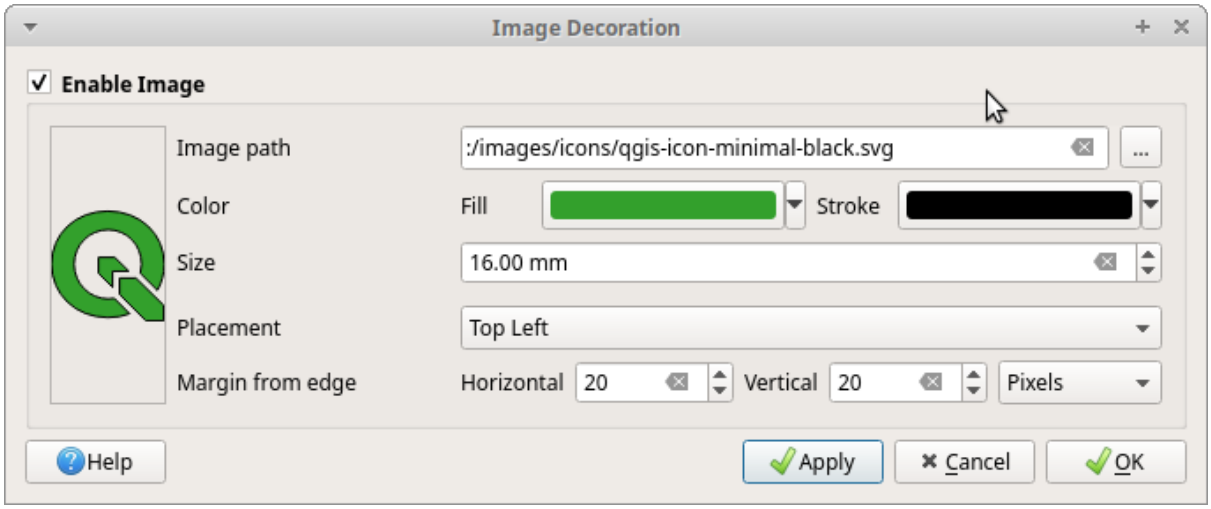


Fig. 7.8: The Image Decoration Dialog

2. Make sure *Enable Image* is checked
3. Select a bitmap (e.g. png or jpg) or SVG image using the ... Browse button
4. If you have chosen a parameter enabled SVG then you can also set a *Fill* or *Stroke* (outline) color. For bitmap images, the color settings are disabled.
5. Set a *Size* of the image in mm. The width of selected image is used to resize it to given *Size*.
6. Choose where you want to place the image on the map canvas with the *Placement* combo box. The default position is *Top Left*.
7. Set the *Horizontal* and *Vertical Margin from (Canvas) Edge*. These values can be set in **Millimeters**, **Pixels** or as a **Percentage** of the width or height of the map canvas.
8. Click *Apply* to verify that it looks as expected and *OK* if you're satisfied.

North Arrow

North Arrow allows you to add a north arrow on the map canvas.

To add a north arrow:

1. Select menu option *View* ► *Decorations* ► *North Arrow*... to open the dialog.

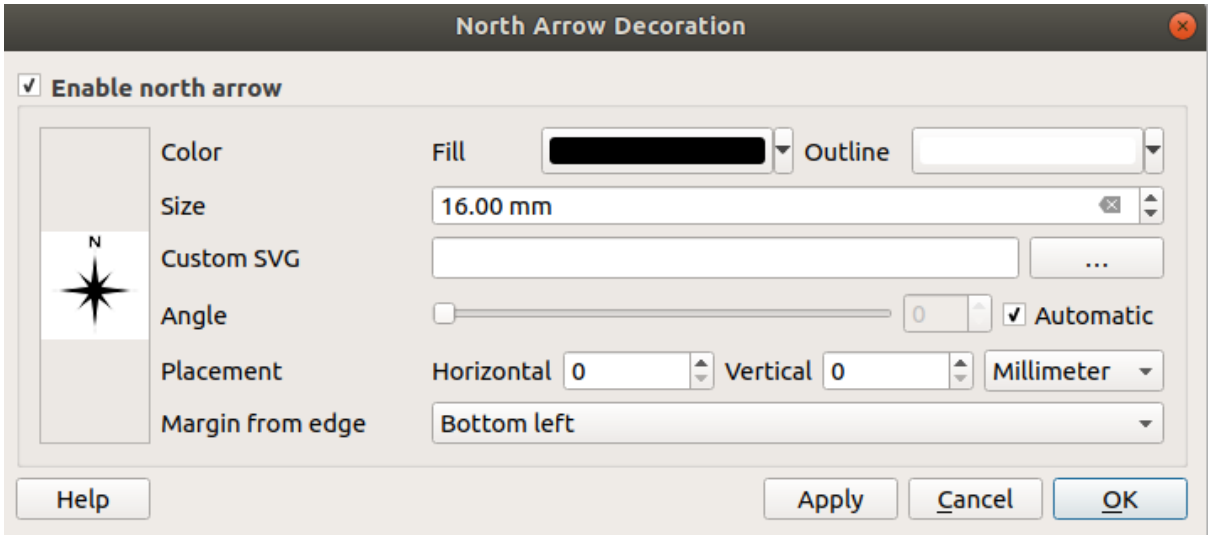


Fig. 7.9: The North Arrow Dialog

2. Make sure *Enable north arrow* is checked
3. Optionally change the color and size, or choose a custom SVG
4. Optionally change the angle or choose **Automatic** to let QGIS determine the direction
5. Optionally choose the placement from the Placement combo box
6. Optionally refine the placement of the arrow by setting a horizontal and/or vertical *Margin from (Canvas) Edge*. These values can be in **Millimeters** or **Pixels** or set as a **Percentage** of the width or height of the map canvas.
7. Click *Apply* to verify that it looks as expected and *OK* if you're satisfied.

Scale Bar

Scale Bar adds a simple scale bar to the map canvas. You can control the style and placement, as well as the labelling of the bar. The scale bar respects the active *project's distance unit* as defined in *Project properties* ► *General* ► *Units for distance measurement*.

To add a scale bar:

1. Select menu option *View* ► *Decorations* ► *Scale Bar...* to open the dialog

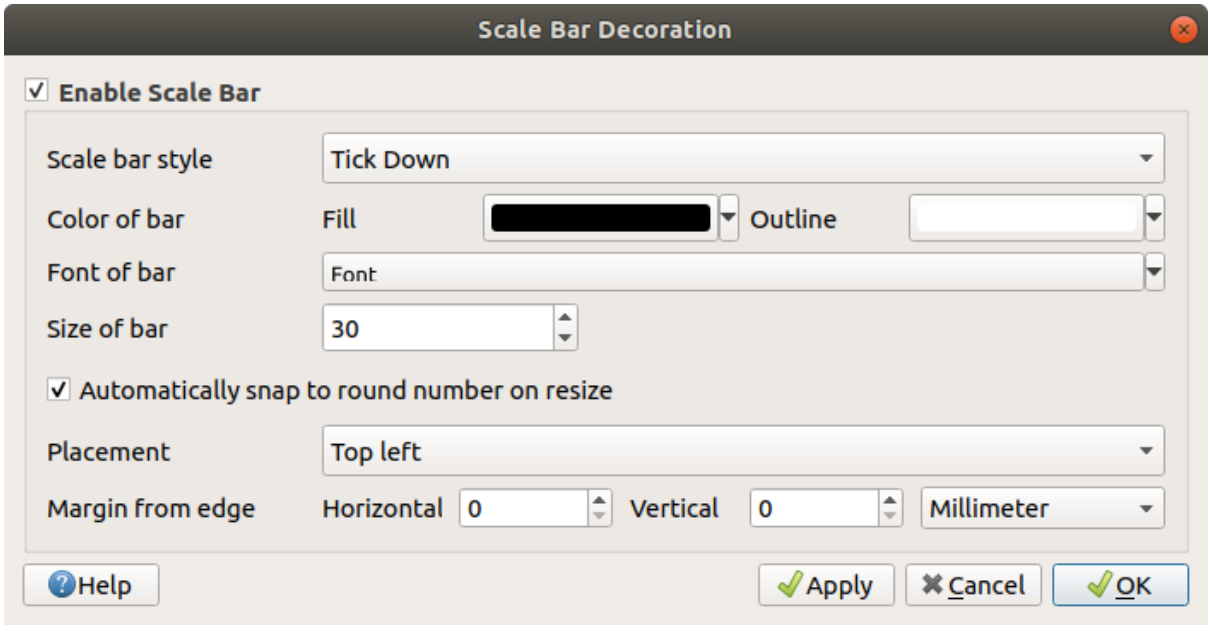


Fig. 7.10: The Scale Bar Dialog

2. Make sure *Enable scale bar* is checked
3. Choose a style from the *Scale bar style* combo box
4. Select the *Color of bar* by choosing a fill color (default: black) and an outline color (default: white). The scale bar fill and outline can be made opaque by clicking on the down arrow to the right of the color input.
5. Select the font for the scale bar from the *Font of bar* combo box
6. Set the *Size of bar* in the active unit
7. Optionally check *Automatically snap to round number on resize* to display easy-to-read values
8. Choose the placement from the *Placement* combo box
9. You can refine the placement of the item by setting a horizontal and/or vertical *Margin from (Canvas) Edge*. These values can be in **Millimeters** or **Pixels** or set as a **Percentage** of the width or height of the map canvas.
10. Click *Apply* to verify that it looks as expected or *OK* if you're satisfied.

Layout Extents

Layout Extents adds the extents of *map item(s)* in print layout(s) to the canvas. When enabled, the extents of all map items within all print layouts are shown using a lightly dotted border labeled with the name of the print layout and map item. You can control the style and labeling of the displayed layout extents. This decoration is useful when you are tweaking the positioning of map elements such as labels, and need to know the actual visible region of print layouts.

Fig. 7.11: Example of layout extents displayed in a QGIS project with two print layouts. The print layout named 'Sights' contains two map items, while the other print layout contains one map item.

To add layout extent(s):

1. Select *View* ► *Decorations* ► *Layout Extents* to open the dialog

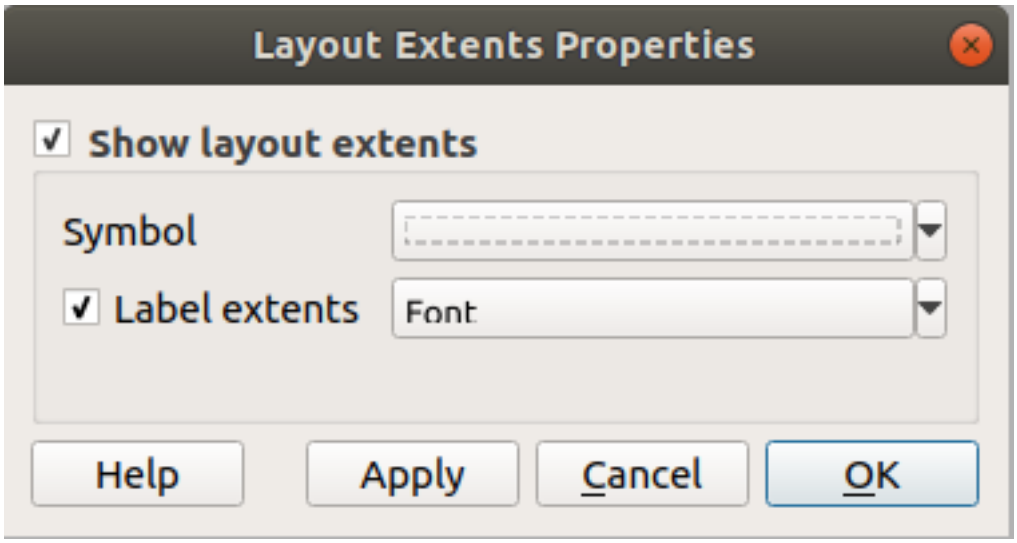


Fig. 7.12: The Layout Extents Dialog

2. Make sure *Show layout extents* is checked.

3. Optionally change the symbol and labeling of the extents.
4. Click *Apply* to verify that it looks as expected and *OK* if you're satisfied.

> **Tip**
>
> **Decorations Settings**
>
> When you save a QGIS project file, any changes you have made to Grid, North Arrow, Scale Bar, Copyright and Layout Extents will be saved in the project and restored the next time you load the project.

7.1.7 Annotation Tools

Annotations are another type of elements added onto the map canvas to provide additional information that can not be depicted by the rendered layers. Unlike *labels* that rely on attribute values stored in vector layers, annotations are independent details, stored within the project file itself.

Two families of annotations are available in QGIS:

- **Feature annotations**: they are actual georeferenced features of text, marker, line or polygon type stored within a special layer type called "annotation layer". They are tied to a particular geographic location, meaning that moving your map, changing the scale or changing projection won't cause your annotations to jump around the map. Rather, they'll be locked in place to the location you've drawn them.
- **Balloon annotations**: these are individual HTML or form annotations inside a bubble. They can be associated to any layer for their visibility and drawn on top of the map canvas. The size is dependent from the map canvas scale, and its position can be anchored.

> **Tip**
>
> **Layout the map with annotations**
>
> You can print or export annotations with your map to various formats using:
>
> - map canvas export tools available in the *Project* menu
> - *print layout*, in which case you need to check *Draw map canvas items* in the corresponding map item properties

The *Annotations Toolbar* provides a set of tools to create and interact with both families of annotations.

Table 7.3: The Annotations Toolbar actions

工具	用法	Scope
New Annotation Layer	Create a new layer to store annotations	*Feature annotations*
Main Annotation Layer Properties	Control settings of the Main Annotation Layer	
Modify Annotations	Select, move, resize and modify symbology properties of annotations	
Create Polygon Annotation	Create an annotation as a polygon feature	
Create Line Annotation	Create an annotation as a polyline feature	
Create Marker Annotation	Create an annotation as a point feature	
Create Text Annotation at Point	Create an annotation as a text label anchored at a point	
Create Text Annotation along Line	Create an annotation as a curved text along a line feature	

continues on next page

Table 7.3 – continued from previous page

工具	用法	Scope
Create Text Annotation in Rectangle	Create an annotation as a text label in a rectangular text box	*Balloon annotations*
Create Picture Annotation	Create an annotation representing an `.SVG` or raster image	
HTML Annotation	Select and create annotation with a `HTML` formatted text	
Form Annotation	Select and create annotation showing attributes of a vector layer in a custom form file	

Note

Starting with QGIS 3.40, it is no longer possible to create balloon annotations of SVG or text type. When loading old projects, any of these annotations will automatically be converted to the newer picture and rectangular text annotation item types.

Feature Annotations

Feature annotations are stored in **annotation layers**. Unlike conventional layers, an annotation layer is available only in the current project and can contain features of different types (text, marker, line, polygon, picture). The layer has no attributes and no symbology associated, but instead each feature can be symbolized on an item-by-item basis, through *Layer Styling* panel.

Two types of annotation layer are available in QGIS:

- A common *Annotation Layer*: you can create one using the New Annotation Layer tool. It is listed in the *Layers* panel, allowing you to control its features' visibility, move it to show above or below particular layers in your map, like any common layer. Double-click the layer and you can access its properties.
- The *Main Annotation Layer*: By default, this is where annotations are stored when no annotation layer is available in the project or is selected at creation time. This layer is always drawn on the very top of your map and you won't see it listed in the *Layers* panel alongside the other layers in your project, meaning that its features are always visible. The *Main Annotation Layer Properties* entry on the *Annotations* toolbar helps you open its properties dialog.

图层属性

The properties dialog of an annotation layer provides the following tabs:

- *Information*: a read-only dialog representing an interesting place to quickly grab summarized information and metadata on the current layer. This may include the layer extent, count of items per annotation type and total count, CRS details, ...
- *Source*: defines general settings for the annotation layer. You can:
  - Set a *Layer name* that will be used to identify the layer in the project (in the *Layers Panel*, with expressions, ...)
  - Display the layer's *Assigned Coordinate Reference System (CRS)*: you can change the layer's CRS, selecting a recently used one in the drop-down list or clicking on Select CRS button (see 坐标参照系选择器). Use this process only if the CRS applied to the layer is a wrong one or if none was applied.
- *Rendering*:

- You can set the *Maximum (inclusive)* and *Minimum (exclusive)* scale, defining a range of scale in which features will be visible. Out of this range, they are hidden. The Set to current canvas scale button helps you use the current map canvas scale as boundary of the range visibility. See 可见比例选择器 for more information.
- *Linked layer*: allows you to optionally set a linked visibility layer for the annotation layer. If set, then the annotations will only be drawn when the linked layer is visible in the map.
- *Opacity*: You can make the underlying layer in the map canvas visible with this tool. Use the slider to adapt the visibility of your vector layer to your needs. You can also make a precise definition of the percentage of visibility in the menu beside the slider.
- *Blending mode* at the *Layer* level: You can achieve special rendering effects with these tools that you may previously only know from graphics programs. The pixels of your overlaying and underlying layers are mixed through the settings described in *Blending Modes*.
- Apply *paint effects* on all the layer features with the *Draw Effects* button.

Some of these options are accessible from the feature annotation *Symbology* properties.

与要素交互

The Feature annotations have dedicated tools for creation depending on their type.

Table 7.4: Creating feature annotation

Family / based on	Annotation tool	步骤
Geometry	Create Polygon Annotation	This is like digitizing a point, line or polygon vector feature. All the usual shortcuts for creating features apply when creating annotation items. A line or polygon annotation is drawn by left-clicking once for each vertex, with a final right mouse click to complete the shape. Snapping can be enabled while you draw, you can use the *Advanced Digitizing Tools* to precisely place vertices, and even switch the *drawing tools* to for example the streaming mode for completely free-form shapes.
	Create Line Annotation	
	Create Marker Annotation	
Text	Create Text Annotation at Point	Left-click on the map canvas.
	Create Text Annotation along Line	Left-click to add vertices to the text basement line geometry, and right-click to end. As above, digitizing features capabilities are also available.
	Create Text Annotation In Rectangle	Draw the rectangle box of the text, by left-clicking to add a first corner and left-clicking again to add the opposite one.
Picture	Create Picture Annotation	Draw the rectangle box of the picture, by left-clicking to add a first corner and left-clicking again to add the opposite one. Then pick an image or SVG file from the file explorer.

Unlike common layers, an annotation layer does not need to be active before you select its features. Simply grab the Modify Annotations tool and you will be able to interact with any feature annotation:

- **Selection**: left-click on the annotation. By default, annotations are rendered in the order of creation, meaning that recent annotations will be placed on top of older. You may need to play with the Z-index property of features in order to select ones they would sit above.
- **Moving**: Left click on a selected annotation item to start moving it. A right-click or pressing `Esc` key cancels

the move, while a second left click will confirm the move. The displacement can also be controlled pressing the cursor keys:

  - `Shift+key` for big movement
  - `Alt+key` for `1 px` movement

- **Geometry modification**: move over an annotation and purple squares are displayed on nodes of its underlying geometry. Left-click on the square, move and click again. For linear or polygonal annotations, double-clicking a segment will add a new vertex.
- **Delete**: Pressing the `Del` or `Backspace` key while an annotation is selected will delete that annotation
- *Change feature symbology*

Feature symbology

A selected annotation will display its *Symbology* properties in the *Layer styling* panel.

- For polygon, polyline and marker annotations, you can modify the appearance using full capabilities of the *symbol properties*
- For text-based annotations, an area with text editing tools helps you construct the string to display. It can be multiline, use HTML formatting and rely on QGIS expression functions. You can moreover modify the appearance using full capabilities of the *text format properties*.

  Depending on the type of annotation, additional options are available.

  - For text annotation at point, it is also possible to:
    - set the text *Alignment* (left, center or right of the anchor point)
    - configure whether the text should *Ignore map rotation* or *Rotate with map*. In both cases, a custom *Angle* can be set for the feature orientation.
  - For text annotation along a line, configure an *Offset from line* in the unit of your choice
  - For text annotation in rectangle, it is also possible to:
    - set the text *Horizontal alignment* (left, center, right or justify) and *Vertical alignment* (top, vertical center or bottom) in the rectangle box
    - configure for the rectangle, the *Margins* distance in the units of your choice, the *Frame* and *Background* colors.
- Picture annotations allow you to display a *Raster image* or an *SVG image*, using the 远程或嵌入式文件选择器. The placement and size of the picture can be set as:
  - *Scale dependent size*: the picture scales with the map, appearing larger when zooming in and smaller when zooming out. The size is defined by a single reference size in map units (e.g. meters or degrees, depending on the layer's CRS). With *Lock aspect ratio*, you can keep the annotation size proportional to the embedded picture size, or stretch the picture to fill the annotation extent.
  - *Fixed size*: the picture annotation maintains a constant size in screen or output units (e.g. pixels or millimeters), regardless of the map scale. The *Width* and *Height* of the annotation can be filled separatey or adjusted proportionally to the embedded picture size.
  - *Relative to map*: the picture annotation will always be rendered at the same position relative to the map canvas bounds, regardless of the map scale. The *Width* and *Height* of the annotation can be filled separatey or adjusted proportionally to the embedded picture size.

  Moreover, you can enable display of the *Frame* and *Background* of the rectangle box, and configure them using *fill symbols*.
- For text at point, text inside rectangle and picture annotations, you can *Show callout* when the picture or text anchor point is offset from its default placement. To create a callout for an annotation:
  1. Select the annotation item

2. Move the anchor point, i.e., click the central and green `X` node and click at the new placement
3. In the annotation properties, press the ... button to configure the *callout properties*.

> **Tip**
>
> Applying a balloon callout to a text or picture annotation will make you get their old-style balloon annotation display.

- Configure a *Reference scale*: indicates the map scale at which symbol or text sizes which use paper-based units (such as millimeters or points) relate to. The sizes will be scaled accordingly whenever the map is viewed at a different scale. For instance, a line feature wide of 2mm at 1:2000 *Reference scale* will be rendered using 4mm when the map is viewed at 1:1000.
- Set a *Z-index*: a feature with a higher index is placed on top of features with lower index. A convenient setting for both feature display and selection.
- Modify some of the *Layer rendering* settings

Balloon annotations

You can add balloon annotations through *Edit* ► *Add Annotation* ► menu or from the *Annotations Toolbar*:

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

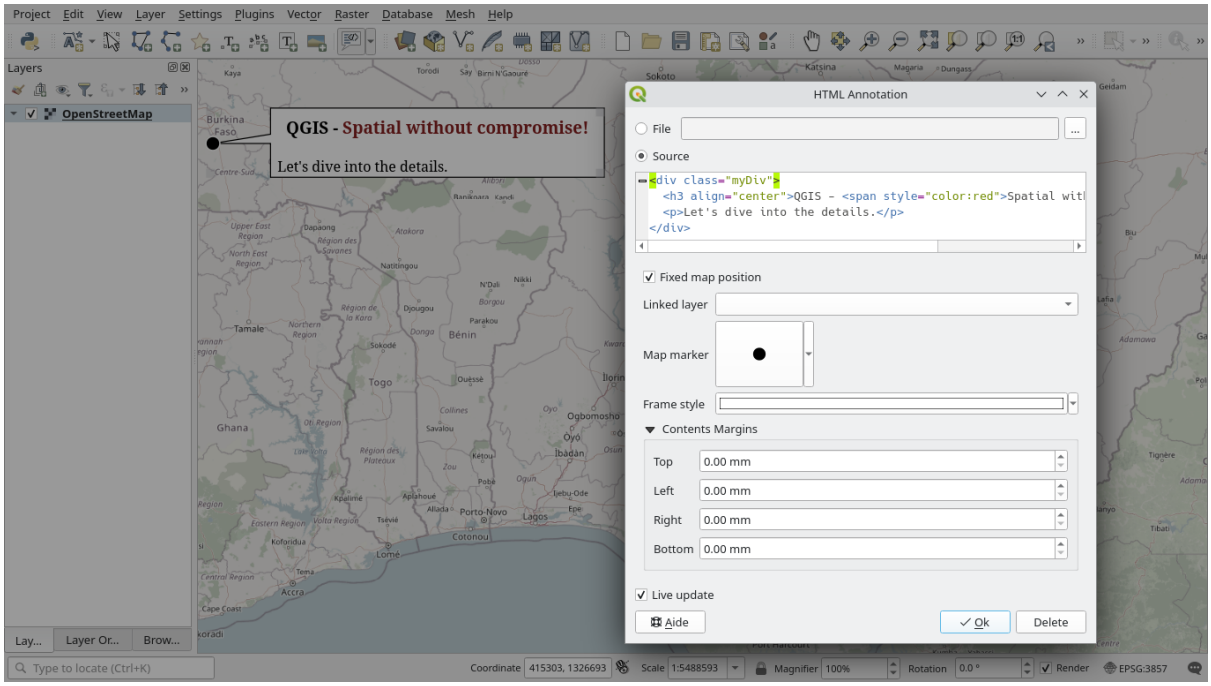


Fig. 7.13: Example of a HTML balloon annotation

To add a balloon annotation, select the corresponding tool and click on the map canvas. An empty balloon is added. Double-click on it and a dialog opens with various options. This dialog is almost the same for all the annotation types:

- *File*: Fill it with the path to a `.html` or `.ui` file, depending on the type of annotation, to serve as contents of the annotation.

- *Source*: Only available for HTML annotations, it allows to serve the annotation with a plain or HTML-formatted text instead of a file.
- *Fixed map position*: when unchecked, the balloon placement is based on a screen position (instead of the map), meaning that it's always shown regardless of the map canvas extent.
- *Linked layer*: associates the annotation with a map layer, making it visible only when that layer is visible.
- *Map marker*: using *QGIS symbols*, sets the symbol to display at the balloon anchor position (shown only when *Fixed map position* is checked).
- *Frame style*: sets the frame background color, transparency, stroke color or width of the balloon using QGIS symbols.
- *Contents margins*: sets interior margins of the annotation frame.
- *Live update* allows you to live preview your changes.

When a balloon annotation tool is active, you can also:

- Select an annotation
- Resize an annotation
- Move an annotation by map position (by dragging the map marker) or by moving only the balloon.
- Delete an annotation: select it and either press the `Del` or `Backspace` button, or double-click it and press the *Delete* button in the properties dialog.
- Right-click and in the contextual menu:
  - *Copy coordinate* of the annotation's map marker in various CRS
  - *Edit* properties of the annotation. Same as double-clicking the annotation.
  - *Delete* the annotation

7.1.8 Measuring

General information

QGIS provides four means of measuring geometries:

- interactive measurement tools
- measuring in the Field Calculator
- derived measurements in the 识别要素 tool
- the vector analysis tool: *Vector* ► *Geometry Tools* ► *Export/Add Geometry Columns*

Measuring works within projected coordinate systems (e.g., UTM) and unprojected data. The first three measuring tools behave equally to global project settings:

- Unlike most other GIS, the default measurement metric is ellipsoidal, using the ellipsoid defined in *Project* ► *Properties...* ► *General*. This is true both when geographic and projected coordinate systems are defined for the project.
- If you want to calculate the projected/planimetric area or distance using cartesian maths, the measurement ellipsoid has to be set to "None/Planimetric" (*Project* ► *Properties...* ► *General*). However, with a geographic (ie unprojected) CRS defined for the data and project, area and distance measurement will be ellipsoidal.

However, neither the identify tool nor the field calculator will transform your data to the project CRS before measuring. If you want to achieve this, you have to use the vector analysis tool: *Vector* ► *Geometry Tools* ► *Add Geometry Attributes...*. Here, measurement is planimetric, unless you choose the ellipsoidal measurement.

Measure length, areas, bearings and angles interactively

Click the icon in the Attribute toolbar to begin measurements. The down arrow near the icon switches between length, area, bearing or angle. The default unit used in the dialog is the one set in *Project* ► *Properties...* ► *General* menu.

For the *Measure Line* and the *Measure Area* the measurements can be done in *Cartesian* or *Ellipsoidal* measure.

> **Note**
>
> **Configuring the measure tool**
>
> While measuring length or area, clicking the *Configuration* button at the bottom of the widget opens the *Settings* ► *Options* ► *Map Tools* menu, where you can select the rubberband color, the precision of the measurements and the unit behavior. You can also choose your preferred measurement or angle units, but keep in mind that those values are overridden in the current project by the selection made in the *Project* ► *Properties...* ► *General* menu, and by the selection made in the measurement widget.

All measuring modules use the snapping settings from the digitizing module (see section 设置捕捉容差和搜索半径). So, if you want to measure exactly along a line feature, or around a polygon feature, first set its layer snapping tolerance. Now, when using the measuring tools, each mouse click (within the tolerance setting) will snap to that layer.

The Measure Line measures distances between given points. The tool then allows you to click points on the map. Each segment length, as well as the total, shows up in the measure window. In the measure window, you will see coordinates for all your points and distances. Keep in mind that the first row will contain only coordinates, as it represents your starting point. Now it is possible to copy all your line measurements at once to the clipboard using the *Copy* button. Clicking the *Configuration* button you will access to *Measure Tool Copy Settings* where you can set up *copy options*. To stop measuring, click the right mouse button.

Note that you can use the drop-down list near the total to change the *measurement units* interactively while working with the measure tool. This unit is retained for the widget until a new project is created or another project is opened.

The *Info* section in the dialog explains how calculations are made according to the CRS settings available.

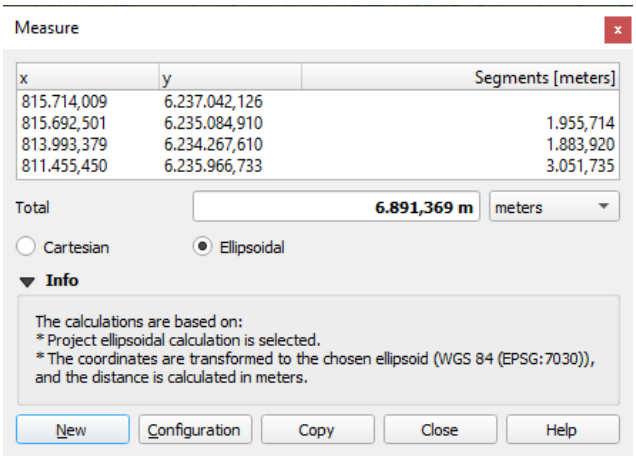


Fig. 7.14: Measure Distance

Measure Area: Areas can also be measured. In the measure window, the accumulated area size appears. Right-click to stop drawing. The Info section is also available as well as the ability to switch between different *area units*.

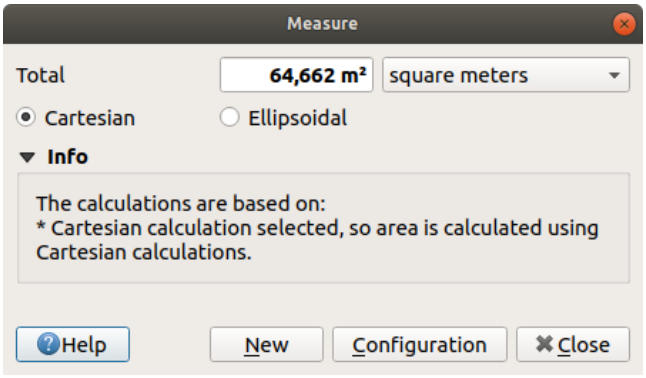


Fig. 7.15: Measure Area

Measure Bearing: You can also measure bearings. The cursor becomes cross-shaped. Click to draw the first point of the bearing, then move the cursor to draw the second point. The measurement is displayed in a pop-up dialog.

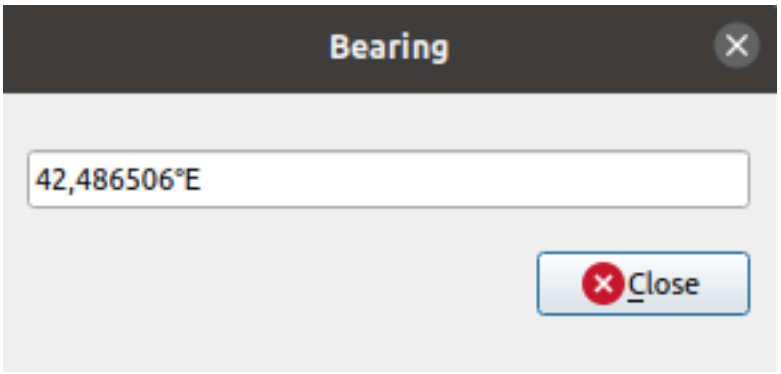


Fig. 7.16: Measure Bearing

Measure Angle: You can also measure angles. The cursor becomes cross-shaped. Click to draw the first segment of the angle you wish to measure, then move the cursor to draw the desired angle. The measurement is displayed in a pop-up dialog.

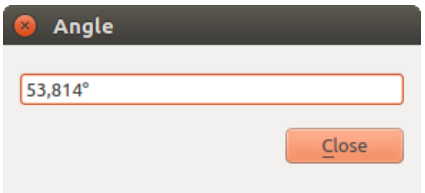


Fig. 7.17: Measure Angle

7.1.9 设置其他地图视图

It is also possible to open additional map views whose content could diverge from the *Layers* panel current state. To add a new map view, go to *View* ► *New Map View*. A new floating widget, mimicking the main map view's rendering, is added to QGIS. You can add as many map views as you need. They can be kept floating, placed side by side or stacked on top of each other.

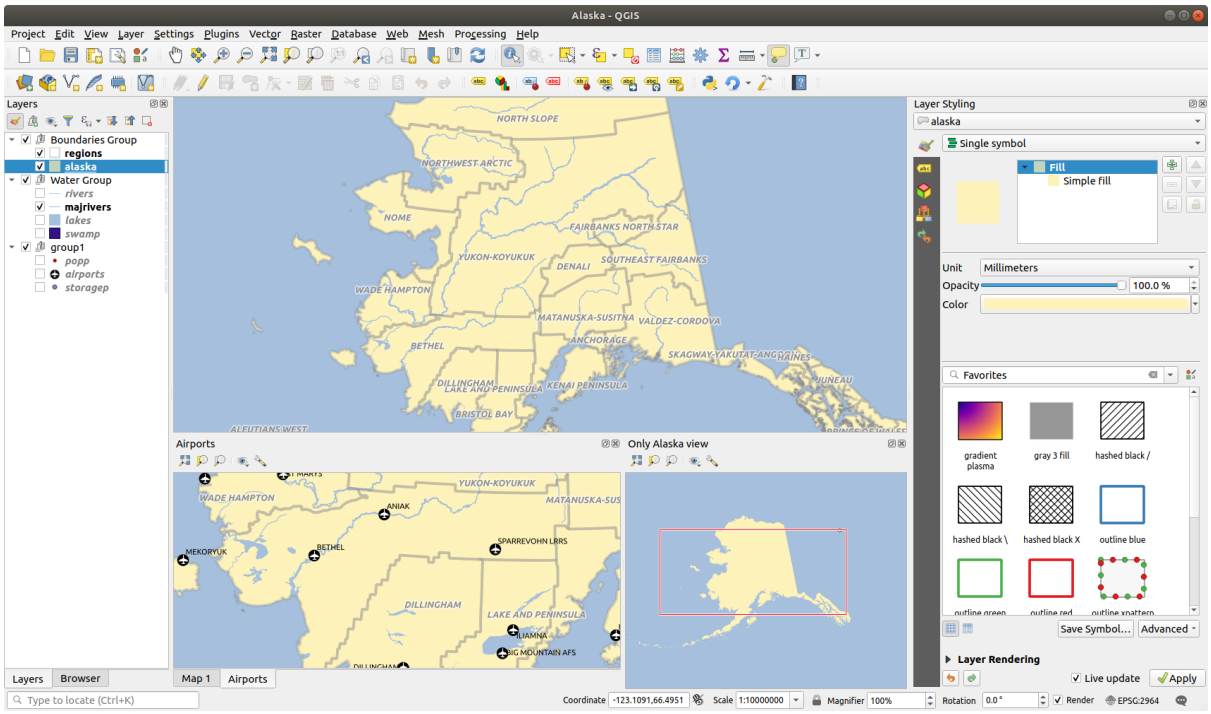


Fig. 7.18: 具有不同设置的多个地图视图

在新建的地图画布顶部，有一个具有以下功能的工具栏：

- 全图显示, 缩放到选中要素的范围 和 缩放到图层 可在视图中导航
- 设置视图主题 以选择要在地图视图中显示的:ref: 地图主题 <*map_themes*>。如果设置为（无），视图将跟随 图层面板的变化而变化。
- 视图设置 用来配置地图视图:
  - 视图中心与主地图同步: 在不更改比例的情况下同步地图视图的中心，这允许您在主画布中心有一个鹰眼图样式或放大的地图。
  - 将视图与所选内容同步: 与" 缩放到选中的区域" 作用相同
  - 比例
  - 旋转角度
  - 放大
  - 同步比例与主地图比例一致。然后可以应用 比例因子，使您可以拥有一个例如始终是主画布比例 2 倍的视图。
  - 显示注记
  - 显示指针位置
  - 显示主要画布范围

- 显示标注: 允许隐藏标注，无论它们在显示图层的属性中如何设置
- 更改地图 *CRS*...
- 重命名视图...

7.1.10 导出地图视图

使用打印布局或报告 的高级功能，可以将您制作的地图布局导出为各种格式，也可以不使用布局直接导出当前渲染。这个地图视图的快速" 截图" 有一些方便的功能。

要使用当前渲染导出地图画布，请执行以下操作:

1. 转到 工程 ► 导入/导出
2. 根据您的输出格式，选择
   - 导出地图为图像...
   - 或 导出地图为 *PDF* 文件...

这两个工具为您提供了一组通用选项。在打开的对话框中:

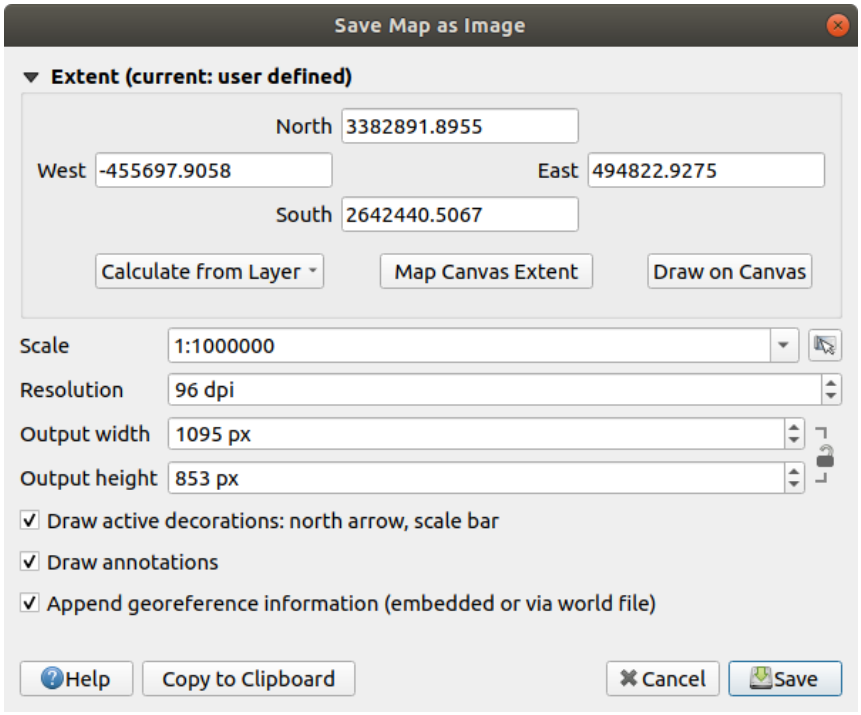


Fig. 7.19: 地图另存为图像对话框

1. 选择要导出的 范围: 可以是当前视图范围 (默认)、图层范围或在地图画布上绘制的自定义范围。将显示所选区域的坐标，并可手动编辑。
2. 输入地图 比例或从预定义比例 选择: 更改比例将调整要导出范围的大小 (从中心开始)。
3. 设置输出的 分辨率
4. 以像素为单位控制图像的 输出宽度和 输出高度: 默认是基于当前分辨率和范围，可以自定义并将调整地图范围的大小 (从中心开始)。大小比值可以锁定，这在画布上绘制范围时可能特别方便。
5. 绘制活动装饰: 使用中的装饰 (比例尺、标题、网格、指北针等) 与地图一起导出
6. 绘制注记导出全部注记

7. 附加配准信息 (嵌入或者借助于坐标信息文件): 根据输出格式，一个同名的坐标信息文件 (扩展名为 PNGW 表示 PNG 图像，扩展名为 JPGW 表示 JPG, ...) 与您的图像保存在同一文件夹中。PDF 格式将信息嵌入到 PDF 文件中。
8. 导出为 PDF 时，地图另存为 *PDF...* 对话框提供了更多选项：

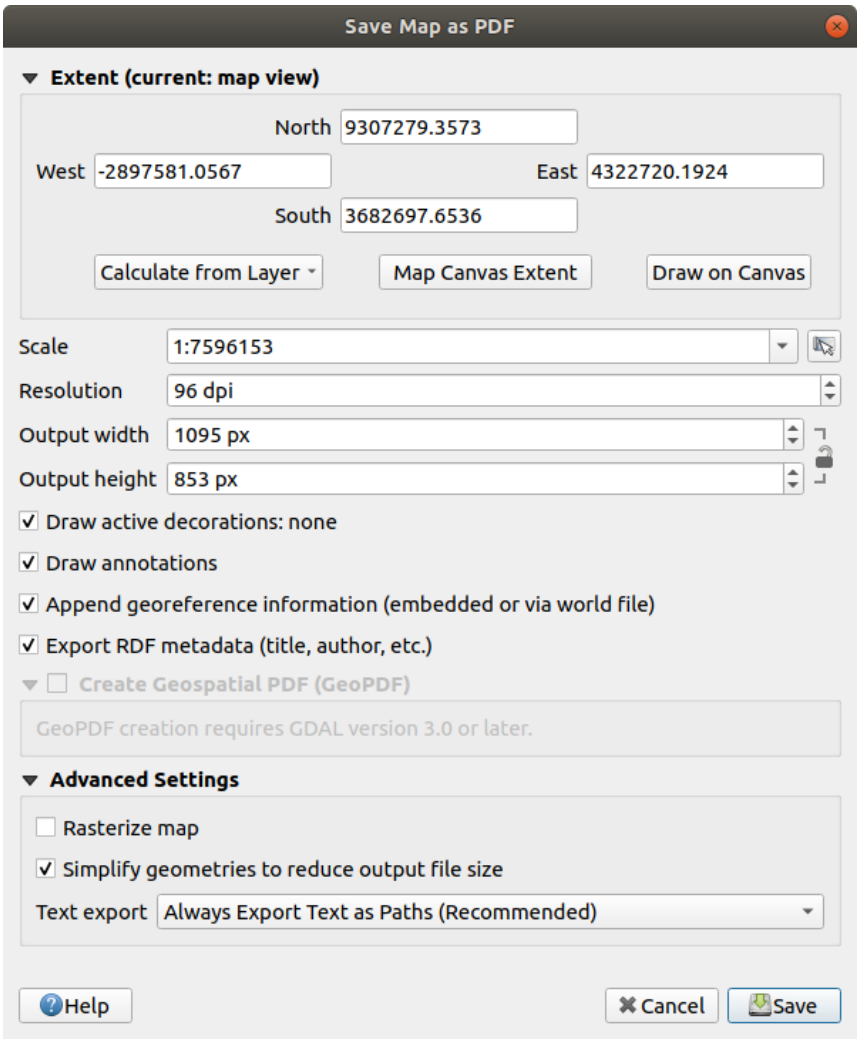


Fig. 7.20: 地图另存为 PDF 的对话框

- 文档的 导出 *RDF* 元数据，例如标题、作者、日期、描述...
- *Create Geospatial PDF*: Generate a georeferenced PDF file. You can:
  - Choose the Geospatial PDF *Format*
  - *Include vector feature information* in the Geospatial PDF file: will include all the geometry and attribute information from features visible within the map in the output Geospatial PDF file.

> **Note**
>
> A Geospatial PDF file can also be used as a data source. For more on Geospatial PDF support in QGIS, see https://north-road.com/2019/09/03/qgis-3-10-loves-geospatialpdf/.

- 地图栅格化
- *Simplify geometries to reduce output file size*: Geometries will be simplified while exporting the map

by removing vertices that are not discernibly different at the export resolution (e.g. if the export resolution is `300 dpi`, vertices that are less than `1/600 inch` apart will be removed). This can reduce the size and complexity of the export file (very large files can fail to load in other applications).

- Set the *Text export*: controls whether text labels are always or preferably exported as *text or outline objects*.

9. 单击 保存后可以选择文件位置、名称和格式。

   导出到图像时，还可以将上述设置的预期结果 复制到剪贴板，并将地图粘贴到另一应用程序中，如 LibreOffice、GIMP...

7.2 3D 地图视图

3D visualization support is offered through the 3D map view. You can create, manage and open 3D map views via *View* ► *3D Map Views* ► menu:

1. By clicking on *New 3D Map View* you can create a new 3D map view. A floating and dockable QGIS panel will appear (see *3D 地图视图对话框*). It has the same extent and view as the 2D main map canvas and provides a set of navigation tools to turn the view into 3D.
2. By clicking on *Manage 3D Map Views* you get in the 3D Map Views Manager. Here you get the ability to open, duplicate, remove and rename 3D map views.
3. If you created one or more 3D map views, you see them listed in *3D Map Views*. You can turn them on and off by clicking on. They will be saved by saving the project, even if they are turned off.

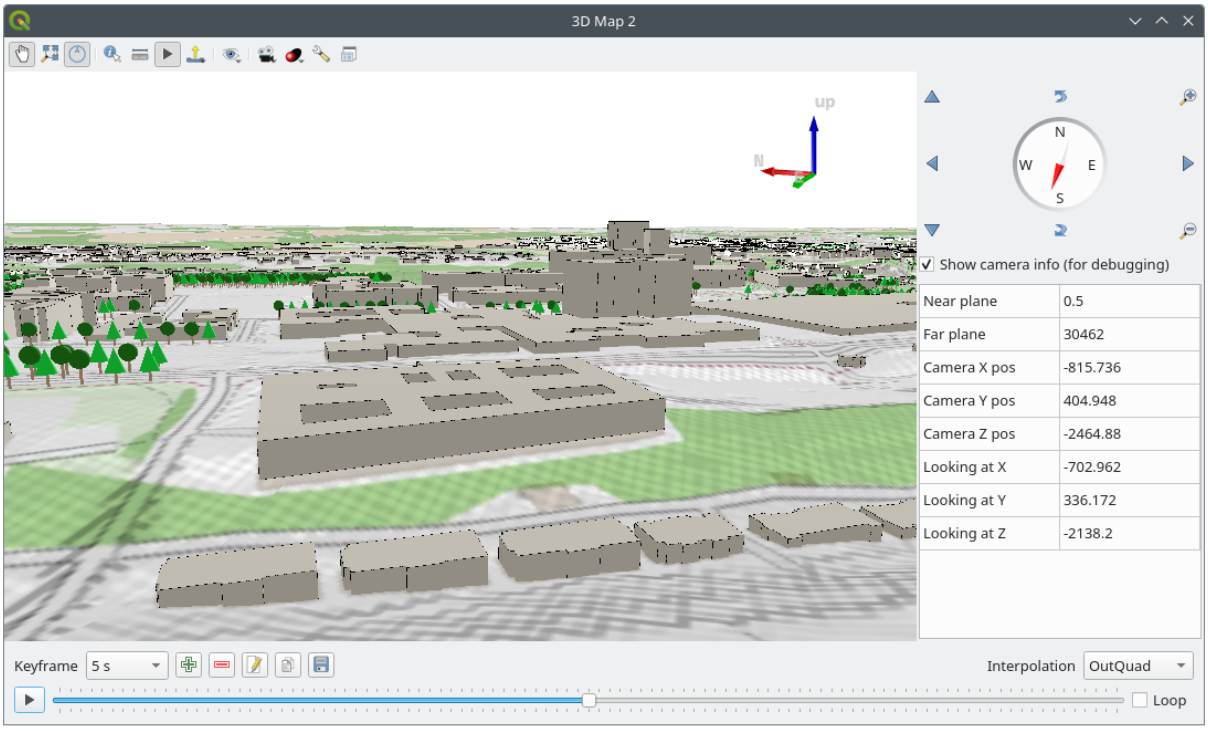


Fig. 7.21: 3D 地图视图对话框

3D 地图视图面板顶部提供了以下工具：

- Camera Control: moves the view, keeping the same angle and direction of the camera
- Zoom Full: resizes the view to the whole layers' extent, or the *reference extent* if set

- Toggle On-Screen Notification: shows/hides the navigation widget (that is meant to ease controlling of the map view)
- 识别: 返回有关地形的单击点或单击 3D 要素的信息 -- 更多详细信息请访问识别要素
- Measurement Line: measures the horizontal, vertical, and total 3D distance between points
  - Click to add points
  - Right-click to finish the measurement
  - Right-clicking twice or pressing *New* will reset the table
- 动画: 显示/隐藏动画播放 控件
- Export menu allows to export the scene to various formats:
  - Save as Image...: exports the current view to an image file format
  - Export 3D Scene: exports the current view as a 3D scene (`.obj` file), allowing post-processing in applications like Blender... The terrain and vector features are exported as 3D objects. The export settings, overriding the layers *properties* or map view *configuration*, include:
    - 场景名称和目标 文件夹
    - 地形分辨率
    - 地形纹理分辨率
    - 模型比例
    - 平滑边
    - 导出法线
    - 导出纹理
- 设置视图主题: 允许您从预定义的地图主题 中选择要在地图视图中显示的图层集。
- The Camera menu helps you control relation between the 2D and 3D views:
  - Synchronize the views (*2D map view follows 3D camera* and/or *3D camera follows 2D Map view*)
  - *Show visible camera area in 2D map view*
  - *Set 3D scene on 2D map view*: allows to clip the 3D scene and display only the terrain and features intersecting an extent drawn on the 2D map canvas. More options are available in the *General configuration* tab.
  - Cross Section Tool creates a cross section in 3D scene by defining an area on the terrain in the 2D map canvas. Click to set the start point, move the mouse to define the line direction and click again to set the end point. You can adjust the size of the section by moving the mouse after selecting the start and end points. Press `Esc` or right-click to reset the selection.
  - Use Disable Cross Section to remove the cross section.
- Effects adds visual effects to the 3D rendering, such as showing *shadows*, *eye dome lighting* or *ambient occlusion*.
- The Configure... button opens the dialog to configure the 3D map view *settings*.
- Dock 3D Map View: switch from docked widget to top level window

7.2.1 场景配置

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

在 3D 配置窗口，有各种选项可微调 3D 场景：

General

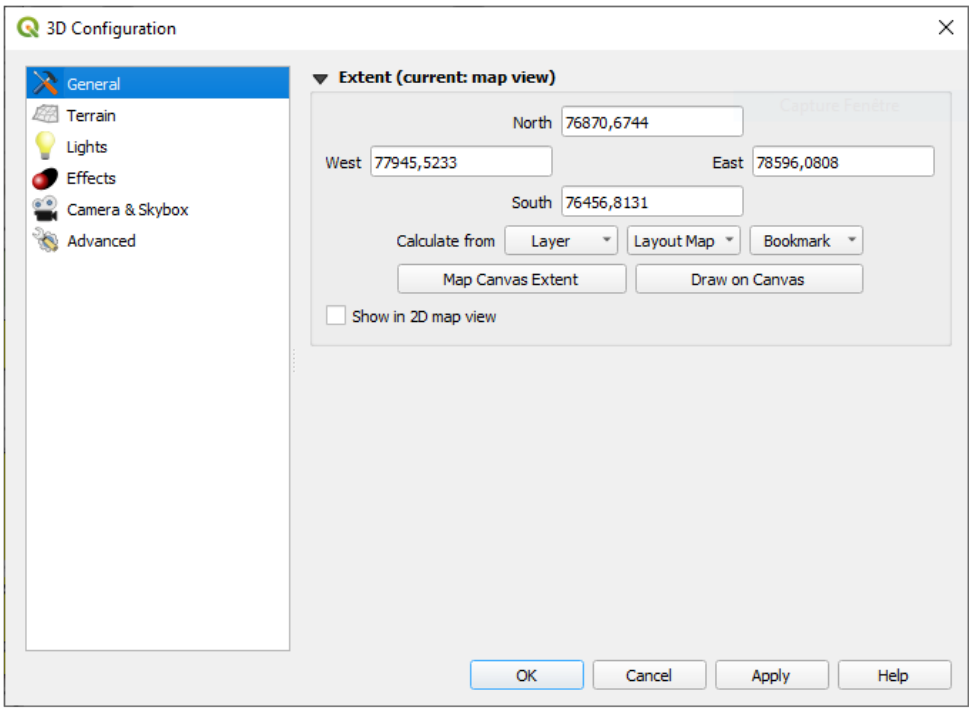


Fig. 7.22: The 3D Map General Configuration dialog

Under the *General* tab, you can:

- Limit the 3D scene's contents to a 2D map extent, using the *spatial extent selector*: terrain and features are clipped at the specified extent and only the parts falling within the extent are loaded in the scene. Elevation range of the terrain, mesh and pointcloud layers is also taken into account so that the camera is not positioned below the scene's contents, particularly when using the terrain's vertical scale setting to exaggerate the elevation differences.

  The set extent is also used as reference extent when pressing the Zoom full button in 3D map view.

- Check *Show in 2D map view* to display in the main map canvas a rubberband corresponding to the current extent of the 3D scene.

地形

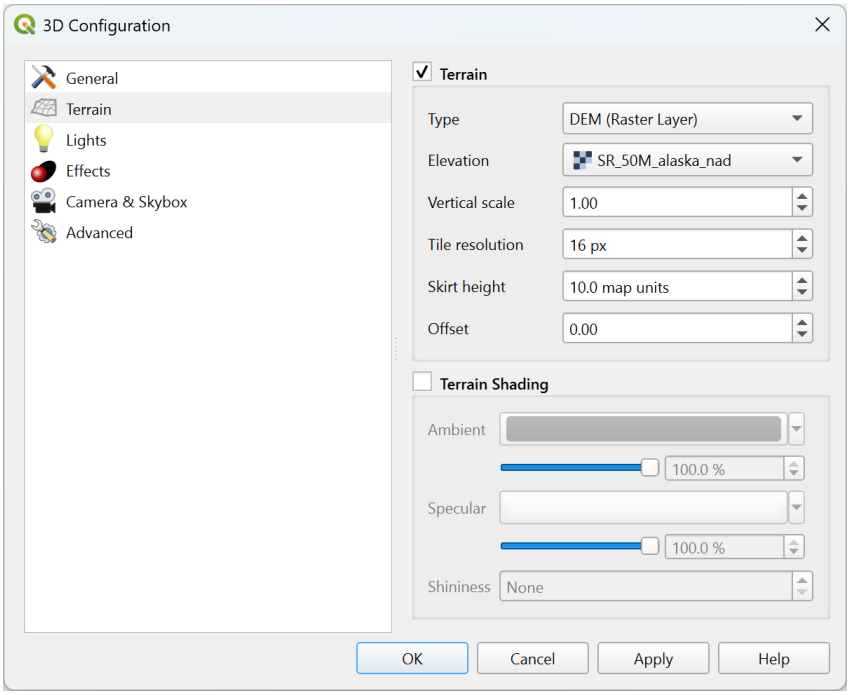


Fig. 7.23: The 3D Map Terrain Configuration dialog

- 地形: 在深入研究细节之前，值得注意的是，3D 视图中的地形是由地形瓦片的层次结构表示，随着相机靠近地形，没有足够细节的现有瓦片将被具有更多细节的较小瓦片替换。每个瓦片都是源自高程栅格图层的网孔几何图形和 2D 地图图层的纹理。
  - 高程地形 类型可以是:
    - a 平坦地形
    - 已加载的 *DEM (栅格图层)*
    - 在线服务，加载由 Mapzen 工具生成的 高程瓦片 -- 更多详细信息，请访问 https://registry.opendata.aws/terrain-tiles/
    - 已加载的 网孔数据集
    - a *Quantized Mesh* layer from the project
  - 高程: 用于生成地形的栅格或网孔图层。栅格图层必须包含表示高程的波段；对于网孔图层，将使用顶点的 Z 值。
  - 垂直比例: 垂直轴的比例因子，增加比例会扩大地貌的高度。
  - 瓦片分辨率: 每个瓦片要使用的地形栅格图层中的样本数。16px 的值意味着每个瓦片的几何图形将由 16x16 个高程样本组成。数字越大，生成的地形瓦片越详细，但渲染复杂度就越高。
  - 裙板高度: 有时可能会在地形的瓦片之间看到小裂缝。提高该值将在地形瓦片周围添加垂直墙(" 裙板 - skirts")，以隐藏裂缝。
  - *Offset*: moves the terrain up or down, e.g. to adjust its elevation with respect to the ground level of other objects in the scene.

    This can be useful when there is a discrepancy between the height of the terrain and the height of layers in your scene (e.g. point clouds which use a relative vertical height only). In this case adjusting the terrain elevation manually to coincide with the elevation of objects in your scene can improve the navigation experience.

- When a mesh layer is used as terrain, you can configure the *Triangles settings* (wireframe display, smooth triangles, level of detail) and the *Rendering colors settings* (as a uniform color or *color ramp based*). More details in the *Mesh layer 3D properties* section.
- 地形晕渲: 允许您选择地形的渲染方式:
  - 禁用晕渲 - 地形颜色仅由地图纹理确定
  - 启用晕渲 - 使用 Phong 的晕渲模型确定地形颜色，同时考虑地图纹理、地形法线矢量、场景灯光以及地形材质的 周围环境和 镜面反射颜色以及 光滑度

灯光

在 灯光选项卡中，按 菜单添加

- 最多八个 点光源: 向各个方向发光，就像一个填充区域的光球。离光源越近的对象越亮，而远离光源的对象就越暗。点光源具有设定的位置 (*X*, *Y* 和 *Z*)、颜色、强度和 衰减等属性
- 最多四个 平行光: : 模拟从远离对象的巨型闪光灯中获得的照明，始终居中且永不熄灭 (例如太阳)。它在一个方向上发射平行光线，但光线延伸到无限远。平行光可以在给定 方位角、海拔、颜色和强度的参数下进行旋转。

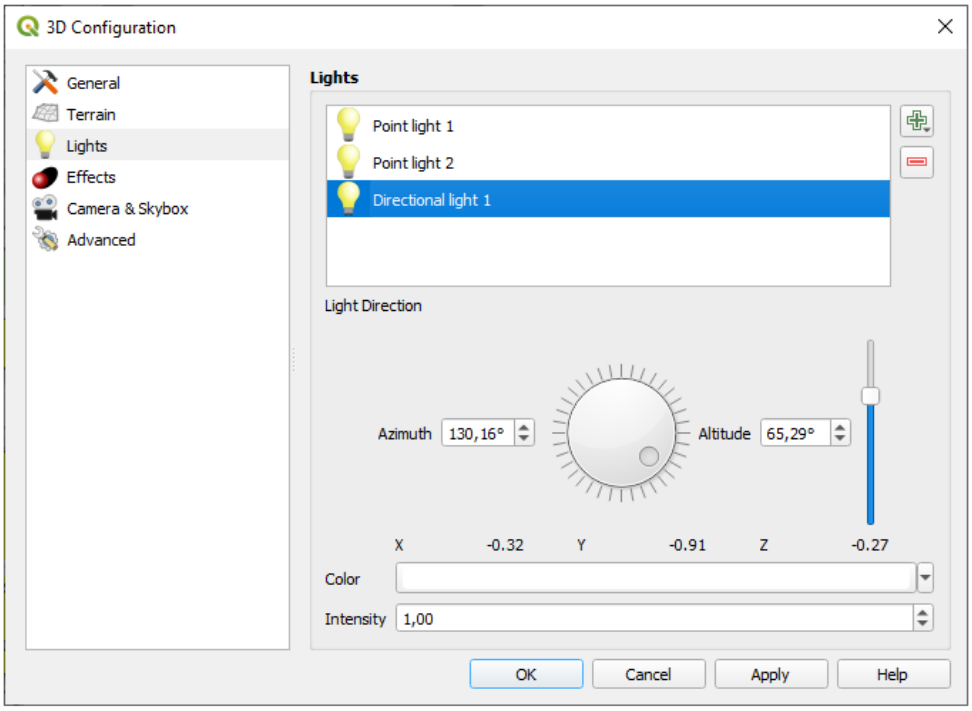


Fig. 7.24: 3D 地图灯光配置对话框

Effects

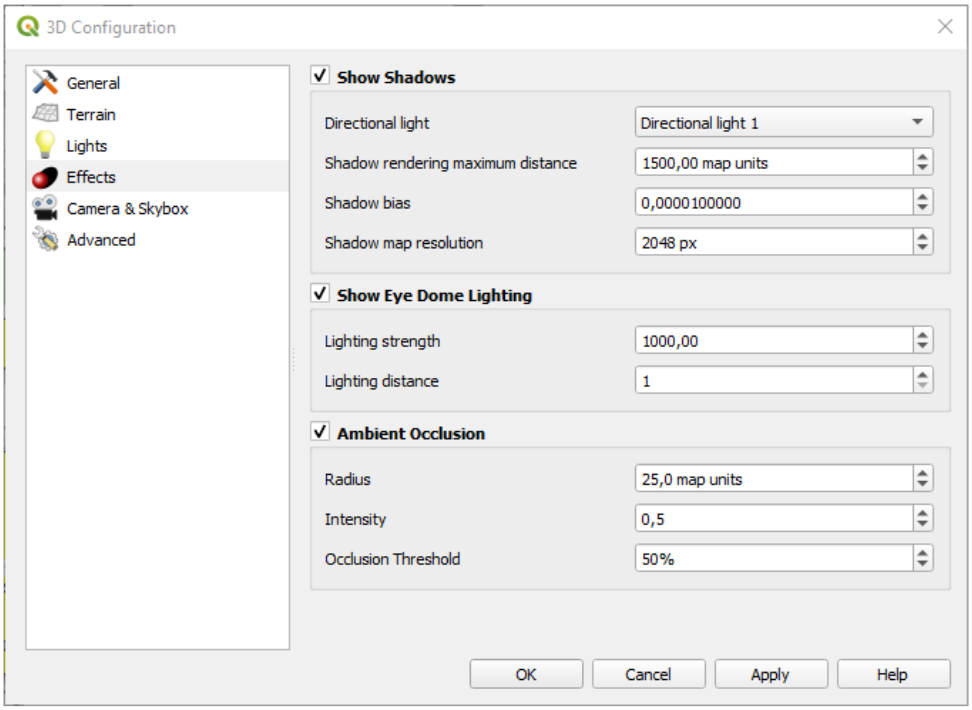


Fig. 7.25: The 3D Map Effects Configuration dialog

- Check *Show shadows* to display shadows within your scene, given:
  - 平行光
  - 阴影渲染最大距离: 避免渲染太远对象的阴影，尤其是当相机沿地平线仰视时
  - 阴影偏移: 避免由于地图大小不同而导致某些区域比其他区域更暗的自阴影效果，越低越好
  - 阴影地图分辨率: 使阴影看起来更清晰。如果分辨率参数太高，可能会导致性能下降。
- *Show Eye Dome Lighting* (EDL): a post processing effect which enhances depth perception. Each pixel's depth (distance off the camera) is compared to its neighboring pixels' depth and gets highlighted according to that depth difference, making the edges stand out. Affects the whole scene and can be combined with *Screen Space Ambient Occlusion*. Following parameters can be controlled:
  - *Lighting strength*: increases the contrast, allowing for better depth perception
  - *Lighting distance*: represents the distance of the used pixels off the center pixel and has the effect of making edges thicker.
- Add screen-space *Ambient Occlusion* (SSAO): a post processing effect which also enhances depth perception by applying a darker shading to areas which are less exposed to ambient lighting. Affects the whole scene and can be combined with *Eye dome Lighting*. Following parameters can be controlled:
  - *Radius*: how far we will reach to calculate ambient occlusion
  - *Intensity*: how strong the effect should be (higher values make things darker)
  - *Occlusion threshold*: how many neighboring points need to be occluded for the effect to appear (lower values than 50% will make the output darker, but possibly providing greater range of occlusion)

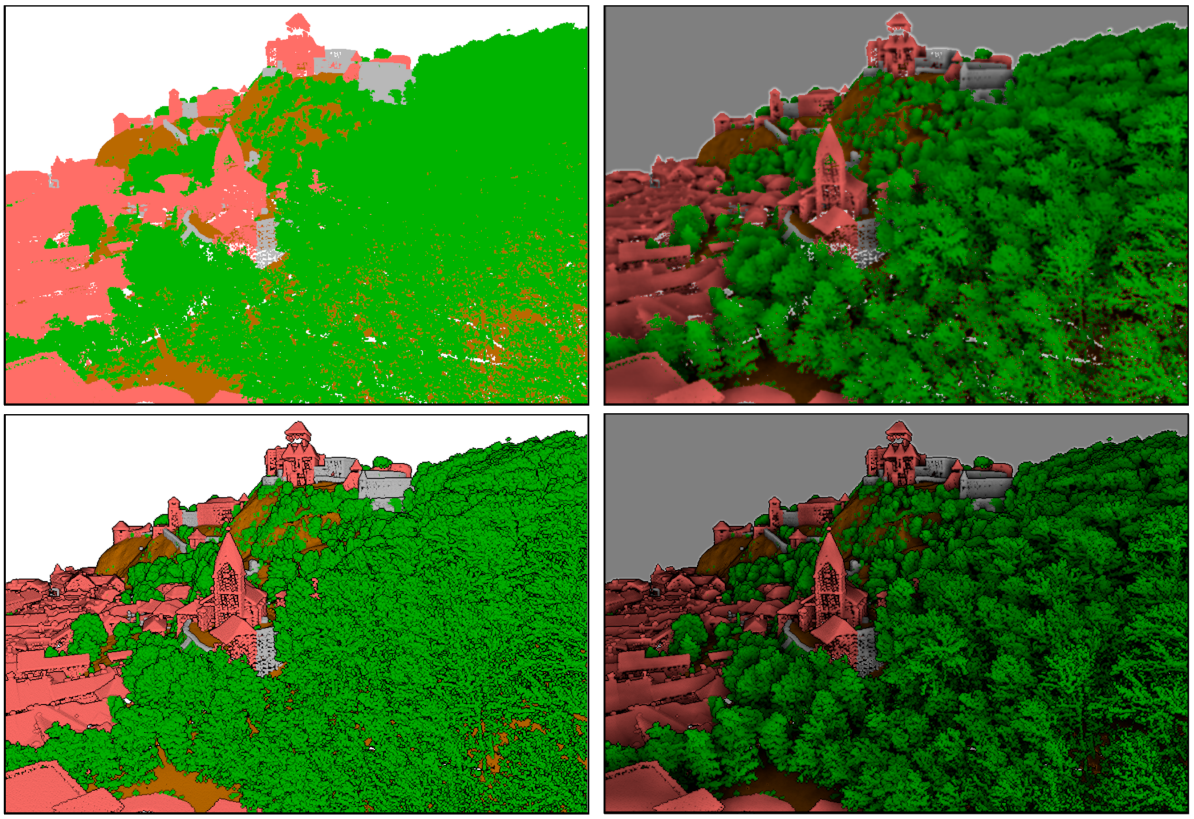


Fig. 7.26: Rendering Point clouds in 3D map using Eye Dome Lighting (EDL) and/or Screen-Space Ambient Occlusion (SSAO)

From top, left to right: No effect -- SSAO only -- EDL only -- SSAO and EDL

相机和天空盒

In this tab, you can control different parameters like camera, 3D axis, navigation synchronization and skybox.

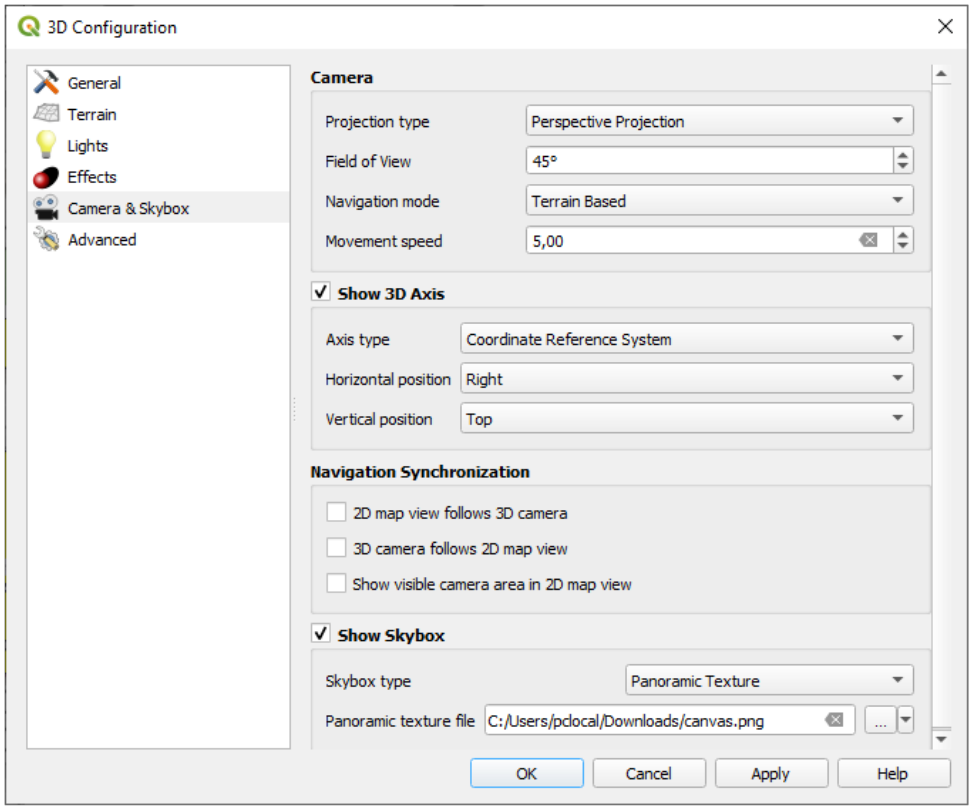


Fig. 7.27: The 3D Map Camera Configuration dialog

- The *Camera* parameter group overrides some *default camera settings* made in the *Settings* ► *Options* ► *3D* dialog.
- Check *Show 3D Axis* to enable 3D axis tool. This parameter group allows to set the axis type and its position.
  - With the *Coordinate Reference System* type an orthogonal axis will be represented.
  - With the *Cube* type, a 3D cube will be represented. The cube faces can be used to change the camera view: for example, click on the *north* face to set the camera to see from the north.

Tip

Right-click the 3D axis to quickly set its position and type, and the camera view.

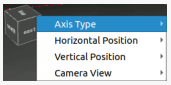


Fig. 7.28: The 3D Axis context menu

- The *Navigation Synchronization* parameter group adds options to synchronize 2D view with 3D camera position or 3D camera position with 2D view or bi directional synchronization. The last option displays the extent visible from the 3D camera over the 2D map view.
- Check *Show skybox* to enable skybox rendering in the scene. The skybox type can be:
  - 全景纹理, 单个文件提供 360° 视野
  - 不同表面, 包含场景的盒子的六个面各有一个纹理文件

Texture image files of the skybox can be files on the disk, remote URLs or embedded in the project (*more details*).

高级

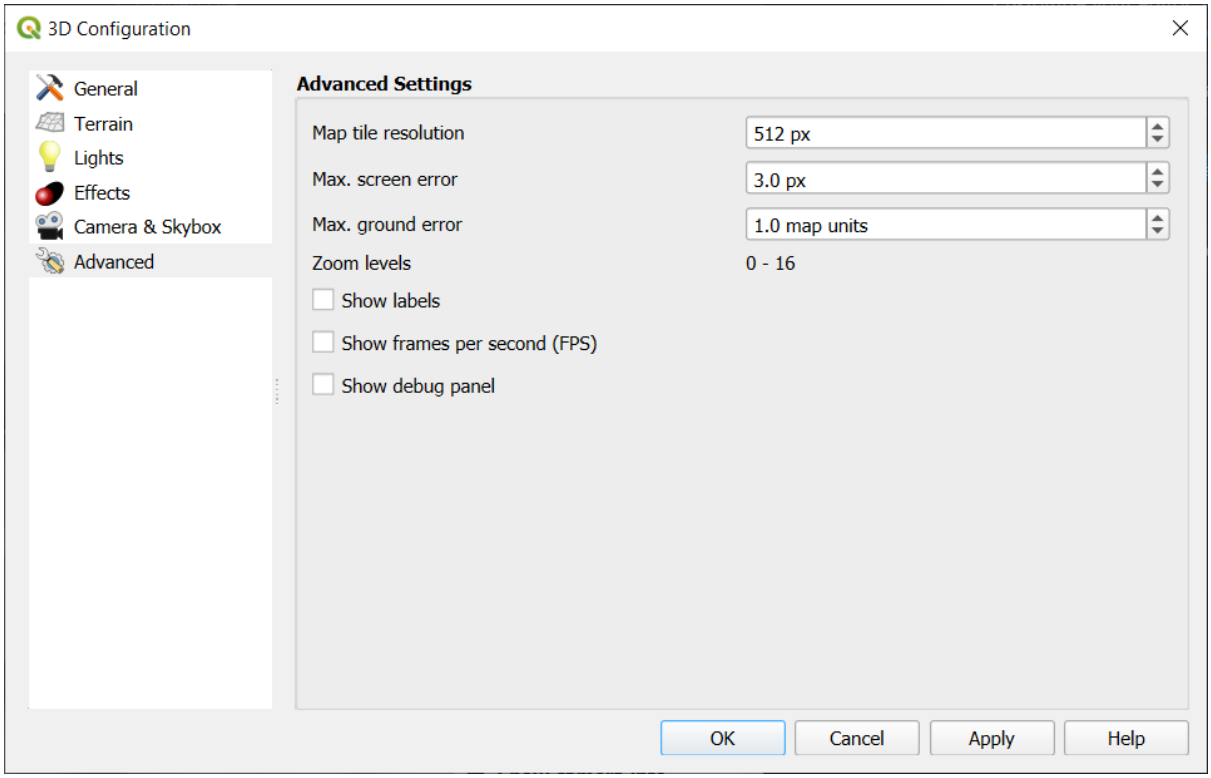


Fig. 7.29: The 3D Map Advanced Configuration dialog

- 地图瓦片分辨率: 用作地形瓦片纹理的 2D 地图图像的宽度和高度。256px 意味着每个瓦片被渲染成 256x256 像素的图像。数字越大，生成的地形地形越详细，但渲染复杂度越高。
- 最大屏幕误差: 确定将地形瓦片与更详细的地形瓦片交换的阈值 (反之亦然) - 即 3D 视图使用更高质量瓦片的时间。数字越小，场景中的细节越多，但渲染复杂度越高。
- 最大地面误差: 地形瓦片的分辨率，在该分辨率下，将瓦片拆分为更详细的瓦片将停止 (分割它们不会引入任何额外的细节)。该值限制了瓦片层次结构的深度：值越低，层次结构越深，渲染复杂度越高。
- 缩放层级: 显示缩放层级的数量 (取决于地图瓦片分辨率和最大地面误差)。
- ☐ 显示标注: 打开或关闭地图标注
- ☐ *Show frames per second (FPS)*
- ☐ *Show debug panel*: allows to show or hide a panel next to the 3D map view displaying options to better control map navigation and debug information:
  - ☐ 显示地图瓦片信息: 包括地形瓦片的边界和瓦片编号 (用于解决地形问题)
  - ☐ 显示边界框: 显示地形瓦片的 3D 边界框 (用于解决地形问题)
  - ☐ 显示相机视图中心
  - ☐ *Show camera's rotation center*

  - 显示光源: 显示光源原点处的球体，相对于场景内容更容易重新定位和放置光源
  - *Stop scene updates*
  - *Show debug overlay*: visual overlay which displays some useful debugging and profiling information. This allows in particular to quickly see the frame graph and the scene graph
  - *Debug Shadow Map*: renders the scene as a red-black image from the point of view of the light used for shadows (for troubleshooting). The widget is set with a proportional *Size* to the 3D map view's, and docked in a *Corner*.
  - *Debug Depth Map*: renders the scene's depth map as an image with nearer pixels being darker (for troubleshooting). The widget is set with a proportional *Size* to the 3D map view's, and docked in a *Corner*.
  - *Show camera info*:
    * *Far plane*: controls how far from the camera rendering stops
    * *Near plane*: controls how close to the camera rendering starts
    * *Camera X/Y/Z pos*: sets the camera's position in 3D space
    * *Looking at X/Y/Z*: sets the target point the camera is looking at

Note

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

7.2.2 导航选项

要在 3D 中浏览地图视图，请执行以下操作：

- 倾斜地形 (围绕穿过窗口中心的水平轴旋转地形)
  - 按下 向上倾斜 和 向下倾斜 工具
  - 按下 `Shift` 并使用向上/向下方向键
  - 按住鼠标中键，向前/向后拖动鼠标
  - 按住 `Shift` 并在按住鼠标左键的情况下向前/向后拖动鼠标
- 旋转地形 (围绕穿过窗口中心的垂直轴旋转地形)
  - 将导航控件的罗盘转到观察方向
  - 按住 `Shift` 并使用左/右键
  - 按住鼠标中键，左右拖动鼠标
  - 按住 `Shift` 并在按住鼠标左键的情况下左右拖动鼠标
- 更改相机位置 (和视图中心)，在水平平面移动相机
  - 按住鼠标左键拖动鼠标，并启用 相机视角控制
  - 按导航控件的方向箭头
  - 使用上/下/左/右键分别向前、向后、向右和向左移动相机
- 更改相机高度：按 `Page Up`/`Page Down` 键
- 更改相机方向 (相机保持在其位置，但视图中心点移动)
  - 按住 `Ctrl` 并使用方向键上下左右转动相机

  - 按住 Ctrl 并在按住鼠标左键的情况下拖动鼠标
- 放大和缩小
  - 按导航控件中相关的 放大 和 缩小 工具
  - 滚动鼠标滚轮 (按住 Ctrl 键可获得更精细的缩放效果)
  - 按住鼠标右键拖动鼠标进行放大 (向下拖动) 和缩小 (向上拖动)

要重置相机视图，请单击 3D 画布面板顶部的 全图显示 按钮。

7.2.3 创建动画

动画基于一组关键帧 - 特定时间的相机位置。要创建动画，请执行以下操作：

1. 切换 动画 工具，显示动画播放控件
2. 单击 添加关键帧 按钮并以秒为单位输入 关键帧时间，关键帧组合框现在显示时间设置。
3. 使用导航工具，将相机移动到与当前关键帧时间相关联的位置。
4. 重复前面的步骤，根据需要添加若干关键帧 (包括时间和位置)。
5. 单击 按钮预览动画。QGIS 将使用相机在设定时间的位置或旋转生成场景，并在这些关键帧之间进行插值。可以使用各种动画 插值模式 (例如，linear、inQuad、outQuad、inCirc…-- 更多详细信息，请访问 https://doc.qt.io/archives/qt-5.15/qeasingcurve.html#EasingFunction-typedef)。

   The animation can also be previewed by moving the time slider. Keeping the *Loop* box checked will repeatedly run the animation while clicking stops a running animation.

单击 导出动画帧 将生成一系列表示场景的图像。除了文件名 模板和 输出目录，还可以设置 帧率数、输出宽度和 输出高度。

7.2.4 3D 矢量图层

通过选中矢量图层属性的 *3D* 视图部分中的 *启用 3D 渲染器*，可以在 3D 地图视图中显示具有高程值的矢量图层。有许多选项可用于控制 3D 矢量图层的渲染。

7.3 Elevation Profile View

The *Elevation Profile* panel is a plotting tool for side view, for visualizing elevation data along a line. It supports vector, raster, mesh and point cloud layers. Data can be of 2D or 3D type.

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

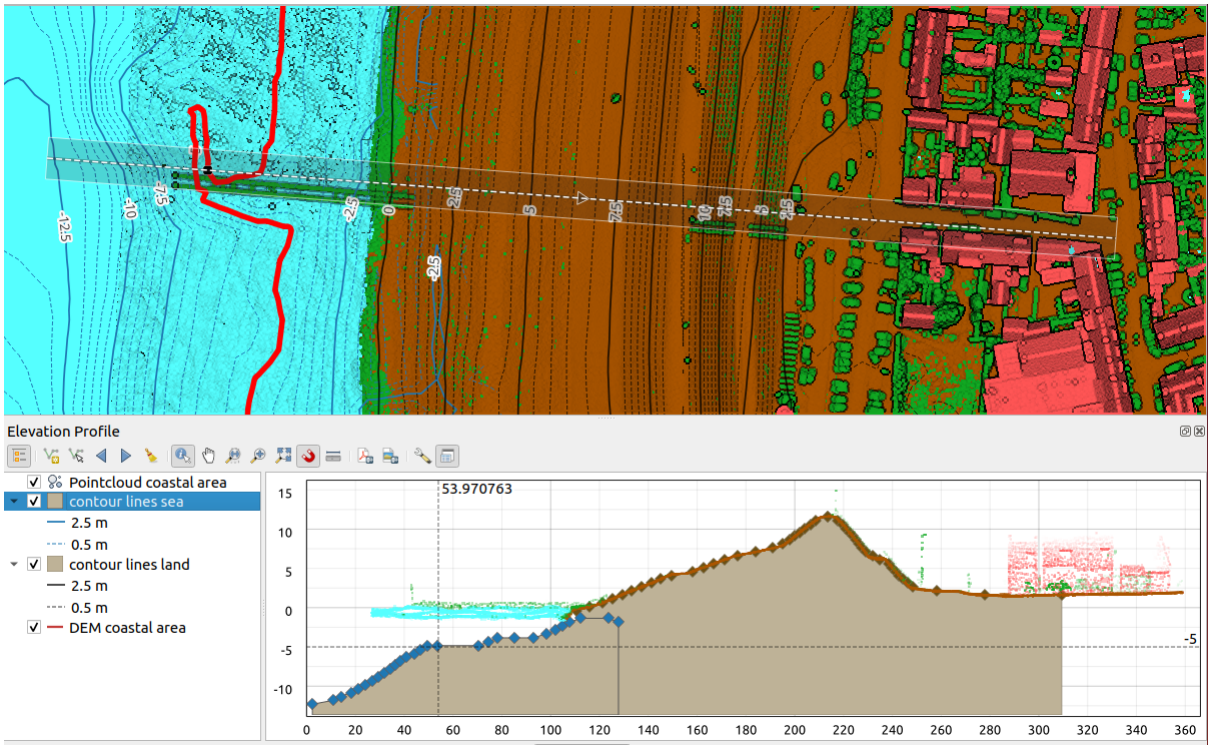


Fig. 7.30: Elevation Profile dialog embedded below main map canvas
Figure shows a coastline with a town behind a sea-dike. A larger tolerance applied to the elevation profile line returns more point cloud features.

7.3.1 The interface

At the top of the *Elevation Profile* panel, a toolbar provides you with the following tools:

Table 7.5: Elevation Profile View toolbar

工具	快捷键	描述
Add Layers	hold `Ctrl` while drag-and-drop	Allows you to add new layers to a plot.
Show Layer Tree		Shows or hides a list of project layers to configure rendering in the profile view.
Capture Curve		Draws interactively a line over the map canvas to represent the profile curve.
Capture Curve From Feature		Picks an existing line feature on the map canvas and generate a profile curve along that line.
Nudge Left	`Ctrl+Alt+,`	Allows you to slowly move the capture line across the map to the left (e.g. to find the optimal profile line based on the elevation).
Nudge Right	`Ctrl+Alt+.`	Allows you to slowly move the capture line across the map to the right (e.g. to find the optimal profile line based on the elevation).
Clear		Removes the profile line and any plot displayed in the *Elevation Profile* view

continues on next page

Table 7.5 – continued from previous page

工具	快捷键	描述
Identify Features		Identifies features in the plot canvas via either a single click, or click-and-drag rectangle. Results are shown in the standard *Identify Results* dock.
Pan	`Space`	Click and drag to pan the plot canvas. Can also be done with the middle mouse button.
Zoom X Axis		Zooms in/out along the horizontal axis, keeping the vertical ratio
zoom	`Ctrl+Space`	Click or click-and-drag a rectangle over the plot to zoom in. Press `Alt` and click to instead zoom out.
Zoom Full		Zooms the *Elevation Profile* view to the extent of the capture line
Enable Snapping		Allows to snap to the edge or vertex of the plot features in the profile view. Convenient for accurate retrieval of coordinates or distance measurements.
Measure Distances		Measures horizontal and vertical distances
Export As PDF		Exports plots to PDF (as high quality vector objects)
Export As Image		Exports plots to several image formats
Export Results		Exports plots as 3D features, 2D Profile and Distance/Elevation Table. (The results can be saved as DXF files, CSV files or any of the standard writable vector spatial formats). More details at *Exporting from the elevation profile*.
Options ►		Provides access to configuration settings of the profile elevation line.
► *Lock distance/elevation scales*		Ensures that the horizontal and vertical scales are always kept equal (so that e.g. a 45° slope will appear as a 45° slope in the profile view).
► *Distance units*		Allows to render distances in the profile chart with units other than the map canvas units.
► *Tolerance*		Sets how far from the actual profile line a feature (vector point, line or polygon, point cloud) can reside to be included in the results. Uses the map units and is ignored by other layer or geometry types. For some layer types (e.g. points, lines, polygons), a custom tolerance can be set in the layer's Elevation properties tab, which replaces the global tolerance and allows more control over feature inclusion based on distance to the profile line.
► *Rename Profile...*		Allows to rename the profile view.
Dock Elevation Profile View		Switch between docked and floating status of the view

In the bottom left, a copy of the *Layers* panel can be displayed pushing the *Show Layer Tree* button. This is however an independent widget, with its own set of visible layers, in a custom stack order. It allows you to control layers rendering and behavior within the plot canvas:

- Tick the box next to the layer name to set whether it should be rendered in the plot canvas
- Drag-and-drop layers up or down to change the order of the layers
- style rendering of layers in the profile view: double-click a layer or right-click and select *Properties...* to open the layer's *Elevation* properties tab for configuration. A summary of elevation settings is displayed as tooltip

when hovering over the layer.

On the right of the layer tree, the plot canvas is the main place you can preview the elevation profile of the enabled layers. It is based on a graduated grid in which the horizontal axis displays the length of the profile line and the vertical axis displays the Z elevation of the observed features. It also allows a set of interactions such as zooming, panning, measuring, identifying features, ... using the tools at the top.

7.3.2 Creating an elevation profile

To create a profile view, you can:

1. Go to *View* ► *Elevation Profile* menu. The *Elevation profile* panel opens.
2. Create the profile line along which the terrain and the features will be rendered. Select a drawing tool:
   - Capture Curve: click left over the main map canvas to add vertices and click right to finalize a line that will be used as profile line
   - or Capture Curve From Feature: click a line feature on the map canvas to select it as the profile line. If multiple features are present at the clicked point then a popup menu will appear allowing you to select among them.

   All the line digitizing capabilities such as the *snapping options*, *tracing*, *digitizing techniques* or the advanced *digitizing panel* are available for use.

   The plot canvas may start rendering some features.
3. The next step is to configure the elevation properties of the layers you want to visualize.
   1. Push on the Show Layer Tree button to display the list of layers. By default, not all the layers of the project are loaded and referenced in the profile tool; e.g., raster layers which are not marked as having elevation data in their properties are ignored. Clicking the Add layers button will show a filtered list of possible layers which can be added to the plot, but which currently aren't in the plot. Applying the dialog with selected layers will automatically mark them as having elevation data and immediately add them to the plot.

      Tip

      Holding `Ctrl` key, it is possible to drag and drop additional layers from the main layer tree to one in the elevation profile tool.

   2. Toggle visibility of the layers you are interested in. These are the only ones rendered in the profile view and selected layers can be different from the main *Layers* panel's.
   3. Double-click a layer name or right-click and select *Properties*. The *Elevation* properties tab of the layer opens. This is the place you configure how each feature or terrain should render on the profile view. Available properties depend on the layer type:
      - *Raster Elevation Properties*
      - *Vector Elevation Properties*
      - *Point Cloud Elevation Properties*
      - *Mesh Elevation Properties*

      The profile view starts rendering terrain or features of active layers crossing the given profile curve, as soon as they have elevation properties configured.
4. Under Options drop-down menu, you can:

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

Note

Limitations with polygon extrusion

Geometry extrusion can be set in the *Elevation* properties of a layer, and rendered in the profile view. When tolerance is enabled, it is however not trivial to render extruded polygons, thus, for now, polygon extrusion is ignored.

7.3.3 Interacting with the profile Views

When an elevation profile line is created, the plot canvas zooms to its full extent. On the X-axis you can see the length of the profile and, on the Y-axis the height range between minimum and maximum height captured, both in map units.

When you move the mouse pointer in the elevation profile view, you will see two crossing dot lines:

- the vertical line shows the height information
- the horizontal line shows the distance from the beginning of the elevation profile Line

When you move the mouse pointer in the elevation profile canvas, you can also see a black dot move along the elevation profile line on the main map canvas. At the middle of the profile line, an arrow indicates its direction.

As for the main map canvas, QGIS provides means to navigate on the plot canvas:

- Pan is used to move the elevation profile extent in any direction you want. Holding `Space` key while moving the mouse also shifts the plot canvas extent.
- Zoom X Axis is used to zoom in along the horizontal axis, keeping the ratio of the vertical axis (the elevation) the same. Left click to stretch out the plot along the X axis, with the clicked point at the center of the axis, or drag a rectangle to stretch out the plot along the X axis to the rectangle width. Hold `Alt` while using Zoom X Axis to zoom out along that axis.
- Zoom is used to zoom in on a certain point (using left click), or to a certain extent (dragging a rectangle on the area). Hold `Alt` while using Zoom to instead zoom out. In combination with the `Ctrl` key you can zoom in or out more smoothly.
- Zoom Full is the default zoom level used at the beginning and shows the full extent of the profile line, with all returned features. Use it to reset the zoom level.

It is also possible to interact with the elements displayed in the plot canvas:

- Press Enable snapping button to accurately catch points, vertices or edges of the features, for an accurate measurement or coordinates report.
- Identify Features is used to identify features of the visible layers in the layer tree. You can drag a rectangle across several features in the profile view to query all of them. When compatible with the format (e.g. vector, point cloud), these features will be highlighted in the main map canvas.

- Measurement Distances: click or select two points in the plot canvas to report the horizontal *Distance*, the *Elevation* and the *Total distance* between them, in map units.
- Nudge Left and Nudge right are used to shift the position of the elevation profile line in the map canvas to its left or right. The plot canvas will be redrawn, showing features and terrain overlapping the profile line buffer. The line is moved sideways using the *Tolerance* value in Options menu.

> **Warning**
>
> Closing an elevation profile view or the project currently removes the view from the project.

For more details, give a look to QGIS elevation profile/cross section tool -- a deep dive!, a presentation done by Nyall Dawson.

7.3.4 Exporting from the elevation profile

The top toolbar of the *Elevation profile* dialog provides a variety of tools to export an elevation profile, allowing you to reuse the plots in another software:

- Export As PDF: Exports plots to PDF (as high quality vector objects). Following options are requested for generating the output file:
  - The *Page size*, selecting a page from a predefined list or setting custom dimensions, and the page orientation
  - The *Chart ranges* to control the extent of the chart to export:
    - on the X axis, the *Minimum distance* and *Maximum distance* from the profile curve starting point
    - On the Y axis, the *Minimum elevation* and *Maximum elevation*
  - The *Distance axis settings*: helps you configure the horizontal graduation of the grid over which the elevation profile is rendered. You can customize the *Label interval*, the *Major grid line interval* and the *Minor grid line interval*.
  - The *Elevation axis settings*, same as above, for vertical graduation
- Export As Image: Exports plots to an image file format. Specific *Export options* such as the *Image width* and *Image height* are requested, along with the *Chart ranges*, *Distance axis settings* and *Elevation axis settings* previously mentioned.
- Export results: Exports the profile results as a set of features of a vector layer (`.DXF`, `.CSV`, `.SHP`, `.GPKG`, ...). Depending on the output format, multiple layers may be returned when the geometry types of exported features differ.
  - *Export 3D features*: Exports profile lines as 3D features, with elevation values taken from the elevation slices and stored in exported geometry Z values.
  - *Export 2D profile*: Exports profiles as 2D profile lines, with elevation stored in exported geometry Y dimension and distance in X dimension (as shown in the elevation profile widget).
  - *Export distance/elevation table*: Exports profiles as a table of sampled distance vs elevation values.

CHAPTER 8

常用工具

8.1 上下文帮助

每当您需要详细的主题帮助时，可以通过对话框中的 *帮助* 按钮访问当前用户手册中的相应页面 --- 请注意，第三方插件指向专用 Web 页面。

8.2 面板

QGIS 默认提供了许多面板供您使用。下文对其中部分面板进行了描述，而其他面板可在文档的不同部分找到。QGIS 提供的默认面板的完整列表可通过 *视图* ► *面板* ► 菜单访问，在面板 中对提到的面板进行了详细介绍。

8.2.1 图层面板

图层面板 (也称为 *地图图例*) 列出了该工程的所有图层，并帮助您管理其可见性和设计地图。按 `Ctrl+1` 可以显示或隐藏面板。

QGIS provides a variety of ways to add layers to a project:

- using the *Add* button from the dedicated data provider tab in the *Data source manager* dialog
- from QGIS *Browser panel* or *DB Manager*: double-click, drag-and-drop files and layers onto QGIS or use the contextual menu
- drag-and-drop files from the Operating System files explorer onto QGIS

In all these scenarios, you can open one or many layers at a time. New layers are added to the *Layers* panel:

1. if dropped over the *Layers* panel, at the exact location they are released
2. in the other cases, at a location that respects the global *behavior used when adding new layers* setting
3. and in case of multiple layers, they are sorted in a way that increases the chance of their stacking being logical and features being visible as most as possible, using the following logic (top to bottom):
    - vector point layers
    - vector line layers

- vector polygon layers
- point cloud layers
- mesh layers
- raster layers

在 图层面板的顶部，有一个工具栏可用于：

- 打开图层样式停靠栏 (F7): 切换图层样式 面板的打开和关闭。
- 添加组: 请参阅与组和图层交互
- 管理地图主题: 控制图层的可见性，并将其排列在不同的地图主题 中。
- 在图例树中的过滤图层：
  - 按地图内容过滤图例: 只有设置为可见并且其要素与当前地图画布相交的图层，其样式才会在图层面板中渲染。否则，将对图层应用通用的 NULL 符号。基于图层符号，这是一种便捷的方式来识别哪些图层覆盖了关注区的要素。
  - 显示私有图层: 一个方便的快捷方式，用于在 图层面板显示私有图层 并与之交互，而无需修改工程设置。
  - *Show Broken Layers Only*: only layers with broken data sources are displayed.
- 按表达式过滤图例: 应用表达式从所选图层树中移除没有满足条件的要素样式。这可用于突出显示另一图层给定区域/要素内的要素。从下拉列表可以编辑和清除当前应用的表达式。
- 在图层面板 全部展开 或 全部折叠 图层和组。
- 从当前所选 移除图层/组 。

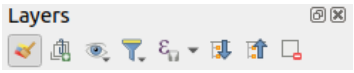


Fig. 8.1: 图层面板中的图层工具栏

> **Note**
>
> 用于管理图层面板的工具也可用于打印布局中的地图和图例项

配置地图主题

通过 管理地图主题 下拉按钮，可以使用方便的快捷方式来控制 图层面板中图层的可见性：

- 显示所有图层
- 隐藏所有图层
- 显示选中的图层
- 隐藏选中的图层
- 切换选中图层: 更改面板中首个选中图层的可见性，并将该状态应用于其他选中图层。也可以通过 空格键快捷方式访问。
- 独自切换选中图层: 更改每个选中图层的可见状态。

- 隐藏未选中的图层

除了简单地控制图层可见性之外，管理地图主题 菜单还允许在图例中配置 **地图主题**，并可以切换地图主题。地图主题是当前地图图例的 **快照**，其中记录：

- 在 图层面板设置为可见的图层
- **并且**对于每个可见图层：
  - 对应用于图层样式 的引用
  - 样式的可见类别，即 图层面板中的图层选中节点项。这适用于单个符号渲染之外的符号化
  - 图层节点及其所在组的折叠/展开状态

要创建地图主题，请执行以下操作：

1. 选中要显示的图层
2. 像往常一样配置图层属性 (符号化、图表、标注等)
3. 展开 样式 ► 菜单，然后单击 添加... 按钮，将设置储存为嵌入工程中的新样式

> **Note**
>
> 地图主题不记录当前属性的详细信息，只保存对样式名称的引用。因此，当启用此样式时，无论何时对图层应用修改 (例如更改符号化渲染)，地图主题都会更新为新信息。

4. 根据需要对其他图层重复上述步骤
5. 若适用，请在 图层面板展开或折叠组与可见图层节点
6. 单击面板顶部的 管理地图主题 按钮，然后 添加主题...
7. 输入地图主题的名称，然后单击 确定

新主题排列在 下拉菜单的中下部。

您可以根据需要创建任意数量的地图主题：每当地图图例 (可见图层、其有效样式、地图图例节点) 中的当前组合与上面定义的任何现有地图主题内容不匹配时，单击 添加主题... 创建新的地图主题，或使用 更换主题 ► 更新地图主题。可以使用 重命名当前主题... 重命名活动地图主题，或者使用 移除当前主题按钮将其删除。

地图主题有助于在不同预配置组合之间快速切换：在列表中选择一个地图主题恢复其组合。所有配置的主题也可以在打印布局中访问，允许您基于特定主题创建不同的地图项，并且独立于当前主画布渲染 (请参阅地图项图层) 。

图层面板上下文菜单概览

在工具栏的下面，图层面板的主要部分是列出添加到工程中的所有图层的框架，并可以按组进行组织。除非设置了基于可见比例 ，否则旁边带有复选框的图层显示的内容与地图画布范围重叠。为更改 Z-排序，可以选择图层并在图例中向上或向下拖动。Z-排序表示靠近图例上部列出的图层绘制在图例下部列出的图层之上。此外，可以在多个 QGIS 实例中拖动图层或图层组。

> **Note**
>
> Z-排序行为可以由图层顺序 面板覆写。

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

Table 8.1: 图层面板项的上下文菜单

选项	组	矢量图层	栅格图层	网孔图层	点云图层	3D Layer
缩放到图层/组	✓	✓	✓	✓	✓	✓
缩放到选中要素的范围		✓				
在鹰眼图中显示		✓	✓	✓	✓	✓
显示要素数量		✓				
显示标注		✓		✓		
复制图层/组	✓	✓	✓	✓	✓	✓
重命名图层/组	✓	✓	✓	✓	✓	✓
缩放到原始分辨率 (100%)			✓			
按当前范围拉伸			✓			
Create Raster Attribute Table			✓			
Open Raster Attribute Table			✓			
Load Raster Attribute Table from VAT.DBF			✓			
更新 *SQL* 图层...		✓				
Execute SQL...		✓				
编辑虚拟图层...		✓				
添加组	✓					
创建图层副本		✓	✓	✓	✓	✓
移除图层/组...	✓	✓	✓	✓	✓	✓
从组中移出		✓	✓	✓	✓	
移到顶层	✓	✓	✓	✓	✓	
移到底层	✓	✓	✓	✓	✓	
勾选及所有父项		✓	✓	✓	✓	
将所选成组		✓	✓	✓	✓	
打开属性表		✓				
切换编辑模式		✓		✓		
当前编辑 ►		✓		✓		
过滤...		✓	✓		✓	
更换数据源...		✓	✓	✓	✓	
修复数据源...		✓	✓	✓	✓	
在所选要素上执行操作 ► (编辑模式)		✓				
► 创建要素副本		✓				
► 创建要素副本并数字化		✓				
设置图层可见比例...		✓	✓	✓	✓	✓
缩放到可见比例		✓	✓	✓	✓	
图层 *CRS* ►		✓	✓	✓	✓	✓

continues on next page

Table 8.1 – continued from previous page

选项	组	矢量图层	栅格图层	网孔图层	点云图层	3D Layer
► 将图层的 *CRS* 设置为工程的 *CRS*		☑	☑	☑	☑	☑
► 设为.. (最近使用的 CRS)				☑	☑	☑
► 设置图层 *CRS*...		☑	☑	☑	☑	☑
设置组 *CRS*...	☑					
设置组的 *WMS* 数据...	☑					
☐ 互斥组	☑					
勾选及所有子项 *(Ctrl-*单击*)*	☑					
取消勾选及所有子项 *(Ctrl-*单击*)*	☑					
永久化		☑				
导出 ►	☑	☑	☑	☑	☑	☑
► 另存为...			☑			
► 要素另存为...		☑				
► 选中的要素另存为...		☑				
► 另存为图层定义文件...	☑	☑	☑	☑	☑	☑
► 另存为 *QGIS* 图层样式文件...		☑	☑	☑	☑	☑
样式 ►		☑	☑	☑	☑	☑
► 复制样式		☑	☑	☑	☑	☑
► 粘贴样式	☑	☑	☑	☑	☑	
► 增加...		☑	☑	☑	☑	☑
► 重命名当前样式...		☑	☑	☑	☑	☑
► 编辑符号...		☑				
► 复制符号		☑				
► 粘贴符号		☑				
添加图层注释...		☑	☑	☑	☑	☑
编辑图层注释...		☑	☑	☑	☑	
移除图层注释		☑	☑	☑	☑	
工程属性...		☑	☑	☑	☑	☑

对于 GRASS 矢量图层， 切换编辑 不可用。有关编辑 GRASS 矢量图层的信息，请参阅*Digitizing and editing a GRASS vector layer* 一节。

与组和图层交互

图例窗口中的图层可以组织成组。有如下不同的方式可以实现：

1. 点击 图标添加新组。键入组的名称，然后按 Enter 键。现在，单击现有图层并将其拖动到组。
2. 选择要成组的图层，然后点击 图标。选中的图层将自动嵌入到新组。
3. 在图例窗口右键单击所选的多个图层，然后选择 *将所选成组*。选中的图层将自动放置在新组中。

若要将图层移出组，请将其拖出，或右键单击并选择 *从组中移出*: 将图层从组中移出并放在其上方。组也可以嵌套在其他组中。如果某个图层放在嵌套组中，*从组中移出*会将该图层从所有嵌套组中移出。

若要将组或图层移到图层面板的顶部，请将其拖动到顶部，或选择 *移到顶层*。如果在嵌套组中的图层上使用此选项，该图层将移到其当前组的顶部。*移到底层*选项遵循相同的逻辑向下移动图层和组。

组的复选框将一键显示或隐藏组中选中的图层。按下 Ctrl 键后，单击复选框将打开或关闭组及其子组中的所有图层。

Ctrl-单击选中/未选中的图层将取消勾选/勾选该图层及其所有父图层。

启用 **互斥组**选项意味着一个组内同时只有一个图层可见。每当组中的某个图层设置为可见时，其他图层将切换为不可见。

按住 Ctrl 键并单击其他图层，可以同时选择多个图层或组。然后可以同时将所有选中的图层移到新组中。

您也可以使用 Ctrl 键选择多项后按 Ctrl+D 同时删除多个图层或组：所有选中的图层或组都将从图层列表中移除。

有关使用指示器图标的图层和组的详细信息

在某些情况下，图层面板中图层或组旁边的格式或图标会发生变化，以提供有关图层/组的更多信息。这些元素是：

- 表示图层处于编辑模式，并且可以修改数据
- 表示正在编辑的图层有一些未保存的更改
- 表示应用于图层的过滤 。将鼠标悬停在图标上可以查看过滤表达式，之后双击更新查询
- 表示工程中必需 的图层，因此不可移除
- 表示嵌入组或图层 及其原始工程文件的路径
- 若要识别其数据源在工程文件打开时不可用的图层 (请参阅处理损坏的文件路径)，请单击图标更新源路径或从图层上下文菜单选择 *修复数据源...* 条目。
- 提示该图层是临时图层 ，当您关闭此工程时，其内容将被丢弃。为了避免数据丢失并使图层永久化，请单击图标以 QGIS 支持的 GDAL 矢量格式储存图层。
- 表示在离线编辑模式 下使用的图层。
- 表示没有/未知 CRS 的图层
- 表示坐标存储在坐标参照系中，本身精度较低的图层 (需要启用相应的设置)
- 表示由画布动画控制的时态图层
- 表示具有关联注释 的图层

- 当地图画布的当前比例超出图层的可见比例范围 (在其 渲染属性中设置) 时，该名称将变灰。选择上下文菜单 缩放到可见比例选项，将地图缩放到图层最近的可见比例范围。

通过分组控制图层渲染

组是工程中树目录结构化图层的一种方式，但它们也会影响其组成图层的渲染方式，即在地图渲染过程中作为单个展平对象。

只要选择了组，就可以在 图层样式面板使用此类渲染的选项。在 符号化 选项卡中，勾选复选框 将图层作为组进行渲染，启用组选项控制子图层的外观，而不是单个图层：

- 不透明度: 子图层中被其他子图层遮挡的要素将保持遮挡状态，并且不透明度仅适用于” 整个组”。

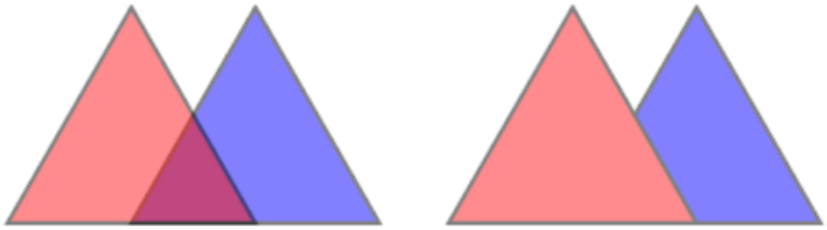


Fig. 8.2: 设置图层与组不透明度的对比
左侧图像显示以 50% 的不透明度渲染的两个图层 (底层要素可见，但顶层 50% 的红色要素半遮盖)。第二幅图像显示了在组中设置不透明度的结果 (蓝色底层子图层的一部分被顶层的红色图层完全遮盖，然后以 50% 的不透明度渲染的结果)。

- 混合模式: 正如不透明度一样，为整个组设置混合模式 (例如正片叠底、叠加等) 首先会导致子图层的拼合要素，上层会遮挡下层的要素。然后通过混合扁平组和位于该组下方的图层获得渲染。
  - 当子图层指定了混合模式时，将在拼合之前应用，但范围仅限于影响该组的其他子图层，而不影响位于整个组下方的其他图层。
  - 在 符号化选项卡，组中的子图层可以使用更多的混合模式 选项，这些选项在渲染过程中对其他子图层执行” 裁剪” 样式的操作。例如，您可以通过第二个” 掩膜” 图层中的内容裁剪第一个图层内容的渲染。
- 图层效果: 仅将效果 应用于子图层的拼合渲染，例如，应用于组的阴影效果对于被遮挡的子图层是不可见的。

当组设置为 将图层作为组进行渲染时，只有该组显示在 图层顺序面板列表中。组子项在此顺序列表不可见，因为它们的顺序是由组图层的位置而决定的。

编辑图层样式

在 图层面板，可以使用快捷方式快速轻松地更改图层渲染。

右键单击图层，然后在列表中选择 样式 ► 以便：

- 查看图层当前可用样式 。如果为图层定义了多个样式，则可以从一个切换到另一个样式，并且将在地图画布上自动更新图层渲染。
- 复制当前样式的一部分或全部，并在适用时从其他图层粘贴复制样式
- 重命名当前... 样式
- 添加新样式 (实际是当前样式的副本)
- 或 移除当前样式 (仅当有多个样式可用时)。

> **Tip**
>
> **快速分享图层样式**
>
> 从上下文菜单，复制图层的样式并将其粘贴到组或图层的选择集：该样式将应用于与原始图层具有相同类型 (矢量、栅格、网孔、点云等) 的所有图层，对于矢量图层而言，则是具有相同几何图形的类型 (点、线或多边形)。

当使用基于要素分类的符号时 (例如 分类或基于规则 的矢量图层, 或点云的分类), 右键单击 图层面板中的类条目可以编辑类 (及其要素) 的可见性，并避免逐个将其 (取消) 选中：

- 切换项
- 显示所有项
- 隐藏所有项

对于矢量图层，类叶条目的上下文菜单还可以访问：

- 选择要素: 在图层中选择与该类匹配的所有要素
- 在属性表中显示: 打开仅过滤与该类匹配的要素的属性表
- 使用 **色轮**更新符号颜色 。为了方便起见，最近使用的颜色也可以在色轮的下面找到。
- 编辑符号...: 打开符号选择器 对话框更改要素符号 (符号、大小、颜色等)。
- 复制符号
- 粘贴符号

> **Tip**
>
> 双击类叶条目也可以打开 符号选择器对话框。

8.2.2 图层样式面板

图层样式面板 (也可以使用 Ctrl+3 激活) 是 图层属性对话框某些功能的快捷方式。它提供了快速而简单的方式定义图层的渲染和行为，并可视化其效果，而无需打开图层属性对话框。

除了避免阻塞 (或" 模式") 图层属性对话框外，图层样式面板还避免了对话框使屏幕杂乱，并包含多数样式功能 (颜色选择器、效果属性、规则编辑、标注替换等)。例如，单击图层样式面板内的颜色按钮会使颜色选择器对话框在图层样式面板内打开，而不是作为单独的对话框启动。

从图层面板中当前图层的下拉列表中，选择一项并：

- 根据活动项，设置：
  - 组的 符号化 (请参阅通过分组控制图层渲染)
  - 栅格图层的 符号化, 透明度, 和 直方图属性。这些等同于*Raster Properties Dialog* 中的选项。
  - 矢量图层的 符号化, 标注, 掩膜和 *3D* 视图属性。这些与矢量属性对话框 中的选项相同，并且可以通过第三方插件引入的自定义属性进行扩展。
  - *Symbology*, *Labels* and *3D View* properties for mesh layer. These options are the same as in the *Mesh Dataset Properties*.

  - *Symbology*, *3D View* and *Elevation* properties for point cloud or 3D Tiles layer. These options are the same as in the *Point Clouds Properties* or *3D Tiles Properties*.
  - *Symbology* and *Labels* properties for vector tiles layer. These options are the same as in the 矢量瓦片数据集属性.
  - *Symbology* properties for annotations layer. These options are the same as in the *feature annotation properties*.
- Enable and configure *global map shading* properties
- 在 样式管理器中管理关联的样式 (更多信息请参阅管理自定义样式)。
- 查看当前工程应用于图层样式更改的 历史记录: 您可以通过在列表中选择某条记录并单击 应用 取消或恢复到任何之前状态。

此面板的另一个强大功能是 实时更新复选框。对其勾选可以立即在地图画布渲染更改，您不再需要单击 应用按钮。

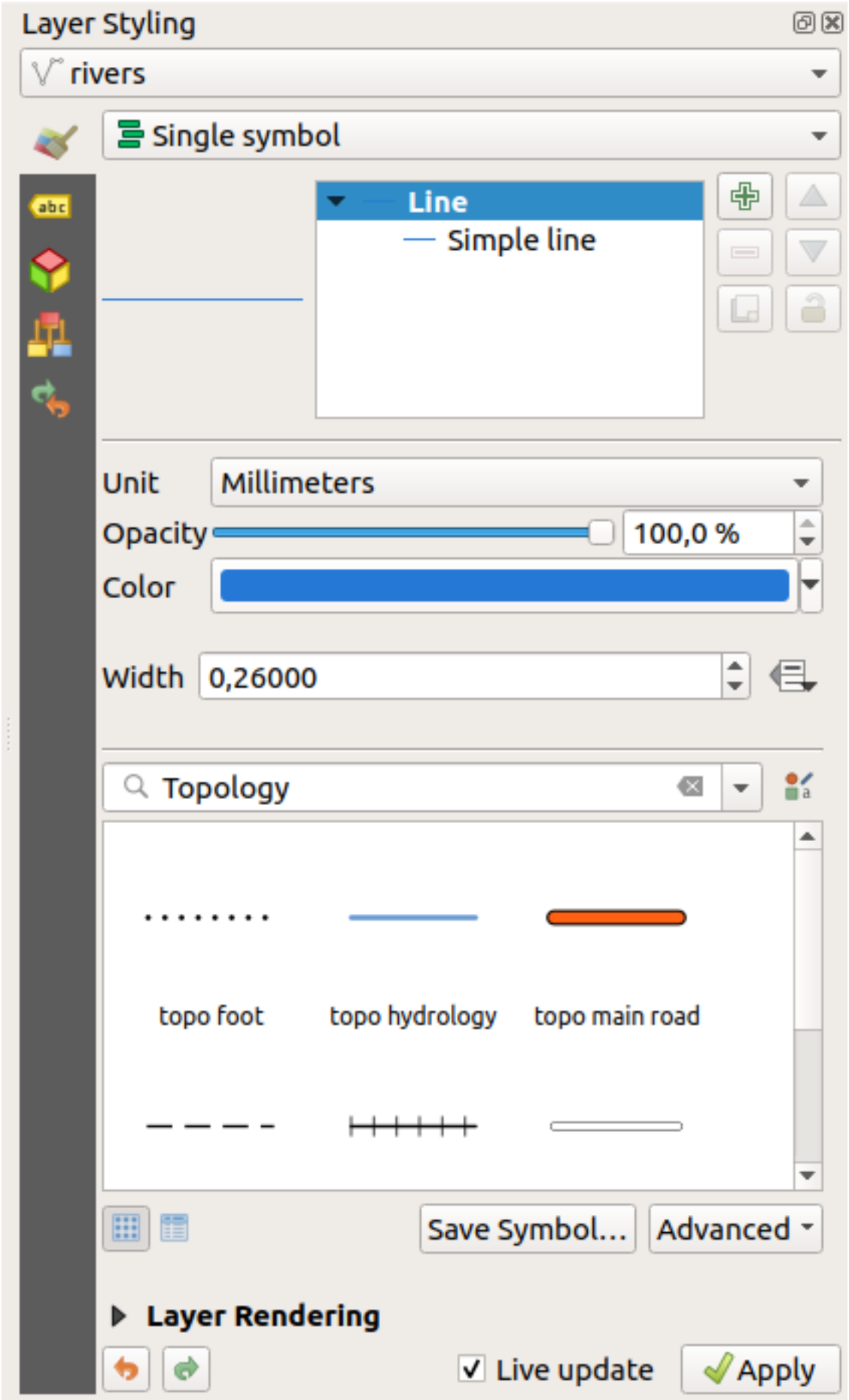


Fig. 8.3: 从图层样式面板定义图层的符号

8.2.3 图层顺序面板

默认情况下，QGIS 地图画布上显示的图层是按 图层面板的顺序绘制的：图层在面板中的位置越靠前，它在地图视图中的位置就越高 (因此越明显)。

您可以为图层定义独立于图层面板排序的图层绘制顺序，并在 视图 ► 面板 ► 菜单或按 Ctrl+9 启用 图层顺序面板。在图层列表下方勾选 控制渲染顺序，然后根据需要重新组织面板中的图层。此排序将成为应用于地图画布的渲染顺序。例如，在 Fig. 8.4 中，您可以看到 airports 要素显示在 alaska 多边形上面，尽管这些图层分别位于图层面板的不同位置。

取消勾选 控制渲染顺序将恢复为默认行为。

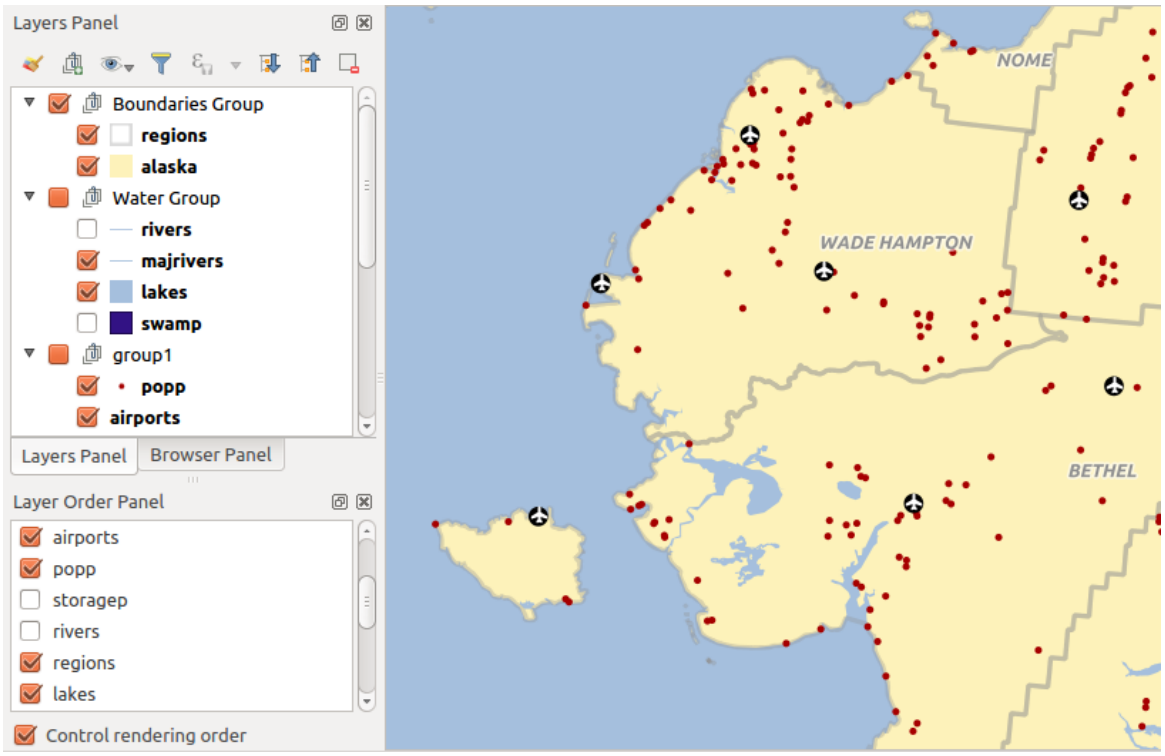


Fig. 8.4: 定义独立于图例的图层顺序

8.2.4 概览图面板

概览图面板 (Ctrl+8) 显示某些图层的完整范围视图的地图。使用 图层菜单或图层上下文菜单中的 在概览图中显示选项，将在面板中填充概览地图。在视图中，红色矩形显示当前地图画布范围，帮助您快速确定当前正在查看的整个地图的具体区域。如果单击并拖动概览图框架中的红色矩形，将相应更新主地图视图范围。

请注意，即使已为地图概览图中使用的图层设置了标注，标注也不会在地图概览图中渲染。

8.2.5 日志消息面板

加载或处理某些操作时，可以使用 日志消息面板跟踪和关注出现在不同选项卡中的消息。可以使用底部状态栏最右侧的图标激活。

8.2.6 撤消/重做面板

对于处于编辑状态的每个图层，撤销/重做 (Ctrl+5) 面板显示已执行的操作列表，允许您通过选择面板内列出的操作快速撤消或重做。更多信息，请参阅撤消和重做编辑 。

8.2.7 统计摘要面板

统计面板 (Ctrl+6) 提供有关矢量图层的摘要信息。此面板允许您选择：

- 用于计算统计数据的矢量图层：可以从顶部的下拉菜单选择相关图层，也可以使用统计下拉列表底部的 遵循所选图层复选框与 图层面板中的活动图层进行同步
- 使用字段或 表达式：对于每个图层，将记忆最后的条目，并在重新选择图层时自动计算
- 使用对话框右下角的下拉按钮返回统计面板。根据字段 (或表达式的值) 的类型，可用的统计字段包括：

Table 8.2: 可用统计字段类型

统计	字符串	整数	浮点数	日期
计数	✓	✓	✓	✓
计数不同值	✓			✓
计数缺失值	✓	✓	✓	✓
Sum		✓	✓	
均值		✓	✓	✓
标准差		✓	✓	
样品标准差		✓	✓	
最小值	✓	✓	✓	✓
最大值	✓	✓	✓	✓
范围		✓	✓	✓
寡数	✓	✓	✓	
众数	✓	✓	✓	
种类		✓	✓	
第一个四分位数		✓	✓	
第三个四分位数		✓	✓	
四分位距		✓	✓	
最小长度	✓			
最大长度	✓			
平均长度	✓			

统计摘要可以是：

- 返回整个图层或 仅选中的要素

- 复制到剪贴板并粘贴为其他应用程序的表
- 当基础数据源出现变化 (例如，新建或移除的要素/字段、属性修改) 时，使用 按钮重新计算

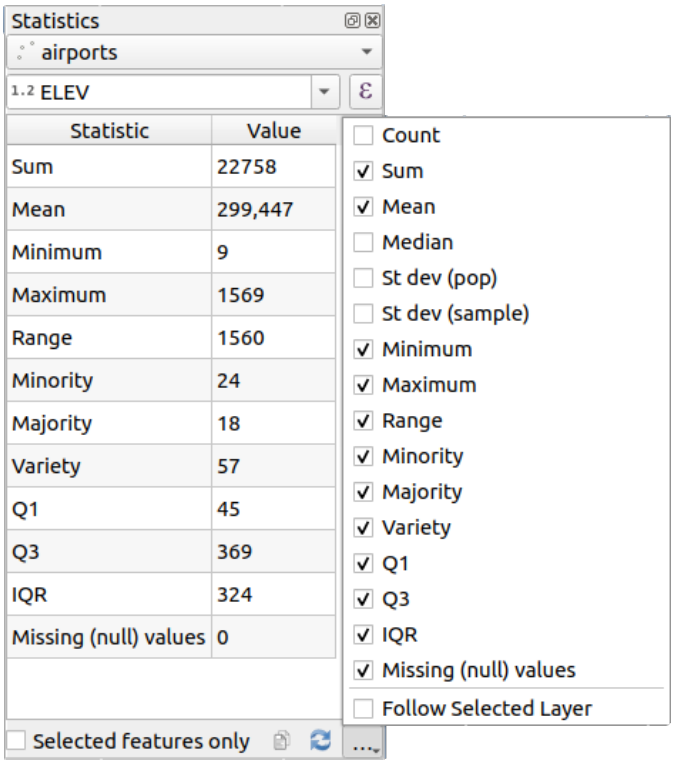


Fig. 8.5: 显示统计的字段

8.2.8 调试/开发工具面板

调试/开发工具面板 (F12) 为 QGIS 的处理和调试操作提供了一致的位置。可用工具按以下选项卡排序进行组织：

- *API Documentation*: embeds developer oriented documentation
- *Network Logger*: records and displays network requests and interactions
- *Query Logger*: logs SQL commands issued by data providers and duration of execution
- *Profiler*: provides load times for every actions in order to detect causes of slow down

Note

插件作者可以使用自定义选项卡扩展面板，用于调试和开发自己的插件。这是使用 `registerDevToolWidgetFactory` 方法完成的。

API Documentation

The *API Documentation* tab can display either the PyQGIS API documentation or the Qt API documentation. These are embedded in the widget, allowing you to browse and search the docs without leaving QGIS. It is also the place you can display specific help page for objects or classes in your code sample; from the Python interactive console or text editor dialog, select or hover a text and press `F1` or right-click and select Contextual help to get its API help. If you wish to read the active page in the system web browser, then press Open in web browser.

Read *contextual help* for details on the widget configuration.

网络日志工具

网络日志工具选项卡可帮助您记录和显示网络请求，以及大量有用的详细信息，如请求和回复状态、标头、错误、SSL 配置错误、超时、缓存状态等。

在其顶部工具栏，您可以：

- 记录日志: 将启动或停止日志记录。
- 清除日志: 将清除日志历史记录。
- 保存日志...: 将首先显示重大警告，说明日志是敏感的，应该被视为机密，然后允许您保存日志。
- 点击 设置下拉菜单选择 显示成功的请求, 显示超时和 显示来自缓存的响应。
- 禁用缓存: 将禁用缓存，以便必须执行每个请求。
- 过滤请求: 基于 URL 字符串子集或请求状态

右键单击请求，您可以执行：

- 打开 *URL*: 将在默认浏览器打开 URL。
- 复制 *URL*
- 复制为 *cURL*: 以便在终端使用。
- 复制为 *JSON*: 将树值作为 json 字符串复制到剪贴板，以便粘贴到 Bug 报告或用于远程协助。

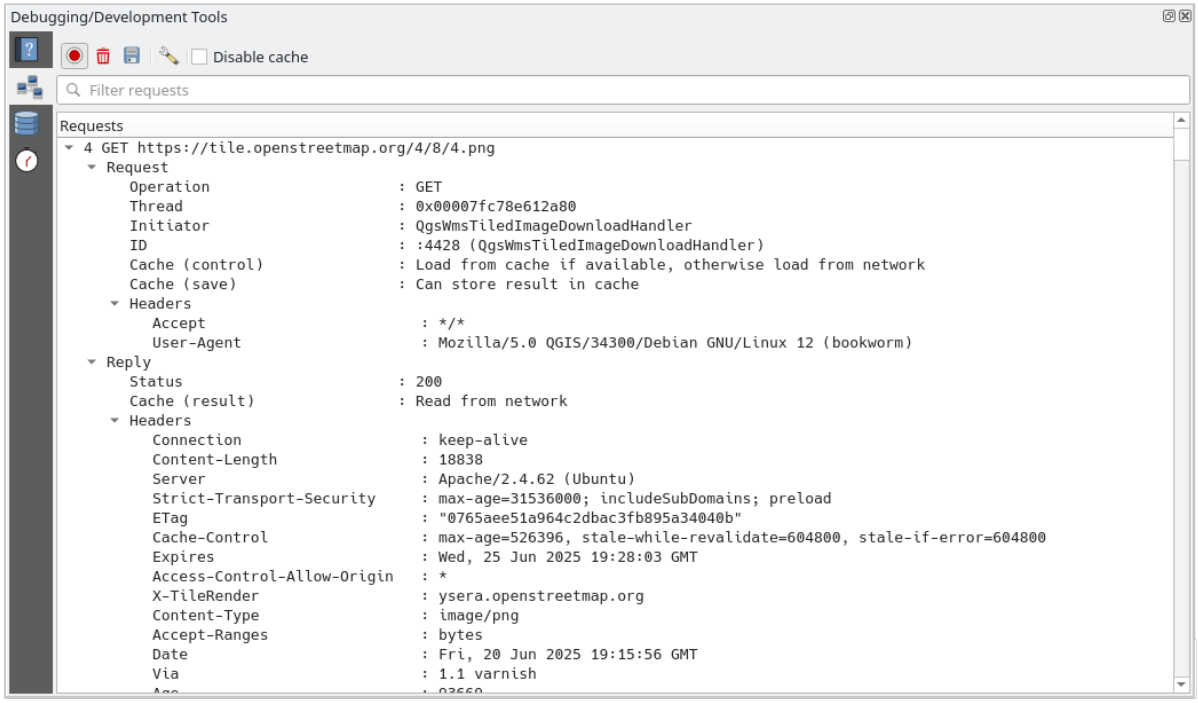


Fig. 8.6: 网络日志工具输出 GET 请求

查询记录器

查询记录器用于记录数据提供程序发送的 SQL 命令和 API 到后端数据库的连接，以及 QGIS 测量的执行时间 (即在发送命令的客户端中)。当在 QGIS 算法或插件的调试或开发过程中研究特定图层的性能时，这非常有用。

在其顶部工具栏，您可以:

- 记录日志: 将启动或停止日志记录。
- 清除日志: 将清除日志历史记录。
- 保存日志...: 将首先显示重大警告，说明日志是敏感的，应该被视为机密，然后允许您保存日志。
- 过滤查询基于查询字符串子集或详细信息，例如提供程序类型、开始时间、启动程序等。

右键单击报告查询，可以:

- 复制 *SQL*: QGIS 在数据库调用的命令
- 复制为 *JSON*: 将树值作为 json 字符串复制到剪贴板，以便粘贴到 Bug 报告或用于远程协助。

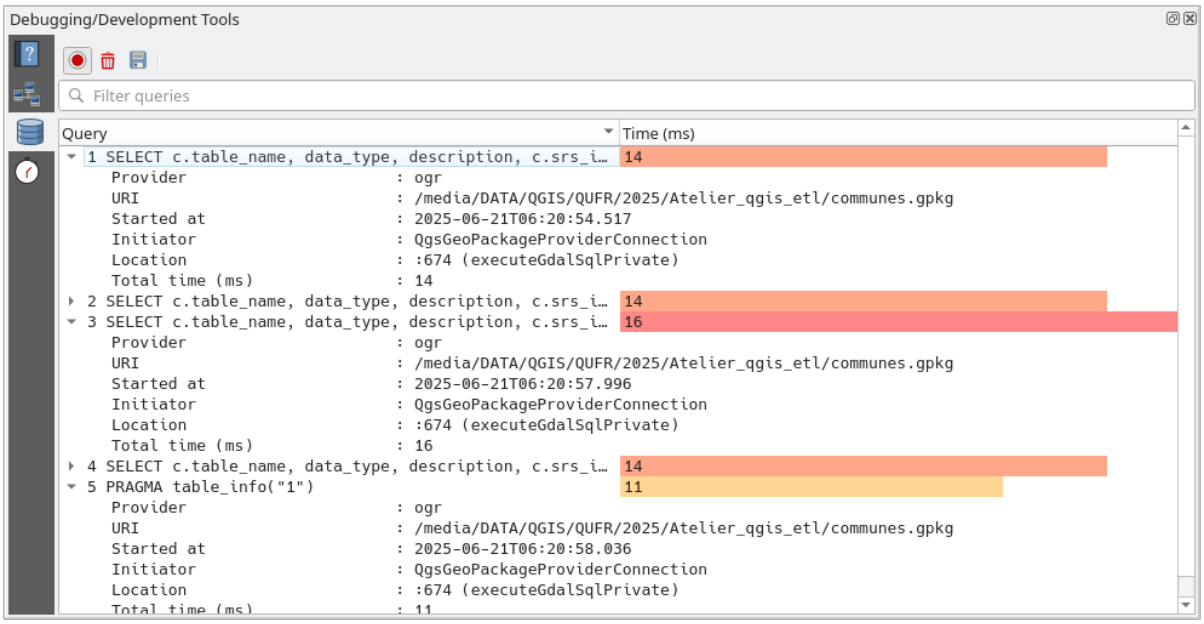


Fig. 8.7: 输出查询记录

分析工具

分析工具选项卡允许获取用户请求的动作中涉及的每个操作的加载时间。根据上下文，这些操作可以是设置读取、菜单、地图画布或 3D 视图创建、解析地图图层引用、书签或布局加载等。这有助于识别出加载时间缓慢的原因。

Default supported actions can be selected from the *Category* drop-down menu:

- *Startup* helps you diagnose (and fix) occasional long startup times of QGIS.
- *Project Load* allows you to get a breakdown of the various stages of project load, in order to identify the causes of slow project load times.
- *Map Render* tool allows you to identify pain points in your map rendering and track down exactly which layers are causing long map redraws.

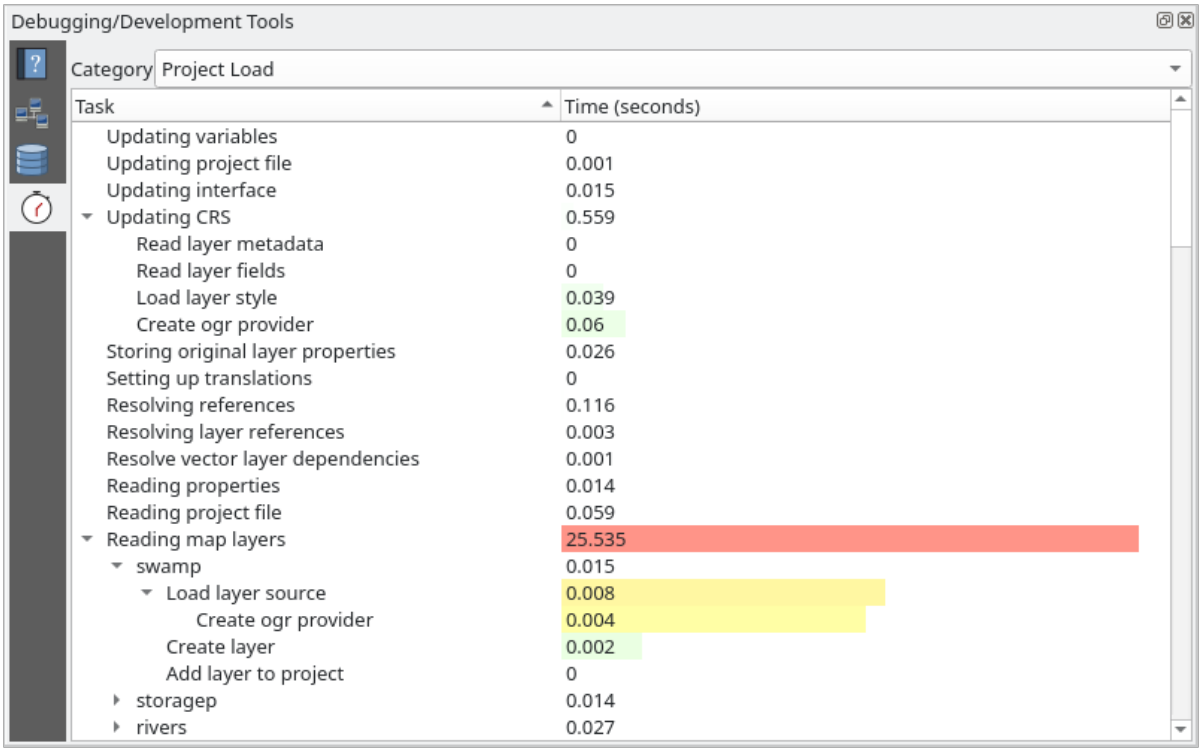


Fig. 8.8: QGIS 启动的分析工具

8.3 从外部工程嵌入图层

有时，希望将某些图层保存在不同的工程中，并具有相同的样式。您可以为这些图层创建默认样式，也可以从其他工程中嵌入，以节省时间和精力。

与样式设置相比，从现有工程嵌入图层和组具有一定优势：

- 可以添加所有类型的图层 (矢量或栅格、本地或在线...)
- 获取组和图层，可以在不同的工程中保持” 背景” 图层相同的树结构
- 虽然嵌入图层是可编辑的，但您无法更改其属性，例如符号、标注、表单、默认值和动作，从而确保工程之间的一致性
- 修改原始工程中的项，并将更改传导至所有其他工程

如果要将其他工程文件的内容嵌入到工程中，请选择 *图层* ► *嵌入图层和组*:

1. 单击 ... 按钮查找工程：您可以查看工程的内容 (请参阅 Fig. 8.9)
2. 按住 `Ctrl` (或 `Cmd`) 并单击要检索的图层和组
3. 单击 *确定*

所选图层和组将嵌入到 *图层*面板中，并在地图画布上显示。在其名称旁边添加 图标进行识别，将鼠标悬停其上会显示原始工程文件路径的提示信息。

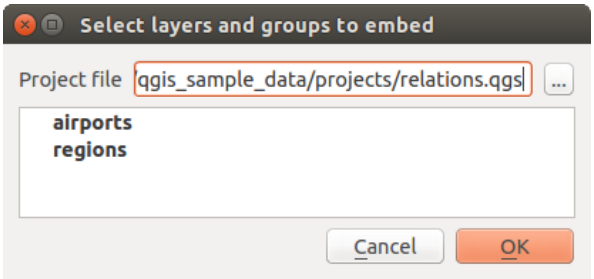


Fig. 8.9: 选择要嵌入的图层和组

与其他图层一样，可以右键单击图层并单击 移除 项，从工程中移除嵌入的图层。

Tip

更改嵌入图层的渲染

除非在原始工程文件中进行修改，否则无法更改嵌入图层的渲染。但是，可以右键单击图层并选择复制项，这将创建一个功能齐全且不依赖于原始工程的图层。然后可以安全地移除链接的图层。

8.4 与要素交互

8.4.1 选择要素

QGIS 提供了多种工具选择地图画布上的要素。选择工具位于 编辑 ► 选择菜单或 选择工具栏中。

Note

选择工具可用于当前活动图层。

在地图画布上手动选择

要用鼠标选择一个或多个要素，可以使用以下某个工具：

- 框选或单击选择要素
- 按多边形选择要素
- 自由手绘选择要素
- 按半径选择要素

Note

除了 按多边形选择要素 外，这些手动选择工具还允许您通过单击在地图画布上选择要素。

Note

按多边形选择要素 工具可以使用现有多边形要素 (来自任何图层) 选择活动图层中的重叠要素。右键单击多边形，然后从上下文菜单进行选择，该菜单显示包含单击点的所有多边形的列表。将选取活动图层中的所有重叠要素。

Tip

使用 *编辑* ► *选择* ► *重新选择要素*工具可以重做最近选择。当您煞费苦心地做出选择，却意外单击其他地方清除现有选择时非常有用。

While using the *Select Feature(s)* tool, holding `Shift` or `Ctrl` (`Cmd` on macOS) toggles whether a feature is selected (i.e., either adds to the current selection or removes from it).

对于其他工具，按住以下键可以执行不同的行为：

- `Shift`: 将要素添加到当前选择集
- `Ctrl`/`Cmd`: subtract features from the current selection
- `Ctrl+Shift`/`Cmd+Shift`: intersect with the current selection, i.e., only keep overlapping features from the current selection
- `Alt`: select features that are totally within the selection shape. Combined with `Shift` or `Ctrl`/`Cmd` keys, you can add or subtract features to/from the current selection.

自动选择

其他选择工具 (其中大多数可从*属性表* 中获得) 根据要素的属性或其选择状态执行选择 (请注意，属性表和地图画布显示相同的信息，因此如果在属性表中选择一个要素，该要素也将在地图画布上被选中)：

- Select Features By Expression... or press `Ctrl+F3` to select features using the expression dialog
- 按值选择要素... 或按 `F3`
- 在所有图层中取消对要素的选择 或按 `Ctrl+Alt+A` 取消选择所有图层中的所有选中的要素
- 从当前活动图层中取消选择要素 或按 `Ctrl+Shift+A`
- 选择全部要素 或按 `Ctrl+A` 选择当前图层中的所有要素
- 反选要素 反选当前图层中的选择
- 按位置选择 根据要素与其他要素的空间关系选择要素 (在同一或另一图层中 - 请参阅*Select by location*)
- Select within distance to select features wherever they are within the specified maximum distance from referenced features - see *Select within distance*)

例如，如果要从 QGIS 样本数据的 `regions.shp` 中查找属于行政区的区域，则可以：

1. 使用 使用表达式选择要素 图标选择要素
2. 展开 *字段和值*组
3. 双击要查询的字段 (”TYPE_2”)
4. 单击面板中右侧显示的 *所有唯一值*
5. 在列表中，双击’Borough’。在 *表达式*编辑器字段中，编写以下查询：

```
"TYPE_2"  =  'Borough'
```

6. 单击 选择要素

在表达式构建器对话框中，还可以使用 函数列表 ► 最近使用 (选择集) 进行您以前使用过的选择。该对话框会记忆最近使用的 20 个表达式。更多信息和示例，请参阅表达式 。

Tip

将选择集保存到新文件

用户可以将所需格式使用 编辑 ► 要素复制和 编辑 ► 粘贴要素为将所选要素保存到 **新建临时图层**或 **新建矢量图层**。

按值选择要素

该选择工具打开图层的要素表单，允许用户选择要为每个字段查找的值，搜索是否应区分大小写，以及使用的操作。此工具还具有自动补全功能，可以使用现有值自动填充搜索框。

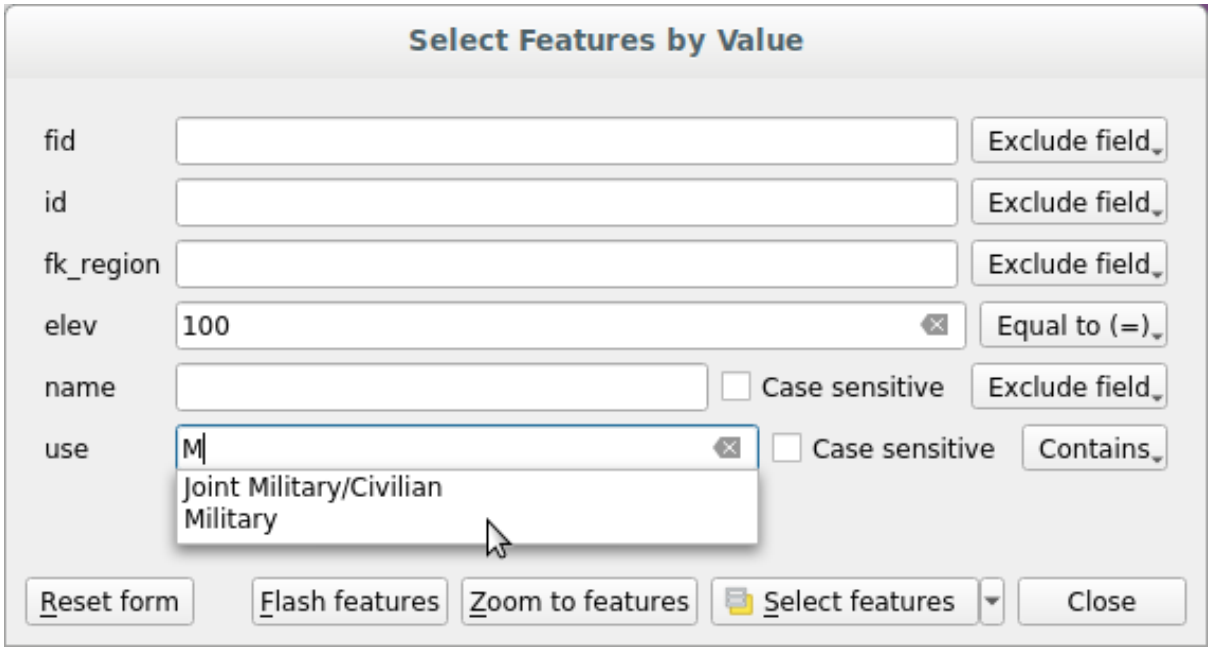


Fig. 8.10: 使用表单对话框过滤/选择要素

在每个字段旁边，都有一个下拉列表，其中包含用于控制搜索行为的选项：

Table 8.3: 每种数据类型的查询运算符

字段搜索选项	字符串	Numeric	日期
从搜索中 排除字段	☑	☑	☑
等于 (=)	☑	☑	☑
不等于 (≠)	☑	☑	☑
大于 (>)		☑	☑
小于 (<)		☑	☑
大于或等于 (≥)		☑	☑
小于或等于 (≤)		☑	☑
介于 (包含)		☑	☑
非介于 (包含)		☑	☑
包含	☑		
不包含	☑		
缺失 (空)	☑	☑	☑
未缺失 (非空)	☑	☑	☑
始于	☑		
终于	☑		

对于字符串比较，也可以使用 ☑ 区分大小写选项。

设置完所有搜索选项后，单击 选择要素以选择匹配的要素。下拉选项包括：

- 选择要素
- 添加到当前选择集
- 从当前选择集移除
- 过滤当前选择集

还可以使用 重置表单按钮清除所有搜索选项。

设置条件后，您还可以：

- 无需预选即可在地图画布上 缩放到要素范围
- 闪烁要素: 突出显示匹配的要素。这是一种无需选择或使用识别工具即可识别要素的便捷方法。请注意，闪烁不会改变地图画布的范围，并且仅当要素在当前地图画布的范围内时才可见。

8.4.2 识别要素

The Identify Features tool allows you to interact with the map canvas and get information on features or pixels in a pop-up window. It can be used to query most of the layer types supported by QGIS (vector, raster, mesh, point cloud, wms, wfs, ...). To identify an element, use either:

- 视图 ► 识别要素
- `Ctrl+Shift+I` (或 `Cmd+Shift+I`)
- Identify Features button on the *Attributes* toolbar

Then click on a feature or pixel of the active layer. The identified item gets highlighted in the map canvas while the *Identify Results* dialog opens with detailed information on it. The dialog also shows a set of buttons for advanced configuration.

识别结果对话框

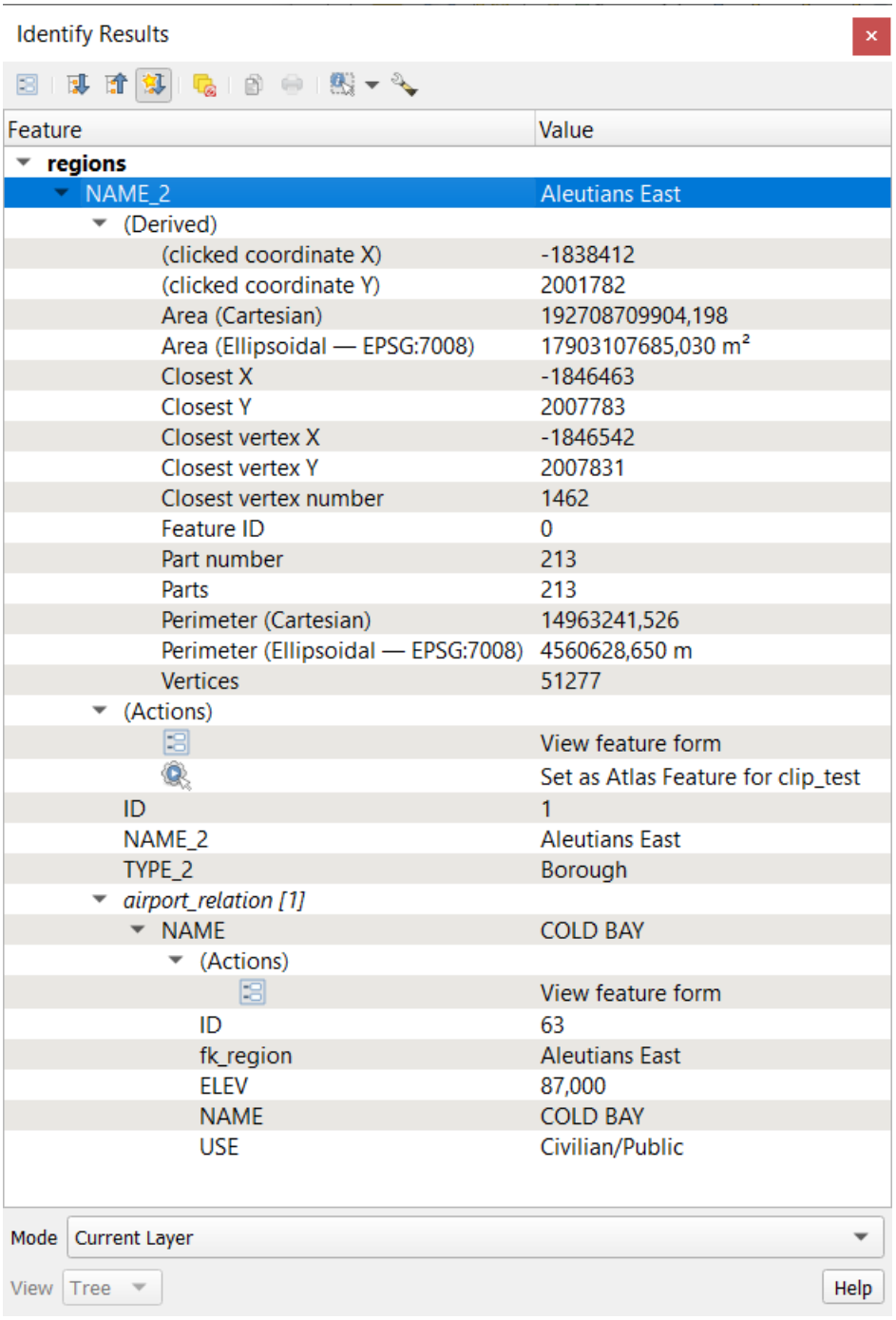


Fig. 8.11: 识别结果对话框

From bottom to top:

- The *View* controls the general aspect of the dialog and the formatting of the results; it can be set as:
  - **Tree**: this is the default view, and returns the results in a tree-structure where the first item is the name of the layer and its children are its identified element(s).
  - **Table**: available only for raster-based layers, it allows to display the results as a table whose columns are `Layer`, `FID`, `Attribute` and `Value`
  - or **Graph**: available only for raster-based layers

- The *Mode* helps you select the layers from which results could be returned. These layers should be set visible, displaying data in the map canvas, and set *identifiable* from the *Project properties* ► *Data Sources* ► *Layers capabilities*. Available modes are:
  - **Current layer**: only the layer(s) selected in the *Layers* panel return results. If a group is selected, then results are picked from its leaf layers.
  - **Top down, stop at first**: results are from the layer of the top most feature or pixel under the mouse.
  - **Top down**: results are from the layers with feature or pixel under the mouse.
  - **Layer selection**: opens a contextual menu where the user selects the layer to identify features from. If only a single feature is under the mouse, then the results are automatically displayed.
- In the upper part, the information widget: when you identify a data in the map canvas, this is the place where the *Identify Results* dialog will list details about the clicked (or hovered over, depending on the tool in use) items. Their formatting relies on the *selected view*.

  The information displayed by the identify tool will depend on the type of layer you have selected, whether it is:
  - a *vector layer* (including vector tiles or point cloud data),
  - a *raster layer*,
  - a *mesh layer*.
- When in a tree view, the upper part also displays a handful of tools above the results:
  - 当前要素的 打开表单
  - 展开树
  - 折叠树
  - 默认展开新结果 定义下一个已识别要素的信息是折叠还是展开
  - 清除结果
  - Copy the identified feature to clipboard, suitable for pasting in a spreadsheet.
  - Print selected HTML response: a text-based formatting of the results to print on paper or save as a `.PDF` file
  - the *interactive identifying tools*: a drop-down menu with tools for selecting on the map canvas features or pixels to identify
  - Under Identify Settings, you can activate whether to:
    - *Auto open form for single feature results*: If checked, each time a single feature is identified, a form opens showing its attributes. This is a handy way to quickly edit a feature's attributes.
    - *Hide derived attributes from results* to only show fields actually defined in the layer
    - *Hide NULL values from results*
    - *Show Relations* to show relations in the vector layer *Identify results tree*
  - Help to access the current documentation

使用识别要素工具

In its default display (*View: Tree*), the *Identify Results* panel offers several tools to interact with the layers to query. A smart combination of these tools with the *target layers selector* may greatly improve identification operations:

- Identify Feature(s) by single click or click-and-drag
  - single click or click-and-drag: overlaying features in the target layers are returned
  - right-click: overlaying features from target layers are listed in the contextual menu, grouped by layers. You can then choose to:
    - display the result for a specific feature,
    - display the result for all the features of a specific layer,
    - for vector layers, it is also possible to open its attribute table filtered to the returned features
    - or show all of the returned features.
- Identify Features on Mouse over: move over the map canvas and hovered items in the target layers get highlighted and returned in the results panel.
- Identify Features by Polygon: returns items overlapping a drawn or selected polygon.
  - Draw a polygon (left click to add point, right click to close the polygon) and all the overlaying features from target layers are highlighted and returned in the results panel.
  - Right-click and you get the list of all visible polygon features in the project under the click. Pick an entry and QGIS will return all the features from the target layers that overlap the selected polygon.
- Identify Features by Freehand: returns items overlapping a polygon drawn by freehand. Draw a polygon (left-click to start, move the pointer to shape the area and right-click to close the polygon). All the overlaying features from target layers are highlighted and returned in the results panel.
- Identify Features by Radius returns items overlapping a drawn circle. Draw a circle (left-click to indicate the center point, move the pointer to shape the area or enter the radius in the pop-up text box and left-click or press `Enter` to validate the circle). All the overlaying features from target layers are highlighted and returned in the results panel.

> **Tip**
>
> **使用识别要素工具过滤要查询的图层**
>
> Under *Project* ► *Properties...* ► *Data Sources* ► *Layer Capabilities*, uncheck the *Identifiable* column next to a layer to avoid it being queried when using the Identify Features tool. This is a handy way to return features from only layers that are of interest to you.

Results contextual menu

Other functions can be found in the context menu of the identified item. For example, depending on the layer type, you can:

- interact with the individual identified item:
  - *View feature form* or *Edit feature form*, for vector layers
  - *Zoom to feature*: Zooms the map canvas to the extent of the identified feature or pixel
  - *Identify feature*: Restarts and recenters the search results on a *referenced or referencing vector feature* of the identified feature. Only available if *Show Relations* is checked in Identify Settings.

  - *Copy feature*: Copies the vector geometry and attributes, or the pixel geometry
  - *Toggle feature selection*: Adds or removes identified feature to/from the selection
  - *Copy attribute value*: Copies the value in the *Value* column of the tree item you click on
  - *Copy feature attributes*: Copies the attribute names and values of the identified item
  - *Select features by attribute value*: Selects all vector features in the layer that match the selected attribute value
- control the rendering of the layers, the results panel or the map canvas:
  - *Clear result*: Removes results in the window
  - *Clear highlights*: Removes items highlighted on the map
  - *Highlight all*: Highlights in the map canvas all the items identified in the results dialog, regardless of their layer
  - *Highlight layer*: Highlights in the map canvas all the items identified in the layer of the selected entry
  - *Activate layer*: Selects the layer of the identified item in the *Layers* panel
  - *Layer properties*: Opens the layer properties window of the identified item
  - *Expand all* or *Collapse all* the results tree

8.5 保存和共享图层属性

8.5.1 管理自定义样式

将矢量图层添加到地图画布时，QGIS 默认使用随机符号/颜色渲染其要素。但是，您可以在 工程 ► 工程属性... ► 默认符号 设置默认样式，这将根据其几何图形类型应用于每个新添加的图层。

不过，多数情况，您更希望拥有可自定义且更复杂的样式，并可以自动或手动应用于图层 (只需较少的工作量)。请使用图层属性对话框底部的 样式菜单实现此目的。此菜单为您提供了创建、加载和管理样式的功能。

A style stores various information set in the layer or project properties dialog to render or interact with the layer. This includes:

Table 8.4: Components of stored style and their references

Category	Vector	Raster
Layer Configuration	*Layers capabilities* *Display Properties*	*Display Properties*
Symbology	*Symbology Properties*	*Symbology Properties* ([1])
3D Symbology	*3D View Properties*	N/A
Labels	*Labels Properties*	N/A
Fields	*Fields Properties* and *Constraints*	N/A
Attributes Form	*Attributes Form Properties*	N/A
Actions	*Actions Properties*	N/A
Map Tips	*Display Properties*	*Display Properties*
Diagrams	*Diagrams Properties*	N/A
Attribute Table Configuration	属性表操作	
Rendering	*Rendering Properties*	*Rendering Properties* ([1])
Custom Properties	信息属性	信息属性
Geometry Options	*Digitizing Properties*	N/A
Relations	*Setting relations between multiple layers*	N/A
Temporal Properties	*Temporal Properties*	*Temporal Properties*
Legend Settings	*Legend Properties*	N/A
Elevation Properties	*Elevation Properties*	*Elevation Properties*
Notes	图层注解	

默认情况下，应用于加载图层的样式命名为 默认。您为图层获得理想且合适的渲染后，可以通过单击 样式组合框对其保存并选择：

- **重命名当前样式**: 活动样式将被重命名，并使用当前选项进行更新
- **添加**: 使用当前选项创建新样式，默认将保存在 QGIS 工程文件中。请参阅以下内容，将样式保存在另一个文件或数据库
- **移除**: 如果为图层定义了多个样式，请删除不需要的样式。

在样式下拉列表的底部，可以看到为当前活动样式的图层设置的样式名称。

请注意，每次验证图层属性对话框时，活动样式都会根据所做的更改进行更新。

可以根据需要为图层创建任意数量的样式，但每次仅有一个样式处于活动状态。与地图主题 结合使用，将提供快速而强大的方式管理复杂的工程，而无需复制地图图例中的任何图层。

Note

鉴于无论何时对图层属性应用修改，更改都存储在活动样式中，因此请确保编辑正确的样式，避免错误变更地图主题 中使用的样式。

[1] For raster layers, symbology and rendering items are melted together, meaning that saving/loading one would also save/load items of/from the other.

> **Tip**
>
> **从图层上下文菜单管理样式**
>
> Right-click on the layer in the *Layers* panel to copy, paste, add or rename layer styles. For both vector and raster layers, you can choose which style categories to copy/paste. Choose *All Style Categories* or one of the available categories, depending on the layer type. Note that for raster layers, the *Symbology* and *Rendering* categories are always copied together.

8.5.2 在文件或数据库中存储样式

虽然从 样式组合框创建的样式默认保存在工程内部，并且可以在工程的图层之间复制和粘贴，但也可以将其保存在工程外部，以便加载到另一个工程中。

另存为文本文件

单击 样式 ► 保存样式, 可以将样式另存为:

- QGIS 图层样式文件 (`.qml`)
- SLD 文件 (`.sld`), 仅适用于矢量图层

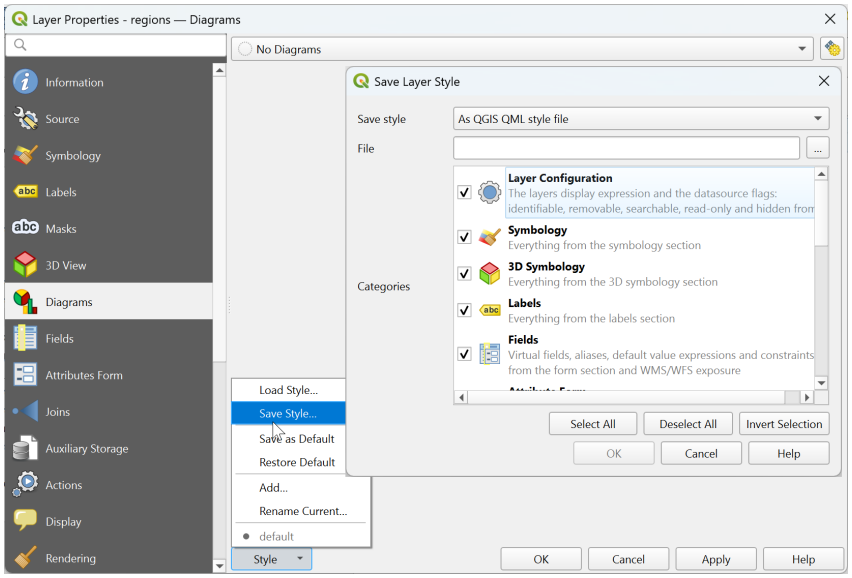


Fig. 8.12: 矢量图层样式组合框选项

在基于文件格式图层 (`.shp`, `.tab`...) 上使用, 另存为默认为该图层生成 `.qml` 文件 (具有相同名称)。SLD 可以将所有渲染器的类型 (单符号、分类、分级或基于规则) 导出，但在导入 SLD 时，会创建单符号或基于规则的渲染器。这意味着分类或分级样式将转换为基于规则的样式。如果要保留这些渲染器，必须使用 QML 格式。另一方面，有时将样式转换为基于规则的样式会非常方便。

保存在数据库

Vector layer styles can also be stored in a database if the layer datasource is a database provider. Supported formats are PostgreSQL, GeoPackage, SpatiaLite, MS SQL Server and Oracle. The layer style is saved inside a table (named `layer_styles`) in the database. Click on *Save Style...* ► *Save in database* then fill in the dialog to define a style name, add a description, a `.ui` file if applicable and to check if the style should be the default style.

可以为数据库中的单个表保存多个样式。但是，每个表只能有一个默认样式。默认样式可以保存到图层数据库或 `qgis.db` (当前*用户配置* 目录中的本地 SQLite 数据库)。

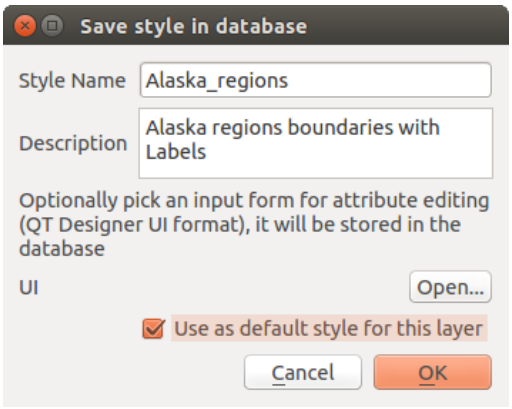


Fig. 8.13: 在数据库中保存样式对话框

> **Tip**
>
> **在数据库之间共享样式文件**
>
> 仅当图层来自此数据库时，才能将样式保存到数据库。您不能混淆数据库 (例如 Oracle 的图层和 MS SQL Server 的样式)。如果希望在数据库之间共享样式，请改用纯文本文件。

> **Note**
>
> 从 PostgreSQL 数据库备份中恢复 `layer_styles` 表时可能会遇到问题。请参照*QGIS layer_style 表和数据库备份* 试图解决。

加载样式

QGIS 加载图层时，如果该图层已经存在默认样式，QGIS 将使用该样式加载此图层。此外，还可以使用*样式* ► *重置为默认*查找并加载文件，替换图层的当前样式。

样式 ► *加载样式*可帮助您将保存的样式应用于图层。虽然文本文件样式 (`.sld` 或 `.qml`) 可以应用于任何图层，无论是何种格式，只有当图层来自同一数据库或样式存储在 QGIS 本地数据库中时，才能加载存储在数据库中的样式。

*数据库样式管理器*对话框显示与在数据库中找到的图层以及保存在其中的所有其他样式相关的样式列表以及名称和描述。

> **Tip**
>
> **在工程中快速共享图层样式**

您也可以在工程中共享图层样式，而无需导入文件或数据库样式：右键单击 图层面板中的图层，然后从 样式组合框复制图层的样式并将其粘贴到组或所选图层 - 样式应用于与原始图层具有相同类型 (矢量与栅格) 的所有图层，如果是矢量图层，则具有相同的几何图形类型 (点、线或多边形)。

8.5.3 图层定义文件

图层定义可以使用活动图层的上下文菜单的 导出 ► 另存为图层定义文件... 保存为 图层定义文件 (.qlr) 。图层定义文件 (.qlr) 包含对图层的数据源及其样式的引用。.qlr 文件显示在浏览器面板中，可用于将图层 (使用保存的样式) 添加到图层面板。您还可以将 .qlr 文件从系统文件管理器拖放到地图画布。

8.6 编写您的数据

除了显示和符号化图层中的数据外，QGIS 还允许您填写：

- **metadata**: information to help people find and understand the dataset, how they can access and use it... these are properties of the datasource and can live out of the QGIS project. Metadata can also be defined for the project file itself.
- **注解**: 关于当前工程中图层的说明和注释

8.6.1 元数据

In the project or layer properties dialog, the *Metadata* tab provides you with options to create and edit a metadata report on the element. Information to fill concern:

- 数据 识别: 数据集的基本属性 (父标识符、标识符、标题、摘要、语言等);
- 数据所属的 类别。除了 **ISO** 类别，您还可以添加自定义类别;
- 关键字按照基于标准的词汇检索数据和相关概念;
- 访问数据集 (许可证、版权、限制和约束);
- 数据集的 范围，空间范围 (CRS、地图范围、海拔) 或时态;
- 数据集所有者的 联系人;
- 附属资源和相关信息的 链接;
- 数据集的 历史记录。

验证选项卡中提供了填写信息的摘要，可帮助您识别与表单相关的潜在问题。之后，可以修复或忽略这些问题。

默认元数据保存在工程文件中，元数据下拉列表提供了从 .qmd 文件加载/保存元数据以及在" 默认" 位置加载/保存元数据的选项。

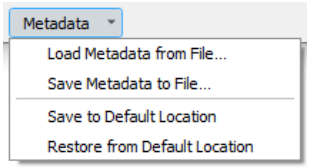


Fig. 8.14: 元数据加载/保存选项

The "Default" location used by *Save to Default Location* and *Restore from Default Location* changes depending on the underlying data source and on its configuration:

- 对于 PostgreSQL 数据源，如果勾选配置选项 *允许在数据库中保存/加载 QGIS 图层元数据*，元数据将存储在数据库的专用表中。
- For GeoPackage data sources *Save to Default Location* always saves the metadata in the internal metadata tables of the GeoPackage.

  当元数据保存到 PostgreSQL 或地理包的内部表中时，它们就可以在浏览器和图层元数据搜索面板中进行搜索和过滤。
- For all other file based data sources *Save to Default Location* saves the metadata in a `.qmd` file alongside the file.
- In all other cases *Save to Default Location* saves the metadata in a local `.sqlite` database.

8.6.2 图层注解

图层注解允许记录当前工程的图层。可用于储存工程的用户的重要消息，如待办事项列表、说明、警告等。

从 图层面板的图层上下文菜单中，选择 *添加图层注解...* 并用必要的文本填写打开的对话框。

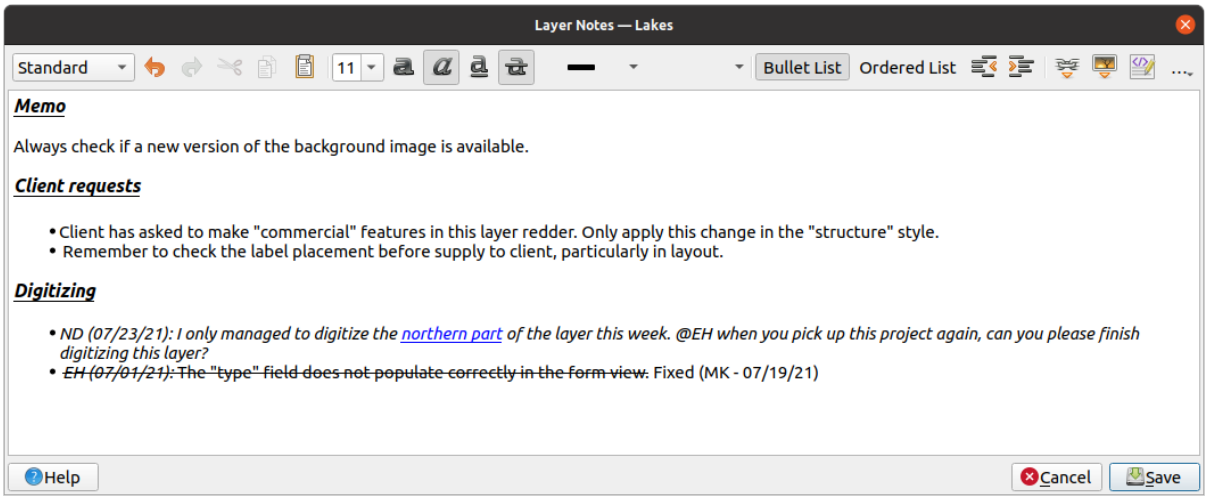


Fig. 8.15: 向图层添加注解

*添加图层注解*对话框提供了基于 HTML 格式的多行文本框，其中包含一套完整的工具，用于：

- 文本操作：剪切、复制、粘贴、撤消、重做
- 字符格式，应用于全部或部分内容：字号和颜色、粗体、斜体、下划线、删除线、背景色、URL 突出显示
- 段落结构：项目符号和有序列表、缩进、预定义标题
- 文件插入，即使用拖放
- 使用 HTML 编码进行编辑

在工具栏的最右侧 ... 下拉菜单，您可以：

- *移除所有格式*
- *移除字符格式*
- *清除所有内容*

在 图层面板中，带有注解的图层将被指定为 图标，鼠标悬停于该图标时将显示注解内容。单击图标将编辑注解。您也可以右键单击图层并 *编辑图层注解...* 或 *移除图层注解*。

> **Note**
>
> 注解是图层样式的一部分，可以保存在 .qml 或 .qlr 文件中。在复制粘贴图层样式时，可以将它们从一个图层传递到另一个图层。

8.7 将值存储在变量中

在 QGIS 中，您可以使用变量来储存可在表达式中使用的复发值 (例如，工程的标题或用户的全称)。变量可以在应用程序的全局级别、工程级别、图层级别、处理建模级别、布局级别和布局项级别进行定义。就如 CSS 级联规则，变量也可以被盖写 - 例如，工程级别变量将盖写任何具有相同名称的应用程序全局级别变量集。您可以使用这些变量在变量名前加 @ 字符构建文本字符串或其他自定义表达式。例如，在打印布局中使用以下内容创建标注:

```
This map was made using QGIS [% @qgis_version %]. The project file for this
map is: [% @project_path %]
```

将按如下渲染标注:

```
This map was made using QGIS 3.4.4-Madeira. The project file for this map is:
/gis/qgis-user-conference-2019.qgs
```

除了预设只读变量 外，还可以为上述任何级别定义自己的自定义变量。您可以管理:

- **全局变量**在 设置 ► 选项菜单中设置
- **工程变量**在 工程属性对话框中设置 (请参阅工程属性)
- 图层属性对话框中的 **矢量图层变量** (请参阅矢量属性对话框);
- 模型构建器对话框中的 **模型变量** (请参阅*The model designer*);
- 打印布局中 布局面板的 **布局变量** (请参阅*The Layout Panel*);
- 打印布局中 项属性面板的 **布局项变量** (请参阅*Layout Items Common Options*) 。

为了区别可编辑变量，只读变量名称和值以斜体显示。而由较低级变量盖写较高级变量时添加删除线显示。

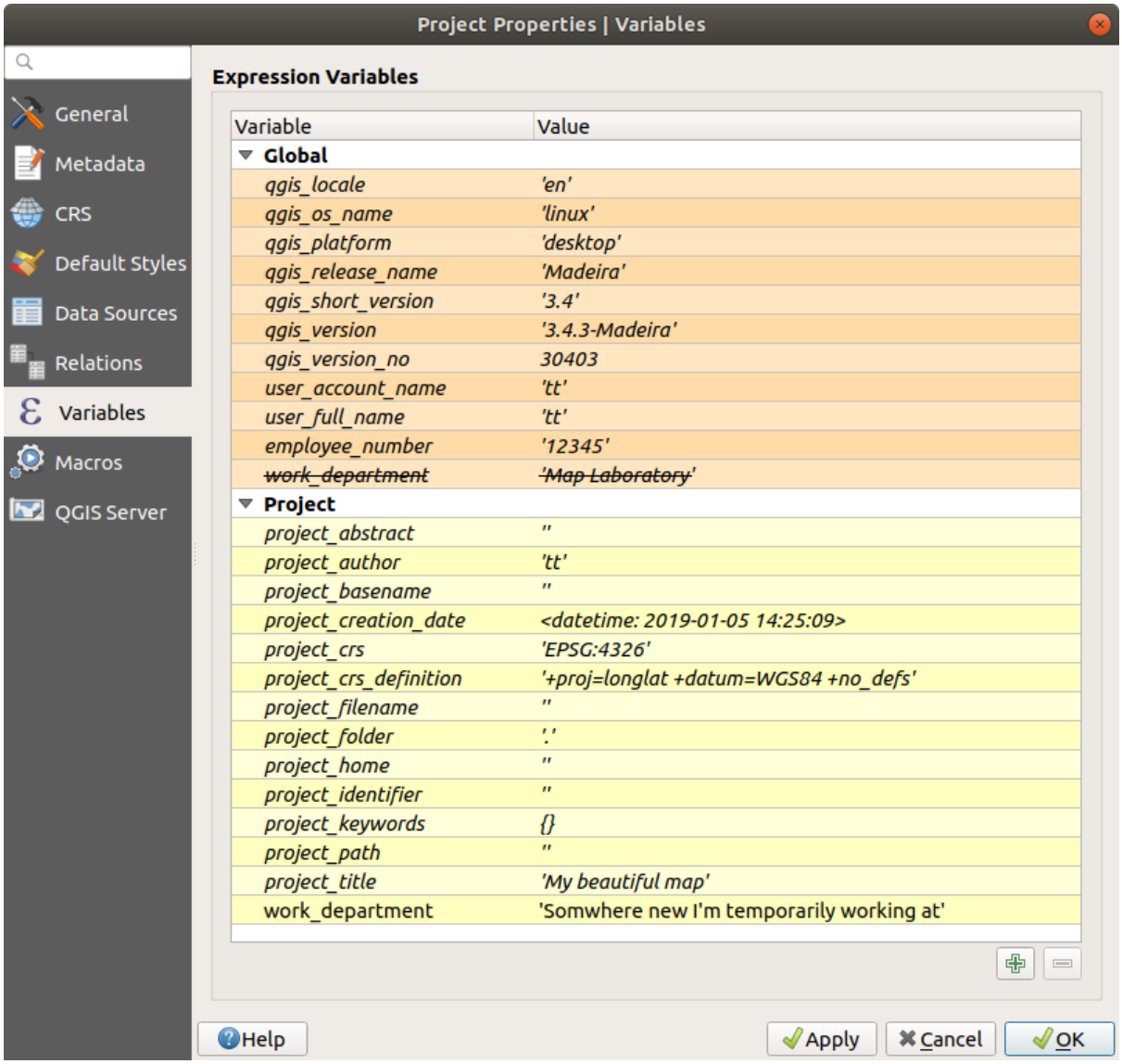


Fig. 8.16: 工程级别的变量编辑器

Note

您可以在 Nyall Dawson 的 探索 QGIS 2.12 中的变量, part 1, part 2 和 part 3 博客文章中阅读有关变量的更多信息并查找一些示例。

8.8 常用控件

在 QGIS 中，您经常需要使用一些选项。为了方便起见，QGIS 提供了如下所示的专用控件。

8.8.1 颜色选择器

颜色对话框

每当单击 图标选择颜色时，就会显示 *选择颜色* 对话框。此对话框的功能取决于 *设置* ► *选项...* ► *常规* 中 *使用操作系统颜色选择对话框* 参数复选框的状态。勾选时，使用的颜色对话框是运行 QGIS 操作系统的原生对话框。否则，将使用 QGIS 自定义颜色选择器。

自定义颜色选择器对话框有四个不同的选项卡，允许您按 颜色渐变， 色轮， 色卡 或 拾色器 选择颜色。前两个选项卡，可以浏览到所有的颜色组合，并将您的选择应用于项。

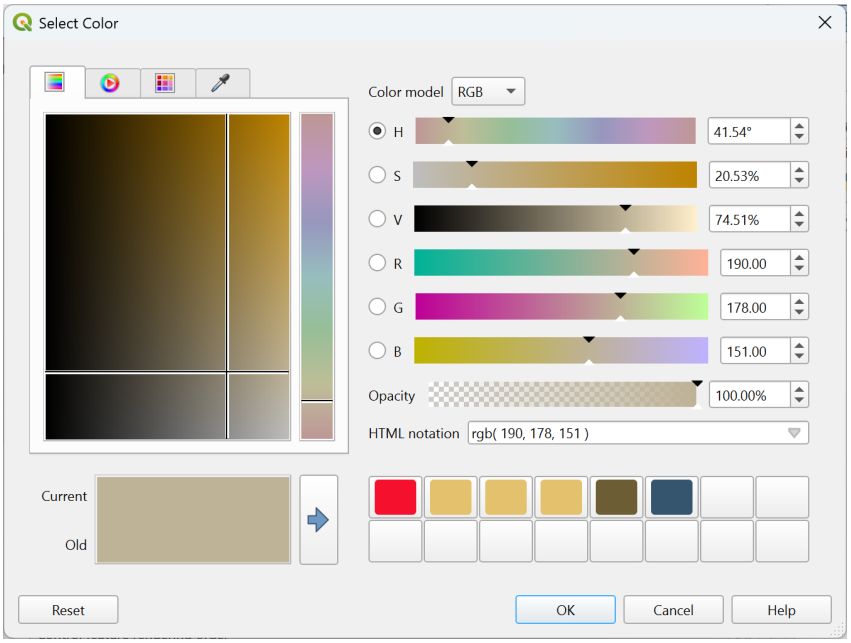


Fig. 8.17: 颜色选择器渐变选项卡

在 色卡 选项卡，可以从调色板列表中进行选择 (详细信息请参阅*Colors settings*) 。除了 *最近使用过的颜色* 调色板外，其他的都可以通过框架底部的 添加当前颜色 和 移除选中的颜色 按钮进行修改。

调色板组合框旁边的 ... 按钮还提供了几个选项：

- 复制、粘贴、导入或导出颜色
- 创建、导入或移除调色板
- 使用 *在颜色按钮中显示项* 将自定义调色板添加到颜色选择器控件 (请参阅 Fig. 8.19)

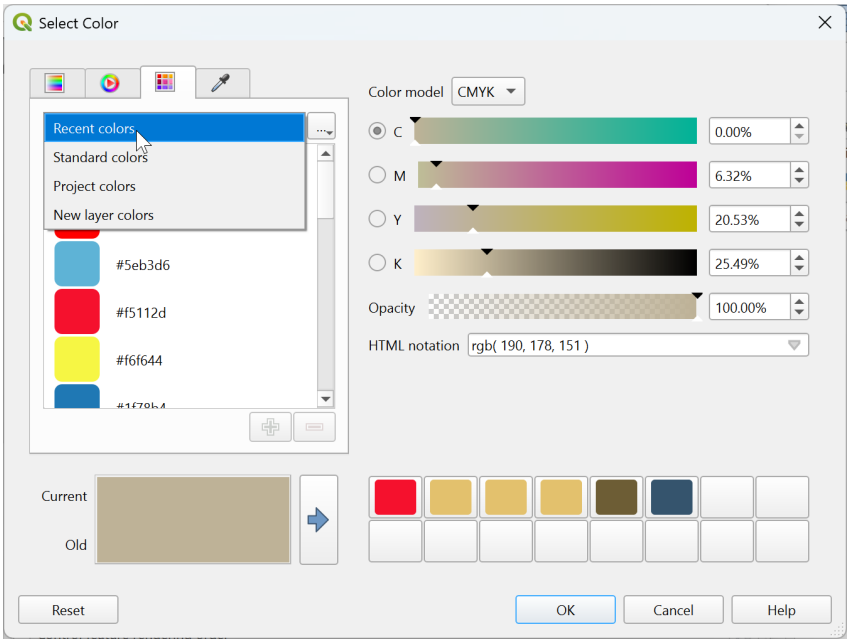


Fig. 8.18: 颜色选择器色卡选项卡

另一个选项是使用 拾色器，它允许您从 QGIS 用户界面的任何部分的鼠标光标位置，甚至从另一个应用程序中拾取颜色：在选项卡处于活动状态时按下空格键，将鼠标移到所需颜色上并单击，或再次按下空格键。也可以单击 *拾取颜色* 按钮激活拾色器。

Whatever method you use, the selected color is always described through options of a *Color model* that defaults to the *project color model*:

- *RGB*: color is set using sliders for `HSV` (Hue, Saturation, Value) or `RGB` (Red, Green, Blue) values.
- *CMYK* (Cyan, Magenta, Yellow, Black)

修改颜色非常简单，只需单击色轮或渐变或任意颜色参数滑块即可。您可以使用旁边的输入框或在相应滑块上滚动鼠标滚轮来调整这些参数，也可以在 HTML 标记键入颜色。最后，还可以使用 *不透明度* 滑块设置透明度级别。

该对话框还提供了 *旧颜色* (对象原颜色) 和 *新颜色* (正在选择) 之间的视觉对比。使用拖放或点击 *将颜色添加到色卡* 按钮，可以将选择的颜色保存在一个色卡中，便于以后使用。

> **Tip**
>
> **快速颜色修改**
>
> 将颜色选择器控件拖放到另一个控件上，应用其颜色。

颜色下拉快捷方式

单击 颜色按钮右侧的下拉箭头，可以显示用于快速选择颜色的控件。此快捷方式可访问：

- 从色轮拾取颜色
- 用于更改颜色不透明度的 Alpha 滑块
- 之前设置为 *在颜色按钮中显示* 的颜色调色板
- 复制当前颜色并将其粘贴到另一个控件

- 从计算机显示器的任意位置拾取颜色
- 从颜色选择器对话框选择颜色
- 将颜色从一个控件拖放到另一个控件，进行快速修改

Tip

将鼠标滚轮在颜色选择器控件上滚动，可以快速修改关联颜色的不透明度。

Note

当颜色控件通过数据定义重设属性设置为*工程颜色*时，上述用于更改颜色的功能不可用。您首先需要 *取消颜色链接*或 *清除定义*。

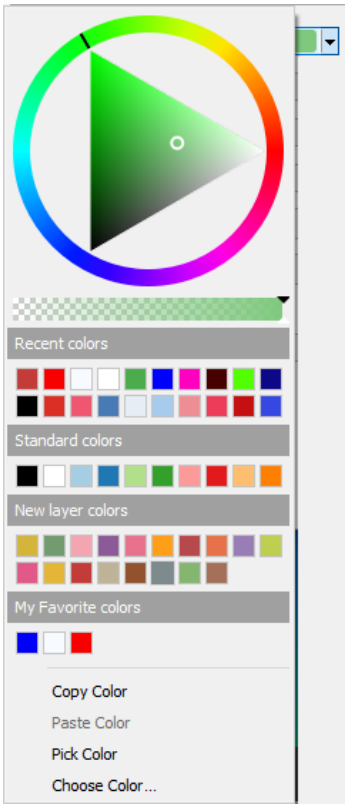


Fig. 8.19: 快速颜色选择器菜单

颜色渐变下拉快捷方式

颜色渐变是将一组颜色应用于一个或多个要素的实用方法。其创建方式在设置颜色渐变 章节进行了描述，对于颜色，点击 颜色渐变按钮将打开相应的颜色渐变类型对话框，并允许您更改其属性。

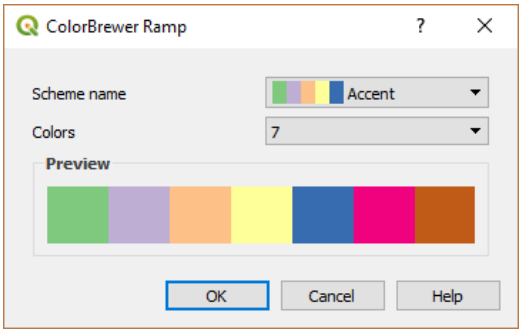


Fig. 8.20: 自定义 ColorBrewer 渐变

按钮右侧的下拉菜单可快速访问更多的颜色渐变和选项集：

- 反转颜色渐变
- 清除当前渐变：取消为控件设置的颜色渐变 (仅在某些环境中可用)
- 随机颜色渐变：仅在某些环境中可用 (例如，当颜色渐变用于图层符号化时)，勾选此条目将创建并应用具有随机颜色的颜色渐变。如果当前的随机颜色渐变不令人满意，还可以启用 打乱随机颜色条目重新生成新的随机颜色渐变。
- 梯度的预览或在 样式管理器对话框中标记为 **收藏夹**的 目录分类： `cpt-city` 的颜色渐变
- 全部颜色渐变：访问兼容的颜色渐变数据库
- 可在当前控件使用的受支持的 新建颜色渐变... 类型 (请注意，除非您将其保存到库，否则此颜色渐变无法在其他地方使用)
- 编辑颜色渐变...：与单击整个颜色渐变按钮的作用相同
- 保存颜色渐变...：将当前颜色渐变及其自定义项保存到样式库

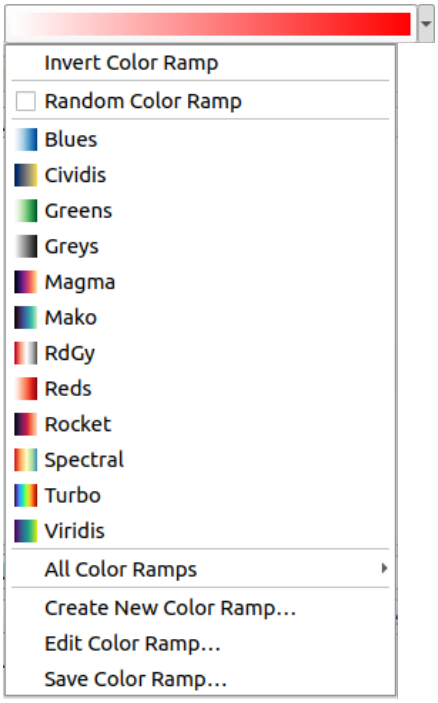


Fig. 8.21: 快速颜色渐变选择控件

8.8.2 符号控件

符号选择器控件是用于设置要素符号属性的快捷方式。单击下拉箭头将显示以下符号选项以及颜色下拉控件 的功能：

- 配置符号...: 与点击符号选择器控件相同，将打开对话框设置符号参数 。
- 从当前项 复制符号
- 粘贴符号到当前项，加速配置
- 清除当前符号取消为控件设置的指定符号 (仅在某些上下文中可用)

Tip

将鼠标滚轮在标记或线符号控件上滚动，可以快速修改关联符号的大小。

8.8.3 远程或嵌入式文件选择器

随同文件选择器控件, ... 按钮有时会显示下拉箭头。这通常在使用时可用：

- 符号或标注中的 SVG 文件
- 用于自定义符号、标注、纹理或装饰的栅格图像

按下箭头将为您提供一个菜单：

- *Select File*⋯, to load the file from the file system. The file is identified through the file path and QGIS needs to resolve the path in order to display the corresponding image
- *From URL*⋯, to load the file from a remote URL. As above, the image will only be loaded on successful retrieval of the remote resource

- *Embed File*⋯, to embed the file inside the current project, style database, or print layout template. The file is then always rendered as part of the item. This is a convenient way to create self-contained projects with custom symbols which can be easily shared amongst different users and installations of QGIS.
- *Extract Embedded File*⋯, to extract the embedded file from the widget and save it on disk.

8.8.4 可见比例选择器

可见比例选择器提供了用于控制元素在地图画布中可见比例的选项。在指定比例范围之外，元素不会显示。可以从其 渲染属性选项卡进行设置，并应用于例如图层、标注或图表等。

1. 勾选 依赖可见比例复选框
2. 用所需缩小比例的最大值填充 最小比例 (不含) 框，键入或从预定义比例 中选择值
3. 和/或使用所需放大比例的最大值填充 最大比例 (包含) 框

   比例框旁边的 设置为当前画布比例 按钮将当前地图画布比例设置为范围可见边界。点击按钮旁边的箭头可以访问布局地图中的比例，并重新使用其填充框。

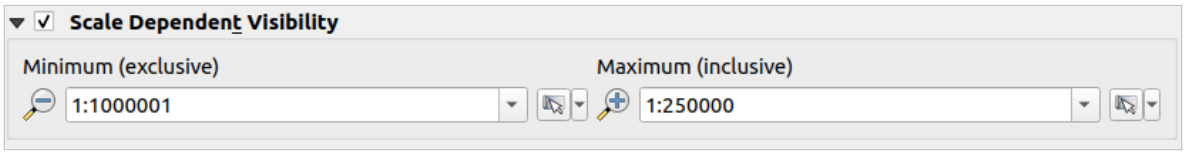


Fig. 8.22: 可见比例选择器控件

8.8.5 空间范围选择器

当您想选择要分配给图层的空间范围或限制要运行的动作时，范围选择器控件是便捷的快捷方式。根据上下文，提供如下选项：

- *Current Layer Extent*: e.g. when exporting a layer
- *Calculate from Layer* ►: uses extent of a layer loaded in the current project
- Use current *Map Canvas Extent*
- *Draw on Canvas*: a rectangle whose coordinates are then used
- *Calculate from Bookmark*: uses extent of a saved *bookmark*
- *Calculate from Layout Map*: uses extent of a *layout map*
- Enter or edit the coordinates as `xmin, xmax, ymin, ymax`

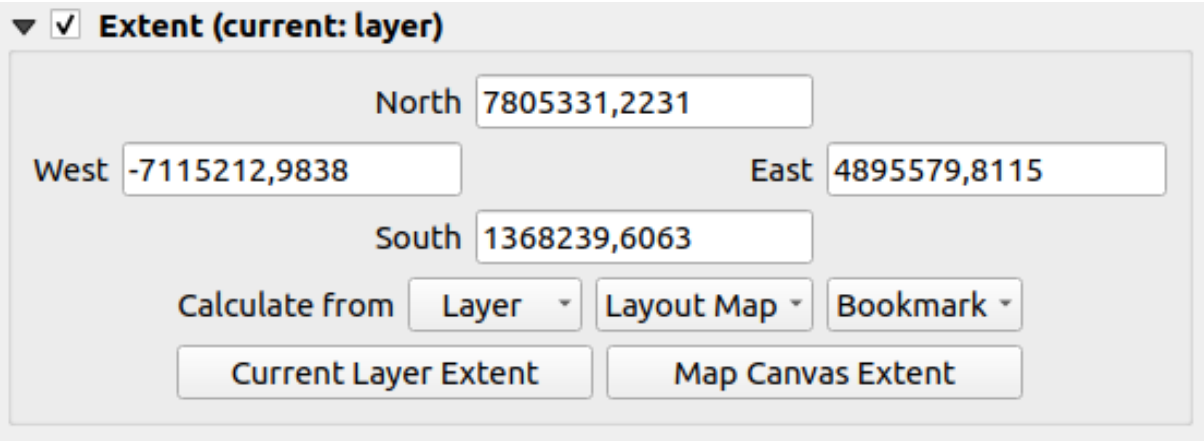


Fig. 8.23: Extent selector widget

8.8.6 Font Selector

The *Font* selector widget is a convenient shortcut when you want to set font properties for textual information (feature labels, decoration labels, map legend text, ...). Clicking the drop-down arrow shows some or all of the following options:

Fig. 8.24: Font selector drop-down menu

- *Clear Current Text Format* to unset any assigned text format to the widget (available only in some contexts)
- *Font Size* in the associated unit
- *Recent Fonts* ► menu with the active font checked (at the top)
- *Configure Format...*: same as pressing the font selector widget. It opens the *Text format* dialog, providing advanced formatting options such as color, opacity, orientation, HTML notation, buffer, background, shadow, ...

- *Copy Format* of the text
- *Paste Format* to the text, speeding configuration
- the *color widget* for quick color setting

> **Tip**
>
> Scroll the mouse wheel over a font selector widget to quickly modify the font size of the associated text.

8.8.7 Unit Selector

Size properties of the items (labels, symbols, layout elements, ...) in QGIS are not necessarily bound to either the project units or the units of a particular layer. For a large set of properties, the *Unit* selector drop-down menu allows you to tweak their values according to the rendering you want (based on screen resolution, paper size, or the terrain). Available units are:

- 毫米
- *Points*
- 像素
- 英寸
- *Percentage*: allows you to set some properties as a percent of another one. For example, this is useful for creation of text formats where the components (buffer size, shadow radius...) nicely scale as the text size is changed, instead of having constant buffer/shadow sizes. So you don't need to adjust those sizes, when the text size changes.
- *Meters at Scale*: This allows you to always set the size in meters, regardless of what the underlying map units are (e.g. they can be in inches, feet, geographic degrees, ...). The size in meters is calculated based on the current project ellipsoid setting and a projection of the distances in meters at the center of the current map extent. For maps in a projected coordinate system this is calculated using projected units. For maps in a geographic (latitude/longitude) based system the size is approximated by calculating meter sizes using ellipsoidal calculations for the vertical scale of the map.
- and *Map Units*: The size is scaled according to the map view scale. Because this can lead to too big or too small values, use the button next to the entry to constrain the size to a range of values based on:
  - The *Minimum scale* and the *Maximum scale*: The value is scaled based on the map view scale until you reach any of these scale limits. Out of the range of scale, the value at the nearest scale limit is kept.
  - and/or The *Minimum size* and the *Maximum size* in mm: The value is scaled based on the map view scale until it reaches any of these limits; Then the limit size is kept.

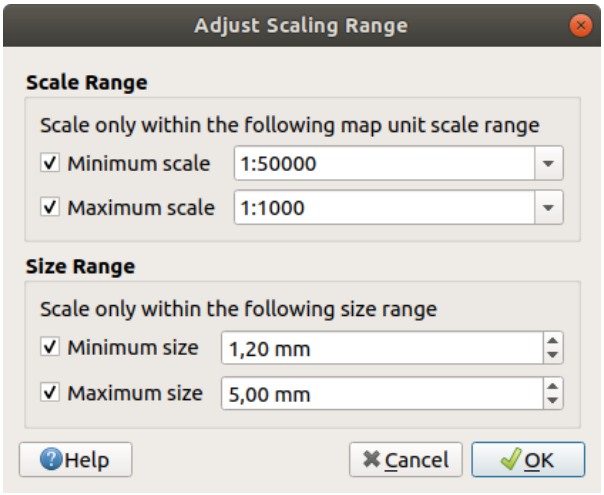


Fig. 8.25: Adjust scaling range dialog

8.8.8 Number Formatting

Numeric formatters allow formatting of numeric values for display, using a variety of different formatting techniques (for instance scientific notation, currency values, percentage values, custom formatting, etc). One use of this is to set text in a layout scale bar or table, a color ramp legend, an elevation profile plot, ...

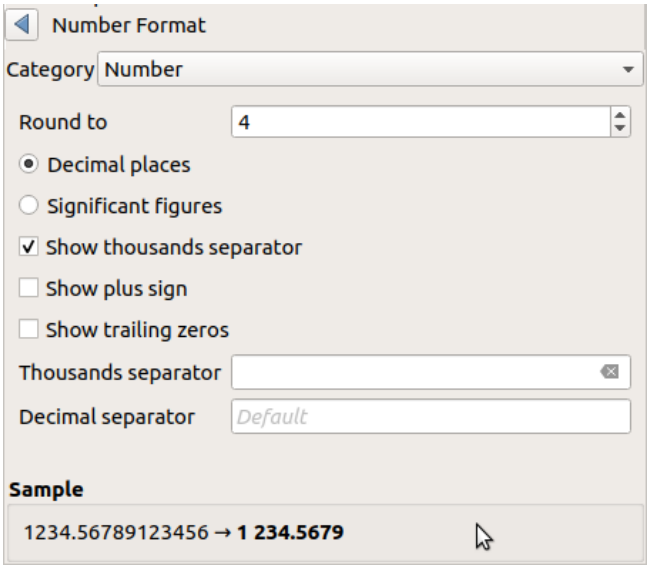


Fig. 8.26: Formatting numeric value

Different categories of formats are supported. For most of them, you can set part or all of the following numeric options:

- 显示千位分隔符
- 显示加号
- 显示尾随零

But they can also have their custom settings. Provided categories are:

- *General*, the default category: has no setting and displays values as set in the parent widget properties or using the global settings.
- *Custom expression*: allows you to craft a custom *QGIS expression* to format numbers. The expression can use the `@value` variable to retrieve the value to be formatted, and then use any standard QGIS expression function to format this as desired.
- *Number*
  - The value can be *Round to* a self defined number of *Decimal places* or their *Significant figures*
  - customize the *Thousands separator* and *Decimal separator*
- *Bearing* for a text representation of a direction/bearing using:
  - *Format*: possible ranges of values are `0 to 180°, with E/W suffix`, `-180 to +180°` and `0 to 360°`
  - number of *Decimal places*
- *Currency* for a text representation of a currency value.
  - 前缀
  - 后缀
  - number of *Decimal places*
- *Fraction* for a vulgar fractional representation of a decimal value (e.g. *1/2* instead of *0.5*)
  - ☐ *Use unicode super/subscript* to show. For example $^{1}/_{2}$ instead of 1/2
  - ☐ 使用专用的 *Unicode* 字符
  - customize the *Thousands separator*
- *Percentage* - appends `%` to the values, with setting of:
  - number of *Decimal places*
  - *Scaling* to indicate whether the actual values already represent percentages (then they will be kept as is) or fractions (then they are converted)
- *Scientific* notation in the form `2.56e+03`. The number of *Decimal places* can be set.

A live preview of the settings is displayed under the *Sample* section.

8.8.9 Blending Modes

QGIS offers different options for special rendering effects with these tools that you may previously only know from graphics programs. Blending modes can be applied on layers and features, and also on print layout items:

- *Normal*: This is the standard blend mode, which uses the alpha channel of the top pixel to blend with the pixel beneath it. The colors aren't mixed.
- *Lighten*: This selects the maximum of each component from the foreground and background pixels. Be aware that the results tend to be jagged and harsh.
- *Screen*: Light pixels from the source are painted over the destination, while dark pixels are not. This mode is most useful for mixing the texture of one item with another item (such as using a hillshade to texture another layer).
- *Dodge*: Brighten and saturate underlying pixels based on the lightness of the top pixel. Brighter top pixels cause the saturation and brightness of the underlying pixels to increase. This works best if the top pixels aren't too bright. Otherwise the effect is too extreme.
- *Addition*: Adds pixel values of one item to the other. In case of values above the maximum value (in the case of RGB), white is displayed. This mode is suitable for highlighting features.

- *Darken*: Retains the lowest values of each component of the foreground and background pixels. Like lighten, the results tend to be jagged and harsh.
- *Multiply*: Pixel values of the top item are multiplied with the corresponding values for the bottom item. The results are darker.
- *Burn*: Darker colors in the top item cause the underlying items to darken. Burn can be used to tweak and colorize underlying layers.
- *Overlay*: Combines multiply and screen blending modes. Light parts become lighter and dark parts become darker.
- *Soft light*: Very similar to overlay, but instead of using multiply/screen it uses color burn/dodge. This is supposed to emulate shining a soft light onto an image.
- *Hard light*: Hard light is also very similar to the overlay mode. It's supposed to emulate projecting a very intense light onto an image.
- *Difference*: Subtracts the top pixel from the bottom pixel, or the other way around, in order always to get a positive value. Blending with black produces no change, as the difference with all colors is zero.
- *Subtract*: Subtracts pixel values of one item from the other. In the case of negative values, black is displayed.

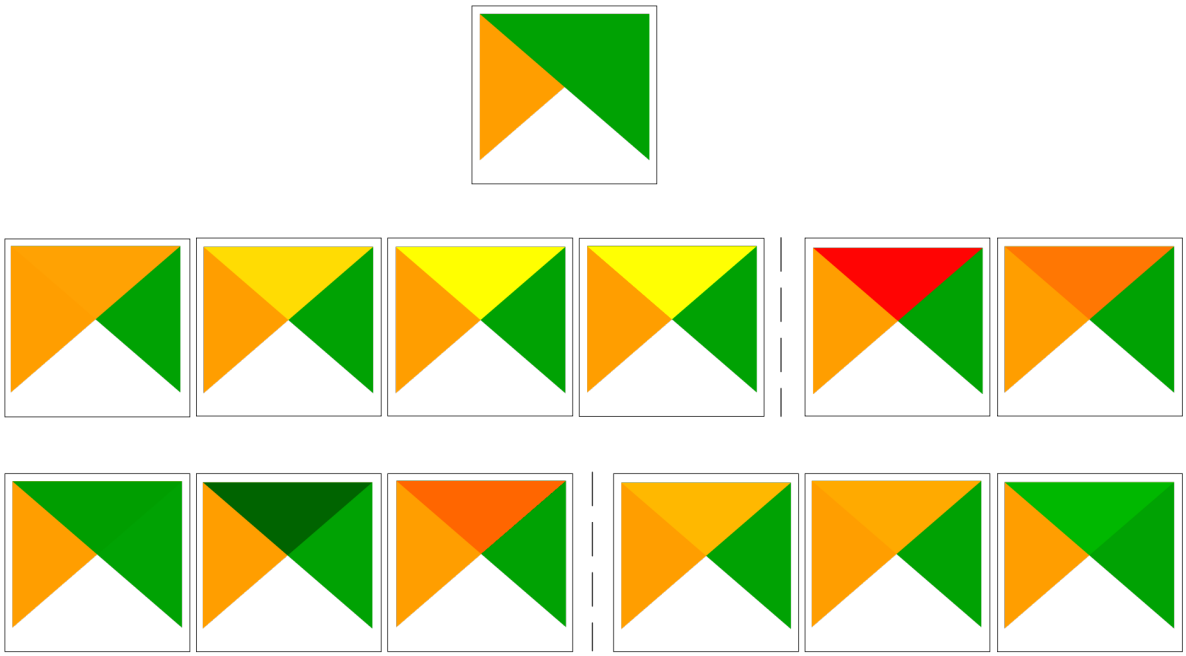


Fig. 8.27: Examples of blend modes applied to a green feature over an orange
From top to bottom, left to right: Normal -- Lighten, Screen, Dodge, Addition -- Difference, Subtract -- Darken, Multiply, Burn -- Overlay, Soft light, Hard light

When a layer is part of a group that *renders layers as a group*, additional blending modes are available for the rendering. They provide methods to clip the render of one layer's content by the content in a second "mask" layer.

- *Masked By Below*: The output is the top pixel, where the opacity is reduced by that of the bottom pixel.
- *Mask Below*: The output is the bottom pixel, where the opacity is reduced by that of the top pixel.
- *Inverse Masked By Below*: The output is the top pixel, where the opacity is reduced by the inverse of the bottom pixel.
- *Inverse Mask Below*: The output is the bottom pixel, where the opacity is reduced by the inverse of the top pixel.

- *Paint Inside Below*: The top pixel is blended on top of the bottom pixel, with the opacity of the top pixel reduced by the opacity of the bottom pixel.
- *Paint Below Inside*: The bottom pixel is blended on top of the top pixel, with the opacity of the bottom pixel reduced by the opacity of the top pixel.

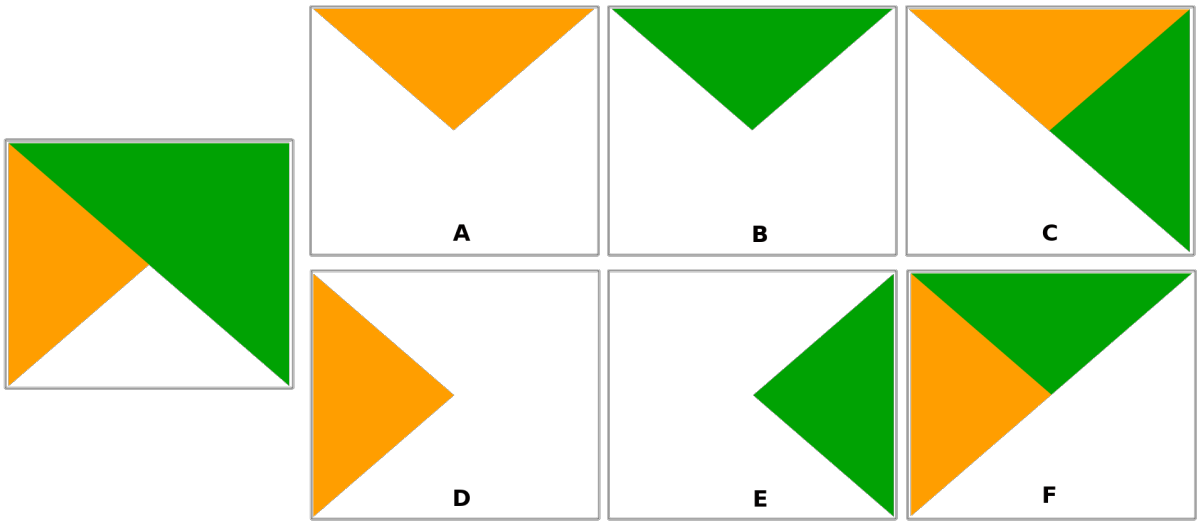


Fig. 8.28: Examples of blend clipping mode applied to top green layer in a group
A: Mask Below **B**: Masked By Below **C**: Paint Below Inside **D**: Inverse Mask Below **E**: Inverse Masked By Below **F**: Paint Inside Below

8.8.10 Data defined override setup

Next to many options in the vector layer properties dialog or settings in the print layout, you will find a Data defined override icon. Using *expressions* based on layer attributes or item settings, prebuilt or custom functions and *variables*, this tool allows you to set dynamic values for parameters. When enabled, the value returned by this widget is applied to the parameter regardless of its normal value (checkbox, textbox, slider...).

The data defined override widget

Clicking the Data defined override icon shows the following entries:

- *Description...* that indicates if the option is enabled, which input is expected, the valid input type and the current definition. Hovering over the widget also pops up this information.
- *Store data in the project*: a button allowing the property to be stored using the *Auxiliary Storage Properties* mechanism.
- *Field type*: an entry to select from the layer's fields that match the valid input type.
- *Color*: when the widget is linked to a color property, this menu gives access to the colors defined as part of the current *project's colors* scheme.
- *Variable*: a menu to access the available user-defined *variables*
- *Edit...* button to create or edit the expression to apply, using the *Expression String Builder* dialog. To help you correctly fill in the expression, a reminder of the expected output's format is provided in the dialog.
- *Paste* and *Copy* buttons.
- *Clear* button to remove the setup.

- For numeric and color properties, *Assistant...* to rescale how the feature data is applied to the property (more details *below*)

> **Tip**
>
> **Use right-click to (de)activate the data override**
>
> When the data-defined override option is set up correctly the icon is yellow or . If it is broken, the icon is red or .
>
> You can enable or disable a configured Data-defined override button by simply clicking the widget with the right mouse button.

> **Tip**
>
> **Use middle-click to create or edit the expression to apply**
>
> You can directly open the *Expression String Builder* dialog to create or edit the expression to apply by simply clicking the Data-defined override widget with the middle mouse button.

Using the data-defined assistant interface

When the Data-defined override button is associated with a size, a width, a rotation, an opacity or a color property, it has an *Assistant...* option that helps you change how the data is applied to the parameter for each feature. Basically, QGIS will scale a range of input values over a range of output values, and render each feature with their corresponding ouput value.

The assistant allows you to:

- Define the *Input* data, i.e.:
  - *Source*: the attribute to represent, using a field or an *expression*
  - the range of values to represent: you can manually enter the values or use the Fetch value range from layer button to fill these fields automatically with the minimum and maximum values returned by the *Source* expression applied to your data. Features whose values are out of the given range will be represented with the property of the closest limit of the output range.
- *Apply transform curve*: by default, output values are applied to input features following a specific scaling method (see below for setting). You can override this logic: enable the transform option, click on the graphic to add break point(s) and drag the point(s) to apply a custom distribution.
- Define the *Output* values: the options vary according to the parameter to define. You can globally set:
  - for color: a linear interpolation of source values is done over a given *color ramp* in order to assign each symbol the correct color. A color to use for symbols with NULL values can also be given.
  - for rotation: a linear interpolation is done over a range of minimum and maximum angles in order to assign each symbol the correct rotation. An angle to use for symbols with NULL values can also be given.
  - for opacity: a polynomial interpolation is done over a range of minimum and maximum opacity values. The *Exponent* field dictates the way source values are mapped to the output range. Power values greater than `1` will cause the output values distribution to start slowly before accelerating as the source values approach their maximum. Smaller exponents (less than `1`) will do the opposite. An opacity value to use for symbols with NULL values can also be given.

- for width (line symbols): an interpolation is done over a range of minimum and maximum values in order to assign each symbol the correct width. The *Scale method* field controls how the interpolation is done and what the output value represents.
  - **Linear**: linearly interpolates the width of the symbol between the *Size from* and the *Size to* when the *Source* expression value varies from *value from* to *value to*. Symbols width are proportional.
  - **Exponential**: Also interpolates the width, but allows to control the exact power used for scaling through the *Exponent* field. Power values greater than `1` will cause the output values distribution to start slowly before accelerating as the source values approach their maximum. Smaller exponents (less than `1`) will do the opposite.

  The width to use for symbols with NULL values can also be given.
- for size (point symbols): an interpolation is done over a range of minimum and maximum values in order to assign each symbol the correct size "diameter". The *Scale method* field controls how the interpolation is done and what the output value represents.
  - **Radius**: linearly interpolates the diameter of the symbol between the *Size from* and the *Size to* when the *Source* expression value varies from *value from* to *value to*. Symbols diameter are proportional.
  - **Surface**: scales the symbol diameter according to the square root (power of `0.5`) of the *Source* expression value to linearly interpolate the area of the symbol between the *Size from* and the *Size to* when the *Source* expression value varies from *value from* to *value to*. Symbols area are proportional. This method reflects human perception of size variation better than the **radius** one.
  - **Flannery**: Applies a compensation factor (power of `0.57`) to the area interpolation, in order to mitigate the eyes' perceptual underestimation of proportional areas. Maybe relevant only for circular symbols.
  - **Exponential**: Also interpolates the area, but allows to control the exact power used for scaling through the *Exponent* field. Power values greater than `1` will cause the output values distribution to start slowly before accelerating as the source values approach their maximum. Smaller exponents (less than `1`) will do the opposite.

  The size to use for symbols with NULL values can also be given.

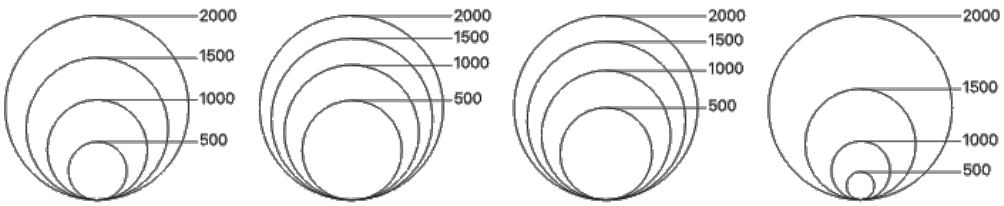


Fig. 8.29: Scaling values using different methods (from left to right): Radius - Surface - Flannery - Exponential of 2

When compatible with the property, a live-update preview is displayed in the right-hand side of the dialog to help you control the value scaling.

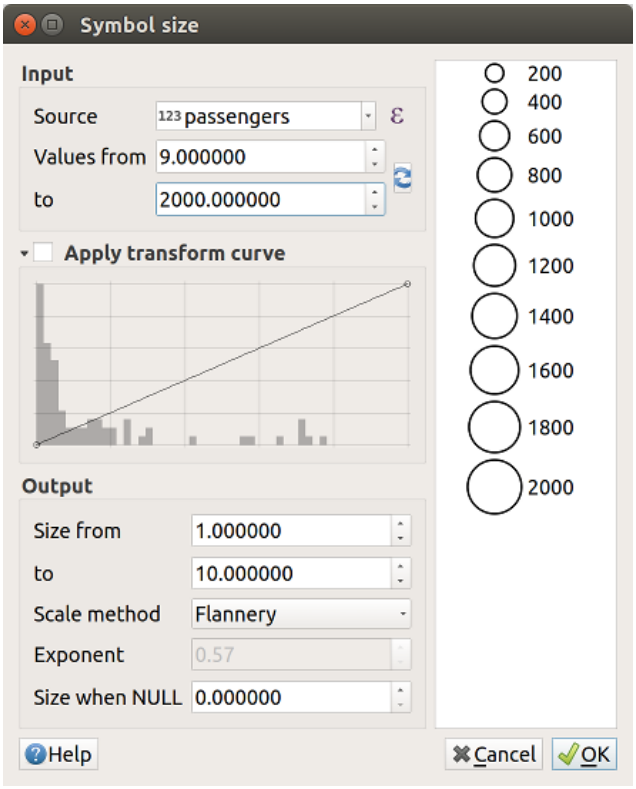


Fig. 8.30: Scaling feature size based on passengers field's value

The values presented in the varying size assistant above will set the size 'Data-defined override' with the editable formula below:

```
coalesce(scale_polynomial("passengers", 9, 2000, 1, 10, 0.57), 0)
```

8.8.11 认证

Many data providers in QGIS (such as databases, web services or file-based services) require credentials in order to connect. The authentication widget in QGIS helps users supply these credentials securely, and it offers two ways to do so:

- entering basic credentials directly,
- or selecting an existing authentication configuration stored in the encrypted authentication database.

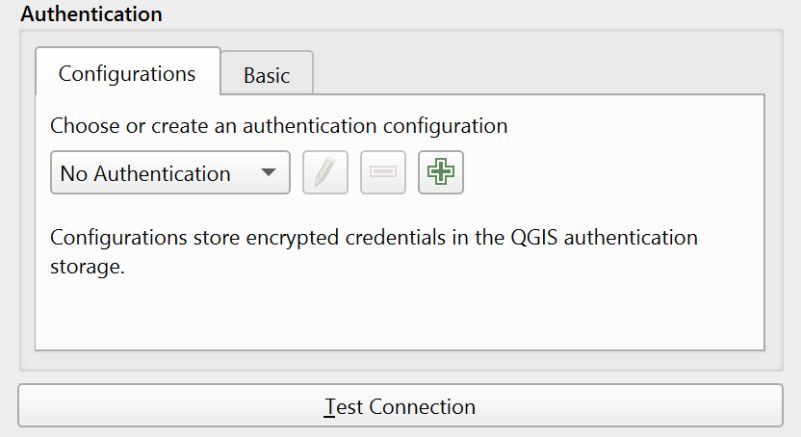


Fig. 8.31: Authentication widget

Basic

When setting authentication credentials, the *Basic* tab allows you to set:

- *User name*: User name used to connect to the data source.
- *Password*: Password used to connect to the data source.

Convert to configuration will save the user name and password in a new authentication configuration.

You can save any or both of the `User name` and `Password` parameters, in which case they will be used by default each time you need to connect to this database or service. If not saved, you'll be prompted to supply the credentials to connect to the database or service in next QGIS sessions. You can cancel the prompt, in which case the connection will fail. Click the *Ignore for session* button to avoid being prompted again during the current QGIS session. It is also possible to *Ignore for 10 seconds*. This will cancel the prompt but you need to wait 10 seconds before you can attempt to connect again. The connection parameters you entered are stored in a temporary internal cache and returned whenever a username/password for the same database or service is requested, until you end the current QGIS session.

> **Warning**
>
> **QGIS User Settings and Security**
>
> In the *Authentication* tab, saving **username** and **password** will keep unprotected credentials in the connection configuration. Those **credentials will be visible** if, for instance, you share the project file with someone. Therefore, it is advisable to save your credentials in an *Authentication configuration* instead (*Configurations* tab - See 认证系统 for more details) or in a service connection file (see *PostgreSQL Service connection file* for example).

Configurations

QGIS can store credentials under *authentication configurations*. Each configuration safely saves your login details in a portable encrypted database, so you can use them again without typing them every time. These configurations can be safely referenced by their ID tokens in projects and settings.

The authentication configuration selector allows users to choose from existing authentication configurations or create new ones. It provides options to:

- Select an existing authentication configuration from a drop-down list.
- Create a new authentication configuration

- Delete selected configuration
- Edit selected configuration

For more information on the authentication system, see 认证系统.

CHAPTER 9

使用表达式提升

9.1 表达式

基于图层数据和预构建或用户定义的函数，**表达式**提供了一种强大的方式来操作属性值、几何图形和变量，以便动态更改几何图形样式、标注的内容或位置、图表的值、布局项的高度、选择一些要素、创建虚拟字段等。

> **Note**
>
> 可以在函数列表 中找到用于编写表达式的默认函数和变量的列表，其中包含详细信息和示例。

9.1.1 表达式字符串构建器

用于构建表达式的主对话框，QGIS 的许多部件都可以使用 表达式字符串构建器，尤其是以下情况下访问特别有用：

- 点击 按钮；
- 使用 按表达式选择... 工具选择要素 ；
- 使用 字段计算器 等工具编辑属性 ；
- 使用 数据定义重写 工具控制符号、标注或布局项参数 (请参阅*Data defined override setup*) ；
- 构建几何图形生成器 符号图层；
- 完成一些地理处理 工作。

通过表达式构建器对话框可以访问：

- *Expression* tab which, thanks to a list of *predefined functions*, helps to write and check the expression to use;
- 函数编辑器选项卡 ，通过创建自定义函数来帮助扩展函数列表。

界面

表达式选项卡提供了使用函数、图层字段和值编写表达式的主界面，包含以下控件：

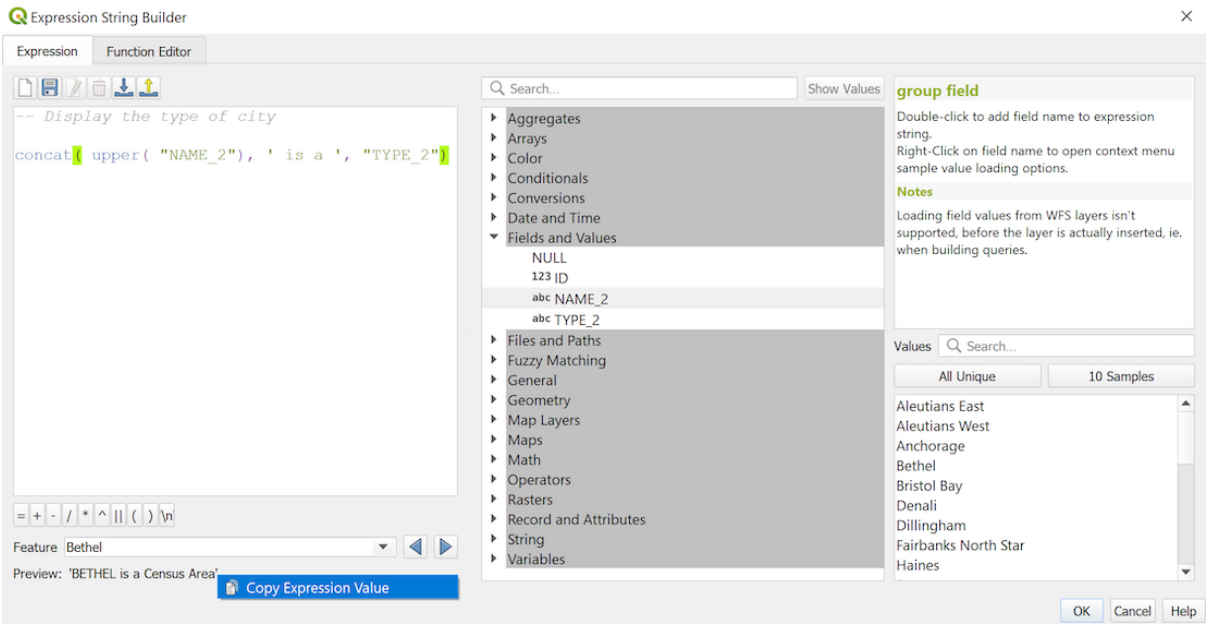


Fig. 9.1: 表达式选项卡

- 用于键入或粘贴表达式的表达式编辑区域，自动补全功能可用于加快表达式的编写速度：
  - 输入文本的对应变量、函数名和字段名，如下所示：使用 上和 下方向箭浏览条目，按 Tab 键插入表达式，或只需单击所需的条目。
  - 填充时显示函数参数。

  QGIS 还会检查表达式的正确性，并使用以下方法突出显示所有错误：
  - 下划线: 表示未知函数、参数错误或无效；
  - 标记: 表示单个位置的每个其他错误 (例如，缺少括号、意外字符)。

> **Tip**
>
> **用备注记录您的表达式**
>
> When using complex expression, it is good practice to add text either as a multiline comment or inline comments to help you remember.

```
/*
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
)
```

- Above the expression editor, a set of tools helps you:
  - Clear the expression editor
  - Create and manage *user expressions*
- Under the expression editor, you find:
  - A set of basic operators to help you build the expression
  - An indication of the expected format of output when you are data-defining feature properties
  - A live *Output preview* of the expression (up to 60 characters), evaluated on the first feature of the Layer by default. To view output preview text exceeding 60 characters, you can hover your cursor over the text to display a tooltip pop-up containing the entire output preview. To copy the output preview text onto your clipboard, right-click on the output preview text and select *Copy Expression Value*.

    You can browse and evaluate other features of the layer using the *Feature* combobox (the values are taken from the *display name* property of the layer).

    In case of error, it indicates it and you can access the details with the provided hyperlink.
- A function selector displays the list of functions, variables, fields... organized in groups. A search box is available to filter the list and quickly find a particular function or field. Double-clicking an item adds it to the expression editor.
- A help panel displays help for each selected item in the function selector.

> **Tip**
>
> Press `Ctrl+Click` when hovering a function name in an expression to automatically display its help in the dialog.

A field's values widget shown when a field is selected in the function selector helps to fetch features attributes:

- Look for a particular field value
- Display the list of *All Unique* or *10 Samples* values. Also available from right-click.

  When the field is mapped with another layer or a set of values, i.e. if the *field widget* is of *RelationReference*, *ValueRelation* or *ValueMap* type, it's possible to list all the values of the mapped field (from the referenced layer, table or list). Moreover, you can filter this list to *Only show values in use* in the current field.

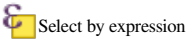
Double-clicking a field value in the widget adds it to the expression editor.

> **Tip**
>
> The right panel, showing functions help or field values, can be collapsed (invisible) in the dialog. Press the *Show Values* or *Show Help* button to get it back.

Writing an expression

QGIS expressions are used to select features or set values. Writing an expression in QGIS follows some rules:

1. **The dialog defines the context**: if you are used to SQL, you probably know queries of the type *select features from layer where condition* or *update layer set field = new_value where condition*. A QGIS expression also needs all these information but the tool you use to open the expression builder dialog provides parts of them. For example, giving a layer (`buildings`) with a field (`height`):

   - pressing the  tool means that you want to "select features from buildings". The **condition** is the only information you need to provide in the expression text widget, e.g. type `"height" > 20` to select buildings that are higher than 20.
   - with this selection made, pressing the button and choosing "height" as *Update existing field*, you already provide the command "update buildings set height = ??? where height > 20". The only remaining bits you have to provide in this case is the **new value**, e.g. just enter `50` in the expression editor textbox to set the height of the previously selected buildings.

2. **Pay attention to quotes**: single quotes return a literal, so a text placed between single quotes (`'145'`) is interpreted as a string. Double quotes will give you the value of that text so use them for fields (`"myfield"`). Fields can also be used without quotes (`myfield`). No quotes for numbers (`3.16`).

Note

Functions normally take as argument a string for field name. Do:

```
attribute( @atlas_feature, 'height' ) -- returns the value stored in the
↪"height" attribute of the current atlas feature
```

And not:

```
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.
```

Tip

Use named parameters to ease expression reading

Some functions require many parameters to be set. The expression engine supports the use of named parameters. This means that instead of writing the cryptic expression `clamp( 1, 2, 9)`, you can use `clamp( min:=1, value:=2, max:=9)`. This also allows arguments to be switched, e.g. `clamp( value:=2, max:=9, min:=1)`. Using named parameters helps clarify what the arguments for an expression function refer to, which is helpful when you are trying to interpret an expression later!

Some use cases of expressions

- From the Field Calculator, calculate a "pop_density" field using the existing "total_pop" and "area_km2" fields:

```
"total_pop" / "area_km2"
```

- Label or categorize features based on their area:

```
CASE WHEN $area > 10 000 THEN 'Larger' ELSE 'Smaller' END
```

- Update the field "density_level" with categories according to the "pop_density" values:

```
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
```

- Apply a categorized style to all the features according to whether their average house price is smaller or higher than 10000€ per square metre:

```
"price_m2" > 10000
```

- Using the "Select By Expression..." tool, select all the features representing areas of "High population density" and whose average house price is higher than 10000€ per square metre:

```
"density_level" = 'High population density' and "price_m2" > 10000
```

  The previous expression could also be used to define which features to label or show on the map.

- Select features that overlap a natural zone from the "lands" layer:

```
overlay_intersects( layer:='lands', filter:="zone_type"='Natural' )
```

- Count for each feature the number of buildings they contain:

```
array_length( overlay_contains( layer:='buildings', expression:=@id ) )
```

- Create a different symbol (type) for the layer, using the geometry generator:

```
point_on_surface( @geometry )
```

- Given a point feature, generate a closed line (using `make_line`) around its geometry:

```
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 )␣
↪)
  )
)
```

- In a print layout label, display the name of the "airports" features that are within the layout "Map 1" item:

```
with_variable( 'extent',
               map_get( item_variables( 'Map 1' ), 'map_extent' ),
               aggregate( 'airports', 'concatenate', "NAME",
                          intersects( @geometry, @extent ), ' ,'
                        )
             )
```

Saving Expressions

Using the Add current expression to user expressions button above the expression editor frame, you can save important expressions you want to have quick access to. These are available from the **User expressions** group in the middle panel. They are saved under the *user profile* (`<userprofile>/QGIS/QGIS3.ini` file) and available in all expression dialogs inside all projects of the current user profile.

A set of tools available above the expression editor frame helps you manage the user expressions:

- Add the current expression to user expressions: store the expression in the user profile. A label and a help text can be added for easy identification.
- Edit selected expression from user expressions, as well as their help and label
- Remove selected expression from user expressions
- Import user expressions from a `.json` file into the active user profile folder
- Export user expressions as a `.json` file; all the user expressions in the user profile `QGIS3.ini` file are shared

9.1.2 Function Editor

With the *Function Editor* tab, you are able to write your own functions in Python language. This provides a handy and comfortable way to address particular needs that would not be covered by the predefined functions.

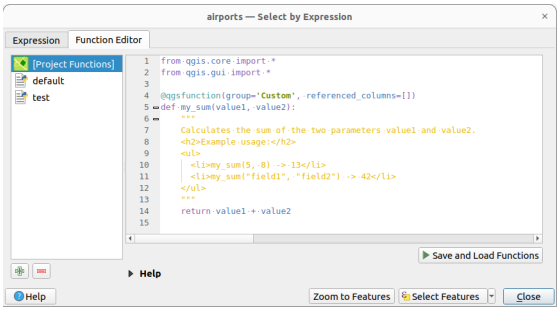


Fig. 9.2: The Function Editor tab

To create a new function:

1. Press the New File button.
2. Select where to store the new function. You can choose to save it either in a *Function file* or in *Project functions*.
   - If you select *Function file*, you need to enter a name for the Python `.py` file, which is based on a QGIS template file and is stored in the `/python/expressions` folder under the active *user profile* directory, thus available for all projects you may open with that profile.

     Custom Python functions stored as Function files are auto loaded whenever you start a new project under the user profile. Be aware that for sharing such Python functions, you need to share the `.py` file in the `/python/expressions` folder.
   - If you select *Project functions*, the new function is stored in the project file and only available for that project.

     Custom Python functions stored as Project functions can be easily shared by sharing the project file where they were saved. When opening a project with project functions, QGIS can load or ignore them depending on whether *embedded Python code is enabled*.

   A new item is added in the left panel of the *Function Editor* tab.

3. The right panel displays the content of the file: a python script template.
4. Uncomment the code block and start editing it according to your needs. You can rely on code autocompletion, code formatting and typing helpers, as defined in *Python settings*. Online API documentation for Qt and PyQGIS objects is also accessible pressing the `F1` key.

   You will find in the *Help* widget general information for setting the @qgsfunction decorator.
5. Press the *Save and Load Functions* button. The function you wrote is added to the functions tree in the *Expression* tab, by default under the `Custom` group.
6. Enjoy your new function.
7. If the function requires improvements, enable the *Function Editor* tab, do the changes and press again the *Save and Load Functions* button to make them available in the file, hence in any expression tab.

To delete a custom function:

1. Enable the *Function Editor* tab
2. Select the function in the list
3. Press the Remove selected function. The function is removed from the list and, depending on the storage, the corresponding `.py` file is deleted from the user profile folder, or the Python functions are removed from the project file.

> **Note**
>
> When a project file with Project functions is closed, the corresponding Project functions are unloaded and are no longer available in the QGIS session.

Example

Here's a short example on how to create your own `my_sum` function that will operate with two values.

```
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
```

The `@qgsfunction` decorator accepts the following arguments:

- The `group` argument indicates the group in which the function should be listed in the Expression dialog.
- `usesgeometry`: Set this to `True` if the expression requires access to the features geometry. By default `False`.
- `handlesnull`: Set this to `True` if the expression has custom handling for NULL values. If `False` (default), the result will always be NULL as soon as any parameter is NULL.
- `referenced_columns`: An array of attribute names that are required to run the function. Defaults to `[QgsFeatureRequest.ALL_ATTRIBUTES]`. Specifying a subset of fields or an empty list will result in a faster execution.

- `params_as_list`: Set this to `True` to pass the function parameters as a list. If `False` (default), the function will receive the parameters as individual arguments.

The function itself takes any number and type of parameters you want to pass to your function. These should be set before the following optional arguments:

- `feature`: the current feature
- `parent`: the `QgsExpression` object
- `context`: If there is an argument called `context` found at the last position, this variable will contain a `QgsExpressionContext` object, that gives access to various additional information like expression variables. E.g. `context.variable( 'layer_id' )`

If those arguments are present in the function signature, they will be automatically passed to the function, without the need to specify them in the expression.

The previous example function can then be used in expressions:

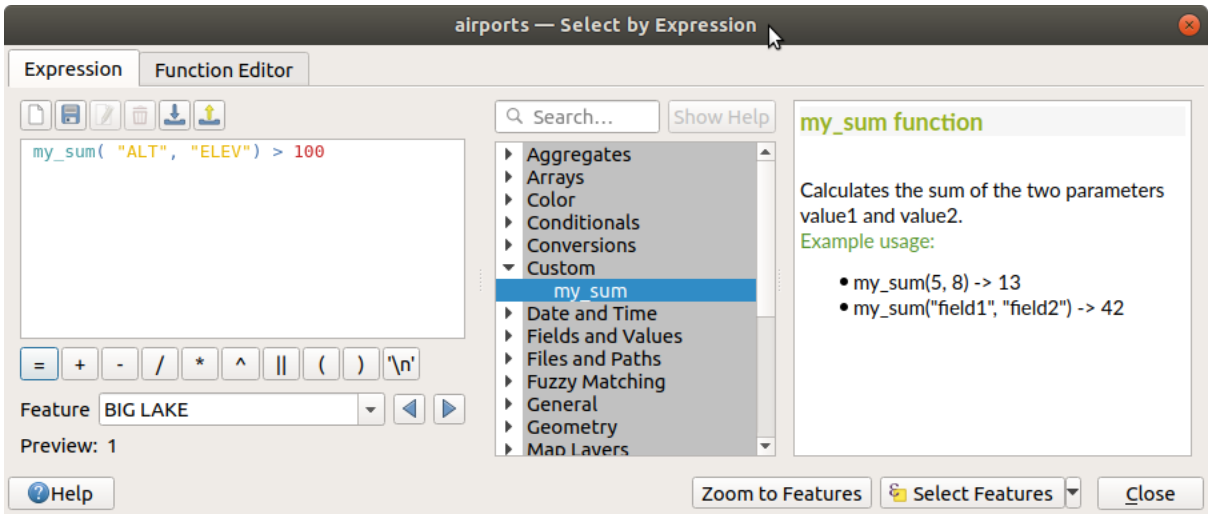


Fig. 9.3: Custom Function added to the Expression tab

Find more examples and additional arguments for custom functions at `qgsfunction`. Further information about creating Python code can be found in the PyQGIS-Developer-Cookbook.

9.2 函数列表

QGIS 中可用的函数、运算符和变量如下所示，并按类别分组。

9.2.1 聚合函数

该组包含在图层和字段上聚合值的函数。

aggregate

返回使用其他图层中的要素计算的聚合值。

语法	aggregate(layer, aggregate, expression, [filter], [concatenator=''], [order_by]) [] 标记可选参数
参数	• **图层** - 字符串, 表示图层名称或图层 ID • **聚合** - 与要计算的聚合相对应的字符串。有效选项为: – count – count_distinct – count_missing – minimum or min – maximum or max – sum – mean – median – stdev – stdevsample – range – minority – majority – q1: 第一四分位数 – q3: 第三四分位数 – iqr: 四分位距 – min_length: 最小字符串长度 – max_length: 最大字符串长度 – concatenate: 使用连接符连接字符串 – concatenate_unique: 使用连接符连接唯一字符串 – collect: 创建聚合的多部件几何图形 – array_agg: create an array of aggregated values • **expression** - sub expression or field name to aggregate • **filter** - optional filter expression to limit the features used for calculating the aggregate. Fields and geometry are from the features on the joined layer. The source feature can be accessed with the variable @parent. • **concatenator** - optional string to use to join values for 'concatenate' and 'concatenate_unique' aggregates • **order_by** - optional filter expression to order the features used for calculating the aggregate. Fields and geometry are from the features on the joined layer. By default, the features will be returned in an unspecified order.
Examples	• `aggregate(layer:='rail_stations',aggregate:='sum', expression:="passengers")` → sum of all values from the passengers field in the rail_stations layer • `aggregate('rail_stations','sum', "passengers"/7)` → calculates a daily average of "passengers" by dividing the "passengers" field by 7 before summing the values • `aggregate(layer:='rail_stations',aggregate:='sum', expression:="passengers",filter:="class">3)` → sums up all values from the "passengers" field from features where the "class" attribute is greater than 3 only • `aggregate(layer:='rail_stations',aggregate:='concatenate', expression:="name", concatenator:=',')` → comma separated list of the name field for all features in the rail_stations layer • `aggregate(layer:='countries', aggregate:='max', expression:="code", filter:=intersects( @geometry, geometry(@parent) ) )` → The country code of an intersecting country on the layer 'countries' • `aggregate(layer:='rail_stations',aggregate:='sum', expression:="passengers",filter:=contains( @atlas_geometry, @geometry ) )` → sum of all values from the passengers field in the rail_stations within the current atlas feature • `aggregate(layer:='rail_stations', aggregate:='collect', expression:=centroid(@geometry), filter:=region_name = attribute(@parent,'name') )` → aggregates centroid geometries of the rail_stations of the same region as current feature

array_agg

Returns an array of aggregated values from a field or expression.

语法	array_agg(expression, [group_by], [filter], [order_by]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate • **order_by** - optional expression to use to order features used to calculate aggregate. By default, the features will be returned in an unspecified order.
Examples	• `array_agg("name",group_by:="state")` → list of name values, grouped by state field

collect

Returns the multipart geometry of aggregated geometries from an expression

语法	collect(expression, [group_by], [filter]) [] 标记可选参数
参数	• **expression** - geometry expression to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate
Examples	• `collect( @geometry )` → multipart geometry of aggregated geometries • `collect( centroid(@geometry), group_by:="region", filter:="use" = 'civilian' )` → aggregated centroids of the civilian features based on their region value

concatenate

Returns all aggregated strings from a field or expression joined by a delimiter.

语法	concatenate(expression, [group_by], [filter], [concatenator], [order_by]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate • **concatenator** - optional string to use to join values. Empty by default. • **order_by** - optional expression to use to order features used to calculate aggregate. By default, the features will be returned in an unspecified order.
Examples	• `concatenate("town_name",group_by:="state",concatenator:=', ')` → comma separated list of town_names, grouped by state field

concatenate_unique

Returns all unique strings from a field or expression joined by a delimiter.

语法	concatenate_unique(expression, [group_by], [filter], [concatenator], [order_by]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate • **concatenator** - optional string to use to join values. Empty by default. • **order_by** - optional expression to use to order features used to calculate aggregate. By default, the features will be returned in an unspecified order.
Examples	• `concatenate_unique("town_name",group_by:="state", concatenator:=',')` → comma separated list of unique town_names, grouped by state field

count

Returns the count of matching features.

语法	count(expression, [group_by], [filter]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate
Examples	• `count("stations",group_by:="state")` → count of stations, grouped by state field

count_distinct

Returns the count of distinct values.

语法	count_distinct(expression, [group_by], [filter]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate
Examples	• `count_distinct("stations",group_by:="state")` → count of distinct stations values, grouped by state field

count_missing

Returns the count of missing (NULL) values.

语法	count_missing(expression, [group_by], [filter]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate
Examples	• `count_missing("stations",group_by:="state")` → count of missing (NULL) station values, grouped by state field

iqr

Returns the calculated inter quartile range from a field or expression.

语法	iqr(expression, [group_by], [filter]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate
Examples	• `iqr("population",group_by:="state")` → inter quartile range of population value, grouped by state field

majority

Returns the aggregate majority of values (most commonly occurring value) from a field or expression.

语法	majority(expression, [group_by], [filter]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate
Examples	• `majority("class",group_by:="state")` → most commonly occurring class value, grouped by state field

max_length

Returns the maximum length of strings from a field or expression.

语法	max_length(expression, [group_by], [filter]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate
Examples	• `max_length("town_name",group_by:="state")` → maximum length of town_name, grouped by state field

maximum

Returns the aggregate maximum value from a field or expression.

语法	maximum(expression, [group_by], [filter]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate
Examples	• `maximum("population",group_by:="state")` → maximum population value, grouped by state field

mean

Returns the aggregate mean value from a field or expression.

语法	mean(expression, [group_by], [filter]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate
Examples	• `mean("population",group_by:="state")` → mean population value, grouped by state field

median

Returns the aggregate median value from a field or expression.

语法	median(expression, [group_by], [filter]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate
Examples	• `median("population",group_by:="state")` → median population value, grouped by state field

min_length

Returns the minimum length of strings from a field or expression.

语法	min_length(expression, [group_by], [filter]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate
Examples	• `min_length("town_name",group_by:="state")` → minimum length of town_name, grouped by state field

minimum

Returns the aggregate minimum value from a field or expression.

语法	minimum(expression, [group_by], [filter]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate
Examples	• `minimum("population",group_by:="state")` → minimum population value, grouped by state field

minority

Returns the aggregate minority of values (least occurring value) from a field or expression.

语法	minority(expression, [group_by], [filter]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate
Examples	• `minority("class",group_by:="state")` → least occurring class value, grouped by state field

q1

Returns the calculated first quartile from a field or expression.

语法	q1(expression, [group_by], [filter]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate
Examples	• `q1("population",group_by:="state")` → first quartile of population value, grouped by state field

q3

Returns the calculated third quartile from a field or expression.

语法	q3(expression, [group_by], [filter]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate
Examples	• `q3("population",group_by:="state")` → third quartile of population value, grouped by state field

range

Returns the aggregate range of values (maximum - minimum) from a field or expression.

语法	range(expression, [group_by], [filter]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate
Examples	• `range("population",group_by:="state")` → range of population values, grouped by state field

relation_aggregate

Returns an aggregate value calculated using all matching child features from a layer relation.

语法	relation_aggregate(relation, aggregate, expression, [concatenator=''], [order_by]) [] 标记可选参数
参数	• **relation** - a string, representing a relation ID • **聚合** - 与要计算的聚合相对应的字符串。有效选项为: – count – count_distinct – count_missing – minimum or min – maximum or max – sum – mean – median – stdev – stdevsample – range – minority – majority – q1: 第一四分位数 – q3: 第三四分位数 – iqr: 四分位距 – min_length: 最小字符串长度 – max_length: 最大字符串长度 – concatenate: 使用连接符连接字符串 – concatenate_unique: 使用连接符连接唯一字符串 – collect: 创建聚合的多部件几何图形 – array_agg: create an array of aggregated values • **expression** - sub expression or field name to aggregate • **concatenator** - optional string to use to join values for 'concatenate' aggregate • **order_by** - optional expression to order the features used for calculating the aggregate. Fields and geometry are from the features on the joined layer. By default, the features will be returned in an unspecified order.
Examples	• `relation_aggregate(relation:='my_relation', aggregate:='mean',expression:="passengers")` → mean value of all matching child features using the 'my_relation' relation • `relation_aggregate('my_relation','sum', "passengers"/7)` → sum of the passengers field divided by 7 for all matching child features using the 'my_relation' relation • `relation_aggregate('my_relation','concatenate', "towns", concatenator:=',')` → comma separated list of the towns field for all matching child features using the 'my_relation' relation • `relation_aggregate('my_relation','array_agg', "id")` → array of the id field from all matching child features using the 'my_relation' relation

Further reading: *Setting relations between multiple layers*

stdev

Returns the aggregate standard deviation value from a field or expression.

语法	stdev(expression, [group_by], [filter]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate
Examples	• `stdev("population",group_by:="state")` → standard deviation of population value, grouped by state field

sum

Returns the aggregate summed value from a field or expression.

语法	sum(expression, [group_by], [filter]) [] 标记可选参数
参数	• **expression** - sub expression of field to aggregate • **group_by** - optional expression to use to group aggregate calculations • **filter** - optional expression to use to filter features used to calculate aggregate
Examples	• `sum("population",group_by:="state")` → summed population value, grouped by state field

9.2.2 Array Functions

This group contains functions to create and manipulate arrays (also known as list data structures). The order of values within the array matters, unlike the *'map' data structure*, where the order of key-value pairs is irrelevant and values are identified by their keys.

array

Returns an array containing all the values passed as parameter.

语法	array(value1, value2, ...)
参数	• **value** - a value
Examples	• `array(2,10)` → [2, 10] • `array(2,10)[0]` → 2

array_all

Returns TRUE if an array contains all the values of a given array.

语法	array_all(array_a, array_b)
参数	• **array_a** - an array • **array_b** - the array of values to search
Examples	• `array_all(array(1,2,3),array(2,3))` → TRUE • `array_all(array(1,2,3),array(1,2,4))` → FALSE

array_append

Returns an array with the given value added at the end.

语法	array_append(array, value)
参数	• **array** - an array • **value** - the value to add
Examples	• `array_append(array(1,2,3),4)` → [1, 2, 3, 4]

array_cat

Returns an array containing all the given arrays concatenated.

语法	array_cat(array1, array2, ...)
参数	• **array** - an array
Examples	• `array_cat(array(1,2),array(2,3))` → [1, 2, 2, 3]

array_contains

Returns TRUE if an array contains the given value.

语法	array_contains(array, value)
参数	• **array** - an array • **value** - the value to search
Examples	• `array_contains(array(1,2,3),2)` → TRUE

array_count

Counts the number of occurrences of a given value in an array.

语法	array_count(array, value)
参数	• **array** - an array • **value** - the value to count
Examples	• `array_count(array('a', 'b', 'c', 'b'), 'b')` → 2

array_distinct

Returns an array containing distinct values of the given array.

语法	array_distinct(array)
参数	• **array** - an array
Examples	• `array_distinct(array(1,2,3,2,1))` → [1, 2, 3]

array_filter

Returns an array with only the items for which the expression evaluates to true.

语法	array_filter(array, expression, [limit=0]) [] 标记可选参数
参数	• **array** - an array • **expression** - an expression to evaluate on each item. The variable *@element* will be replaced by the current value. • **limit** - maximum number of elements to be returned. Use 0 to return all values.
Examples	• `array_filter(array(1,2,3),@element < 3)` → [1, 2] • `array_filter(array(1,2,3),@element < 3, 1)` → [1]

array_find

Returns the lowest index (0 for the first one) of a value within an array. Returns -1 if the value is not found.

语法	array_find(array, value)
参数	• **array** - an array • **value** - the value to search
Examples	• `array_find(array('a', 'b', 'c'), 'b')` → 1 • `array_find(array('a', 'b', 'c', 'b'), 'b')` → 1

array_first

Returns the first value of an array.

语法	array_first(array)
参数	• **array** - an array
Examples	• `array_first(array('a','b','c'))` → 'a'

array_foreach

Returns an array with the given expression evaluated on each item.

语法	array_foreach(array, expression)
参数	• **array** - an array • **expression** - an expression to evaluate on each item. The variable *@element* will be replaced by the current value and the variable *@counter* by the current index (starting with 0).
Examples	• `array_foreach(array('a','b','c'),upper(@element))` → ['A', 'B', 'C'] • `array_foreach(array(1,2,3),@element + 10)` → [11, 12, 13] • `array_foreach(array(1,2,3),@element + @counter)` → [1, 3, 5]

array_get

Returns the Nth value (0 for the first one) or the last -Nth value (-1 for the last one) of an array.

语法	array_get(array, pos)
参数	• **array** - an array • **pos** - the index to get (0 based)
Examples	• `array_get(array('a','b','c'),1)` → 'b' • `array_get(array('a','b','c'),-1)` → 'c'

Hint

You can also use the *index operator ([])* to get a value from an array.

array_insert

Returns an array with the given value added at the given position.

语法	array_insert(array, pos, value)
参数	• **array** - an array • **pos** - the position where to add (0 based) • **value** - the value to add
Examples	• `array_insert(array(1,2,3),1,100)` → [1, 100, 2, 3]

array_intersect

Returns TRUE if at least one element of array1 exists in array2.

语法	array_intersect(array1, array2)
参数	• **array1** - an array • **array2** - another array
Examples	• `array_intersect(array(1,2,3,4),array(4,0,2,5))` → TRUE

array_last

Returns the last value of an array.

语法	array_last(array)
参数	• **array** - an array
Examples	• `array_last(array('a','b','c'))` → 'c'

array_length

Returns the number of elements of an array.

语法	array_length(array)
参数	• **array** - an array
Examples	• `array_length(array(1,2,3))` → 3

array_majority

Returns the most common values in an array.

语法	array_majority(array, [option='all']) [] 标记可选参数
参数	• **array** - an array • **option='all'** - a string specifying the return values handling. Valid options are: – all: Default, all most common values are returned in an array. – any: Returns one of the most common values. – median: Returns the median of the most common values. Non arithmetic values are ignored. – real_majority: Returns the value which occurs more than half the size of the array.
Examples	• `array_majority(array(0,1,42,42,43), 'all')` → [42] • `array_majority(array(0,1,42,42,43,1), 'all')` → [42, 1] • `array_majority(array(0,1,42,42,43,1), 'any')` → 1 or 42 • `array_majority(array(0,1,1,2,2), 'median')` → 1.5 • `array_majority(array(0,1,42,42,43), 'real_majority')` → NULL • `array_majority(array(0,1,42,42,43,42), 'real_majority')` → NULL • `array_majority(array(0,1,42,42,43,42,42), 'real_majority')` → 42

array_max

Returns the maximum value of an array.

语法	array_max(array)
参数	• **array** - an array
Examples	• `array_max(array(0,42,4,2))` → 42

array_mean

Returns the mean of arithmetic values in an array. Non numeric values in the array are ignored.

语法	array_mean(array)
参数	• **array** - an array
Examples	• `array_mean(array(0,1,7,66.6,135.4))` → 42 • `array_mean(array(0,84,'a','b','c'))` → 42

array_median

Returns the median of arithmetic values in an array. Non arithmetic values in the array are ignored.

语法	array_median(array)
参数	• **array** - an array
Examples	• `array_median(array(0,1,42,42,43))` → 42 • `array_median(array(0,1,2,42,'a','b'))` → 1.5

array_min

Returns the minimum value of an array.

语法	array_min(array)
参数	• **array** - an array
Examples	• `array_min(array(43,42,54))` → 42

array_minority

Returns the less common values in an array.

语法	array_minority(array, [option='all']) [] 标记可选参数
参数	• **array** - an array • **option='all'** - a string specifying the return values handling. Valid options are: – all: Default, all less common values are returned in an array. – any: Returns one of the less common values. – median: Returns the median of the less common values. Non arithmetic values are ignored. – real_minority: Returns values which occur less than half the size of the array.
Examples	• `array_minority(array(0,42,42), 'all')` → [0] • `array_minority(array(0,1,42,42), 'all')` → [0, 1] • `array_minority(array(0,1,42,42,43,1), 'any')` → 0 or 43 • `array_minority(array(1,2,3,3), 'median')` → 1.5 • `array_minority(array(0,1,42,42,43), 'real_minority')` → [42, 43, 0, 1] • `array_minority(array(0,1,42,42,43,42), 'real_minority')` → [42, 43, 0, 1] • `array_minority(array(0,1,42,42,43,42,42), 'real_minority')` → [43, 0, 1]

array_prepend

Returns an array with the given value added at the beginning.

语法	array_prepend(array, value)
参数	• **array** - an array • **value** - the value to add
Examples	• `array_prepend(array(1,2,3),0)` → [0, 1, 2, 3]

array_prioritize

Returns an array sorted using the ordering specified in another array. Values which are present in the first array but are missing from the second array will be added to the end of the result.

语法	array_prioritize(array, array_prioritize)
参数	• **array** - an array • **array_prioritize** - an array with values ordered by priority
Examples	• `array_prioritize(array(1, 8, 2, 5), array(5, 4, 2, 1, 3, 8))` → [5, 2, 1, 8] • `array_prioritize(array(5, 4, 2, 1, 3, 8), array(1, 8, 6, 5))` → [1, 8, 5, 4, 2, 3]

array_remove_all

Returns an array with all the entries of the given value removed.

语法	array_remove_all(array, value)
参数	• **array** - an array • **value** - the values to remove
Examples	• `array_remove_all(array('a','b','c','b'),'b')` → ['a', 'c']

array_remove_at

Returns an array with the item at the given index removed. Supports positive (0 for the first element) and negative (the last -Nth value, -1 for the last element) index.

语法	array_remove_at(array, pos)
参数	• **array** - an array • **pos** - the position to remove (0 based)
Examples	• `array_remove_at(array(1, 2, 3), 1)` → [1, 3] • `array_remove_at(array(1, 2, 3), -1)` → [1, 2]

array_replace

Returns an array with the supplied value, array, or map of values replaced.

Value & array variant

Returns an array with the supplied value or array of values replaced by another value or an array of values.

语法	array_replace(array, before, after)
参数	• **array** - the input array • **before** - the value or array of values to replace • **after** - the value or array of values to use as a replacement
Examples	• `array_replace(array('QGIS','SHOULD','ROCK'),'SHOULD', 'DOES')` → ['QGIS', 'DOES', 'ROCK'] • `array_replace(array(3,2,1),array(1,2,3),array(7,8,9))` → [9, 8, 7] • `array_replace(array('Q','G','I','S'),array('Q','S'),'-')` → ['-', 'G', 'I', '-']

Map variant

Returns an array with the supplied map keys replaced by their paired values.

语法	array_replace(array, map)
参数	• **array** - the input array • **map** - the map containing keys and values
Examples	• `array_replace(array('APP', 'SHOULD', 'ROCK'),map('APP', 'QGIS','SHOULD','DOES'))` → ['QGIS', 'DOES', 'ROCK']

Further reading: *replace*, *regexp_replace*

array_reverse

Returns the given array with array values in reversed order.

语法	array_reverse(array)
参数	• **array** - an array
Examples	• `array_reverse(array(2,4,0,10))` → [10, 0, 4, 2]

array_slice

Returns a portion of the array. The slice is defined by the start_pos and end_pos arguments.

语法	array_slice(array, start_pos, end_pos)
参数	• **array** - an array • **start_pos** - the index of the start position of the slice (0 based). The start_pos index is included in the slice. If you use a negative start_pos, the index is counted from the end of the list (-1 based). • **end_pos** - the index of the end position of the slice (0 based). The end_pos index is included in the slice. If you use a negative end_pos, the index is counted from the end of the list (-1 based).
Examples	• `array_slice(array(1,2,3,4,5),0,3)` → [1, 2, 3, 4] • `array_slice(array(1,2,3,4,5),0,-1)` → [1, 2, 3, 4, 5] • `array_slice(array(1,2,3,4,5),-5,-1)` → [1, 2, 3, 4, 5] • `array_slice(array(1,2,3,4,5),0,0)` → [1] • `array_slice(array(1,2,3,4,5),-2,-1)` → [4, 5] • `array_slice(array(1,2,3,4,5),-1,-1)` → [5] • `array_slice(array('Dufour','Valmiera','Chugiak', 'Brighton'),1,2)` → ['Valmiera', 'Chugiak'] • `array_slice(array('Dufour','Valmiera','Chugiak', 'Brighton'),-2,-1)` → ['Chugiak', 'Brighton']

array_sort

Returns the provided array with its elements sorted.

语法	array_sort(array, [ascending=true]) [] 标记可选参数
参数	• **array** - an array • **ascending** - set this parameter to false to sort the array in descending order
Examples	• `array_sort(array(3,2,1))` → [1, 2, 3]

array_sum

Returns the sum of arithmetic values in an array. Non numeric values in the array are ignored.

语法	array_sum(array)
参数	• **array** - an array
Examples	• `array_sum(array(0,1,39.4,1.6,'a'))` → 42.0

array_to_string

Concatenates array elements into a string separated by a delimiter and using optional string for empty values.

语法	array_to_string(array, [delimiter=','], [empty_value='']) [] 标记可选参数
参数	• **array** - the input array • **delimiter** - the string delimiter used to separate concatenated array elements • **empty_value** - the optional string to use as replacement for empty (zero length) matches
Examples	• `array_to_string(array('1','2','3'))` → '1,2,3' • `array_to_string(array(1,2,3),'-')` → '1-2-3' • `array_to_string(array('1','','3'),',','0')` → '1,0,3'

Further reading: *string_to_array*

generate_series

Creates an array containing a sequence of numbers.

语法	generate_series(start, stop, [step=1]) [] 标记可选参数
参数	• **start** - first value of the sequence • **stop** - value that ends the sequence once reached • **step** - value used as the increment between values
Examples	• `generate_series(1,5)` → [1, 2, 3, 4, 5] • `generate_series(5,1,-1)` → [5, 4, 3, 2, 1]

geometries_to_array

Splits a geometry into simpler geometries in an array.

语法	geometries_to_array(geometry)
参数	• **geometry** - the input geometry
Examples	• `geometries_to_array(geom_from_wkt('MultiPoint (1 2, 5 21)'))` → An array containing 'Point (1 2)' and 'Point (5 21)' • `geometries_to_array(geom_from_wkt('GeometryCollection (Polygon ((5 8, 4 1, 3 2, 5 8)),LineString (3 2, 4 2))'))` → an array of a polygon and a line geometries • `geom_to_wkt(geometries_to_array(geom_from_wkt('GeometryCollection (Polygon ((5 8, 4 1, 3 2, 5 8)),LineString (3 2, 4 2))'))[0])` → 'Polygon ((5 8, 4 1, 3 2, 5 8))' • `geometries_to_array(geom_from_wkt('MULTIPOLYGON(((5 5,0 0,0 10,5 5)),((5 5,10 10,10 0,5 5))'))` → an array of two polygon geometries

regexp_matches

Returns an array of all strings captured by capturing groups, in the order the groups themselves appear in the supplied regular expression against a string.

语法	regexp_matches(string, regex, [empty_value='']) [] 标记可选参数
参数	• **string** - the string to capture groups from against the regular expression • **regex** - the regular expression used to capture groups • **empty_value** - the optional string to use as replacement for empty (zero length) matches
Examples	• `regexp_matches('QGIS=>rocks','(.*)=>(.*)')` → ['QGIS', 'rocks'] • `regexp_matches('key=>','(.*)=>(.*)','empty value')` → ['key', 'empty value']

Further reading: *substr*, *regexp_substr*

string_to_array

Splits string into an array using supplied delimiter and optional string for empty values.

语法	string_to_array(string, [delimiter=','], [empty_value='']) [] 标记可选参数
参数	• **string** - the input string • **delimiter** - the string delimiter used to split the input string • **empty_value** - the optional string to use as replacement for empty (zero length) matches
Examples	• `string_to_array('1,2,3',',')` → ['1', '2', '3'] • `string_to_array('1,,3',',','0')` → ['1', '0', '3']

Further reading: *array_to_string*

9.2.3 Color Functions

This group contains functions for manipulating colors.

color_cmyk

Returns a string representation of a color based on its cyan, magenta, yellow and black components

语法	color_cmyk(cyan, magenta, yellow, black)
参数	• **cyan** - cyan component of the color, as a percentage integer value from 0 to 100 • **magenta** - magenta component of the color, as a percentage integer value from 0 to 100 • **yellow** - yellow component of the color, as a percentage integer value from 0 to 100 • **black** - black component of the color, as a percentage integer value from 0 to 100
Examples	• `color_cmyk(100,50,0,10)` → '0,115,230'

color_cmyka

Returns a string representation of a color based on its cyan, magenta, yellow, black and alpha (transparency) components

语法	color_cmyka(cyan, magenta, yellow, black, alpha)
参数	• **cyan** - cyan component of the color, as a percentage integer value from 0 to 100 • **magenta** - magenta component of the color, as a percentage integer value from 0 to 100 • **yellow** - yellow component of the color, as a percentage integer value from 0 to 100 • **black** - black component of the color, as a percentage integer value from 0 to 100 • **alpha** - alpha component as an integer value from 0 (completely transparent) to 255 (opaque).
Examples	• `color_cmyka(100,50,0,10,200)` → '0,115,230,200'

color_cmykf

Returns a color object based on its cyan, magenta, yellow, black and alpha components.

语法	color_cmykf(cyan, magenta, yellow, black, [alpha=1.0]) [] 标记可选参数
参数	• **cyan** - cyan component as a float value from 0.0 to 1.0 • **magenta** - magenta component as a float value from 0.0 to 1.0 • **yellow** - yellow component as a float value from 0.0 to 1.0 • **black** - black component as a float value from 0.0 to 1.0 • **alpha** - alpha component as a float value from 0.0 to 1.0
Examples	• `color_cmykf(1,0.9,0.81,0.62)` → CMYKA: 1.00,0.90,0.81,0.62,1.00

color_grayscale_average

Applies a grayscale filter to a color and returns it. Returned type is the same as color argument, i.e. a color string representation or a color object.

语法	color_grayscale_average(color)
参数	• **color** - a color string representation or a color object
Examples	• `color_grayscale_average('255,100,50')` → '135,135,135,255' • `color_grayscale_average(color_cmykf(0.6,0.5,0.1,0.8))` → CMYKA: 0.40,0.40,0.40,0.80,1.00

color_hsl

Returns a string representation of a color based on its hue, saturation, and lightness attributes.

语法	color_hsl(hue, saturation, lightness)
参数	• **hue** - hue of the color, as an integer value from 0 to 360 • **saturation** - saturation percentage of the color as an integer value from 0 to 100 • **lightness** - lightness percentage of the color as an integer value from 0 to 100
Examples	• `color_hsl(100,50,70)` → '166,217,140'

color_hsla

Returns a string representation of a color based on its hue, saturation, lightness and alpha (transparency) attributes

语法	color_hsla(hue, saturation, lightness, alpha)
参数	• **hue** - hue of the color, as an integer value from 0 to 360 • **saturation** - saturation percentage of the color as an integer value from 0 to 100 • **lightness** - lightness percentage of the color as an integer value from 0 to 100 • **alpha** - alpha component as an integer value from 0 (completely transparent) to 255 (opaque).
Examples	• `color_hsla(100,50,70,200)` → '166,217,140,200'

color_hslf

Returns a color object based on its hue, saturation, and lightness attributes.

语法	color_hslf(hue, saturation, lightness, [alpha=1.0]) [] 标记可选参数
参数	• **hue** - hue of the color, as a float value from 0.0 to 1.0 • **saturation** - saturation of the color as a float value from 0.0 to 1.0 • **lightness** - lightness of the color as a float value from 0.0 to 1.0 • **alpha** - alpha component as a float value from 0.0 to 1.0
Examples	• `color_hslf(0.3,0.52,0.7)` → HSLA: 0.30,0.52,0.70,1.00

color_hsv

Returns a string representation of a color based on its hue, saturation, and value attributes.

语法	color_hsv(hue, saturation, value)
参数	• **hue** - hue of the color, as an integer value from 0 to 360 • **saturation** - saturation percentage of the color as an integer value from 0 to 100 • **value** - value percentage of the color as an integer from 0 to 100
Examples	• `color_hsv(40,100,100)` → '255,170,0'

color_hsva

Returns a string representation of a color based on its hue, saturation, value and alpha (transparency) attributes.

语法	color_hsva(hue, saturation, value, alpha)
参数	• **hue** - hue of the color, as an integer value from 0 to 360 • **saturation** - saturation percentage of the color as an integer value from 0 to 100 • **value** - value percentage of the color as an integer from 0 to 100 • **alpha** - alpha component as an integer value from 0 (completely transparent) to 255 (opaque)
Examples	• `color_hsva(40,100,100,200)` → '255,170,0,200'

color_hsvf

Returns a color object based on its hue, saturation, and value attributes.

语法	color_hsvf(hue, saturation, value, [alpha=1.0]) [] 标记可选参数
参数	• **hue** - hue of the color, as a float value from 0.0 to 1.0 • **saturation** - saturation of the color as a float value from 0.0 to 1.0 • **value** - value of the color as as a float value from 0.0 to 1.0 • **alpha** - alpha component as a float value from 0.0 to 1.0
Examples	• `color_hsvf(0.4,1,0.6)` → HSVA: 0.40,1.00,0.60,1.00

color_mix

Returns a color mixing the red, green, blue, and alpha values of two provided colors based on a given ratio. Returned type is the same as color arguments, i.e. a color string representation or a color object.

语法	color_mix(color1, color2, ratio)
参数	• **color1** - a color string or a color object • **color2** - a color string or a color object • **ratio** - a ratio
Examples	• `color_mix_rgb('0,0,0','255,255,255',0.5)` → '127,127,127,255' • `color_mix(color_cmykf(0.9,0.9,0.9,0.9),color_cmykf(0.1,0.1,0.1,0.1),0.5)` → CMYKA: 0.50,0.50,0.50,0.50,1.00

color_mix_rgb

Returns a string representing a color mixing the red, green, blue, and alpha values of two provided colors based on a given ratio.

语法	color_mix_rgb(color1, color2, ratio)
参数	• **color1** - a color string • **color2** - a color string • **ratio** - a ratio
Examples	• `color_mix_rgb('0,0,0','255,255,255',0.5)` → '127,127,127,255'

color_part

Returns a specific component from a color string or color object, e.g., the red component or alpha component.

语法	color_part(color, component)
参数	• **color** - a color string or a color object • **component** - a string corresponding to the color component to return. Valid options are: – red: RGB red component (0-255) – green: RGB green component (0-255) – blue: RGB blue component (0-255) – alpha: alpha (transparency) value (0-255) – hue: HSV hue (0-360) – saturation: HSV saturation (0-100) – value: HSV value (0-100) – hsl_hue: HSL hue (0-360) – hsl_saturation: HSL saturation (0-100) – lightness: HSL lightness (0-100) – cyan: CMYK cyan component (0-100) – magenta: CMYK magenta component (0-100) – yellow: CMYK yellow component (0-100) – black: CMYK black component (0-100)
Examples	• `color_part('200,10,30','green')` → 10 • `to_int(color_part(color_cmykf(0.1,0.2,0.3,0.9),'black'))` → 90

color_rgb

Returns a string representation of a color based on its red, green, and blue components.

语法	color_rgb(red, green, blue)
参数	• **red** - red component as an integer value from 0 to 255 • **green** - green component as an integer value from 0 to 255 • **blue** - blue component as an integer value from 0 to 255
Examples	• `color_rgb(255,127,0)` → '255,127,0'

color_rgba

Returns a string representation of a color based on its red, green, blue, and alpha (transparency) components.

语法	color_rgba(red, green, blue, alpha)
参数	• **red** - red component as an integer value from 0 to 255 • **green** - green component as an integer value from 0 to 255 • **blue** - blue component as an integer value from 0 to 255 • **alpha** - alpha component as an integer value from 0 (completely transparent) to 255 (opaque).
Examples	• `color_rgba(255,127,0,200)` → '255,127,0,200'

color_rgbf

Returns a color object based on its red, green, blue and alpha components.

语法	color_rgbf(red, green, blue, [alpha=1.0]) [] 标记可选参数
参数	• **red** - red component as a float value from 0.0 to 1.0 • **green** - green component as a float value from 0.0 to 1.0 • **blue** - blue component as a float value from 0.0 to 1.0 • **alpha** - alpha component as a float value from 0.0 to 1.0
Examples	• `color_rgbf(1.0,0.5,0)` → RGBA: 1.00,0.50,0.00,1.00

create_ramp

Returns a gradient ramp from a map of color strings and steps.

语法	create_ramp(map, [discrete=false]) [] 标记可选参数
参数	• **map** - a map of color strings and steps • **discrete** - set this parameter to true to create a discrete color ramp
Examples	• `ramp_color(create_ramp(map(0,'0,0,0',1,'255,0,0')),1)` → '255,0,0,255'

darker

Returns a darker (or lighter) color. Returned type is the same as color arguments, i.e. a color string representation or a color object.

语法	darker(color, factor)
参数	• **color** - a color string or a color object • **factor** - an integer corresponding to the darkening factor: – if the factor is greater than 100, this function returns a darker color (e.g., setting factor to 200 returns a color that is half the brightness); – if the factor is less than 100, the return color is lighter, but using the lighter() function for this purpose is recommended; – if the factor is 0 or negative, the return value is unspecified.
Examples	• `darker('200,10,30', 200)` → '100,5,15,255'

Further reading: *lighter*

lighter

Returns a lighter (or darker) color. Returned type is the same as color arguments, i.e. a color string representation or a color object.

语法	lighter(color, factor)
参数	• **color** - a color string or a color object • **factor** - an integer corresponding to the lightening factor: – if the factor is greater than 100, this function returns a lighter color (e.g., setting factor to 150 returns a color that is 50% brighter); – if the factor is less than 100, the return color is darker, but using the darker() function for this purpose is recommended; – if the factor is 0 or negative, the return value is unspecified.
Examples	• `lighter('200,10,30', 200)` → '255,158,168,255'

Further reading: *darker*

project_color

Returns a color from the project's color scheme.

语法	project_color(name)
参数	• **name** - a color name
Examples	• `project_color('Logo color')` → '20,140,50'

Further reading: *setting project colors*

project_color_object

Returns a color from the project's color scheme. Contrary to project_color which returns a color string representation, project_color_object returns a color object.

语法	project_color_object(name)
参数	• **name** - a color name
Examples	• `project_color_object('Logo color')` → RGBA: 0.08,0.55,0.20,1.00

ramp_color

Returns a string representing a color from a color ramp.

Saved ramp variant

Returns a string representing a color from a saved ramp

语法	ramp_color(ramp_name, value)
参数	• **ramp_name** - the name of the color ramp as a string, for example 'Spectral' • **value** - the position on the ramp to select the color from as a real number between 0 and 1
Examples	• `ramp_color('Spectral',0.3)` → '253,190,115,255'

Note

The color ramps available vary between QGIS installations. This function may not give the expected results if you move your QGIS project between installations.

Expression-created ramp variant

Returns a string representing a color from an expression-created ramp

语法	ramp_color(ramp, value)
参数	• **ramp** - the color ramp • **value** - the position on the ramp to select the color from as a real number between 0 and 1
Examples	• `ramp_color(create_ramp(map(0,'0,0,0',1,'255,0,0')),1)` → '255,0,0,255'

Further reading: 设置颜色渐变, 颜色渐变下拉快捷方式

ramp_color_object

Returns a color object from a color ramp. Contrary to ramp_color which returns a color string representation, ramp_color_object returns a color object.

Saved ramp variant

Returns a color object from a saved ramp

语法	ramp_color_object(ramp_name, value)
参数	• **ramp_name** - the name of the color ramp as a string, for example 'Spectral' • **value** - the position on the ramp to select the color from as a real number between 0 and 1
Examples	• `ramp_color_object('Spectral',0.3)` → RGBA: 0.99,0.75,0.45,1.00

> **Note**
>
> The color ramps available vary between QGIS installations. This function may not give the expected results if you move your QGIS project between installations.

Expression-created ramp variant

Returns a color object from an expression-created ramp

语法	ramp_color_object(ramp, value)
参数	• **ramp** - the color ramp • **value** - the position on the ramp to select the color from as a real number between 0 and 1
Examples	• `ramp_color_object(create_ramp(map(0,color_rgbf(0,0,0),1, color_rgbf(1,0,0))),1)` → RGBA: 1.00,0.00,0.00,1.00

set_color_part

Sets a specific color component for a color string or a color object, e.g., the red component or alpha component.

语法	set_color_part(color, component, value)
参数	• **color** - a color string or a color object • **component** - a string corresponding to the color component to set. Valid options are: – red: RGB red component (0-255) – green: RGB green component (0-255) – blue: RGB blue component (0-255) – alpha: alpha (transparency) value (0-255) – hue: HSV hue (0-360) – saturation: HSV saturation (0-100) – value: HSV value (0-100) – hsl_hue: HSL hue (0-360) – hsl_saturation: HSL saturation (0-100) – lightness: HSL lightness (0-100) – cyan: CMYK cyan component (0-100) – magenta: CMYK magenta component (0-100) – yellow: CMYK yellow component (0-100) – black: CMYK black component (0-100) • **value** - new value for color component, respecting the ranges listed above
Examples	• `set_color_part('200,10,30','green',50)` → '200,50,30,255' • `set_color_part(color_cmykf(0.21,0,0.92,0.70),'black',100)` → CMYKA: 0.21,0.00,0.92,1.00,1.00

9.2.4 Conditional Functions

This group contains functions to handle conditional checks in expressions.

CASE

CASE is used to evaluate a series of conditions and return a result for the first condition met. The conditions are evaluated sequentially, and if a condition is true, the evaluation stops, and the corresponding result is returned. If none of the conditions are true, the value in the ELSE clause is returned. Furthermore, if no ELSE clause is set and none of the conditions are met, NULL is returned.

CASE

WHEN *condition* THEN *result*

[···n]

[ELSE *result*]

END

[ ] marks optional components

参数	• **WHEN condition** - A condition expression to evaluate • **THEN result** - If *condition* evaluates to True then *result* is evaluated and returned. • **ELSE result** - If none of the above conditions evaluated to True then *result* is evaluated and returned.
Examples	• `CASE WHEN "name" IS NULL THEN 'None' END` → Returns the string 'None' if the "name" field is NULL • `CASE WHEN $area > 10000 THEN 'Big property' WHEN $area > 5000 THEN 'Medium property' ELSE 'Small property' END` → Returns the string 'Big property' if the area is bigger than 10000, 'Medium property' if the area is between 5000 and 10000, and 'Small property' for others

coalesce

Returns the first non-NULL value from the expression list.

This function can take any number of arguments.

语法	coalesce(expression1, expression2, ...)
参数	• **expression** - any valid expression or value, regardless of type.
Examples	• `coalesce(NULL, 2)` → 2 • `coalesce(NULL, 2, 3)` → 2 • `coalesce(7, NULL, 3*2)` → 7 • `coalesce("fieldA", "fallbackField", 'ERROR')` → value of fieldA if it is non-NULL else the value of "fallbackField" or the string 'ERROR' if both are NULL

if

Tests a condition and returns a different result depending on the conditional check.

语法	if(condition, result_when_true, result_when_false)
参数	• **condition** - the condition which should be checked • **result_when_true** - the result which will be returned when the condition is true or another value that does not convert to false. • **result_when_false** - the result which will be returned when the condition is false or another value that converts to false like 0 or ". NULL will also be converted to false.
Examples	• `if( 1+1=2, 'Yes', 'No' )` → 'Yes' • `if( 1+1=3, 'Yes', 'No' )` → 'No' • `if( 5 > 3, 1, 0)` → 1 • `if( '', 'It is true (not empty)', 'It is false (empty)' )` → 'It is false (empty)' • `if( ' ', 'It is true (not empty)', 'It is false (empty)' )` → 'It is true (not empty)' • `if( 0, 'One', 'Zero' )` → 'Zero' • `if( 10, 'One', 'Zero' )` → 'One'

nullif

Returns a NULL value if value1 equals value2; otherwise it returns value1. This can be used to conditionally substitute values with NULL.

语法	nullif(value1, value2)
参数	• **value1** - The value that should either be used or substituted with NULL. • **value2** - The control value that will trigger the NULL substitution.
Examples	• `nullif('(none)', '(none)')` → NULL • `nullif('text', '(none)')` → 'text' • `nullif("name", '')` → NULL, if name is an empty string (or already NULL), the name in any other case.

regexp_match

Return the first matching position matching a regular expression within an unicode string, or 0 if the substring is not found.

语法	regexp_match(input_string, regex)
参数	• **input_string** - the string to test against the regular expression • **regex** - The regular expression to test against. Backslash characters must be double escaped (e.g., "\\s" to match a white space character or "\\b" to match a word boundary).
Examples	• `regexp_match('QGIS ROCKS','\\sROCKS')` → 5 • `regexp_match('Budač','udač\\b')` → 2

try

Tries an expression and returns its value if error-free. If the expression returns an error, an alternative value will be returned when provided otherwise the function will return NULL.

语法	try(expression, [alternative]) [] 标记可选参数
参数	• **expression** - the expression which should be run • **alternative** - the result which will be returned if the expression returns an error.
Examples	• `try( to_int( '1' ), 0 )` → 1 • `try( to_int( 'a' ), 0 )` → 0 • `try( to_date( 'invalid_date' ) )` → NULL

9.2.5 Conversions Functions

This group contains functions to convert one data type to another (e.g., string from/to integer, binary from/to string, string to date, ...).

from_base64

Decodes a string in the Base64 encoding into a binary value.

语法	from_base64(string)
参数	• **string** - the string to decode
Examples	• `from_base64('UUdJUw==')` → 'QGIS'

hash

Creates a hash from a string with a given method. One byte (8 bits) is represented with two hex "digits", so 'md4' (16 bytes) produces a 16 * 2 = 32 character long hex string and 'keccak_512' (64 bytes) produces a 64 * 2 = 128 character long hex string.

语法	hash(string, method)
参数	• **string** - the string to hash • **method** - The hash method among 'md4', 'md5', 'sha1', 'sha224', 'sha384', 'sha512', 'sha3_224', 'sha3_256', 'sha3_384', 'sha3_512', 'keccak_224', 'keccak_256', 'keccak_384', 'keccak_512'
Examples	• `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')` → '467f49a5039e7280d5d42fd433e80d203439e338eaabd701f0d6c17c • `hash('QGIS', 'sha3_256')` → '540f7354b6b8a6e735f2845250f15f4f3ba4f666c55574d9e9354575 • `hash('QGIS', 'sha3_384')` → '96052da1e77679e9a65f60d7ead961b28797782314478638 6eb4364 • `hash('QGIS', 'sha3_512')` → '900d079dc69761da113980253aa8ac0414a8bd6d09879a916228f87 • `hash('QGIS', 'keccak_224')` → '5b0ce6acef8b0a121d4ac4f3eaa8503c799ad4e26a3392d1fb2014 • `hash('QGIS', 'keccak_256')` → '991c520aa6815392de24087f61b2ae0fd56abbfeee4a8ca019c101 • `hash('QGIS', 'keccak_384')` → 'c57a3aed9d856fa04e5eeee9b62b6e027cca81ba574116d3cc1f0d

md5

Creates a md5 hash from a string.

语法	md5(string)
参数	• **string** - the string to hash
Examples	• `md5('QGIS')` → '57470aaa9e22adaefac7f5f342f1c6da'

sha256

Creates a sha256 hash from a string.

语法	sha256(string)
参数	• **string** - the string to hash
Examples	• `sha256('QGIS')` → 'eb045cba7a797aaa06ac58830846e40c8e8c780bc0676d3393605fae50c05309'

to_base64

Encodes a binary value into a string, using the Base64 encoding.

语法	to_base64(value)
参数	• **value** - the binary value to encode
Examples	• `to_base64('QGIS')` → 'UUdJUw=='

to_bool

Converts a given value to a boolean. The function will return false if the value is NULL, an empty string, an empty list, or 0.

语法	to_bool(value)
参数	• **value** - value to convert to boolean
Examples	• `to_bool('')` → false • `to_bool('123')` → true • `to_bool('false')` → true • `to_bool(0)` → false • `to_bool(1)` → true • `to_bool(null)` → false

to_date

Converts a string into a date object. An optional format string can be provided to parse the string; see QDate::fromString or the documentation of the format_date function for additional documentation on the format. By default the current QGIS user locale is used.

语法	to_date(string, [format], [language]) [] 标记可选参数
参数	• **string** - string representing a date value • **format** - format used to convert the string into a date • **language** - language (lowercase, two- or three-letter, ISO 639 language code) used to convert the string into a date. By default the current QGIS user locale is used.
Examples	• `to_date('2012-05-04')` → 2012-05-04 • `to_date('June 29, 2019','MMMM d, yyyy')` → 2019-06-29, if the current locale uses the name 'June' for the sixth month, otherwise an error occurs • `to_date('29 juin, 2019','d MMMM, yyyy','fr')` → 2019-06-29

Further reading: *format_date*

to_datetime

Converts a string into a datetime object. An optional format string can be provided to parse the string; see QDate::fromString, QTime::fromString or the documentation of the format_date function for additional documentation on the format. By default the current QGIS user locale is used.

语法	to_datetime(string, [format], [language]) [] 标记可选参数
参数	• **string** - string representing a datetime value • **format** - format used to convert the string into a datetime • **language** - language (lowercase, two- or three-letter, ISO 639 language code) used to convert the string into a datetime. By default the current QGIS user locale is used.
Examples	• `to_datetime('2012-05-04 12:50:00')` → 2012-05-04T12:50:00 • `to_datetime('June 29, 2019 @ 12:34','MMMM d, yyyy @ HH:mm')` → 2019-06-29T12:34, if the current locale uses the name 'June' for the sixth month, otherwise an error occurs • `to_datetime('29 juin, 2019 @ 12:34','d MMMM, yyyy @ HH:mm','fr')` → 2019-06-29T12:34

Further reading: *format_date*

to_decimal

Converts a degree, minute, second coordinate to its decimal equivalent.

语法	to_decimal(value)
参数	• **value** - A degree, minute, second string.
Examples	• `to_decimal('6°21\'16.445')` → 6.3545680555

to_dm

Converts a coordinate to degree, minute.

语法	to_dm(coordinate, axis, precision, [formatting=]) [] 标记可选参数
参数	• **coordinate** - A latitude or longitude value. • **axis** - The axis of the coordinate. Either 'x' or 'y'. • **precision** - Number of decimals. • **formatting** - Designates the formatting type. Acceptable values are NULL (default), 'aligned' or 'suffix'.
Examples	• `to_dm(6.1545681, 'x', 3)` → 6°9.274′ • `to_dm(6.1545681, 'y', 4, 'aligned')` → 6°09.2741′N • `to_dm(6.1545681, 'y', 4, 'suffix')` → 6°9.2741′N

to_dms

Converts a coordinate to degree, minute, second.

语法	to_dms(coordinate, axis, precision, [formatting=]) [] 标记可选参数
参数	• **coordinate** - A latitude or longitude value. • **axis** - The axis of the coordinate. Either 'x' or 'y'. • **precision** - Number of decimals. • **formatting** - Designates the formatting type. Acceptable values are NULL (default), 'aligned' or 'suffix'.
Examples	• `to_dms(6.1545681, 'x', 3)` → 6°9′16.445″ • `to_dms(6.1545681, 'y', 4, 'aligned')` → 6°09′16.4452″N • `to_dms(6.1545681, 'y', 4, 'suffix')` → 6°9′16.4452″N

to_int

Converts a string to integer number. If a value cannot be converted to integer the expression is invalid (e.g '123asd' is invalid).

语法	to_int(string)
参数	• **string** - string to convert to integer number
Examples	• `to_int('123')` → 123

to_interval

Converts a string to an interval type. Can be used to take days, hours, month, etc of a date.

语法	to_interval(string)
参数	• **string** - a string representing an interval. Allowable formats include {n} days {n} hours {n} months.
Examples	• `to_interval('1 day 2 hours')` → interval: 1.08333 days • `to_interval( '0.5 hours' )` → interval: 30 minutes • `to_datetime('2012-05-05 12:00:00') - to_interval('1 day 2 hours')` → 2012-05-04T10:00:00

to_real

Converts a string to a real number. If a value cannot be converted to real the expression is invalid (e.g '123.56asd' is invalid). Numbers are rounded after saving changes if the precision is smaller than the result of the conversion.

语法	to_real(string)
参数	• **string** - string to convert to real number
Examples	• `to_real('123.45')` → 123.45

to_string

Converts a number to string. The conversion is not locale-aware, see 'format_number' for a locale-aware alternative.

语法	to_string(number)
参数	• **number** - Integer or real value. The number to convert to string.
Examples	• `to_string(1.23)` → '1.23'

Further reading: *format_number*

to_time

Converts a string into a time object. An optional format string can be provided to parse the string; see QTime::fromString for additional documentation on the format.

语法	to_time(string, [format], [language]) [] 标记可选参数
参数	• **string** - string representing a time value • **format** - format used to convert the string into a time • **language** - language (lowercase, two- or three-letter, ISO 639 language code) used to convert the string into a time
Examples	• `to_time('12:30:01')` → 12:30:01 • `to_time('12:34','HH:mm')` → 12:34:00 • `to_time('12:34','HH:mm','fr')` → 12:34:00

Further reading: *format_date*

9.2.6 CRS Functions

This group contains functions to operate on coordinate reference system objects.

crs_from_text

Creates a coordinate reference system from a string definition. The string can represent an authority ID, a WKT definition, or a PROJ string definition of the CRS.

语法	crs_from_text(definition)
参数	• **definition** - CRS definition
Examples	• `crs_from_text('EPSG:3857')` → crs value 'EPSG:3857 - WGS 84 / Pseudo-Mercator' • `crs_from_text('PROJ:+proj=merc +a=6378137 +b=6378137 +lat_ts=0 +lon_0=0 +x_0=0 +y_0=0 +k=1 +units=m +nadgrids=@null +wktext +no_defs +type=crs')` → crs value 'EPSG:3857 - WGS 84 / Pseudo-Mercator'

crs_to_authid

Returns the authority ID string for a coordinate reference system.

语法	crs_to_authid(crs)
参数	• **crs** - crs value
Examples	• `crs_to_authid(crs_from_text('PROJ:+proj=merc +a=6378137 +b=6378137 +lat_ts=0 +lon_0=0 +x_0=0 +y_0=0 +k=1 +units=m +nadgrids=@null +wktext +no_defs +type=crs'))` → 'EPSG:3857'

9.2.7 Custom Functions

This group contains functions created by the user. See *Function Editor* for more details.

9.2.8 Date and Time Functions

This group contains functions for handling date and time data. This group shares several functions with the *Conversions Functions* (to_date, to_time, to_datetime, to_interval) and *String Functions* (format_date) groups.

Note

Storing date, datetime and intervals on fields

The ability to store *date*, *time* and *datetime* values directly on fields depends on the data source's provider (e.g., Shapefile accepts *date* format, but not *datetime* or *time* format). The following are some suggestions to overcome this limitation:

- *date*, *datetime* and *time* can be converted and stored in text type fields using the *format_date()* function.
- *Intervals* can be stored in integer or decimal type fields after using one of the date extraction functions (e.g., *day()* to get the interval expressed in days)

age

Returns the difference between two dates or datetimes.

The difference is returned as an `Interval` and needs to be used with one of the following functions in order to extract useful information:

- `year`
- `month`
- `week`
- `day`
- `hour`
- `minute`
- `second`

语法	age(datetime1, datetime2)
参数	• **datetime1** - a string, date or datetime representing the later date • **datetime2** - a string, date or datetime representing the earlier date
Examples	• `day(age('2012-05-12','2012-05-02'))` → 10 • `hour(age('2012-05-12','2012-05-02'))` → 240

datetime_from_epoch

Returns a datetime whose date and time are the number of milliseconds, msecs, that have passed since 1970-01-01T00:00:00.000, Coordinated Universal Time (Qt.UTC), and converted to Qt.LocalTime.

语法	datetime_from_epoch(int)
参数	• **int** - number (milliseconds)
Examples	• `datetime_from_epoch(1483225200000)` → 2017-01-01T00:00:00

day

Extracts the day from a date, or the number of days from an interval.

Date variant

Extracts the day from a date or datetime.

语法	day(date)
参数	• **date** - a date or datetime value
Examples	• `day('2012-05-12')` → 12

Interval variant

Calculates the length in days of an interval.

语法	day(interval)
参数	• **interval** - interval value to return number of days from
Examples	• `day(to_interval('3 days'))` → 3 • `day(to_interval('3 weeks 2 days'))` → 23 • `day(age('2012-01-01','2010-01-01'))` → 730

day_of_week

Returns the day of the week for a specified date or datetime. The returned value ranges from 0 to 6, where 0 corresponds to a Sunday and 6 to a Saturday.

语法	day_of_week(date)
参数	• **date** - date or datetime value
Examples	• `day_of_week(to_date('2015-09-21'))` → 1

epoch

Returns the interval in milliseconds between the unix epoch and a given date value.

语法	epoch(date)
参数	• **date** - a date or datetime value
Examples	• `epoch(to_date('2017-01-01'))` → 1483203600000

format_date

Formats a date type or string into a custom string format. Uses Qt date/time format strings. See QDateTime::toString.

语法

format_date(datetime, format, [language])
[] 标记可选参数

参数

- **datetime** - date, time or datetime value
- **format** - String template used to format the string.

Expression	Output
d	the day as number without a leading zero (1 to 31)
dd	the day as number with a leading zero (01 to 31)
ddd	the abbreviated localized day name (e.g. 'Mon' to 'Sun')
dddd	the long localized day name (e.g. 'Monday' to 'Sunday')
M	the month as number without a leading zero (1-12)
MM	the month as number with a leading zero (01-12)
MMM	the abbreviated localized month name (e.g. 'Jan' to 'Dec')
MMMM	the long localized month name (e.g. 'January' to 'December')
yy	the year as two digit number (00-99)
yyyy	the year as four digit number

These expressions may be used for the time part of the format string:

Expression	Output
h	the hour without a leading zero (0 to 23 or 1 to 12 if AM/PM display)
hh	the hour with a leading zero (00 to 23 or 01 to 12 if AM/PM display)
H	the hour without a leading zero (0 to 23, even with AM/PM display)
HH	the hour with a leading zero (00 to 23, even with AM/PM display)
m	the minute without a leading zero (0 to 59)
mm	the minute with a leading zero (00 to 59)
s	the second without a leading zero (0 to 59)
ss	the second with a leading zero (00 to 59)
z	the milliseconds without trailing zeroes (0 to 999)
zzz	the milliseconds with trailing zeroes (000 to 999)
AP or A	interpret as an AM/PM time. *AP* must be either 'AM' or 'PM'.
ap or a	Interpret as an AM/PM time. *ap* must be either 'am' or 'pm'.

- **language** - language (lowercase, two- or three-letter, ISO 639 language code) used to format the date into a custom string. By default the current QGIS user locale is used.

Examples

- `format_date('2012-05-15','dd.MM.yyyy')` → '15.05.2012'
- `format_date('2012-05-15','d MMMM yyyy','fr')` → '15 mai 2012'
- `format_date('2012-05-15','dddd')` → 'Tuesday', if the current locale is an English variant
- `format_date('2012-05-15 13:54:20','dd.MM.yy')` → '15.05.12'
- `format_date('13:54:20','hh:mm AP')` → '01:54 PM'

hour

Extracts the hour part from a datetime or time, or the number of hours from an interval.

Time variant

Extracts the hour part from a time or datetime.

语法	hour(datetime)
参数	• **datetime** - a time or datetime value
Examples	• `hour( to_datetime('2012-07-22 13:24:57') )` → 13

Interval variant

Calculates the length in hours of an interval.

语法	hour(interval)
参数	• **interval** - interval value to return number of hours from
Examples	• `hour(to_interval('3 hours'))` → 3 • `hour(age('2012-07-22T13:00:00','2012-07-22T10:00:00'))` → 3 • `hour(age('2012-01-01','2010-01-01'))` → 17520

make_date

Creates a date value from year, month and day numbers.

语法	make_date(year, month, day)
参数	• **year** - Year number. Years 1 to 99 are interpreted as is. Year 0 is invalid. • **month** - Month number, where 1=January • **day** - Day number, beginning with 1 for the first day in the month
Examples	• `make_date(2020,5,4)` → date value 2020-05-04

make_datetime

Creates a datetime value from year, month, day, hour, minute and second numbers.

语法	make_datetime(year, month, day, hour, minute, second)
参数	• **year** - Year number. Years 1 to 99 are interpreted as is. Year 0 is invalid. • **month** - Month number, where 1=January • **day** - Day number, beginning with 1 for the first day in the month • **hour** - Hour number • **minute** - Minutes • **second** - Seconds (fractional values include milliseconds)
Examples	• `make_datetime(2020,5,4,13,45,30.5)` → datetime value 2020-05-04 13:45:30.500

make_interval

Creates an interval value from year, month, weeks, days, hours, minute and seconds values.

语法	make_interval([years=0], [months=0], [weeks=0], [days=0], [hours=0], [minutes=0], [seconds=0]) [] 标记可选参数
参数	• **years** - Number of years (assumes a 365.25 day year length). • **months** - Number of months (assumes a 30 day month length) • **weeks** - Number of weeks • **days** - Number of days • **hours** - Number of hours • **minutes** - Number of minutes • **seconds** - Number of seconds
Examples	• `make_interval(hours:=3)` → interval: 3 hours • `make_interval(days:=2, hours:=3)` → interval: 2.125 days • `make_interval(minutes:=0.5, seconds:=5)` → interval: 35 seconds

make_time

Creates a time value from hour, minute and second numbers.

语法	make_time(hour, minute, second)
参数	• **hour** - Hour number • **minute** - Minutes • **second** - Seconds (fractional values include milliseconds)
Examples	• `make_time(13,45,30.5)` → time value 13:45:30.500

minute

Extracts the minutes part from a datetime or time, or the number of minutes from an interval.

Time variant

Extracts the minutes part from a time or datetime.

语法	minute(datetime)
参数	• **datetime** - a time or datetime value
Examples	• `minute( to_datetime('2012-07-22 13:24:57') )` → 24

Interval variant

Calculates the length in minutes of an interval.

语法	minute(interval)
参数	• **interval** - interval value to return number of minutes from
Examples	• `minute(to_interval('3 minutes'))` → 3 • `minute(age('2012-07-22T00:20:00','2012-07-22T00:00:00'))` → 20 • `minute(age('2012-01-01','2010-01-01'))` → 1051200

month

Extracts the month part from a date, or the number of months from an interval.

Date variant

Extracts the month part from a date or datetime.

语法	month(date)
参数	• **date** - a date or datetime value
Examples	• `month('2012-05-12')` → 05

Interval variant

Calculates the length in months of an interval.

语法	month(interval)
参数	• **interval** - interval value to return number of months from
Examples	• `month(to_interval('3 months'))` → 3 • `month(age('2012-01-01','2010-01-01'))` → 4.03333

now

Returns the current date and time. The function is static and will return consistent results while evaluating. The time returned is the time when the expression is prepared.

语法	now()
Examples	• `now()` → 2012-07-22T13:24:57

second

Extracts the seconds part from a datetime or time, or the number of seconds from an interval.

Time variant

Extracts the seconds part from a time or datetime.

语法	second(datetime)
参数	• **datetime** - a time or datetime value
Examples	• `second( to_datetime('2012-07-22 13:24:57') )` → 57

Interval variant

Calculates the length in seconds of an interval.

语法	second(interval)
参数	• **interval** - interval value to return number of seconds from
Examples	• `second(to_interval('3 minutes'))` → 180 • `second(age('2012-07-22T00:20:00','2012-07-22T00:00:00'))` → 1200 • `second(age('2012-01-01','2010-01-01'))` → 63072000

to_date

Converts a string into a date object. An optional format string can be provided to parse the string; see QDate::fromString or the documentation of the format_date function for additional documentation on the format. By default the current QGIS user locale is used.

语法	to_date(string, [format], [language]) [] 标记可选参数
参数	• **string** - string representing a date value • **format** - format used to convert the string into a date • **language** - language (lowercase, two- or three-letter, ISO 639 language code) used to convert the string into a date. By default the current QGIS user locale is used.
Examples	• `to_date('2012-05-04')` → 2012-05-04 • `to_date('June 29, 2019','MMMM d, yyyy')` → 2019-06-29, if the current locale uses the name 'June' for the sixth month, otherwise an error occurs • `to_date('29 juin, 2019','d MMMM, yyyy','fr')` → 2019-06-29

Further reading: *format_date*

to_datetime

Converts a string into a datetime object. An optional format string can be provided to parse the string; see QDate::fromString, QTime::fromString or the documentation of the format_date function for additional documentation on the format. By default the current QGIS user locale is used.

语法	to_datetime(string, [format], [language]) [] 标记可选参数
参数	• **string** - string representing a datetime value • **format** - format used to convert the string into a datetime • **language** - language (lowercase, two- or three-letter, ISO 639 language code) used to convert the string into a datetime. By default the current QGIS user locale is used.
Examples	• `to_datetime('2012-05-04 12:50:00')` → 2012-05-04T12:50:00 • `to_datetime('June 29, 2019 @ 12:34','MMMM d, yyyy @ HH:mm')` → 2019-06-29T12:34, if the current locale uses the name 'June' for the sixth month, otherwise an error occurs • `to_datetime('29 juin, 2019 @ 12:34','d MMMM, yyyy @ HH:mm','fr')` → 2019-06-29T12:34

Further reading: *format_date*

to_interval

Converts a string to an interval type. Can be used to take days, hours, month, etc of a date.

语法	to_interval(string)
参数	• **string** - a string representing an interval. Allowable formats include {n} days {n} hours {n} months.
Examples	• `to_interval('1 day 2 hours')` → interval: 1.08333 days • `to_interval( '0.5 hours' )` → interval: 30 minutes • `to_datetime('2012-05-05 12:00:00') - to_interval('1 day 2 hours')` → 2012-05-04T10:00:00

to_time

Converts a string into a time object. An optional format string can be provided to parse the string; see QTime::fromString for additional documentation on the format.

语法	to_time(string, [format], [language]) [] 标记可选参数
参数	• **string** - string representing a time value • **format** - format used to convert the string into a time • **language** - language (lowercase, two- or three-letter, ISO 639 language code) used to convert the string into a time
Examples	• `to_time('12:30:01')` → 12:30:01 • `to_time('12:34','HH:mm')` → 12:34:00 • `to_time('12:34','HH:mm','fr')` → 12:34:00

Further reading: *format_date*

week

Extracts the week number from a date, or the number of weeks from an interval.

Date variant

Extracts the week number from a date or datetime.

语法	week(date)
参数	• **date** - a date or datetime value
Examples	• `week('2012-05-12')` → 19

Interval variant

Calculates the length in weeks of an interval.

语法	week(interval)
参数	• **interval** - interval value to return number of months from
Examples	• `week(to_interval('3 weeks'))` → 3 • `week(age('2012-01-01','2010-01-01'))` → 104.285

year

Extracts the year part from a date, or the number of years from an interval.

Date variant

Extracts the year part from a date or datetime.

语法	year(date)
参数	• **date** - a date or datetime value
Examples	• `year('2012-05-12')` → 2012

Interval variant

Calculates the length in years of an interval.

语法	year(interval)
参数	• **interval** - interval value to return number of years from
Examples	• `year(to_interval('3 years'))` → 3 • `year(age('2012-01-01','2010-01-01'))` → 1.9986

Some examples:

Besides these functions, subtracting dates, datetimes or times using the − (minus) operator will return an interval.

Adding or subtracting an interval to dates, datetimes or times, using the + (plus) and − (minus) operators, will return a datetime.

- Get the number of days until QGIS 3.0 release:

```
to_date('2017-09-29') - to_date(now())
-- Returns <interval: 203 days>
```

- The same with time:

```
to_datetime('2017-09-29 12:00:00') - now()
-- Returns <interval: 202.49 days>
```

- Get the datetime of 100 days from now:

```
now() + to_interval('100 days')
-- Returns <datetime: 2017-06-18 01:00:00>
```

9.2.9 Fields and Values

Contains a list of fields from the active layer, and special values. Fields list includes the ones stored in the dataset, *virtual* and *auxiliary* ones as well as from *joins*.

Double-click a field name to have it added to your expression. You can also type the field name (preferably inside double quotes) or its *alias*.

To retrieve fields values to use in an expression, select the appropriate field and, in the shown widget, choose between *10 Samples* and *All Unique*. Requested values are then displayed and you can use the *Search* box at the top of the list to filter the result. Sample values can also be accessed via right-clicking on a field.

To add a value to the expression you are writing, double-click on it in the list. If the value is of a string type, it should be simple quoted, otherwise no quote is needed.

NULL

Equates to a NULL value.

语法	NULL
Examples	• `NULL` → a NULL value

Note

To test for NULL use an *IS NULL* or *IS NOT NULL* expression.

9.2.10 Files and Paths Functions

This group contains functions which manipulate file and path names.

base_file_name

Returns the base name of the file without the directory or file suffix.

语法	base_file_name(path)
参数	• **path** - a file path or a map layer value. If a map layer value is specified then the file source of the layer will be used.
Examples	• `base_file_name('/home/qgis/data/country_boundaries.shp')` → 'country_boundaries'

exif

Retrieves exif tag values from an image file.

语法	exif(path, [tag]) [] 标记可选参数
参数	• **path** - An image file path or a map layer value. If a map layer value is specified then the file source of the layer will be used. • **tag** - The tag to return. If empty, a map with all exif tag values will be returned.
Examples	• `exif('/my/photo.jpg','Exif.Image.Orientation')` → 0

file_exists

Returns TRUE if a file path exists.

语法	file_exists(path)
参数	• **path** - a file path or a map layer value. If a map layer value is specified then the file source of the layer will be used.
Examples	• `file_exists('/home/qgis/data/country_boundaries.shp')` → TRUE

file_name

Returns the name of a file (including the file extension), excluding the directory.

语法	file_name(path)
参数	• **path** - a file path or a map layer value. If a map layer value is specified then the file source of the layer will be used.
Examples	• `file_name('/home/qgis/data/country_boundaries.shp')` → 'country_boundaries.shp'

file_path

Returns the directory component of a file path. This does not include the file name.

语法	file_path(path)
参数	• **path** - a file path or a map layer value. If a map layer value is specified then the file source of the layer will be used.
Examples	• `file_path('/home/qgis/data/country_boundaries.shp')` → '/home/qgis/data'

file_size

Returns the size (in bytes) of a file.

语法	file_size(path)
参数	• **path** - a file path or a map layer value. If a map layer value is specified then the file source of the layer will be used.
Examples	• `file_size('/home/qgis/data/country_boundaries.geojson')` → 5674

file_suffix

Returns the file suffix (extension) from a file path.

语法	file_suffix(path)
参数	• **path** - a file path or a map layer value. If a map layer value is specified then the file source of the layer will be used.
Examples	• `file_suffix('/home/qgis/data/country_boundaries.shp')` → 'shp'

is_directory

Returns TRUE if a path corresponds to a directory.

语法	is_directory(path)
参数	• **path** - a file path or a map layer value. If a map layer value is specified then the file source of the layer will be used.
Examples	• `is_directory('/home/qgis/data/country_boundaries.shp')` → FALSE • `is_directory('/home/qgis/data/')` → TRUE

is_file

Returns TRUE if a path corresponds to a file.

语法	is_file(path)
参数	• **path** - a file path or a map layer value. If a map layer value is specified then the file source of the layer will be used.
Examples	• `is_file('/home/qgis/data/country_boundaries.shp')` → TRUE • `is_file('/home/qgis/data/')` → FALSE

9.2.11 Form Functions

This group contains functions that operate exclusively under the attribute form context. For example, in *field's widgets* settings.

current_parent_value

Only usable in an embedded form context, this function returns the current, unsaved value of a field in the parent form currently being edited. This will differ from the parent feature's actual attribute values for features which are currently being edited or have not yet been added to a parent layer. When used in a value-relation widget filter expression, this function should be wrapped into a 'coalesce()' that can retrieve the actual parent feature from the layer when the form is not used in an embedded context.

语法	current_parent_value(field_name)
参数	• **field_name** - a field name in the current parent form
Examples	• `current_parent_value( 'FIELD_NAME' )` → The current value of a field 'FIELD_NAME' in the parent form.

current_value

Returns the current, unsaved value of a field in the form or table row currently being edited. This will differ from the feature's actual attribute values for features which are currently being edited or have not yet been added to a layer.

语法	current_value(field_name)
参数	• **field_name** - a field name in the current form or table row
Examples	• `current_value( 'FIELD_NAME' )` → The current value of field 'FIELD_NAME'.

9.2.12 Fuzzy Matching Functions

This group contains functions for fuzzy comparisons between values.

hamming_distance

Returns the Hamming distance between two strings. This equates to the number of characters at corresponding positions within the input strings where the characters are different. The input strings must be the same length, and the comparison is case-sensitive.

语法	hamming_distance(string1, string2)
参数	• **string1** - a string • **string2** - a string
Examples	• `hamming_distance('abc','xec')` → 2 • `hamming_distance('abc','ABc')` → 2 • `hamming_distance(upper('abc'),upper('ABC'))` → 0 • `hamming_distance('abc','abcd')` → NULL

levenshtein

Returns the Levenshtein edit distance between two strings. This equates to the minimum number of character edits (insertions, deletions or substitutions) required to change one string to another.

The Levenshtein distance is a measure of the similarity between two strings. Smaller distances mean the strings are more similar, and larger distances indicate more different strings. The distance is case sensitive.

语法	levenshtein(string1, string2)
参数	• **string1** - a string • **string2** - a string
Examples	• `levenshtein('kittens','mitten')` → 2 • `levenshtein('Kitten','kitten')` → 1 • `levenshtein(upper('Kitten'),upper('kitten'))` → 0

longest_common_substring

Returns the longest common substring between two strings. This substring is the longest string that is a substring of the two input strings. For example, the longest common substring of ”ABABC” and ”BABCA” is ”BABC”. The substring is case sensitive.

语法	longest_common_substring(string1, string2)
参数	• **string1** - a string • **string2** - a string
Examples	• `longest_common_substring('ABABC','BABCA')` → ’BABC’ • `longest_common_substring('abcDeF','abcdef')` → ’abc’ • `longest_common_substring(upper('abcDeF'),upper('abcdex'))` → ’ABCDE’

soundex

Returns the Soundex representation of a string. Soundex is a phonetic matching algorithm, so strings with similar sounds should be represented by the same Soundex code.

语法	soundex(string)
参数	• **string** - a string
Examples	• `soundex('robert')` → 'R163' • `soundex('rupert')` → 'R163' • `soundex('rubin')` → 'R150'

9.2.13 General Functions

This group contains general assorted functions.

env

Gets an environment variable and returns its content as a string. If the variable is not found, NULL will be returned. This is handy to inject system specific configuration like drive letters or path prefixes. Definition of environment variables depends on the operating system, please check with your system administrator or the operating system documentation how this can be set.

语法	env(name)
参数	• **name** - The name of the environment variable which should be retrieved.
Examples	• `env( 'LANG' )` → 'en_US.UTF-8' • `env( 'MY_OWN_PREFIX_VAR' )` → 'Z:' • `env( 'I_DO_NOT_EXIST' )` → NULL

eval

Evaluates an expression which is passed in a string. Useful to expand dynamic parameters passed as context variables or fields.

语法	eval(expression)
参数	• **expression** - an expression string
Examples	• `eval('\'nice\'')` → 'nice' • `eval(@expression_var)` → [whatever the result of evaluating @expression_var might be…]

eval_template

Evaluates a template which is passed in a string. Useful to expand dynamic parameters passed as context variables or fields.

语法	eval_template(template)
参数	• **template** - a template string
Examples	• `eval_template('QGIS [% upper(\'rocks\') %]')` → QGIS ROCKS

is_layer_visible

Returns TRUE if a specified layer is visible.

语法	is_layer_visible(layer)
参数	• **图层** - 字符串, 表示图层名称或图层 ID
Examples	• `is_layer_visible('baseraster')` → TRUE

mime_type

Returns the mime type of the binary data.

语法	mime_type(bytes)
参数	• **bytes** - the binary data
Examples	• `mime_type('<html><body></body></html>')` → text/html • `mime_type(from_base64('R0lGODlhAQABAAAAACH5BAEKAAEALAAAAAABAAEAAAIAOw==` → image/gif

var

Returns the value stored within a specified variable.

语法	var(name)
参数	• **name** - a variable name
Examples	• `var('qgis_version')` → '2.12'

Further reading: List of default *variables*

with_variable

This function sets a variable for any expression code that will be provided as 3rd argument. This is only useful for complicated expressions, where the same calculated value needs to be used in different places.

语法	with_variable(name, value, expression)
参数	• **name** - the name of the variable to set • **value** - the value to set • **expression** - the expression for which the variable will be available
Examples	• `with_variable('my_sum', 1 + 2 + 3, @my_sum * 2 + @my_sum * 5)` → 42

9.2.14 Geometry Functions

This group contains functions that operate on geometry objects (e.g. buffer, transform, $area).

affine_transform

Returns the geometry after an affine transformation. Calculations are in the Spatial Reference System of this geometry. The operations are performed in a scale, rotation, translation order. If there is a Z or M offset but the coordinate is not present in the geometry, it will be added.

语法	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]) [] 标记可选参数
参数	• **geometry** - a geometry • **delta_x** - x-axis translation • **delta_y** - y-axis translation • **rotation_z** - rotation around z-axis in degrees counter-clockwise • **scale_x** - x-axis scale factor • **scale_y** - y-axis scale factor • **delta_z** - z-axis translation • **delta_m** - m-axis translation • **scale_z** - z-axis scale factor • **scale_m** - m-axis scale factor
Examples	• `geom_to_wkt(affine_transform(geom_from_wkt('LINESTRING(1 1, 2 2)'), 2, 2, 0, 1, 1))` → 'LineString (3 3, 4 4)' • `geom_to_wkt(affine_transform(geom_from_wkt('POLYGON((0 0, 0 3, 2 2, 0 0))'), 0, 0, -90, 1, 2))` → 'Polygon ((0 0, 6 0, 4 -2, 0 0))' • `geom_to_wkt(affine_transform(geom_from_wkt('POINT(3 1)'), 0, 0, 0, 1, 1, 5, 0))` → 'PointZ (3 1 5)'

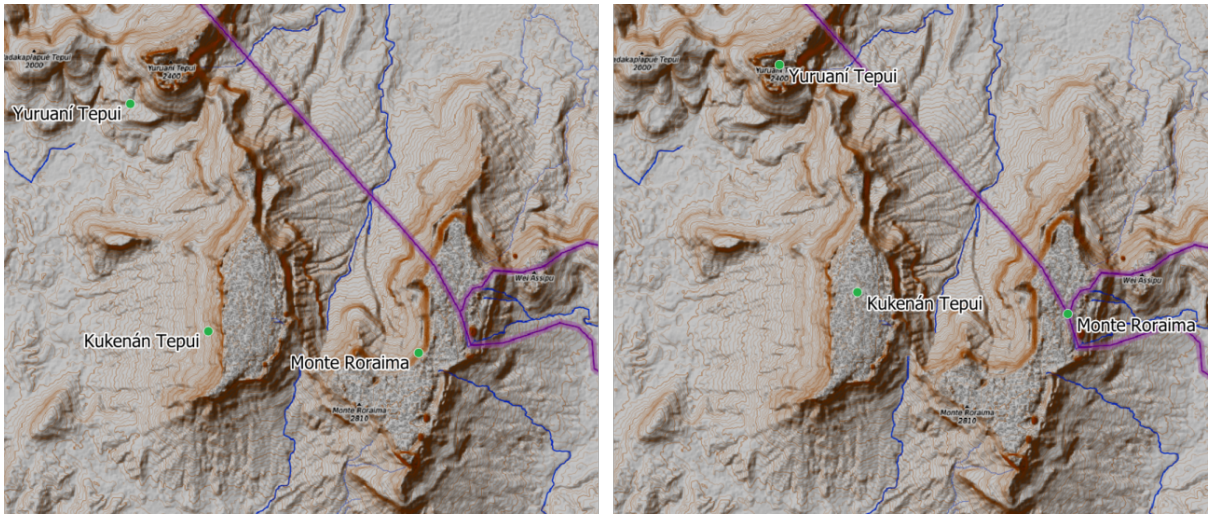


Fig. 9.4: Vector point layer (green dots) before (left), and after (right) an affine transformation (translation).

angle_at_vertex

Returns the bisector angle (average angle) to the geometry for a specified vertex on a linestring geometry. Angles are in degrees clockwise from north.

语法	angle_at_vertex(geometry, vertex)
参数	• **geometry** - a linestring geometry • **vertex** - vertex index, starting from 0; if the value is negative, the selected vertex index will be its total count minus the absolute value
Examples	• `angle_at_vertex(geometry:=geom_from_wkt('LineString(0 0, 10 0, 10 10)'),vertex:=1)` → 45.0

apply_dash_pattern

Applies a dash pattern to a geometry, returning a MultiLineString geometry which is the input geometry stroked along each line/ring with the specified pattern.

语法	apply_dash_pattern(geometry, pattern, [start_rule=no_rule], [end_rule=no_rule], [adjustment=both], [pattern_offset=0]) [] 标记可选参数
参数	• **geometry** - a geometry (accepts (multi)linestrings or (multi)polygons). • **pattern** - dash pattern, as an array of numbers representing dash and gap lengths. Must contain an even number of elements. • **start_rule** - optional rule for constraining the start of the pattern. Valid values are 'no_rule', 'full_dash', 'half_dash', 'full_gap', 'half_gap'. • **end_rule** - optional rule for constraining the end of the pattern. Valid values are 'no_rule', 'full_dash', 'half_dash', 'full_gap', 'half_gap'. • **adjustment** - optional rule for specifying which part of patterns are adjusted to fit the desired pattern rules. Valid values are 'both', 'dash', 'gap'. • **pattern_offset** - Optional distance specifying a specific distance along the pattern to commence at.
Examples	• `geom_to_wkt(apply_dash_pattern(geom_from_wkt('LINESTRING(1 1, 10 1)'), array(3, 1)))` → MultiLineString ((1 1, 4 1),(5 1, 8 1),(9 1, 10 1, 10 1)) • `geom_to_wkt(apply_dash_pattern(geom_from_wkt('LINESTRING(1 1, 10 1)'), array(3, 1), start_rule:='half_dash'))` → MultiLineString ((1 1, 2.5 1),(3.5 1, 6.5 1),(7.5 1, 10 1, 10 1))

$area

Returns the area of the current feature. The area calculated by this function respects both the current project's ellipsoid setting and area unit settings. For example, if an ellipsoid has been set for the project then the calculated area will be ellipsoidal, and if no ellipsoid is set then the calculated area will be planimetric.

语法	$area
Examples	• `$area` → 42

area

Returns the area of a geometry polygon object. Calculations are always planimetric in the Spatial Reference System (SRS) of this geometry, and the units of the returned area will match the units for the SRS. This differs from the calculations performed by the $area function, which will perform ellipsoidal calculations based on the project's ellipsoid and area unit settings.

语法	area(geometry)
参数	• **geometry** - polygon geometry object
Examples	• `area(geom_from_wkt('POLYGON((0 0, 4 0, 4 2, 0 2, 0 0))'))` → 8.0

azimuth

Returns the north-based azimuth as the angle in radians measured clockwise from the vertical on point_a to point_b.

语法	azimuth(point_a, point_b)
参数	• **point_a** - point geometry • **point_b** - point geometry
Examples	• `degrees( azimuth( make_point(25, 45), make_point(75, 100) ) )` → 42.273689 • `degrees( azimuth( make_point(75, 100), make_point(25,45) ) )` → 222.273689

bearing

Returns the north-based bearing as the angle in radians measured clockwise on the ellipsoid from the vertical on point_a to point_b.

语法	bearing(point_a, point_b, [source_crs], [ellipsoid]) [] 标记可选参数
参数	• **point_a** - point geometry • **point_b** - point geometry • **source_crs** - an optional string or CRS object representing the source CRS of the points. By default the current layer's CRS is used. • **ellipsoid** - an optional string representing the acronym or the authority ID (e.g., 'EPSG:7030') of the ellipsoid on which the bearing should be measured. By default the current project's ellipsoid setting is used.
Examples	• `degrees( bearing( make_point(16198544, -4534850), make_point(18736872, -1877769), 'EPSG:3857', 'EPSG:7030') )` → 49.980071 • `degrees( bearing( make_point(18736872, -1877769), make_point(16198544, -4534850), 'EPSG:3857', 'WGS84') )` → 219.282386

boundary

Returns the closure of the combinatorial boundary of the geometry (ie the topological boundary of the geometry). For instance, a polygon geometry will have a boundary consisting of the linestrings for each ring in the polygon. Some geometry types do not have a defined boundary, e.g., points or geometry collections, and will return NULL.

语法	boundary(geometry)
参数	• **geometry** - a geometry
Examples	• `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))'

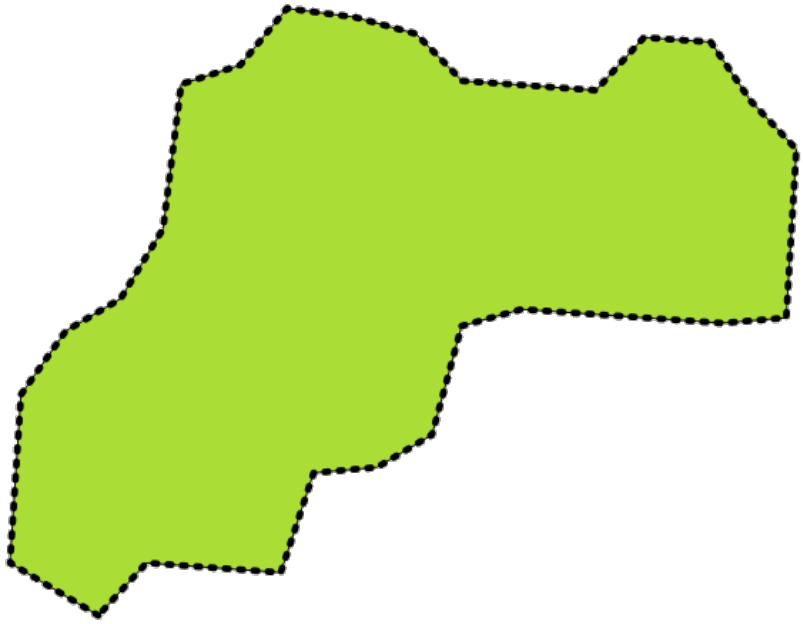


Fig. 9.5: Boundary (black dashed line) of the source polygon layer

Further reading: *Boundary* algorithm

bounds

Returns a geometry which represents the bounding box of an input geometry. Calculations are in the Spatial Reference System of this geometry.

语法	bounds(geometry)
参数	• **geometry** - a geometry
Examples	• `bounds(@geometry)` → bounding box of the current feature's geometry • `geom_to_wkt(bounds(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))')))` → 'Polygon ((-1 0, 1 0, 1 1, -1 1, -1 0))'

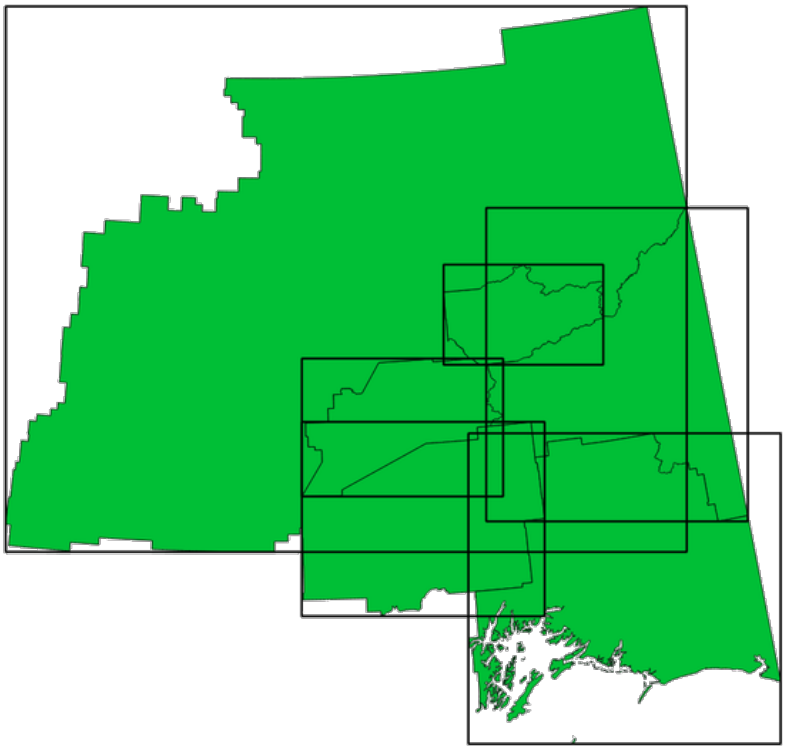


Fig. 9.6: Black lines represent the bounding boxes of each polygon feature

Further reading: *Bounding boxes* algorithm

bounds_height

Returns the height of the bounding box of a geometry. Calculations are in the Spatial Reference System of this geometry.

语法	bounds_height(geometry)
参数	• **geometry** - a geometry
Examples	• `bounds_height(@geometry)` → height of bounding box of the current feature's geometry • `bounds_height(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))'))` → 1

bounds_width

Returns the width of the bounding box of a geometry. Calculations are in the Spatial Reference System of this geometry.

语法	bounds_width(geometry)
参数	• **geometry** - a geometry
Examples	• `bounds_width(@geometry)` → width of bounding box of the current feature's geometry • `bounds_width(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))'))` → 2

buffer

Returns a geometry that represents all points whose distance from this geometry is less than or equal to distance. Calculations are in the Spatial Reference System of this geometry.

语法	buffer(geometry, distance, [segments=8], [cap='round'], [join='round'], [miter_limit=2]) [] 标记可选参数
参数	• **geometry** - a geometry • **distance** - buffer distance in layer units • **segments** - number of segments to use to represent a quarter circle when a round join style is used. A larger number results in a smoother buffer with more nodes. • **cap** - end cap style for buffer. Valid values are 'round', 'flat' or 'square' • **join** - join style for buffer. Valid values are 'round', 'bevel' or 'miter'. • **miter_limit** - miter distance limit, for use when the join style is set to 'miter'
Examples	• `buffer(@geometry, 10.5)` → polygon of the current feature's geometry buffered by 10.5 units

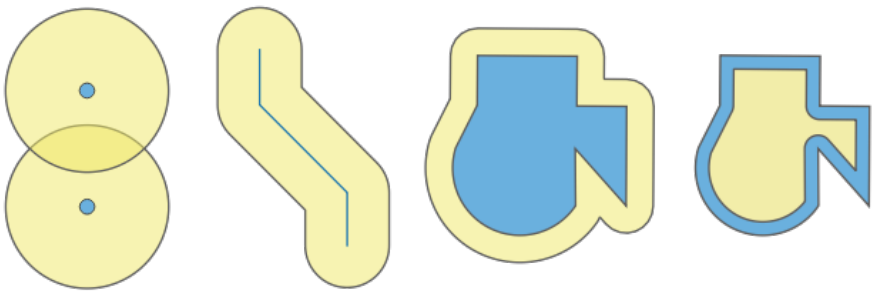


Fig. 9.7: Buffer (in yellow) of points, line, polygon with positive buffer, and polygon with negative buffer

Further reading: *Buffer* algorithm

buffer_by_m

Creates a buffer along a line geometry where the buffer diameter varies according to the m-values at the line vertices.

语法	buffer_by_m(geometry, [segments=8]) [] 标记可选参数
参数	• **geometry** - input geometry. Must be a (multi)line geometry with m values. • **segments** - number of segments to approximate quarter-circle curves in the buffer.
Examples	• `buffer_by_m(geometry:=geom_from_wkt('LINESTRINGM(1 2 0.5, 4 2 0.2)'),segments:=8)` → A variable width buffer starting with a diameter of 0.5 and ending with a diameter of 0.2 along the linestring geometry.

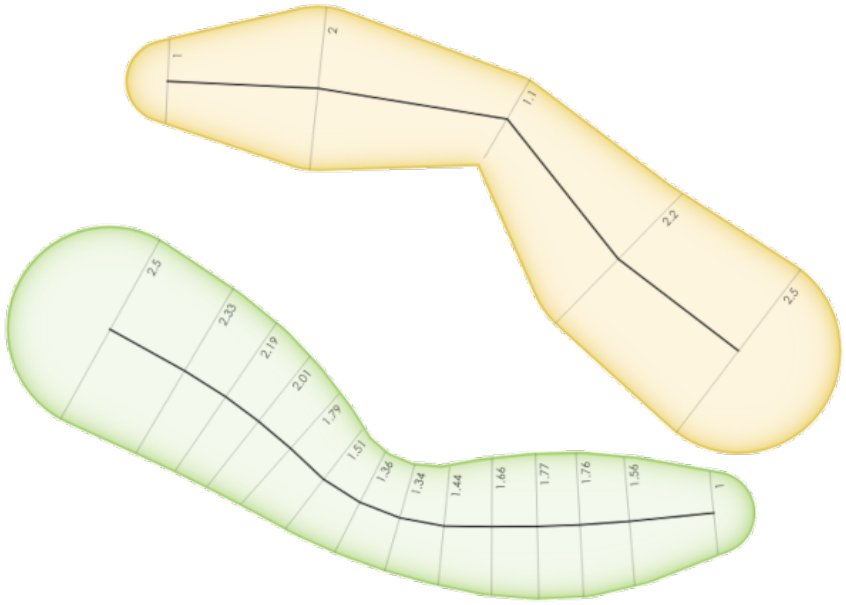


Fig. 9.8: Buffering line features using the m value on the vertices

Further reading: *Variable width buffer (by M value)* algorithm

centroid

Returns the geometric center of a geometry.

语法	centroid(geometry)
参数	• **geometry** - a geometry
Examples	• `centroid(@geometry)` → a point geometry

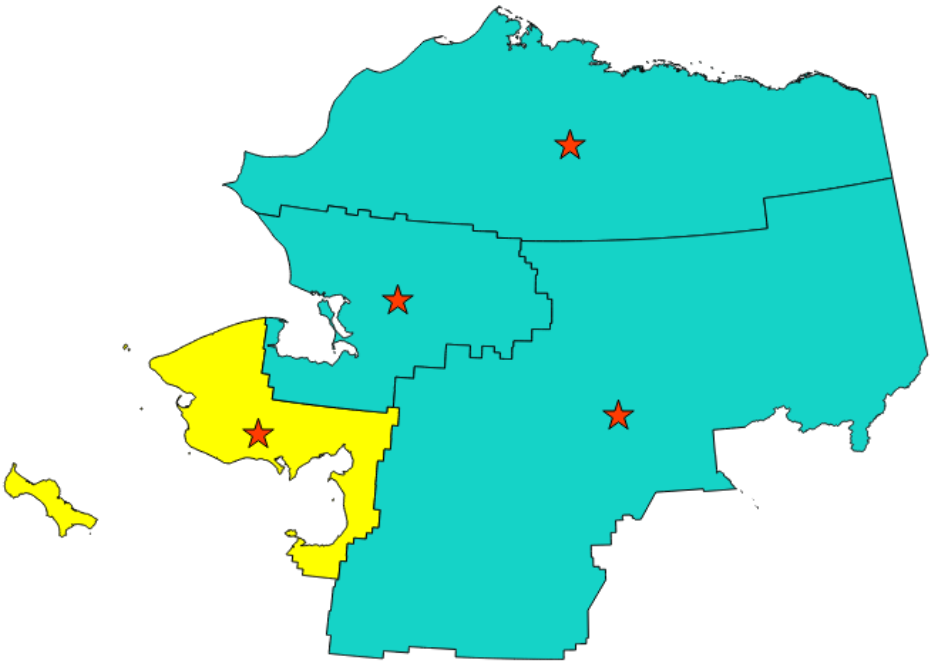


Fig. 9.9: The red stars represent the centroids of the features of the input layer.

Further reading: *Centroids* algorithm

close_line

Returns a closed line string of the input line string by appending the first point to the end of the line, if it is not already closed. If the geometry is not a line string or multi line string then the result will be NULL.

语法	close_line(geometry)
参数	• **geometry** - a line string geometry
Examples	• `geom_to_wkt(close_line(geom_from_wkt('LINESTRING(0 0, 1 0, 1 1)')))` → 'LineString (0 0, 1 0, 1 1, 0 0)' • `geom_to_wkt(close_line(geom_from_wkt('LINESTRING(0 0, 1 0, 1 1, 0 0)')))` → 'LineString (0 0, 1 0, 1 1, 0 0)'

closest_point

Returns the point on geometry1 that is closest to geometry2.

语法	closest_point(geometry1, geometry2)
参数	• **geometry1** - geometry to find closest point on • **geometry2** - geometry to find closest point to
Examples	• `geom_to_wkt(closest_point(geom_from_wkt('LINESTRING (20 80, 98 190, 110 180, 50 75 )'),geom_from_wkt('POINT(100 100)')))` → 'Point(73.0769 115.384)'

collect_geometries

Collects a set of geometries into a multi-part geometry object.

List of arguments variant

Geometry parts are specified as separate arguments to the function.

语法	collect_geometries(geometry1, geometry2, ...)
参数	• **geometry** - a geometry
Examples	• `geom_to_wkt(collect_geometries(make_point(1,2), make_point(3,4), make_point(5,6)))` → 'MultiPoint ((1 2),(3 4),(5 6))'

Array variant

Geometry parts are specified as an array of geometry parts.

语法	collect_geometries(array)
参数	• **array** - array of geometry objects
Examples	• `geom_to_wkt(collect_geometries(array(make_point(1,2), make_point(3,4), make_point(5,6))))` → 'MultiPoint ((1 2),(3 4),(5 6))'

Further reading: *Collect geometries* algorithm

combine

Returns the combination of two geometries.

语法	combine(geometry1, geometry2)
参数	• **geometry1** - a geometry • **geometry2** - a geometry
Examples	• `geom_to_wkt( combine( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 2 1)' ) ) )` → 'MULTILINESTRING((4 4, 2 1), (3 3, 4 4), (4 4, 5 5))' • `geom_to_wkt( combine( geom_from_wkt( 'LINESTRING(3 3, 4 4)' ), geom_from_wkt( 'LINESTRING(3 3, 6 6, 2 1)' ) ) )` → 'LINESTRING(3 3, 4 4, 6 6, 2 1)'

concave_hull

Returns a possibly concave polygon that contains all the points in the geometry

语法	concave_hull(geometry, target_percent, [allow_holes=False]) [] 标记可选参数
参数	• **geometry** - a geometry • **target_percent** - the percentage of area of the convex hull the solution tries to approach. A target_percent of 1 gives the same result as the convex hull. A target_percent between 0 and 0.99 produces a result that should have a smaller area than the convex hull. • **allow_holes** - optional argument specifying whether to allow holes within the output geometry. Defaults to FALSE, set to TRUE to allow including holes in the output geometry.
Examples	• `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))'

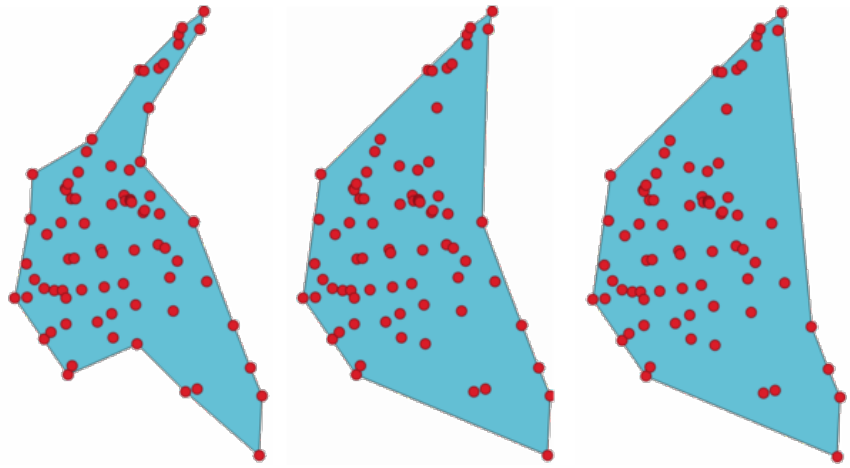


Fig. 9.10: Concave hulls with increasing target_percent parameter

Further reading: *convex_hull*, *Concave hull* algorithm

contains

Tests whether a geometry contains another. Returns TRUE if and only if no points of geometry2 lie in the exterior of geometry1, and at least one point of the interior of geometry2 lies in the interior of geometry1.

语法	contains(geometry1, geometry2)
参数	• **geometry1** - a geometry • **geometry2** - a geometry
Examples	• `contains( geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))' ), geom_from_wkt( 'POINT(0.5 0.5 )' ) )` → TRUE • `contains( geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → FALSE

Further reading: *overlay_contains*

convex_hull

Returns the convex hull of a geometry. It represents the minimum convex geometry that encloses all geometries within the set.

语法	convex_hull(geometry)
参数	• **geometry** - a geometry
Examples	• `geom_to_wkt( convex_hull( geom_from_wkt( 'LINESTRING(3 3, 4 4, 4 10)' ) ) )` → 'POLYGON((3 3, 4 10, 4 4, 3 3))'

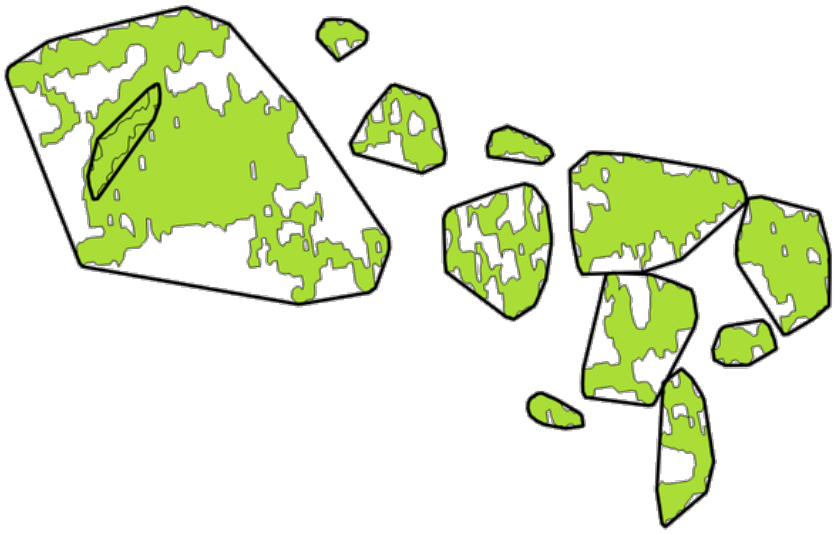


Fig. 9.11: Black lines identify the convex hull for each feature

Further reading: *concave_hull*, *Convex hull* algorithm

crosses

Tests whether a geometry crosses another. Returns TRUE if the supplied geometries have some, but not all, interior points in common.

语法	crosses(geometry1, geometry2)
参数	• **geometry1** - a geometry • **geometry2** - a geometry
Examples	• `crosses( geom_from_wkt( 'LINESTRING(3 5, 4 4, 5 3)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE • `crosses( geom_from_wkt( 'POINT(4 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → FALSE

Further reading: *overlay_crosses*

densify_by_count

Takes a polygon or line layer geometry and generates a new one in which the geometries have a larger number of vertices than the original one.

语法	densify_by_count(geometry, vertices)
参数	• **geometry** - a geometry (accepts (multi)linestrings or (multi)polygons). • **vertices** - number of vertices to add (per segment)
Examples	• `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)

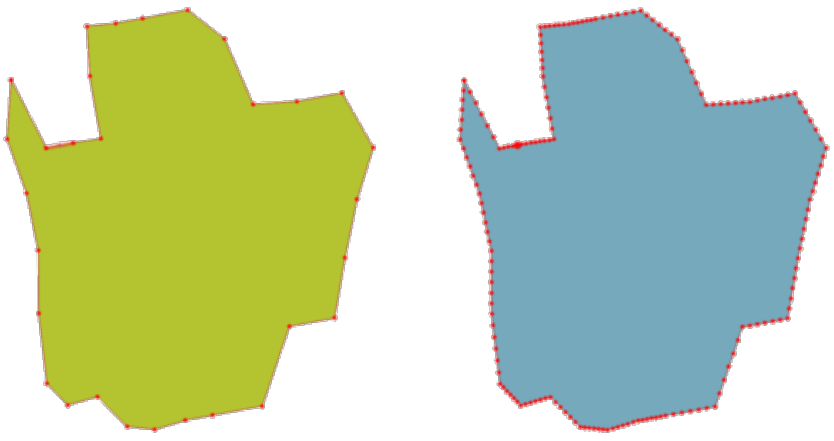


Fig. 9.12: Red points show the vertices before and after the densify

Further reading: *Densify by count* algorithm

densify_by_distance

Takes a polygon or line layer geometry and generates a new one in which the geometries are densified by adding additional vertices on edges that have a maximum distance of the specified interval distance.

语法	densify_by_distance(geometry, distance)
参数	• **geometry** - a geometry (accepts (multi)linestrings or (multi)polygons). • **distance** - maximum interval distance between vertices in output geometry
Examples	• `geom_to_wkt(densify_by_distance(geom_from_wkt('LINESTRING(1 1, 10 1)'), 4))` → LineString (1 1, 4 1, 7 1, 10 1)

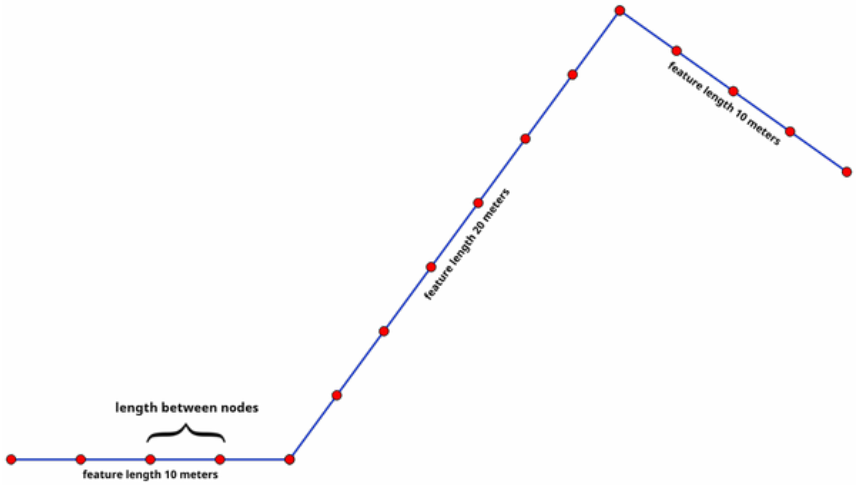


Fig. 9.13: Densify geometry at a given interval

Further reading: *Densify by interval* algorithm

difference

Returns a geometry that represents that part of geometry1 that does not intersect with geometry2.

语法	difference(geometry1, geometry2)
参数	• **geometry1** - a geometry • **geometry2** - a geometry
Examples	• `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)'

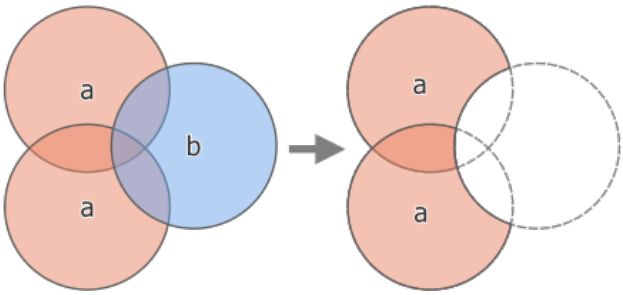


Fig. 9.14: Difference operation between a two-features input layer ‘a’ and a single feature overlay layer ‘b’ (left) - resulting in a new layer with the modified ‘a’ features (right)

Further reading: *Difference* algorithm

disjoint

Tests whether geometries do not spatially intersect. Returns TRUE if the geometries do not share any space together.

语法	disjoint(geometry1, geometry2)
参数	• **geometry1** - a geometry • **geometry2** - a geometry
Examples	• `disjoint( geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0 ))' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE • `disjoint( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'POINT(4 4)' ))` → FALSE

Further reading: *overlay_disjoint*

distance

Returns the minimum distance (based on spatial reference) between two geometries in projected units.

语法	distance(geometry1, geometry2)
参数	• **geometry1** - a geometry • **geometry2** - a geometry
Examples	• `distance( geom_from_wkt( 'POINT(4 4)' ), geom_from_wkt( 'POINT(4 8)' ) )` → 4

distance_to_vertex

Returns the distance along the geometry to a specified vertex.

语法	distance_to_vertex(geometry, vertex)
参数	• **geometry** - a linestring geometry • **vertex** - vertex index, starting from 0; if the value is negative, the selected vertex index will be its total count minus the absolute value
Examples	• `distance_to_vertex(geometry:=geom_from_wkt('LineString(0 0, 10 0, 10 10)'),vertex:=1)` → 10.0

end_point

Returns the last node from a geometry.

语法	end_point(geometry)
参数	• **geometry** - geometry object
Examples	• `geom_to_wkt(end_point(geom_from_wkt('LINESTRING(4 0, 4 2, 0 2)')))` → 'Point (0 2)'

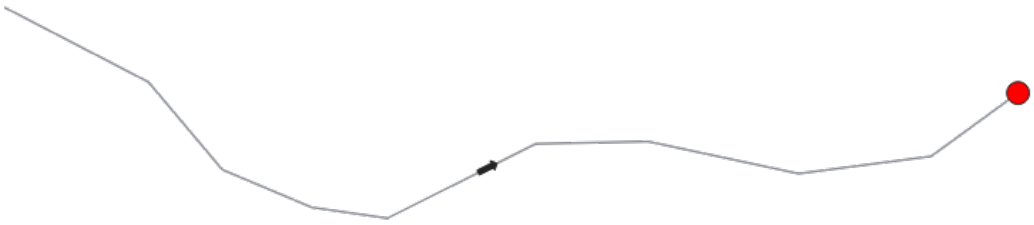


Fig. 9.15: End point of a line feature

Further reading: *start_point*, *Extract specific vertices* algorithm

exif_geotag

Creates a point geometry from the exif geotags of an image file.

语法	exif_geotag(path)
参数	• **path** - An image file path or a map layer value. If a map layer value is specified then the file source of the layer will be used.
Examples	• `geom_to_wkt(exif_geotag('/my/photo.jpg'))` → 'Point (2 4)'

extend

Extends the start and end of a linestring geometry by a specified amount. Lines are extended using the bearing of the first and last segment in the line. For a multilinestring, all the parts are extended. Distances are in the Spatial Reference System of this geometry.

语法	extend(geometry, start_distance, end_distance)
参数	• **geometry** - a (multi)linestring geometry • **start_distance** - distance to extend the start of the line • **end_distance** - distance to extend the end of the line.
Examples	• `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))'

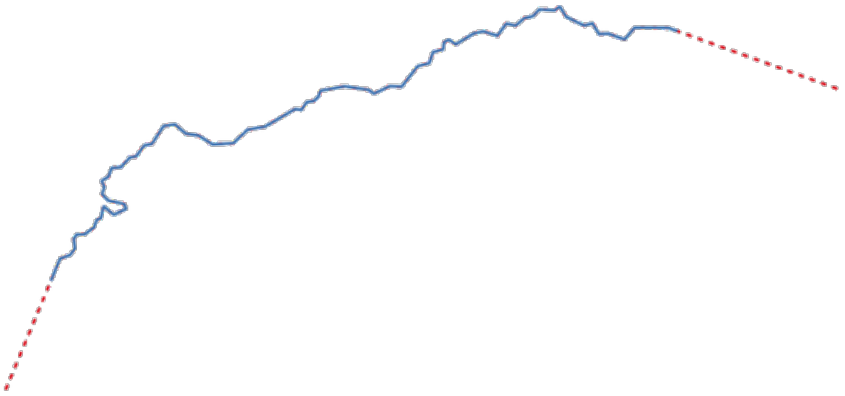


Fig. 9.16: The red dashes represent the initial and final extension of the original layer

Further reading: *Extend lines* algorithm

exterior_ring

Returns a line string representing the exterior ring of a polygon geometry. If the geometry is not a polygon then the result will be NULL.

语法	exterior_ring(geometry)
参数	• **geometry** - a polygon geometry
Examples	• `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)'

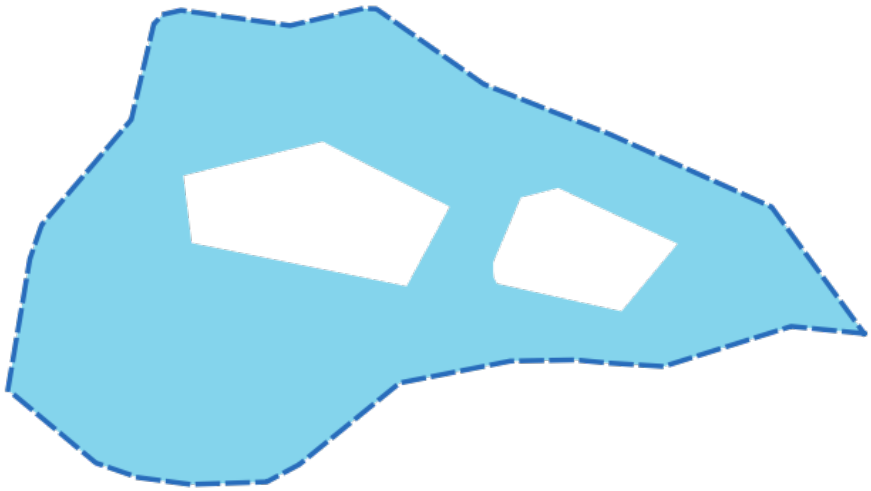


Fig. 9.17: The dashed line represents the exterior ring of the polygon

Further reading: *Boundary* algorithm, *interior_ring_n*

extrude

Returns an extruded version of the input (Multi-)Curve or (Multi-)Linestring geometry with an extension specified by x and y.

语法	extrude(geometry, x, y)
参数	• **geometry** - a curve or linestring geometry • **x** - x extension, numeric value • **y** - y extension, numeric value
Examples	• `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)))'

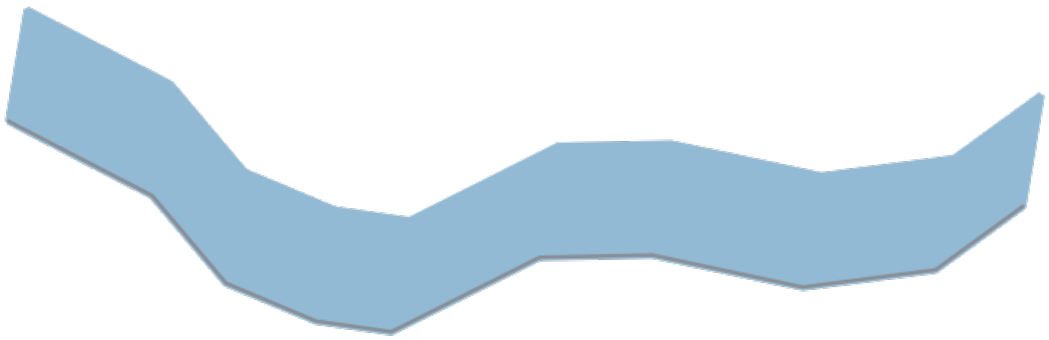


Fig. 9.18: Generating a polygon by extruding a line with offset in x and y directions

flip_coordinates

Returns a copy of the geometry with the x and y coordinates swapped. Useful for repairing geometries which have had their latitude and longitude values reversed.

语法	flip_coordinates(geometry)
参数	• **geometry** - a geometry
Examples	• `geom_to_wkt(flip_coordinates(make_point(1, 2)))` → 'Point (2 1)' • `geom_to_wkt(flip_coordinates(geom_from_wkt('LineString(0 2, 1 0, 1 6)')))` → 'LineString (2 0, 0 1, 6 1)'

Further reading: *Swap X and Y coordinates* algorithm

force_polygon_ccw

Forces a geometry to respect the convention where exterior rings are counter-clockwise, interior rings are clockwise.

语法	force_polygon_ccw(geometry)
参数	• **geometry** - a geometry. Any non-polygon geometries are returned unchanged.
Examples	• `geom_to_wkt(force_polygon_ccw(geometry:=geom_from_wkt('Polygon ((-1 -1, 0 2, 4 2, 4 0, -1 -1))')))` → 'Polygon ((-1 -1, 4 0, 4 2, 0 2, -1 -1))'

Further reading: *force_polygon_cw*, *force_rhr*

force_polygon_cw

Forces a geometry to respect the convention where exterior rings are clockwise, interior rings are counter-clockwise.

语法	force_polygon_cw(geometry)
参数	• **geometry** - a geometry. Any non-polygon geometries are returned unchanged.
Examples	• `geom_to_wkt(force_polygon_cw(geometry:=geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1))')))` → 'Polygon ((-1 -1, 0 2, 4 2, 4 0, -1 -1))'

Further reading: *force_polygon_ccw*, *force_rhr*

force_rhr

Forces a geometry to respect the Right-Hand-Rule, in which the area that is bounded by a polygon is to the right of the boundary. In particular, the exterior ring is oriented in a clockwise direction and the interior rings in a counter-clockwise direction. Due to the inconsistency in the definition of the Right-Hand-Rule in some contexts it is recommended to use the explicit force_polygon_cw function instead.

语法	force_rhr(geometry)
参数	• **geometry** - a geometry. Any non-polygon geometries are returned unchanged.
Examples	• `geom_to_wkt(force_rhr(geometry:=geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1))')))` → 'Polygon ((-1 -1, 0 2, 4 2, 4 0, -1 -1))'

Further reading: *Force right-hand-rule* algorithm, *force_polygon_ccw*, *force_polygon_cw*

geom_from_gml

Returns a geometry from a GML representation of geometry.

语法	geom_from_gml(gml)
参数	• **gml** - GML representation of a geometry as a string
Examples	• `geom_from_gml('<gml:LineString srsName="EPSG:4326"><gml:coordinates>4, 4 5,5 6,6</gml:coordinates></gml:LineString>')` → a line geometry object

geom_from_wkb

Returns a geometry created from a Well-Known Binary (WKB) representation.

语法	geom_from_wkb(binary)
参数	• **binary** - Well-Known Binary (WKB) representation of a geometry (as a binary blob)
Examples	• `geom_from_wkb( geom_to_wkb( make_point(4,5) ) )` → a point geometry object

geom_from_wkt

Returns a geometry created from a Well-Known Text (WKT) representation.

语法	geom_from_wkt(text)
参数	• **text** - Well-Known Text (WKT) representation of a geometry
Examples	• `geom_from_wkt( 'POINT(4 5)' )` → a geometry object

geom_to_wkb

Returns the Well-Known Binary (WKB) representation of a geometry

语法	geom_to_wkb(geometry)
参数	• **geometry** - a geometry
Examples	• `geom_to_wkb( @geometry )` → binary blob containing a geometry object

geom_to_wkt

Returns the Well-Known Text (WKT) representation of the geometry without SRID metadata.

语法	geom_to_wkt(geometry, [precision=8]) [] 标记可选参数
参数	• **geometry** - a geometry • **precision** - numeric precision
Examples	• `geom_to_wkt( make_point(6, 50) )` → 'POINT(6 50)' • `geom_to_wkt(centroid(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))')))` → 'POINT(0 0.66666667)' • `geom_to_wkt(centroid(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))')), 2)` → 'POINT(0 0.67)'

$geometry

Returns the geometry of the current feature. Can be used for processing with other functions. **WARNING: This function is deprecated. It is recommended to use the replacement @geometry variable instead.**

语法	$geometry
Examples	• `geom_to_wkt( $geometry )` → 'POINT(6 50)'

geometry

Returns a feature's geometry.

语法	geometry(feature)
参数	• **feature** - a feature object
Examples	• `geometry( @feature )` → the geometry of the current feature. Prefer using @geometry. • `geom_to_wkt( geometry( get_feature_by_id( 'streets', 1 ) ) )` → the geometry in WKT of the feature with the id 1 on the layer "streets", e.g. 'POINT(6 50)' • `intersects( @geometry, geometry( get_feature( 'streets', 'name', 'Main St.' ) ) )` → TRUE if the current feature spatially intersects the 'Main St.' named feature in the "streets" layer

geometry_n

Returns a specific geometry from a geometry collection, or NULL if the input geometry is not a collection. Also returns a part from a multipart geometry.

语法	geometry_n(geometry, index)
参数	• **geometry** - geometry collection • **index** - index of geometry to return, where 1 is the first geometry in the collection
Examples	• `geom_to_wkt(geometry_n(geom_from_wkt('GEOMETRYCOLLECTION(POINT(0 1), POINT(0 0), POINT(1 0), POINT(1 1))'),3))` → 'Point (1 0)'

geometry_type

Returns a string value describing the type of a geometry (Point, Line or Polygon)

语法	geometry_type(geometry)
参数	• **geometry** - a geometry
Examples	• `geometry_type( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 'Line' • `geometry_type( geom_from_wkt( 'MULTILINESTRING((2 5, 3 6, 4 8), (1 1, 0 0))') )` → 'Line' • `geometry_type( geom_from_wkt( 'POINT(2 5)') )` → 'Point' • `geometry_type( geom_from_wkt( 'POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1))') )` → 'Polygon'

hausdorff_distance

Returns the Hausdorff distance between two geometries. This is basically a measure of how similar or dissimilar 2 geometries are, with a lower distance indicating more similar geometries.

The function can be executed with an optional densify fraction argument. If not specified, an approximation to the standard Hausdorff distance is used. This approximation is exact or close enough for a large subset of useful cases. Examples of these are:

- computing distance between Linestrings that are roughly parallel to each other, and roughly equal in length. This occurs in matching linear networks.
- Testing similarity of geometries.

If the default approximate provided by this method is insufficient, specify the optional densify fraction argument. Specifying this argument performs a segment densification before computing the discrete Hausdorff distance. The parameter sets the fraction by which to densify each segment. Each segment will be split into a number of equal-length subsegments, whose fraction of the total length is closest to the given fraction. Decreasing the densify fraction parameter will make the distance returned approach the true Hausdorff distance for the geometries.

语法	hausdorff_distance(geometry1, geometry2, [densify_fraction]) [] 标记可选参数
参数	• **geometry1** - a geometry • **geometry2** - a geometry • **densify_fraction** - densify fraction amount
Examples	• `hausdorff_distance( geometry1:= geom_from_wkt('LINESTRING (0 0, 2 1)'),geometry2:=geom_from_wkt('LINESTRING (0 0, 2 0)'))` → 2 • `hausdorff_distance( geom_from_wkt('LINESTRING (130 0, 0 0, 0 150)'),geom_from_wkt('LINESTRING (10 10, 10 150, 130 10)'))` → 14.142135623 • `hausdorff_distance( geom_from_wkt('LINESTRING (130 0, 0 0, 0 150)'),geom_from_wkt('LINESTRING (10 10, 10 150, 130 10)'),0.5)` → 70.0

inclination

Returns the inclination measured from the zenith (0) to the nadir (180) on point_a to point_b.

语法	inclination(point_a, point_b)
参数	• **point_a** - point geometry • **point_b** - point geometry
Examples	• `inclination( make_point( 5, 10, 0 ), make_point( 5, 10, 5 ) )` → 0.0 • `inclination( make_point( 5, 10, 0 ), make_point( 5, 10, 0 ) )` → 90.0 • `inclination( make_point( 5, 10, 0 ), make_point( 50, 100, 0 ) )` → 90.0 • `inclination( make_point( 5, 10, 0 ), make_point( 5, 10, -5 ) )` → 180.0

interior_ring_n

Returns a specific interior ring from a polygon geometry, or NULL if the geometry is not a polygon.

语法	interior_ring_n(geometry, index)
参数	• **geometry** - polygon geometry • **index** - index of interior to return, where 1 is the first interior ring
Examples	• `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)'

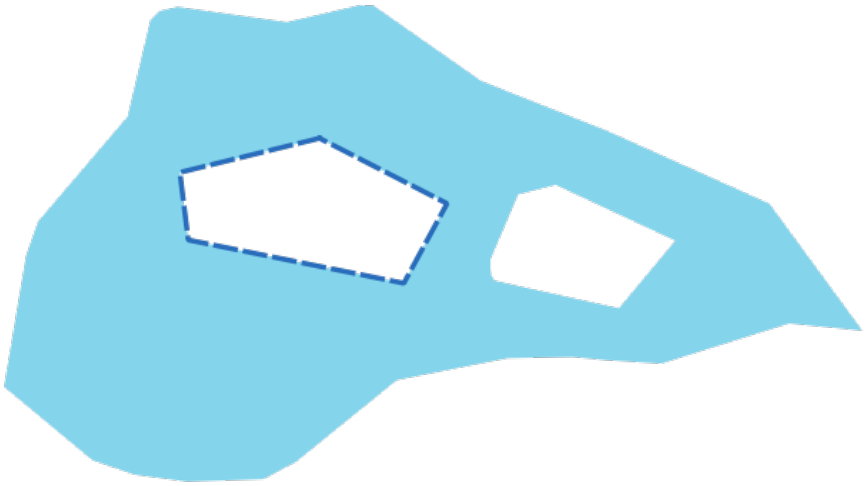


Fig. 9.19: The dashed line represents a specific interior ring of the polygon

Further reading: *exterior_ring*

intersection

Returns a geometry that represents the shared portion of two geometries.

语法	intersection(geometry1, geometry2)
参数	• **geometry1** - a geometry • **geometry2** - a geometry
Examples	• `geom_to_wkt( intersection( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4)' ) ) )` → 'LINESTRING(3 3, 4 4)' • `geom_to_wkt( intersection( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'MULTIPOINT(3.5 3.5, 4 5)' ) ) )` → 'POINT(3.5 3.5)'

Further reading: *Intersection* algorithm

intersects

Tests whether a geometry intersects another. Returns TRUE if the geometries spatially intersect (share any portion of space) and false if they do not.

语法	intersects(geometry1, geometry2)
参数	• **geometry1** - a geometry • **geometry2** - a geometry
Examples	• `intersects( geom_from_wkt( 'POINT(4 4)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE • `intersects( geom_from_wkt( 'POINT(4 5)' ), geom_from_wkt( 'POINT(5 5)' ) )` → FALSE

Further reading: *overlay_intersects*

intersects_bbox

Tests whether a geometry's bounding box overlaps another geometry's bounding box. Returns TRUE if the geometries spatially intersect the bounding box defined and false if they do not.

语法	intersects_bbox(geometry1, geometry2)
参数	• **geometry1** - a geometry • **geometry2** - a geometry
Examples	• `intersects_bbox( geom_from_wkt( 'POINT(4 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE • `intersects_bbox( geom_from_wkt( 'POINT(6 5)' ), geom_from_wkt( 'POLYGON((3 3, 4 4, 5 5, 3 3))' ) )` → FALSE

is_closed

Returns TRUE if a line string is closed (start and end points are coincident), or false if a line string is not closed. If the geometry is not a line string then the result will be NULL.

语法	is_closed(geometry)
参数	• **geometry** - a line string geometry
Examples	• `is_closed(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)'))` → FALSE • `is_closed(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2, 0 0)'))` → TRUE

is_empty

Returns TRUE if a geometry is empty (without coordinates), false if the geometry is not empty and NULL if there is no geometry. See also is_empty_or_null.

语法	is_empty(geometry)
参数	• **geometry** - a geometry
Examples	• `is_empty(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)'))` → FALSE • `is_empty(geom_from_wkt('LINESTRING EMPTY'))` → TRUE • `is_empty(geom_from_wkt('POINT(7 4)'))` → FALSE • `is_empty(geom_from_wkt('POINT EMPTY'))` → TRUE

Further reading: *is_empty_or_null*

is_empty_or_null

Returns TRUE if a geometry is NULL or empty (without coordinates) or false otherwise. This function is like the expression '@geometry IS NULL or is_empty(@geometry)'

语法	is_empty_or_null(geometry)
参数	• **geometry** - a geometry
Examples	• `is_empty_or_null(NULL)` → TRUE • `is_empty_or_null(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)'))` → FALSE • `is_empty_or_null(geom_from_wkt('LINESTRING EMPTY'))` → TRUE • `is_empty_or_null(geom_from_wkt('POINT(7 4)'))` → FALSE • `is_empty_or_null(geom_from_wkt('POINT EMPTY'))` → TRUE

Further reading: *is_empty*, *NULL*

is_multipart

Returns TRUE if the geometry is of Multi type.

语法	is_multipart(geometry)
参数	• **geometry** - a geometry
Examples	• `is_multipart(geom_from_wkt('MULTIPOINT ((0 0),(1 1),(2 2))'))` → TRUE • `is_multipart(geom_from_wkt('POINT (0 0)'))` → FALSE

is_valid

Returns TRUE if a geometry is valid; if it is well-formed in 2D according to the OGC rules.

语法	is_valid(geometry)
参数	• **geometry** - a geometry
Examples	• `is_valid(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2, 0 0)'))` → TRUE • `is_valid(geom_from_wkt('LINESTRING(0 0)'))` → FALSE

Further reading: *make_valid*, *Check validity* algorithm

$length

Returns the length of a linestring. If you need the length of a border of a polygon, use $perimeter instead. The length calculated by this function respects both the current project's ellipsoid setting and distance unit settings. For example, if an ellipsoid has been set for the project then the calculated length will be ellipsoidal, and if no ellipsoid is set then the calculated length will be planimetric.

语法	$length
Examples	• `$length` → 42.4711

length

Returns the number of characters in a string or the length of a geometry linestring.

String variant

Returns the number of characters in a string.

语法	length(string)
参数	• **string** - string to count length of
Examples	• `length('hello')` → 5

Geometry variant

Calculate the length of a geometry line object. Calculations are always planimetric in the Spatial Reference System (SRS) of this geometry, and the units of the returned length will match the units for the SRS. This differs from the calculations performed by the $length function, which will perform ellipsoidal calculations based on the project's ellipsoid and distance unit settings.

语法	length(geometry)
参数	• **geometry** - line geometry object
Examples	• `length(geom_from_wkt('LINESTRING(0 0, 4 0)'))` → 4.0

Further reading: *straight_distance_2d*

length3D

Calculates the 3D length of a geometry line object. If the geometry is not a 3D line object, it returns its 2D length. Calculations are always planimetric in the Spatial Reference System (SRS) of this geometry, and the units of the returned length will match the units for the SRS. This differs from the calculations performed by the $length function, which will perform ellipsoidal calculations based on the project's ellipsoid and distance unit settings.

语法	length3D(geometry)
参数	• **geometry** - line geometry object
Examples	• `length3D(geom_from_wkt('LINESTRINGZ(0 0 0, 3 0 4)'))` → 5.0

line_interpolate_angle

Returns the angle parallel to the geometry at a specified distance along a linestring geometry. Angles are in degrees clockwise from north.

语法	line_interpolate_angle(geometry, distance)
参数	• **geometry** - a linestring geometry • **distance** - distance along line to interpolate angle at
Examples	• `line_interpolate_angle(geometry:=geom_from_wkt('LineString(0 0, 10 0)'),distance:=5)` → 90.0

line_interpolate_point

Returns the point interpolated by a specified distance along a linestring geometry.

语法	line_interpolate_point(geometry, distance)
参数	• **geometry** - a linestring geometry • **distance** - distance along line to interpolate
Examples	• `geom_to_wkt(line_interpolate_point(geometry:=geom_from_wkt('LineString(0 0, 8 0)'), distance:=5))` → 'Point (5 0)' • `geom_to_wkt(line_interpolate_point(geometry:=geom_from_wkt('LineString(0 0, 1 1, 2 0)'), distance:=2.1))` → 'Point (1.48492424 0.51507576)' • `geom_to_wkt(line_interpolate_point(geometry:=geom_from_wkt('LineString(0 0, 1 0)'), distance:=2))` → NULL

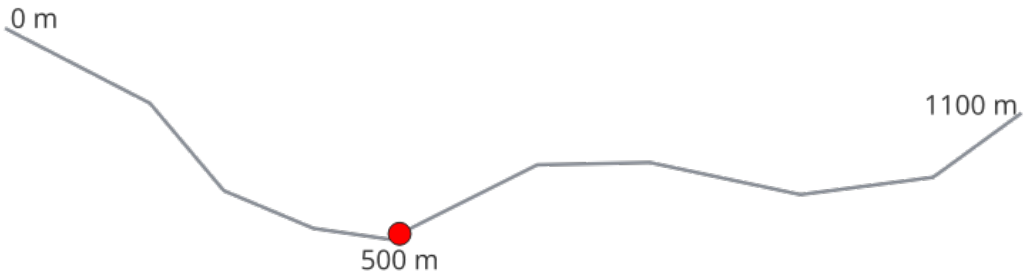


Fig. 9.20: Interpolated point at 500m of the beginning of the line

Further reading: *Interpolate point on line* algorithm

line_interpolate_point_by_m

Returns the point interpolated by a matching M value along a linestring geometry.

语法	line_interpolate_point_by_m(geometry, m, [use_3d_distance=false]) [] 标记可选参数
参数	• **geometry** - a linestring geometry • **m** - an M value • **use_3d_distance** - controls whether 2D or 3D distances between vertices should be used during interpolation (this option is only considered for lines with z values)
Examples	• `geom_to_wkt(line_interpolate_point_by_m(geom_from_wkt('LineStringM(0 0 0, 10 10 10)'), m:=5))` → 'Point (5 5)'

line_locate_m

Returns the distance along a linestring corresponding to the first matching interpolated M value.

语法	line_locate_m(geometry, m, [use_3d_distance=false]) [] 标记可选参数
参数	• **geometry** - a linestring geometry • **m** - an M value • **use_3d_distance** - controls whether 2D or 3D distances between vertices should be used during interpolation (this option is only considered for lines with z values)
Examples	• `line_locate_m(geometry:=geom_from_wkt('LineStringM(0 0 0, 10 10 10)'),m:=5)` → 7.07106

line_locate_point

Returns the distance along a linestring corresponding to the closest position the linestring comes to a specified point geometry.

语法	line_locate_point(geometry, point)
参数	• **geometry** - a linestring geometry • **point** - point geometry to locate closest position on linestring to
Examples	• `line_locate_point(geometry:=geom_from_wkt('LineString(0 0, 10 0)'),point:=geom_from_wkt('Point(5 0)'))` → 5.0

line_merge

Returns a LineString or MultiLineString geometry, where any connected LineStrings from the input geometry have been merged into a single linestring. This function will return NULL if passed a geometry which is not a LineString/MultiLineString.

语法	line_merge(geometry)
参数	• **geometry** - a LineString/MultiLineString geometry
Examples	• `geom_to_wkt(line_merge(geom_from_wkt('MULTILINESTRING((0 0, 1 1),(1 1, 2 2))')))` → 'LineString(0 0,1 1,2 2)' • `geom_to_wkt(line_merge(geom_from_wkt('MULTILINESTRING((0 0, 1 1),(11 1, 21 2))')))` → 'MultiLineString((0 0, 1 1),(11 1, 21 2)'

line_substring

Returns the portion of a line (or curve) geometry which falls between the specified start and end distances (measured from the beginning of the line). Z and M values are linearly interpolated from existing values.

语法	line_substring(geometry, start_distance, end_distance)
参数	• **geometry** - a linestring or curve geometry • **start_distance** - distance to start of substring • **end_distance** - distance to end of substring
Examples	• `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. 9.21: Substring line with starting distance set at 0 meters and the ending distance at 250 meters.

Further reading: *Line substring* algorithm

m

Returns the m (measure) value of a point geometry.

语法	m(geometry)
参数	• **geometry** - a point geometry
Examples	• `m( geom_from_wkt( 'POINTM(2 5 4)' ) )` → 4

m_at

Retrieves a m coordinate of the geometry, or NULL if the geometry has no m value.

语法	m_at(geometry, vertex)
参数	• **geometry** - geometry object • **vertex** - index of the vertex of the geometry (indices start at 0; negative values apply from the last index, starting at -1)
Examples	• `m_at(geom_from_wkt('LineStringZM(0 0 0 0, 10 10 0 5, 10 10 0 0)'), 1)` → 5

m_max

Returns the maximum m (measure) value of a geometry.

语法	m_max(geometry)
参数	• **geometry** - a geometry containing m values
Examples	• `m_max( make_point_m( 0,0,1 ) )` → 1 • `m_max(make_line( make_point_m( 0,0,1 ), make_point_m( -1,-1,2 ), make_point_m( -2,-2,0 ) ) )` → 2

m_min

Returns the minimum m (measure) value of a geometry.

语法	m_min(geometry)
参数	• **geometry** - a geometry containing m values
Examples	• `m_min( make_point_m( 0,0,1 ) )` → 1 • `m_min(make_line( make_point_m( 0,0,1 ), make_point_m( -1, -1,2 ), make_point_m( -2,-2,0 ) ) )` → 0

main_angle

Returns the angle of the long axis (clockwise, in degrees from North) of the oriented minimal bounding rectangle, which completely covers the geometry.

语法	main_angle(geometry)
参数	• **geometry** - a geometry
Examples	• `main_angle(geom_from_wkt('Polygon ((321577 129614, 321581 129618, 321585 129615, 321581 129610, 321577 129614))'))` → 38.66

make_circle

Creates a circular polygon.

语法	make_circle(center, radius, [segments=36]) [] 标记可选参数
参数	• **center** - center point of the circle • **radius** - radius of the circle • **segments** - optional argument for polygon segmentation. By default this value is 36
Examples	• `geom_to_wkt(make_circle(make_point(10,10), 5, 4))` → 'Polygon ((10 15, 15 10, 10 5, 5 10, 10 15))' • `geom_to_wkt(make_circle(make_point(10,10,5), 5, 4))` → 'PolygonZ ((10 15 5, 15 10 5, 10 5 5, 5 10 5, 10 15 5))' • `geom_to_wkt(make_circle(make_point(10,10,5,30), 5, 4))` → 'PolygonZM ((10 15 5 30, 15 10 5 30, 10 5 5 30, 5 10 5 30, 10 15 5 30))'

make_ellipse

Creates an elliptical polygon.

语法	make_ellipse(center, semi_major_axis, semi_minor_axis, azimuth, [segments=36]) [] 标记可选参数
参数	• **center** - center point of the ellipse • **semi_major_axis** - semi-major axis of the ellipse • **semi_minor_axis** - semi-minor axis of the ellipse • **azimuth** - orientation of the ellipse • **segments** - optional argument for polygon segmentation. By default this value is 36
Examples	• `geom_to_wkt(make_ellipse(make_point(10,10), 5, 2, 90, 4))` → 'Polygon ((15 10, 10 8, 5 10, 10 12, 15 10))' • `geom_to_wkt(make_ellipse(make_point(10,10,5), 5, 2, 90, 4))` → 'PolygonZ ((15 10 5, 10 8 5, 5 10 5, 10 12 5, 15 10 5))' • `geom_to_wkt(make_ellipse(make_point(10,10,5,30), 5, 2, 90, 4))` → 'PolygonZM ((15 10 5 30, 10 8 5 30, 5 10 5 30, 10 12 5 30, 15 10 5 30))'

make_line

Creates a line geometry from a series of point geometries.

List of arguments variant

Line vertices are specified as separate arguments to the function.

语法	make_line(point1, point2, ...)
参数	• **point** - a point geometry (or array of points)
Examples	• `geom_to_wkt(make_line(make_point(2,4),make_point(3,5)))` → 'LineString (2 4, 3 5)' • `geom_to_wkt(make_line(make_point(2,4),make_point(3,5),make_point(9,7)))` → 'LineString (2 4, 3 5, 9 7)'

Array variant

Line vertices are specified as an array of points.

语法	make_line(array)
参数	• **array** - array of points
Examples	• `geom_to_wkt(make_line(array(make_point(2,4),make_point(3,5),make_point(9,7))))` → 'LineString (2 4, 3 5, 9 7)'

make_point

Creates a point geometry from an x and y (and optional z and m) value.

语法	make_point(x, y, [z], [m]) [] 标记可选参数
参数	• **x** - x coordinate of point • **y** - y coordinate of point • **z** - optional z coordinate of point • **m** - optional m value of point
Examples	• `geom_to_wkt(make_point(2,4))` → 'Point (2 4)' • `geom_to_wkt(make_point(2,4,6))` → 'PointZ (2 4 6)' • `geom_to_wkt(make_point(2,4,6,8))` → 'PointZM (2 4 6 8)'

make_point_m

Creates a point geometry from an x, y coordinate and m value.

语法	make_point_m(x, y, m)
参数	• **x** - x coordinate of point • **y** - y coordinate of point • **m** - m value of point
Examples	• `geom_to_wkt(make_point_m(2,4,6))` → 'PointM (2 4 6)'

make_polygon

Creates a polygon geometry from an outer ring and optional series of inner ring geometries.

语法	make_polygon(outerRing, [innerRing1], [innerRing2], ...) [] 标记可选参数
参数	• **outerRing** - closed line geometry for polygon's outer ring • **innerRing** - optional closed line geometry for inner ring
Examples	• `geom_to_wkt(make_polygon(geom_from_wkt('LINESTRING( 0 0, 0 1, 1 1, 1 0, 0 0 )')))` → 'Polygon ((0 0, 0 1, 1 1, 1 0, 0 0))' • `geom_to_wkt(make_polygon(geom_from_wkt('LINESTRING( 0 0, 0 1, 1 1, 1 0, 0 0 )'),geom_from_wkt('LINESTRING( 0.1 0.1, 0.1 0.2, 0.2 0.2, 0.2 0.1, 0.1 0.1 )'), geom_from_wkt('LINESTRING( 0.8 0.8, 0.8 0.9, 0.9 0.9, 0.9 0.8, 0.8 0.8 )')))` → 'Polygon ((0 0, 0 1, 1 1, 1 0, 0 0),(0.1 0.1, 0.1 0.2, 0.2 0.2, 0.2 0.1, 0.1 0.1),(0.8 0.8, 0.8 0.9, 0.9 0.9, 0.9 0.8, 0.8 0.8))'

make_rectangle_3points

Creates a rectangle from 3 points.

语法	make_rectangle_3points(point1, point2, point3, [option=0]) [] 标记可选参数
参数	• **point1** - First point. • **point2** - Second point. • **point3** - Third point. • **option** - An optional argument to construct the rectangle. By default this value is 0. Value can be 0 (distance) or 1 (projected). Option distance: Second distance is equal to the distance between 2nd and 3rd point. Option projected: Second distance is equal to the distance of the perpendicular projection of the 3rd point on the segment or its extension.
Examples	• `geom_to_wkt(make_rectangle_3points(make_point(0, 0), make_point(0,5), make_point(5, 5), 0))` → 'Polygon ((0 0, 0 5, 5 5, 5 0, 0 0))' • `geom_to_wkt(make_rectangle_3points(make_point(0, 0), make_point(0,5), make_point(5, 3), 1))` → 'Polygon ((0 0, 0 5, 5 5, 5 0, 0 0))'

make_regular_polygon

Creates a regular polygon.

语法	make_regular_polygon(center, radius, number_sides, [circle=0]) [] 标记可选参数
参数	• **center** - center of the regular polygon • **radius** - second point. The first if the regular polygon is inscribed. The midpoint of the first side if the regular polygon is circumscribed. • **number_sides** - Number of sides/edges of the regular polygon • **circle** - Optional argument to construct the regular polygon. By default this value is 0. Value can be 0 (inscribed) or 1 (circumscribed)
Examples	• `geom_to_wkt(make_regular_polygon(make_point(0,0), make_point(0,5), 5))` → 'Polygon ((0 5, 4.76 1.55, 2.94 -4.05, -2.94 -4.05, -4.76 1.55, 0 5))' • `geom_to_wkt(make_regular_polygon(make_point(0,0), project(make_point(0,0), 4.0451, radians(36)), 5))` → 'Polygon ((0 5, 4.76 1.55, 2.94 -4.05, -2.94 -4.05, -4.76 1.55, 0 5))'

make_square

Creates a square from a diagonal.

语法	make_square(point1, point2)
参数	• **point1** - First point of the diagonal • **point2** - Last point of the diagonal
Examples	• `geom_to_wkt(make_square( make_point(0,0), make_point(5, 5)))` → 'Polygon ((0 0, -0 5, 5 5, 5 0, 0 0))' • `geom_to_wkt(make_square( make_point(5,0), make_point(5, 5)))` → 'Polygon ((5 0, 2.5 2.5, 5 5, 7.5 2.5, 5 0))'

make_triangle

Creates a triangle polygon.

语法	make_triangle(point1, point2, point3)
参数	• **point1** - first point of the triangle • **point2** - second point of the triangle • **point3** - third point of the triangle
Examples	• `geom_to_wkt(make_triangle(make_point(0,0), make_point(5, 5), make_point(0,10)))` → 'Triangle ((0 0, 5 5, 0 10, 0 0))' • `geom_to_wkt(boundary(make_triangle(make_point(0,0), make_point(5,5), make_point(0,10))))` → 'LineString (0 0, 5 5, 0 10, 0 0)'

make_valid

Returns a valid geometry or an empty geometry if the geometry could not be made valid.

语法	make_valid(geometry, [method=structure], [keep_collapsed=false]) [] 标记可选参数
参数	• **geometry** - a geometry • **method** - repair algorithm. May be either 'structure' or 'linework'. The 'linework' option combines all rings into a set of noded lines and then extracts valid polygons from that linework. The 'structure' method first makes all rings valid and then merges shells and subtracts holes from shells to generate valid result. Assumes that holes and shells are correctly categorized. • **keep_collapsed** - if set to true, then components that have collapsed into a lower dimensionality will be kept. For example, a ring collapsing to a line, or a line collapsing to a point.
Examples	• `geom_to_wkt(make_valid(geom_from_wkt('POLYGON((3 2, 4 1, 5 8, 3 2, 4 2))')))` → 'Polygon ((3 2, 5 8, 4 1, 3 2))' • `geom_to_wkt(make_valid(geom_from_wkt('POLYGON((3 2, 4 1, 5 8, 3 2, 4 2))'), 'linework'))` → 'GeometryCollection (Polygon ((5 8, 4 1, 3 2, 5 8)),LineString (3 2, 4 2))' • `geom_to_wkt(make_valid(geom_from_wkt('POLYGON((3 2, 4 1, 5 8))'), method:='linework'))` → 'Polygon ((3 2, 4 1, 5 8, 3 2))' • `make_valid(geom_from_wkt('LINESTRING(0 0)'))` → An empty geometry

Further reading: *is_valid*, *Fix geometries* algorithm

minimal_circle

Returns the minimal enclosing circle of a geometry. It represents the minimum circle that encloses all geometries within the set.

语法	minimal_circle(geometry, [segments=36]) [] 标记可选参数
参数	• **geometry** - a geometry • **segments** - optional argument for polygon segmentation. By default this value is 36
Examples	• `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))'

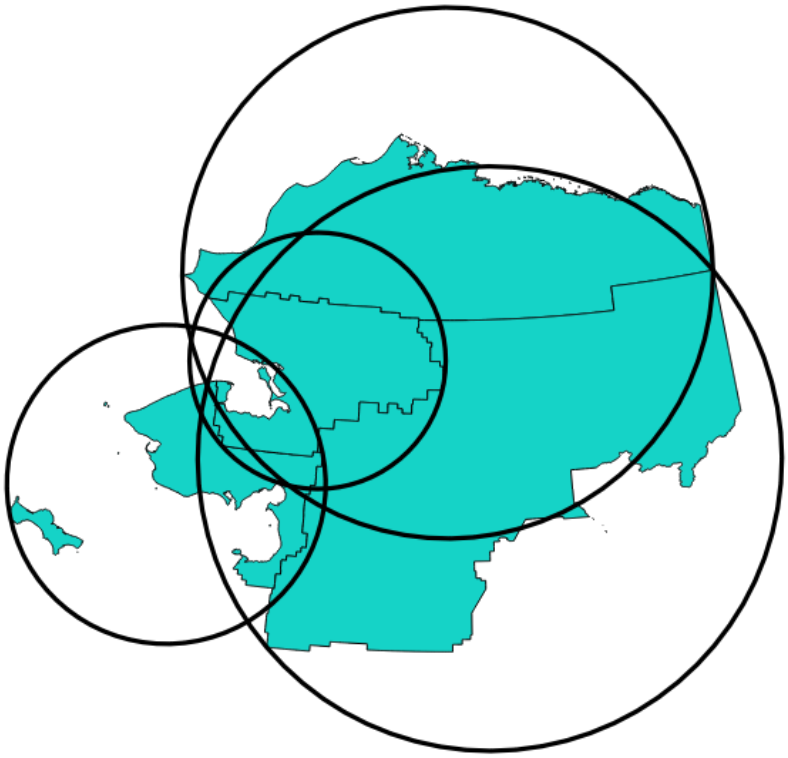


Fig. 9.22: Minimal enclosing circle of each feature

Further reading: *Minimum enclosing circles* algorithm

nodes_to_points

Returns a multipoint geometry consisting of every node in the input geometry.

语法	nodes_to_points(geometry, [ignore_closing_nodes=false]) [] 标记可选参数
参数	• **geometry** - geometry object • **ignore_closing_nodes** - optional argument specifying whether to include duplicate nodes which close lines or polygons rings. Defaults to false, set to true to avoid including these duplicate nodes in the output collection.
Examples	• `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))'

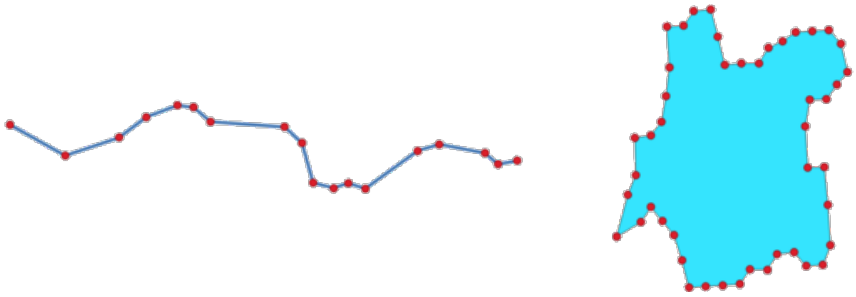


Fig. 9.23: Multi-point feature extracted from vertices

Further reading: *Extract vertices* algorithm

num_geometries

Returns the number of geometries in a geometry collection, or the number of parts in a multi-part geometry. The function returns NULL if the input geometry is not a collection.

语法	num_geometries(geometry)
参数	• **geometry** - geometry collection or multi-part geometry
Examples	• `num_geometries(geom_from_wkt('GEOMETRYCOLLECTION(POINT(0 1), POINT(0 0), POINT(1 0), POINT(1 1))'))` → 4 • `num_geometries(geom_from_wkt('MULTIPOINT((0 1), (0 0), (1 0))'))` → 3

num_interior_rings

Returns the number of interior rings in a polygon or geometry collection, or NULL if the input geometry is not a polygon or collection.

语法	num_interior_rings(geometry)
参数	• **geometry** - input geometry
Examples	• `num_interior_rings(geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1),(-0.1 -0.1, 0.4 0, 0.4 0.2, 0 0.2, -0.1 -0.1))'))` → 1

num_points

Returns the number of vertices in a geometry.

语法	num_points(geometry)
参数	• **geometry** - a geometry
Examples	• `num_points(@geometry)` → number of vertices in the current feature's geometry

num_rings

Returns the number of rings (including exterior rings) in a polygon or geometry collection, or NULL if the input geometry is not a polygon or collection.

语法	num_rings(geometry)
参数	• **geometry** - input geometry
Examples	• `num_rings(geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1),(-0.1 -0.1, 0.4 0, 0.4 0.2, 0 0.2, -0.1 -0.1))'))` → 2

offset_curve

Returns a geometry formed by offsetting a linestring geometry to the side. Distances are in the Spatial Reference System of this geometry.

语法	offset_curve(geometry, distance, [segments=8], [join=1], [miter_limit=2.0]) [] 标记可选参数
参数	• **geometry** - a (multi)linestring geometry • **distance** - offset distance. Positive values will be buffered to the left of lines, negative values to the right • **segments** - number of segments to use to represent a quarter circle when a round join style is used. A larger number results in a smoother line with more nodes. • **join** - join style for corners, where 1 = round, 2 = miter and 3 = bevel • **miter_limit** - limit on the miter ratio used for very sharp corners (when using miter joins only)
Examples	• `offset_curve(@geometry, 10.5)` → line offset to the left by 10.5 units • `offset_curve(@geometry, -10.5)` → line offset to the right by 10.5 units • `offset_curve(@geometry, 10.5, segments:=16, join:=1)` → line offset to the left by 10.5 units, using more segments to result in a smoother curve • `offset_curve(@geometry, 10.5, join:=3)` → line offset to the left by 10.5 units, using a beveled join

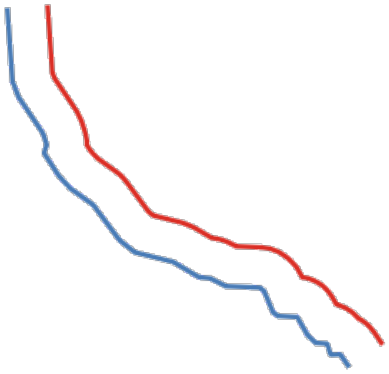


Fig. 9.24: In blue the source layer, in red the offset one

Further reading: *Offset lines* algorithm

order_parts

Orders the parts of a MultiGeometry by a given criteria

语法	order_parts(geometry, orderby, [ascending=true]) [] 标记可选参数
参数	• **geometry** - a multi-type geometry • **orderby** - an expression string defining the order criteria • **ascending** - boolean, True for ascending, False for descending
Examples	• `geom_to_wkt(order_parts(geom_from_wkt('MultiPolygon (((1 1, 5 1, 5 5, 1 5, 1 1)),((1 1, 9 1, 9 9, 1 9, 1 1)))'), 'area(@geometry)', False))` → 'MultiPolygon (((1 1, 9 1, 9 9, 1 9, 1 1)),((1 1, 5 1, 5 5, 1 5, 1 1)))' • `geom_to_wkt(order_parts(geom_from_wkt('LineString(1 2, 3 2, 4 3)'), '1', True))` → 'LineString(1 2, 3 2, 4 3)'

oriented_bbox

Returns a geometry which represents the minimal oriented bounding box of an input geometry.

语法	oriented_bbox(geometry)
参数	• **geometry** - a geometry
Examples	• `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))'

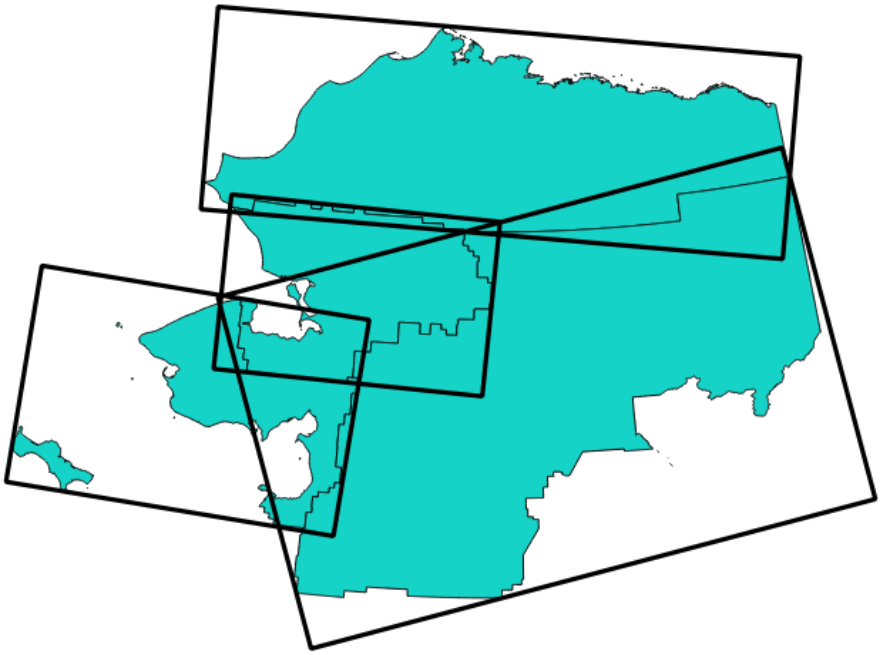


Fig. 9.25: Oriented minimum bounding box

Further reading: *Oriented minimum bounding box* algorithm

overlaps

Tests whether a geometry overlaps another. Returns TRUE if the geometries share space, are of the same dimension, but are not completely contained by each other.

语法	overlaps(geometry1, geometry2)
参数	• **geometry1** - a geometry • **geometry2** - a geometry
Examples	• `overlaps( geom_from_wkt( 'LINESTRING(3 5, 4 4, 5 5, 5 3)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE • `overlaps( geom_from_wkt( 'LINESTRING(0 0, 1 1)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → FALSE

overlay_contains

Returns whether the current feature spatially contains at least one feature from a target layer, or an array of expression-based results for the features in the target layer contained in the current feature.

Read more on the underlying GEOS ”Contains” predicate, as described in PostGIS ST_Contains function.

语法	overlay_contains(layer, [expression], [filter], [limit], [cache=false]) [] 标记可选参数
参数	• **layer** - the layer whose overlay is checked • **expression** - an optional expression to evaluate on the features from the target layer. If not set, the function will just return a boolean indicating whether there is at least one match. • **filter** - an optional expression to filter the target features to check. If not set, all the features will be checked. • **limit** - an optional integer to limit the number of matching features. If not set, all the matching features will be returned. • **cache** - set this to true to build a local spatial index (most of the time, this is unwanted, unless you are working with a particularly slow data provider)
Examples	• `overlay_contains('regions')` → TRUE if the current feature spatially contains a region • `overlay_contains('regions', filter:= population > 10000)` → TRUE if the current feature spatially contains a region with a population greater than 10000 • `overlay_contains('regions', name)` → an array of names, for the regions contained in the current feature • `array_to_string(overlay_contains('regions', name))` → a string as a comma separated list of names, for the regions contained in the current feature • `array_sort(overlay_contains(layer:='regions', expression:="name", filter:= population > 10000))` → an ordered array of names, for the regions contained in the current feature and with a population greater than 10000 • `overlay_contains(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions contained in the current feature

Further reading: *contains*, *array manipulation*, *Select by location* algorithm

overlay_crosses

Returns whether the current feature spatially crosses at least one feature from a target layer, or an array of expression-based results for the features in the target layer crossed by the current feature.

Read more on the underlying GEOS ”Crosses” predicate, as described in PostGIS ST_Crosses function.

语法	overlay_crosses(layer, [expression], [filter], [limit], [cache=false]) [] 标记可选参数
参数	• **layer** - the layer whose overlay is checked • **expression** - an optional expression to evaluate on the features from the target layer. If not set, the function will just return a boolean indicating whether there is at least one match. • **filter** - an optional expression to filter the target features to check. If not set, all the features will be checked. • **limit** - an optional integer to limit the number of matching features. If not set, all the matching features will be returned. • **cache** - set this to true to build a local spatial index (most of the time, this is unwanted, unless you are working with a particularly slow data provider)
Examples	• `overlay_crosses('regions')` → TRUE if the current feature spatially crosses a region • `overlay_crosses('regions', filter:= population > 10000)` → TRUE if the current feature spatially crosses a region with a population greater than 10000 • `overlay_crosses('regions', name)` → an array of names, for the regions crossed by the current feature • `array_to_string(overlay_crosses('regions', name))` → a string as a comma separated list of names, for the regions crossed by the current feature • `array_sort(overlay_crosses(layer:='regions', expression:="name", filter:= population > 10000))` → an ordered array of names, for the regions crossed by the current feature and with a population greater than 10000 • `overlay_crosses(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions crossed by the current feature

Further reading: *crosses*, *array manipulation*, *Select by location* algorithm

overlay_disjoint

Returns whether the current feature is spatially disjoint from all the features of a target layer, or an array of expression-based results for the features in the target layer that are disjoint from the current feature.

Read more on the underlying GEOS ”Disjoint” predicate, as described in PostGIS ST_Disjoint function.

语法	overlay_disjoint(layer, [expression], [filter], [limit], [cache=false]) [] 标记可选参数
参数	• **layer** - the layer whose overlay is checked • **expression** - an optional expression to evaluate on the features from the target layer. If not set, the function will just return a boolean indicating whether there is at least one match. • **filter** - an optional expression to filter the target features to check. If not set, all the features will be checked. • **limit** - an optional integer to limit the number of matching features. If not set, all the matching features will be returned. • **cache** - set this to true to build a local spatial index (most of the time, this is unwanted, unless you are working with a particularly slow data provider)
Examples	• `overlay_disjoint('regions')` → TRUE if the current feature is spatially disjoint from all the regions • `overlay_disjoint('regions', filter:= population > 10000)` → TRUE if the current feature is spatially disjoint from all the regions with a population greater than 10000 • `overlay_disjoint('regions', name)` → an array of names, for the regions spatially disjoint from the current feature • `array_to_string(overlay_disjoint('regions', name))` → a string as a comma separated list of names, for the regions spatially disjoint from the current feature • `array_sort(overlay_disjoint(layer:='regions', expression:="name", filter:= population > 10000))` → an ordered array of names, for the regions spatially disjoint from the current feature and with a population greater than 10000 • `overlay_disjoint(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions spatially disjoint from the current feature

Further reading: *disjoint*, *array manipulation*, *Select by location* algorithm

overlay_equals

Returns whether the current feature's geometry is exactly equal to at least one feature's geometry from a target layer, or an array of expression-based results for the features in the target layer whose geometry is exactly equal to the current feature's geometry. Note that the order of vertices matters.

语法	overlay_equals(layer, [expression], [filter], [limit], [cache=false]) [] 标记可选参数
参数	• **layer** - the layer whose overlay is checked • **expression** - an optional expression to evaluate on the features from the target layer. If not set, the function will just return a boolean indicating whether there is at least one match. • **filter** - an optional expression to filter the target features to check. If not set, all the features will be checked. • **limit** - an optional integer to limit the number of matching features. If not set, all the matching features will be returned. • **cache** - set this to true to build a local spatial index (most of the time, this is unwanted, unless you are working with a particularly slow data provider)
Examples	• `overlay_equals('regions')` → TRUE if the current feature's geometry is exactly the same as the geometry of a region • `overlay_equals('regions', filter:= population > 10000)` → TRUE if the current feature's geometry is exactly the same as the geometry of a region whose population is greater than 10000 • `overlay_equals('regions', name)` → an array of names, for the regions exactly equal to the current feature • `array_to_string(overlay_equals('regions', name))` → a string as a comma separated list of names, for the regions exactly equal to the current feature • `array_sort(overlay_equals(layer:='regions', expression:="name", filter:= population > 10000))` → an ordered array of names, for the regions exactly equal to the current feature and with a population greater than 10000 • `overlay_equals(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions exactly equal to the current feature

Note

This function requires exact equality of geometries, meaning that the geometries 'LINESTRING(0 0, 1 1)' and 'LINESTRING(1 1, 0 0)' are different.

Further reading: *array manipulation*, *Select by location* algorithm

overlay_intersects

Returns whether the current feature spatially intersects at least one feature from a target layer, or an array of expression-based results for the features in the target layer intersected by the current feature.

Read more on the underlying GEOS "Intersects" predicate, as described in PostGIS ST_Intersects function.

语法	overlay_intersects(layer, [expression], [filter], [limit], [cache=false], [min_overlap], [min_inscribed_circle_radius], [return_details], [sort_by_intersection_size]) [] 标记可选参数
参数	• **layer** - the layer whose overlay is checked • **expression** - an optional expression to evaluate on the features from the target layer. If not set, the function will just return a boolean indicating whether there is at least one match. • **filter** - an optional expression to filter the target features to check. If not set, all the features will be checked. • **limit** - an optional integer to limit the number of matching features. If not set, all the matching features will be returned. • **cache** - set this to true to build a local spatial index (most of the time, this is unwanted, unless you are working with a particularly slow data provider) • **min_overlap** - defines an optional exclusion filter: – for polygons, a minimum area in current feature squared units for the intersection. If the intersection results in multiple polygons the intersection will be returned if at least one of the polygons has an area greater or equal to the value – for lines, a minimum length in current feature units. If the intersection results in multiple lines the intersection will be returned if at least one of the lines has a length greater or equal to the value. • **min_inscribed_circle_radius** - defines an optional exclusion filter (for polygons only): minimum radius in current feature units for the maximum inscribed circle of the intersection. If the intersection results in multiple polygons the intersection will be returned if at least one of the polygons has a radius for the maximum inscribed circle greater or equal to the value. Read more on the underlying GEOS predicate, as described in PostGIS ST_MaximumInscribedCircle function. This argument requires GEOS >= 3.9. • **return_details** - Set this to true to return a list of maps containing (key names in quotes) the feature 'id', the expression 'result' and the 'overlap' value (of the largest element in case of multipart). The 'radius' of the maximum inscribed circle is also returned when the target layer is a polygon. Only valid when used with the expression parameter • **sort_by_intersection_size** - only valid when used with an expression, set this to 'des' to return the results ordered by the overlap value in descending order or set this to 'asc' for ascending order.
Examples	• `overlay_intersects('regions')` → TRUE if the current feature spatially intersects a region • `overlay_intersects('regions', filter:= population > 10000)` → TRUE if the current feature spatially intersects a region with a population greater than 10000 • `overlay_intersects('regions', name)` → an array of names, for the regions intersected by the current feature • `array_to_string(overlay_intersects('regions', name))` → a string as a comma separated list of names, for the regions intersected by the current feature • `array_sort(overlay_intersects(layer:='regions', expression:="name", filter:= population > 10000))` → an ordered array of names, for the regions intersected by the current feature and with a population greater than 10000 • `overlay_intersects(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions intersected by the current feature • `overlay_intersects(layer:='regions', min_overlap:=0.54)` → TRUE if the current feature spatially intersects a region and the intersection area (of at least one of the parts in case of multipolygons) is greater or equal to 0.54 • `overlay_intersects(layer:='regions', min_inscribed_circle_radius:=0.54)` → TRUE if the current feature spatially intersects a region and the intersection area maximum inscribed circle's radius (of at least one of the parts in case of multipart) is greater or equal to 0.54 • `overlay_intersects(layer:='regions', expression:= geom_to_wkt(@geometry), return_details:=true)` → an array of maps containing 'id', 'result', 'overlap' and 'radius'

Further reading: *intersects*, *array manipulation*, *Select by location* algorithm

overlay_nearest

Returns whether the current feature has feature(s) from a target layer within a given distance, or an array of expression-based results for the features in the target layer within a distance from the current feature.

Note: This function can be slow and consume a lot of memory for large layers.

语法	overlay_nearest(layer, [expression], [filter], [limit=1], [max_distance], [cache=false]) [] 标记可选参数
参数	• **layer** - the target layer • **expression** - an optional expression to evaluate on the features from the target layer. If not set, the function will just return a boolean indicating whether there is at least one match. • **filter** - an optional expression to filter the target features to check. If not set, all the features in the target layer will be used. • **limit** - an optional integer to limit the number of matching features. If not set, only the nearest feature will be returned. If set to -1, returns all the matching features. • **max_distance** - an optional distance to limit the search of matching features. If not set, all the features in the target layer will be used. • **cache** - set this to true to build a local spatial index (most of the time, this is unwanted, unless you are working with a particularly slow data provider)
Examples	• `overlay_nearest('airports')` → TRUE if the "airports" layer has at least one feature • `overlay_nearest('airports', max_distance:= 5000)` → TRUE if there is an airport within a distance of 5000 map units from the current feature • `overlay_nearest('airports', name)` → the name of the closest airport to the current feature, as an array • `array_to_string(overlay_nearest('airports', name))` → the name of the closest airport to the current feature, as a string • `overlay_nearest(layer:='airports', expression:= name, max_distance:= 5000)` → the name of the closest airport within a distance of 5000 map units from the current feature, as an array • `overlay_nearest(layer:='airports', expression:="name", filter:= "Use"='Civilian', limit:=3)` → an array of names, for up to the three closest civilian airports ordered by distance • `overlay_nearest(layer:='airports', expression:="name", limit:= -1, max_distance:= 5000)` → an array of names, for all the airports within a distance of 5000 map units from the current feature, ordered by distance

Further reading: *array manipulation*, *Join attributes by nearest* algorithm

overlay_touches

Returns whether the current feature spatially touches at least one feature from a target layer, or an array of expression-based results for the features in the target layer touched by the current feature.

Read more on the underlying GEOS "Touches" predicate, as described in PostGIS ST_Touches function.

语法	overlay_touches(layer, [expression], [filter], [limit], [cache=false]) [] 标记可选参数
参数	• **layer** - the layer whose overlay is checked • **expression** - an optional expression to evaluate on the features from the target layer. If not set, the function will just return a boolean indicating whether there is at least one match. • **filter** - an optional expression to filter the target features to check. If not set, all the features will be checked. • **limit** - an optional integer to limit the number of matching features. If not set, all the matching features will be returned. • **cache** - set this to true to build a local spatial index (most of the time, this is unwanted, unless you are working with a particularly slow data provider)
Examples	• `overlay_touches('regions')` → TRUE if the current feature spatially touches a region • `overlay_touches('regions', filter:= population > 10000)` → TRUE if the current feature spatially touches a region with a population greater than 10000 • `overlay_touches('regions', name)` → an array of names, for the regions touched by the current feature • `array_to_string(overlay_touches('regions', name))` → a string as a comma separated list of names, for the regions touched by the current feature • `array_sort(overlay_touches(layer:='regions', expression:="name", filter:= population > 10000))` → an ordered array of names, for the regions touched by the current feature and with a population greater than 10000 • `overlay_touches(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions touched by the current feature

Further reading: *touches*, *array manipulation*, *Select by location* algorithm

overlay_within

Returns whether the current feature is spatially within at least one feature from a target layer, or an array of expression-based results for the features in the target layer that contain the current feature.

Read more on the underlying GEOS "Within" predicate, as described in PostGIS ST_Within function.

语法	overlay_within(layer, [expression], [filter], [limit], [cache=false]) [] 标记可选参数
参数	• **layer** - the layer whose overlay is checked • **expression** - an optional expression to evaluate on the features from the target layer. If not set, the function will just return a boolean indicating whether there is at least one match. • **filter** - an optional expression to filter the target features to check. If not set, all the features will be checked. • **limit** - an optional integer to limit the number of matching features. If not set, all the matching features will be returned. • **cache** - set this to true to build a local spatial index (most of the time, this is unwanted, unless you are working with a particularly slow data provider)
Examples	• `overlay_within('regions')` → TRUE if the current feature is spatially within a region • `overlay_within('regions', filter:= population > 10000)` → TRUE if the current feature is spatially within a region with a population greater than 10000 • `overlay_within('regions', name)` → an array of names, for the regions containing the current feature • `array_to_string(overlay_within('regions', name))` → a string as a comma separated list of names, for the regions containing the current feature • `array_sort(overlay_within(layer:='regions', expression:="name", filter:= population > 10000))` → an ordered array of names, for the regions containing the current feature and with a population greater than 10000 • `overlay_within(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions containing the current feature

Further reading: *within*, *array manipulation*, *Select by location* algorithm

$perimeter

Returns the perimeter length of the current feature. The perimeter calculated by this function respects both the current project's ellipsoid setting and distance unit settings. For example, if an ellipsoid has been set for the project then the calculated perimeter will be ellipsoidal, and if no ellipsoid is set then the calculated perimeter will be planimetric.

语法	$perimeter
Examples	• `$perimeter` → 42

perimeter

Returns the perimeter of a geometry polygon object. Calculations are always planimetric in the Spatial Reference System (SRS) of this geometry, and the units of the returned perimeter will match the units for the SRS. This differs from the calculations performed by the $perimeter function, which will perform ellipsoidal calculations based on the project's ellipsoid and distance unit settings.

语法	perimeter(geometry)
参数	• **geometry** - polygon geometry object
Examples	• `perimeter(geom_from_wkt('POLYGON((0 0, 4 0, 4 2, 0 2, 0 0))'))` → 12.0

point_n

Returns a specific node from a geometry.

语法	point_n(geometry, index)
参数	• **geometry** - geometry object • **index** - index of node to return, where 1 is the first node; if the value is negative, the selected vertex index will be its total count minus the absolute value
Examples	• `geom_to_wkt(point_n(geom_from_wkt('POLYGON((0 0, 4 0, 4 2, 0 2, 0 0))'),2))` → 'Point (4 0)'

Further reading: *Extract specific vertices* algorithm

point_on_surface

Returns a point guaranteed to lie on the surface of a geometry.

语法	point_on_surface(geometry)
参数	• **geometry** - a geometry
Examples	• `point_on_surface(@geometry)` → a point geometry

Further reading: *Point on Surface* algorithm

pole_of_inaccessibility

Calculates the approximate pole of inaccessibility for a surface, which is the most distant internal point from the boundary of the surface. This function uses the 'polylabel' algorithm (Vladimir Agafonkin, 2016), which is an iterative approach guaranteed to find the true pole of inaccessibility within a specified tolerance. More precise tolerances require more iterations and will take longer to calculate.

语法	pole_of_inaccessibility(geometry, tolerance)
参数	• **geometry** - a geometry • **tolerance** - maximum distance between the returned point and the true pole location
Examples	• `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. 9.26: Pole of inaccessibility

Further reading: *Pole of inaccessibility* algorithm

project

Returns a point projected from a start point using a distance, a bearing (azimuth) and an elevation in radians.

语法	project(point, distance, azimuth, [elevation]) [] 标记可选参数
参数	• **point** - start point • **distance** - distance to project • **azimuth** - azimuth in radians clockwise, where 0 corresponds to north • **elevation** - angle of inclination in radians
Examples	• `geom_to_wkt(project(make_point(1, 2), 3, radians(270)))` → 'Point(-2, 2)'

Further reading: *Project points (Cartesian)* algorithm

relate

Tests the Dimensional Extended 9 Intersection Model (DE-9IM) representation of the relationship between two geometries.

Relationship variant

Returns the Dimensional Extended 9 Intersection Model (DE-9IM) representation of the relationship between two geometries.

语法	relate(geometry, geometry)
参数	• **geometry** - a geometry • **geometry** - a geometry
Examples	• `relate( geom_from_wkt( 'LINESTRING(40 40,120 120)' ), geom_from_wkt( 'LINESTRING(40 40,60 120)' ) )` → 'FF1F00102'

Pattern match variant

Tests whether the DE-9IM relationship between two geometries matches a specified pattern.

语法	relate(geometry, geometry, pattern)
参数	• **geometry** - a geometry • **geometry** - a geometry • **pattern** - DE-9IM pattern to match
Examples	• `relate( geom_from_wkt( 'LINESTRING(40 40,120 120)' ), geom_from_wkt( 'LINESTRING(40 40,60 120)' ), '**1F001**' )` → TRUE

reverse

Reverses the direction of a line string or reverses a string of text.

String variant

Reverses the order of characters in a string.

语法	reverse(string)
参数	• **string** - string to reverse
Examples	• `reverse('hello')` → 'olleh'

Geometry variant

Reverses the direction of a line string by reversing the order of its vertices.

语法	reverse(geometry)
参数	• **geometry** - a geometry
Examples	• `geom_to_wkt(reverse(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)')))` → 'LINESTRING(2 2, 1 1, 0 0)'

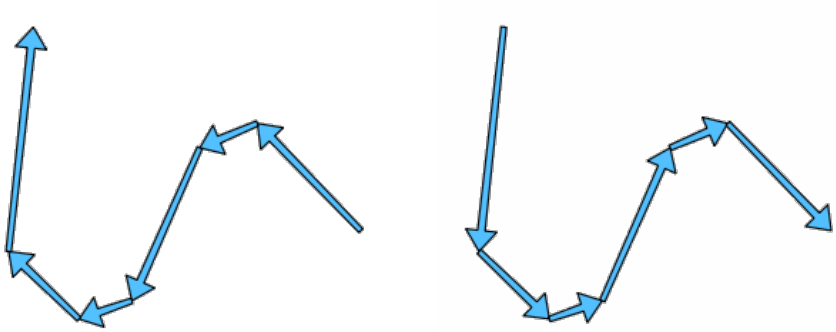


Fig. 9.27: Reversing line direction

Further reading: *Reverse line direction* algorithm

rotate

Returns a rotated version of a geometry. Calculations are in the Spatial Reference System of this geometry.

语法	rotate(geometry, rotation, [center=NULL], [per_part=false]) [] 标记可选参数
参数	• **geometry** - a geometry • **rotation** - clockwise rotation in degrees • **center** - rotation center point. If not specified, the center of the geometry's bounding box is used. • **per_part** - apply rotation per part. If true, then rotation will apply around the center of each part's bounding box when the input geometry is multipart and an explicit rotation center point is not specified.
Examples	• `rotate(@geometry, 45, make_point(4, 5))` → geometry rotated 45 degrees clockwise around the (4, 5) point • `rotate(@geometry, 45)` → geometry rotated 45 degrees clockwise around the center of its bounding box

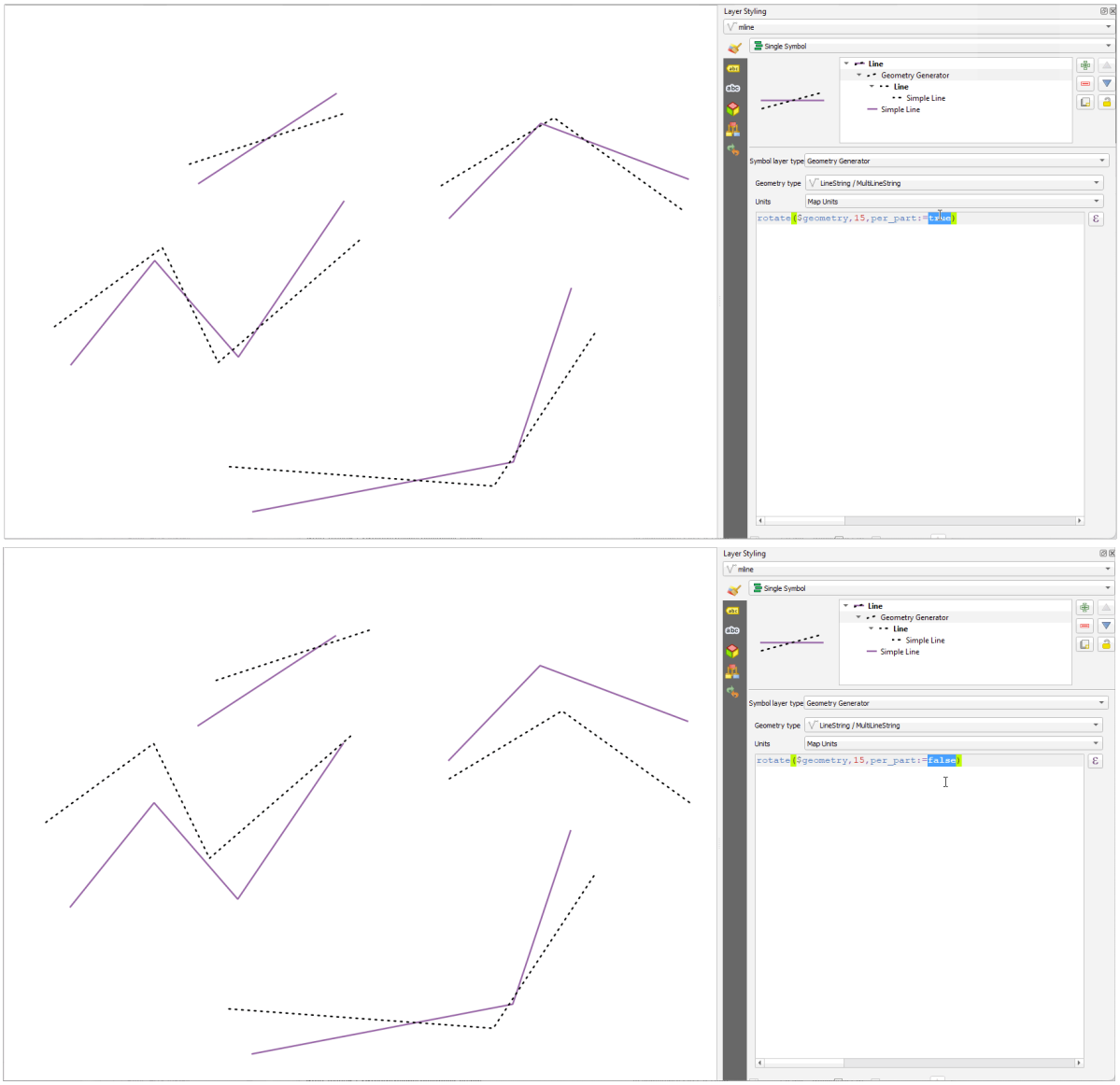


Fig. 9.28: Rotating features

roundness

Calculates how close a polygon shape is to a circle. The function returns 1 when the polygon shape is a perfect circle and 0 when it is completely flat.

语法	roundness(geometry)
参数	• **geometry** - a polygon
Examples	• `round(roundness(geom_from_wkt('POLYGON(( 0 0, 0 1, 1 1, 1 0, 0 0))')), 3)` → 0.785 • `round(roundness(geom_from_wkt('POLYGON(( 0 0, 0 0.1, 1 0.1, 1 0, 0 0))')), 3)` → 0.260

Further reading: *Roundness* algorithm

scale

Returns a scaled version of a geometry. Calculations are in the Spatial Reference System of this geometry.

语法	scale(geometry, x_scale, y_scale, [center]) [] 标记可选参数
参数	• **geometry** - a geometry • **x_scale** - x-axis scaling factor • **y_scale** - y-axis scaling factor • **center** - scaling center point. If not specified, the center of the geometry's bounding box is used.
Examples	• `scale(@geometry, 2, 0.5, make_point(4, 5))` → geometry scaled twice horizontally and halved vertically, around the (4, 5) point • `scale(@geometry, 2, 0.5)` → geometry twice horizontally and halved vertically, around the center of its bounding box

segments_to_lines

Returns a multi line geometry consisting of a line for every segment in the input geometry.

语法	segments_to_lines(geometry)
参数	• **geometry** - geometry object
Examples	• `geom_to_wkt(segments_to_lines(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)')))` → 'MultiLineString ((0 0, 1 1),(1 1, 2 2))'

Further reading: *Explode lines* algorithm

shared_paths

Returns a collection containing paths shared by the two input geometries. Those going in the same direction are in the first element of the collection, those going in the opposite direction are in the second element. The paths themselves are given in the direction of the first geometry.

语法	shared_paths(geometry1, geometry2)
参数	• **geometry1** - a LineString/MultiLineString geometry • **geometry2** - a LineString/MultiLineString geometry
Examples	• `geom_to_wkt(shared_paths(geom_from_wkt('MULTILINESTRING((26 125,26 200,126 200,126 125,26 125),(51 150,101 150,76 175,51 150))'),geom_from_wkt('LINESTRING(151 100,126 156.25,126 125,90 161, 76 175)')))` → 'GeometryCollection (MultiLineString ((126 156.25, 126 125),(101 150, 90 161),(90 161, 76 175)),MultiLineString EMPTY)' • `geom_to_wkt(shared_paths(geom_from_wkt('LINESTRING(76 175,90 161,126 125,126 156.25,151 100)'), geom_from_wkt('MULTILINESTRING((26 125,26 200,126 200,126 125,26 125),(51 150,101 150,76 175,51 150))')))` → 'GeometryCollection (MultiLineString EMPTY,MultiLineString ((76 175, 90 161),(90 161, 101 150),(126 125, 126 156.25)))'

shortest_line

Returns the shortest line joining geometry1 to geometry2. The resultant line will start at geometry1 and end at geometry2.

语法	shortest_line(geometry1, geometry2)
参数	• **geometry1** - geometry to find shortest line from • **geometry2** - geometry to find shortest line to
Examples	• `geom_to_wkt(shortest_line(geom_from_wkt('LINESTRING (20 80, 98 190, 110 180, 50 75 )'),geom_from_wkt('POINT(100 100)')))` → 'LineString(73.0769 115.384, 100 100)'

simplify

Simplifies a geometry by removing nodes using a distance based threshold (ie, the Douglas Peucker algorithm). The algorithm preserves large deviations in geometries and reduces the number of vertices in nearly straight segments.

语法	simplify(geometry, tolerance)
参数	• **geometry** - a geometry • **tolerance** - maximum deviation from straight segments for points to be removed
Examples	• `geom_to_wkt(simplify(geometry:=geom_from_wkt('LineString(0 0, 5 0.1, 10 0)'),tolerance:=5))` → 'LineString(0 0, 10 0)'

Fig. 9.29: From left to right, source layer and increasing simplification tolerances

Further reading: *Simplify* algorithm

simplify_vw

Simplifies a geometry by removing nodes using an area based threshold (ie, the Visvalingam-Whyatt algorithm). The algorithm removes vertices which create small areas in geometries, e.g., narrow spikes or nearly straight segments.

语法	simplify_vw(geometry, tolerance)
参数	• **geometry** - a geometry • **tolerance** - a measure of the maximum area created by a node for the node to be removed
Examples	• `geom_to_wkt(simplify_vw(geometry:=geom_from_wkt('LineString(0 0, 5 0, 5.01 10, 5.02 0, 10 0)'),tolerance:=5))` → 'LineString(0 0, 10 0)'

Further reading: *Simplify* algorithm

single_sided_buffer

Returns a geometry formed by buffering out just one side of a linestring geometry. Distances are in the Spatial Reference System of this geometry.

语法	single_sided_buffer(geometry, distance, [segments=8], [join=1], [miter_limit=2.0]) [] 标记可选参数
参数	• **geometry** - a (multi)linestring geometry • **distance** - buffer distance. Positive values will be buffered to the left of lines, negative values to the right • **segments** - number of segments to use to represent a quarter circle when a round join style is used. A larger number results in a smoother buffer with more nodes. • **join** - join style for corners, where 1 = round, 2 = miter and 3 = bevel • **miter_limit** - limit on the miter ratio used for very sharp corners (when using miter joins only)
Examples	• `single_sided_buffer(@geometry, 10.5)` → line buffered to the left by 10.5 units • `single_sided_buffer(@geometry, -10.5)` → line buffered to the right by 10.5 units • `single_sided_buffer(@geometry, 10.5, segments:=16, join:=1)` → line buffered to the left by 10.5 units, using more segments to result in a smoother buffer • `single_sided_buffer(@geometry, 10.5, join:=3)` → line buffered to the left by 10.5 units, using a beveled join

Fig. 9.30: Left versus right side buffer on the same vector line layer

Further reading: *Single sided buffer* algorithm

sinuosity

Returns the sinuosity of a curve, which is the ratio of the curve length to the straight (2D) distance between its endpoints.

语法	sinuosity(geometry)
参数	• **geometry** - Input curve (circularstring, linestring)
Examples	• `round(sinuosity(geom_from_wkt('LINESTRING(2 0, 2 2, 3 2, 3 3)')), 3)` → 1.265 • `sinuosity(geom_from_wkt('LINESTRING( 3 1, 5 1)'))` → 1.0

smooth

Smooths a geometry by adding extra nodes which round off corners in the geometry. If input geometries contain Z or M values, these will also be smoothed and the output geometry will retain the same dimensionality as the input geometry.

语法	smooth(geometry, [iterations=1], [offset=0.25], [min_length=-1], [max_angle=180]) [] 标记可选参数
参数	• **geometry** - a geometry • **iterations** - number of smoothing iterations to apply. Larger numbers result in smoother but more complex geometries. • **offset** - value between 0 and 0.5 which controls how tightly the smoothed geometry follow the original geometry. Smaller values result in a tighter smoothing, larger values result in looser smoothing. • **min_length** - minimum length of segments to apply smoothing to. This parameter can be used to avoid placing excessive additional nodes in shorter segments of the geometry. • **max_angle** - maximum angle at node for smoothing to be applied (0-180). By lowering the maximum angle intentionally sharp corners in the geometry can be preserved. For instance, a value of 80 degrees will retain right angles in the geometry.
Examples	• `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)'

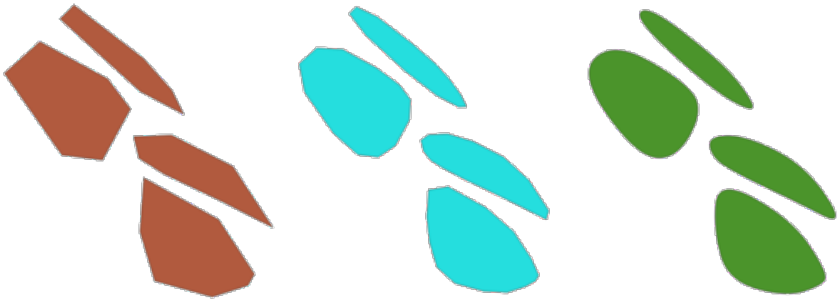


Fig. 9.31: Increasing number of iterations causes smoother geometries

Further reading: *Smooth* algorithm

square_wave

Constructs square/rectangular waves along the boundary of a geometry.

语法	square_wave(geometry, wavelength, amplitude, [strict=False]) [] 标记可选参数
参数	• **geometry** - a geometry • **wavelength** - wavelength of square waveform • **amplitude** - amplitude of square waveform • **strict** - By default the wavelength argument is treated as a "maximum wavelength", where the actual wavelength will be dynamically adjusted so that an exact number of square waves are created along the boundaries of the geometry. If the strict argument is set to true then the wavelength will be used exactly and an incomplete pattern may be used for the final waveform.
Examples	• `square_wave(geom_from_wkt('LineString(0 0, 10 0)'), 3, 1)` → Square waves with wavelength 3 and amplitude 1 along the linestring

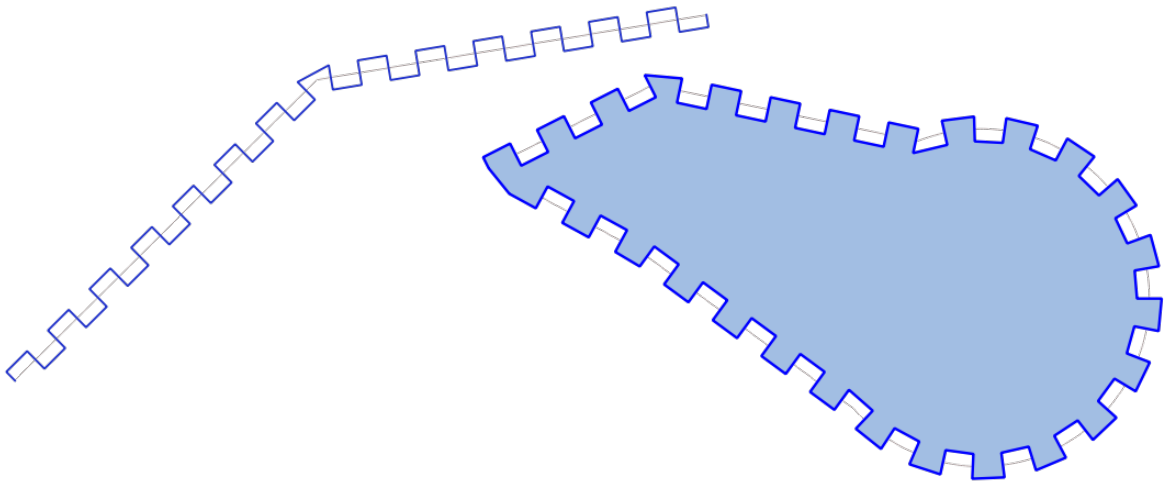


Fig. 9.32: Symbolizing features with square waves

square_wave_randomized

Constructs randomized square/rectangular waves along the boundary of a geometry.

语法	square_wave_randomized(geometry, min_wavelength, max_wavelength, min_amplitude, max_amplitude, [seed=0]) [] 标记可选参数
参数	• **geometry** - a geometry • **min_wavelength** - minimum wavelength of waves • **max_wavelength** - maximum wavelength of waves • **min_amplitude** - minimum amplitude of waves • **max_amplitude** - maximum amplitude of waves • **seed** - specifies a random seed for generating waves. If the seed is 0, then a completely random set of waves will be generated.
Examples	• `square_wave_randomized(geom_from_wkt('LineString(0 0, 10 0)'), 2, 3, 0.1, 0.2)` → Randomly sized square waves with wavelengths between 2 and 3 and amplitudes between 0.1 and 0.2 along the linestring

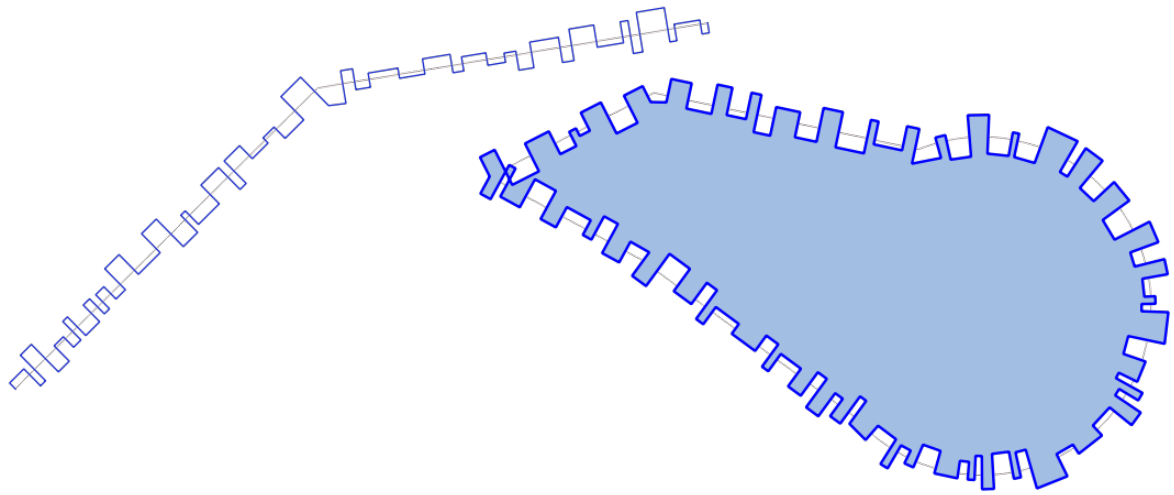


Fig. 9.33: Symbolizing features with square randomized waves

start_point

Returns the first node from a geometry.

语法	start_point(geometry)
参数	• **geometry** - geometry object
Examples	• `geom_to_wkt(start_point(geom_from_wkt('LINESTRING(4 0, 4 2, 0 2)')))` → 'Point (4 0)'

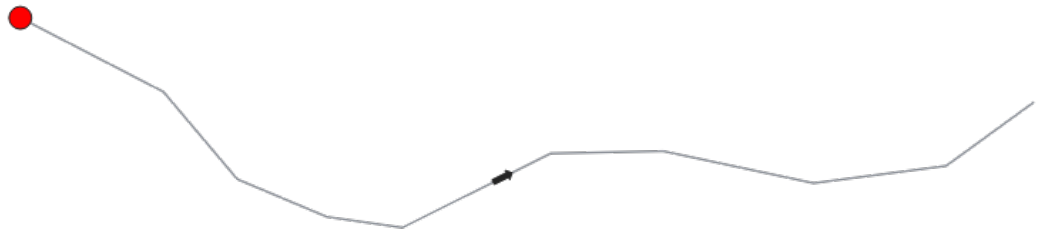


Fig. 9.34: Starting point of a line feature

Further reading: *end_point*, *Extract specific vertices* algorithm

straight_distance_2d

Returns the direct/euclidean distance between the first and last vertex of a geometry. The geometry must be a curve (circularstring, linestring).

语法	straight_distance_2d(geometry)
参数	• **geometry** - The geometry.
Examples	• `straight_distance_2d(geom_from_wkt('LINESTRING(1 0, 1 1)'))` → 1 • `round(straight_distance_2d(geom_from_wkt('LINESTRING(1 4, 3 5, 5 0)')), 3)` → 5.657

Further reading: *length*

sym_difference

Returns a geometry that represents the portions of two geometries that do not intersect.

语法	sym_difference(geometry1, geometry2)
参数	• **geometry1** - a geometry • **geometry2** - a geometry
Examples	• `geom_to_wkt( sym_difference( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'LINESTRING(3 3, 8 8)' ) ) )` → 'LINESTRING(5 5, 8 8)'

Further reading: *Symmetrical difference* algorithm

tapered_buffer

Creates a buffer along a line geometry where the buffer diameter varies evenly over the length of the line.

语法	tapered_buffer(geometry, start_width, end_width, [segments=8]) [] 标记可选参数
参数	• **geometry** - input geometry. Must be a (multi)line geometry. • **start_width** - width of buffer at start of line, • **end_width** - width of buffer at end of line. • **segments** - number of segments to approximate quarter-circle curves in the buffer.
Examples	• `tapered_buffer(geometry:=geom_from_wkt('LINESTRING(1 2, 4 2)'),start_width:=1,end_width:=2,segments:=8)` → A tapered buffer starting with a diameter of 1 and ending with a diameter of 2 along the linestring geometry.

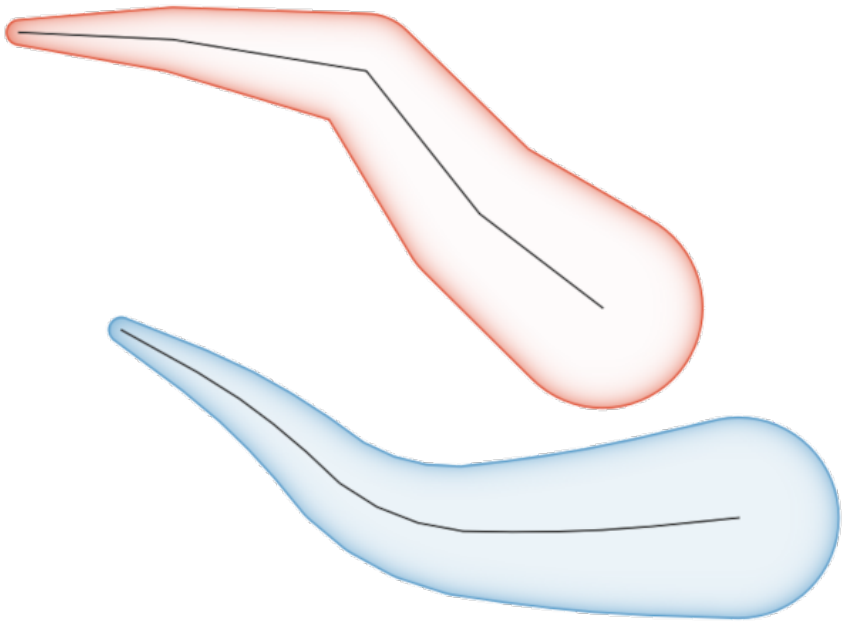


Fig. 9.35: Tapered buffer on line features

Further reading: *Tapered buffers* algorithm

touches

Tests whether a geometry touches another. Returns TRUE if the geometries have at least one point in common, but their interiors do not intersect.

语法	touches(geometry1, geometry2)
参数	• **geometry1** - a geometry • **geometry2** - a geometry
Examples	• `touches( geom_from_wkt( 'LINESTRING(5 3, 4 4)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE • `touches( geom_from_wkt( 'POINT(4 4)' ), geom_from_wkt( 'POINT(5 5)' ) )` → FALSE

Further reading: *overlay_touches*

transform

Returns the geometry transformed from a source CRS to a destination CRS.

语法	transform(geometry, source_auth_id, dest_auth_id)
参数	• **geometry** - a geometry • **source_auth_id** - the source CRS definition or CRS object • **dest_auth_id** - the destination CRS definition or CRS object
Examples	• `geom_to_wkt( transform( make_point(488995.53240249, 7104473.38600835), 'EPSG:2154', 'EPSG:4326' ) )` → 'POINT(0 51)'

Further reading: *Reproject layer* algorithm

translate

Returns a translated version of a geometry. Calculations are in the Spatial Reference System of this geometry.

语法	translate(geometry, dx, dy)
参数	• **geometry** - a geometry • **dx** - delta x • **dy** - delta y
Examples	• `translate(@geometry, 5, 10)` → a geometry of the same type like the original one

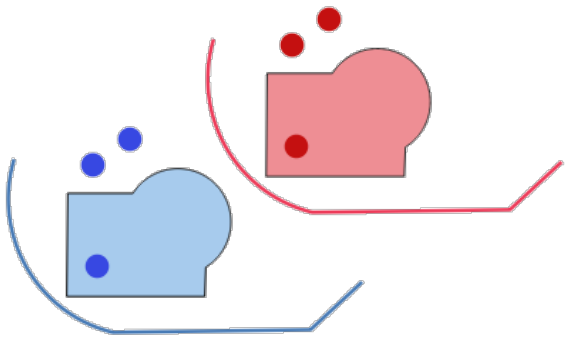


Fig. 9.36: Translating features

Further reading: 翻译 algorithm

triangular_wave

Constructs triangular waves along the boundary of a geometry.

语法	triangular_wave(geometry, wavelength, amplitude, [strict=False]) [] 标记可选参数
参数	• **geometry** - a geometry • **wavelength** - wavelength of triangular waveform • **amplitude** - amplitude of triangular waveform • **strict** - By default the wavelength argument is treated as a "maximum wavelength", where the actual wavelength will be dynamically adjusted so that an exact number of triangular waves are created along the boundaries of the geometry. If the strict argument is set to true then the wavelength will be used exactly and an incomplete pattern may be used for the final waveform.
Examples	• `triangular_wave(geom_from_wkt('LineString(0 0, 10 0)'), 3, 1)` → Triangular waves with wavelength 3 and amplitude 1 along the linestring

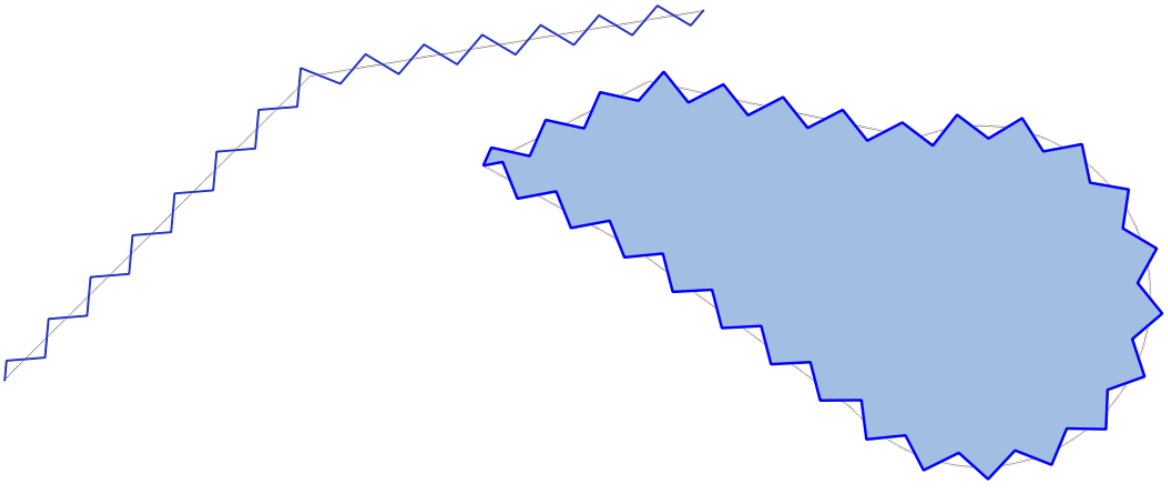


Fig. 9.37: Symbolizing features with triangular waves

triangular_wave_randomized

Constructs randomized triangular waves along the boundary of a geometry.

语法	triangular_wave_randomized(geometry, min_wavelength, max_wavelength, min_amplitude, max_amplitude, [seed=0]) [] 标记可选参数
参数	• **geometry** - a geometry • **min_wavelength** - minimum wavelength of waves • **max_wavelength** - maximum wavelength of waves • **min_amplitude** - minimum amplitude of waves • **max_amplitude** - maximum amplitude of waves • **seed** - specifies a random seed for generating waves. If the seed is 0, then a completely random set of waves will be generated.
Examples	• `triangular_wave_randomized(geom_from_wkt('LineString(0 0, 10 0)'), 2, 3, 0.1, 0.2)` → Randomly sized triangular waves with wavelengths between 2 and 3 and amplitudes between 0.1 and 0.2 along the linestring

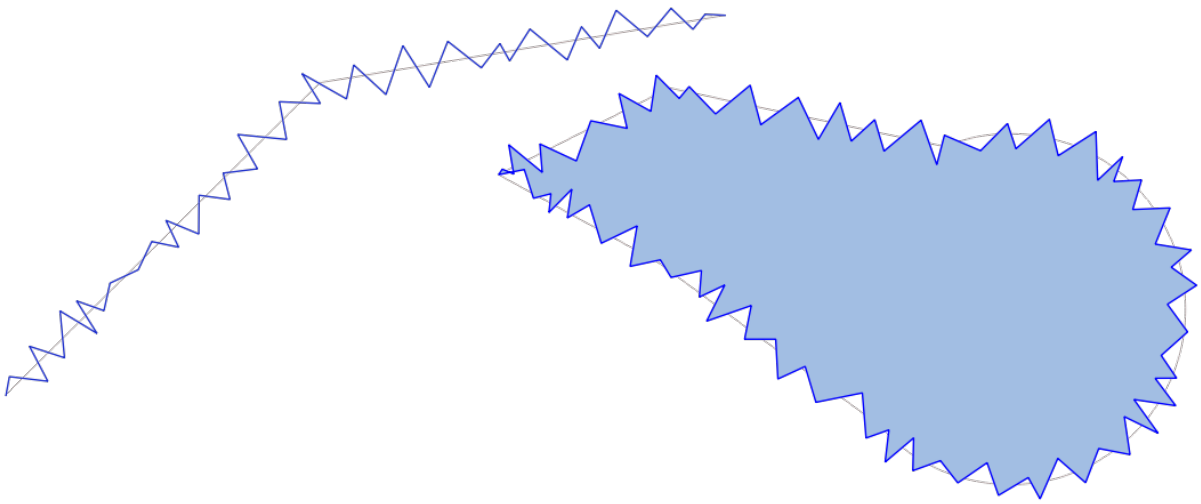


Fig. 9.38: Symbolizing features with triangular randomized waves

union

Returns a geometry that represents the point set union of the geometries.

语法	union(geometry1, geometry2)
参数	• **geometry1** - a geometry • **geometry2** - a geometry
Examples	• `geom_to_wkt( union( make_point(4, 4), make_point(5, 5) ) )` → 'MULTIPOINT(4 4, 5 5)'

wave

Constructs rounded (sine-like) waves along the boundary of a geometry.

语法	wave(geometry, wavelength, amplitude, [strict=False]) [] 标记可选参数
参数	• **geometry** - a geometry • **wavelength** - wavelength of sine-like waveform • **amplitude** - amplitude of sine-like waveform • **strict** - By default the wavelength argument is treated as a "maximum wavelength", where the actual wavelength will be dynamically adjusted so that an exact number of waves are created along the boundaries of the geometry. If the strict argument is set to true then the wavelength will be used exactly and an incomplete pattern may be used for the final waveform.
Examples	• `wave(geom_from_wkt('LineString(0 0, 10 0)'), 3, 1)` → Sine-like waves with wavelength 3 and amplitude 1 along the linestring

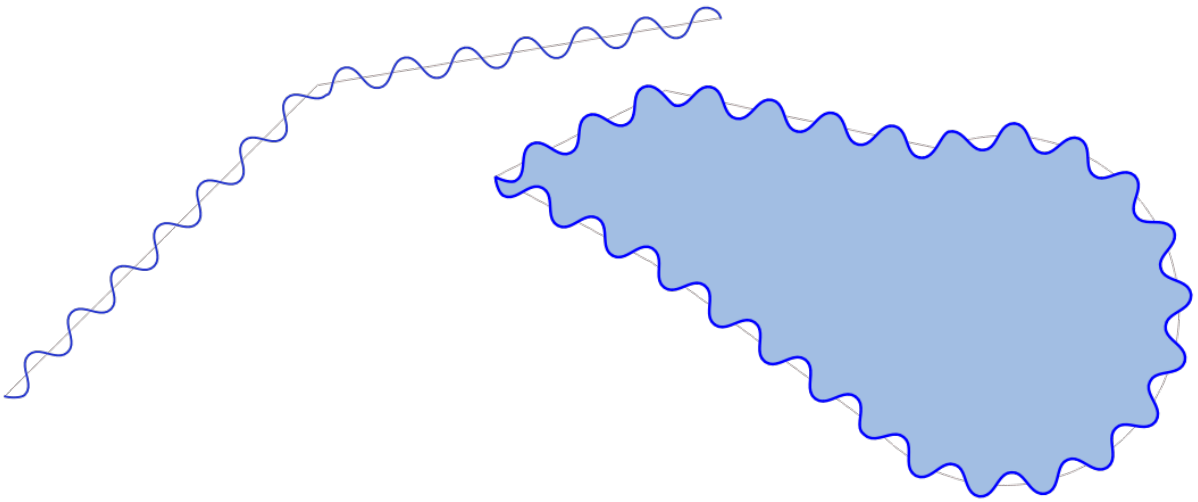


Fig. 9.39: Symbolizing features with waves

wave_randomized

Constructs randomized curved (sine-like) waves along the boundary of a geometry.

语法	wave_randomized(geometry, min_wavelength, max_wavelength, min_amplitude, max_amplitude, [seed=0]) [] 标记可选参数
参数	• **geometry** - a geometry • **min_wavelength** - minimum wavelength of waves • **max_wavelength** - maximum wavelength of waves • **min_amplitude** - minimum amplitude of waves • **max_amplitude** - maximum amplitude of waves • **seed** - specifies a random seed for generating waves. If the seed is 0, then a completely random set of waves will be generated.
Examples	• `wave_randomized(geom_from_wkt('LineString(0 0, 10 0)'), 2, 3, 0.1, 0.2)` → Randomly sized curved waves with wavelengths between 2 and 3 and amplitudes between 0.1 and 0.2 along the linestring

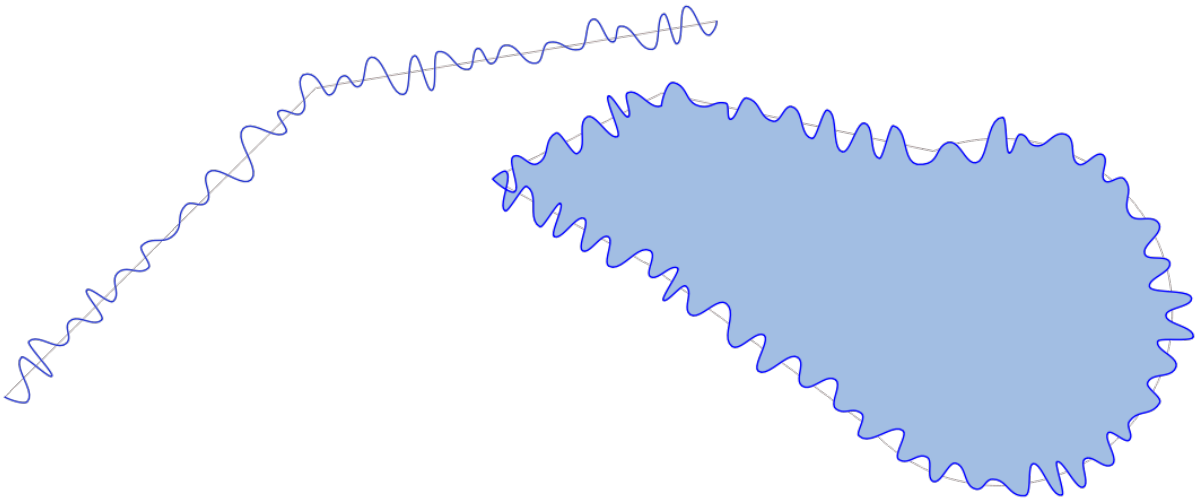


Fig. 9.40: Symbolizing features with randomized waves

wedge_buffer

Returns a wedge shaped buffer originating from a point geometry.

语法	wedge_buffer(center, azimuth, width, outer_radius, [inner_radius=0.0]) [] 标记可选参数
参数	• **center** - center point (origin) of buffer. Must be a point geometry. • **azimuth** - angle (in degrees) for the middle of the wedge to point. • **width** - buffer width (in degrees). Note that the wedge will extend to half of the angular width either side of the azimuth direction. • **outer_radius** - outer radius for buffers • **inner_radius** - optional inner radius for buffers
Examples	• `wedge_buffer(center:=geom_from_wkt('POINT(1 2)'), azimuth:=90,width:=180,outer_radius:=1)` → A wedge shaped buffer centered on the point (1,2), facing to the East, with a width of 180 degrees and outer radius of 1.

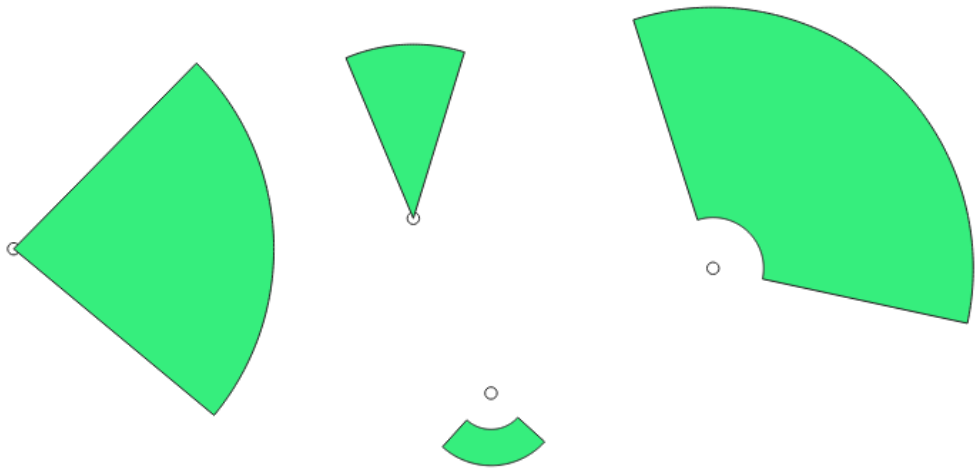


Fig. 9.41: Wedge buffering features

Further reading: *Create wedge buffers* algorithm

within

Tests whether a geometry is within another. Returns TRUE if the geometry1 is completely within geometry2.

语法	within(geometry1, geometry2)
参数	• **geometry1** - a geometry • **geometry2** - a geometry
Examples	• `within( geom_from_wkt( 'POINT( 0.5 0.5)' ), geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))' ) )` → TRUE • `within( geom_from_wkt( 'POINT( 5 5 )' ), geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0 ))' ) )` → FALSE

Further reading: *overlay_within*

$x

Returns the x coordinate of the current point feature. If the feature is a multipoint feature, then the x-coordinate of the first point will be returned. **WARNING: This function is deprecated. It is recommended to use the replacement x() function with @geometry variable instead.**

语法	$x
Examples	• `$x` → 42

Further reading: *x*

x

Returns the x coordinate of a point geometry, or the x coordinate of the centroid for a non-point geometry.

语法	x(geometry)
参数	• **geometry** - a geometry
Examples	• `x( geom_from_wkt( 'POINT(2 5)' ) )` → 2 • `x( @geometry )` → x coordinate of the current feature's centroid

$x_at

Retrieves a x coordinate of the current feature's geometry. **WARNING: This function is deprecated. It is recommended to use the replacement x_at function with @geometry variable instead.**

语法	$x_at(vertex)
参数	• **vertex** - index of the vertex of the current geometry (indices start at 0; negative values apply from the last index, starting at -1)
Examples	• `$x_at(1)` → 5

Further reading: *x_at*

x_at

Retrieves a x coordinate of the geometry.

语法	x_at(geometry, vertex)
参数	• **geometry** - geometry object • **vertex** - index of the vertex of the geometry (indices start at 0; negative values apply from the last index, starting at -1)
Examples	• `x_at( geom_from_wkt( 'POINT(4 5)' ), 0 )` → 4

x_max

Returns the maximum x coordinate of a geometry. Calculations are in the spatial reference system of this geometry.

语法	x_max(geometry)
参数	• **geometry** - a geometry
Examples	• `x_max( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 4

x_min

Returns the minimum x coordinate of a geometry. Calculations are in the spatial reference system of this geometry.

语法	x_min(geometry)
参数	• **geometry** - a geometry
Examples	• `x_min( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 2

$y

Returns the y coordinate of the current point feature. If the feature is a multipoint feature, then the y-coordinate of the first point will be returned. **WARNING: This function is deprecated. It is recommended to use the replacement y() function with @geometry variable instead.**

语法	$y
Examples	• `$y` → 42

Further reading: *y*

y

Returns the y coordinate of a point geometry, or the y coordinate of the centroid for a non-point geometry.

语法	y(geometry)
参数	• **geometry** - a geometry
Examples	• `y( geom_from_wkt( 'POINT(2 5)' ) )` → 5 • `y( @geometry )` → y coordinate of the current feature's centroid

$y_at

Retrieves a y coordinate of the current feature's geometry. **WARNING: This function is deprecated. It is recommended to use the replacement y_at function with @geometry variable instead.**

语法	$y_at(vertex)
参数	• **vertex** - index of the vertex of the current geometry (indices start at 0; negative values apply from the last index, starting at -1)
Examples	• `$y_at(1)` → 2

Further reading: *y_at*

y_at

Retrieves a y coordinate of the geometry.

语法	y_at(geometry, vertex)
参数	• **geometry** - geometry object • **vertex** - index of the vertex of the geometry (indices start at 0; negative values apply from the last index, starting at -1)
Examples	• `y_at( geom_from_wkt( 'POINT(4 5)' ), 0 )` → 5

y_max

Returns the maximum y coordinate of a geometry. Calculations are in the spatial reference system of this geometry.

语法	y_max(geometry)
参数	• **geometry** - a geometry
Examples	• `y_max( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 8

y_min

Returns the minimum y coordinate of a geometry. Calculations are in the spatial reference system of this geometry.

语法	y_min(geometry)
参数	• **geometry** - a geometry
Examples	• `y_min( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 5

$z

Returns the z value of the current point feature if it is 3D. If the feature is a multipoint feature, then the z value of the first point will be returned. **WARNING: This function is deprecated. It is recommended to use the replacement z() function with @geometry variable instead.**

语法	$z
Examples	• `$z` → 123

z

Returns the z coordinate of a point geometry, or NULL if the geometry has no z value.

语法	z(geometry)
参数	• **geometry** - a point geometry
Examples	• `z( geom_from_wkt( 'POINTZ(2 5 7)' ) )` → 7

z_at

Retrieves a z coordinate of the geometry, or NULL if the geometry has no z value.

语法	z_at(geometry, vertex)
参数	• **geometry** - geometry object • **vertex** - index of the vertex of the geometry (indices start at 0; negative values apply from the last index, starting at -1)
Examples	• `z_at(geom_from_wkt('LineStringZ(0 0 0, 10 10 5, 10 10 0)'), 1)` → 5

z_max

Returns the maximum z coordinate of a geometry, or NULL if the geometry has no z value.

语法	z_max(geometry)
参数	• **geometry** - a geometry with z coordinate
Examples	• `z_max( geom_from_wkt( 'POINT ( 0 0 1 )' ) )` → 1 • `z_max( geom_from_wkt( 'MULTIPOINT ( 0 0 1 , 1 1 3 )' ) )` → 3 • `z_max( make_line( make_point( 0,0,0 ), make_point( -1,-1,-2 ) ) )` → 0 • `z_max( geom_from_wkt( 'LINESTRING( 0 0 0, 1 0 2, 1 1 -1 )' ) )` → 2 • `z_max( geom_from_wkt( 'POINT ( 0 0 )' ) )` → NULL

z_min

Returns the minimum z coordinate of a geometry, or NULL if the geometry has no z value.

语法	z_min(geometry)
参数	• **geometry** - a geometry with z coordinate
Examples	• `z_min( geom_from_wkt( 'POINT ( 0 0 1 )' ) )` → 1 • `z_min( geom_from_wkt( 'MULTIPOINT ( 0 0 1 , 1 1 3 )' ) )` → 1 • `z_min( make_line( make_point( 0,0,0 ), make_point( -1,-1, -2 ) ) )` → -2 • `z_min( geom_from_wkt( 'LINESTRING( 0 0 0, 1 0 2, 1 1 -1 )' ) )` → -1 • `z_min( geom_from_wkt( 'POINT ( 0 0 )' ) )` → NULL

9.2.15 Layout Functions

This group contains functions to manipulate print layout items properties.

item_variables

Returns a map of variables from a layout item inside this print layout.

语法	item_variables(id)
参数	• **id** - layout item ID
Examples	• `map_get( item_variables('Map 0'), 'map_scale')` → scale of the item 'Map 0' in the current print layout

Further reading: List of default *variables*

map_credits

Returns a list of credit (usage rights) strings for the layers shown in a layout, or specific layout map item.

语法	map_credits([id], [include_layer_names=false], [layer_name_separator=': ']) [] 标记可选参数
参数	• **id** - Map item ID. If not specified, the layers from all maps in the layout will be used. • **include_layer_names** - Set to true to include layer names before their credit strings • **layer_name_separator** - String to insert between layer names and their credit strings, if include_layer_names is true
Examples	• `array_to_string( map_credits() )` → comma separated list of layer credits for all layers shown in all map items in the layout, e.g 'CC-BY-NC, CC-BY-SA' • `array_to_string( map_credits( 'Main Map' ) )` → comma separated list of layer credits for layers shown in the 'Main Map' layout item, e.g 'CC-BY-NC, CC-BY-SA' • `array_to_string( map_credits( 'Main Map', include_layer_names := true, layer_name_separator := ': ' ) )` → comma separated list of layer names and their credits for layers shown in the 'Main Map' layout item, e.g. 'Railway lines: CC-BY-NC, Basemap: CC-BY-SA'

This function requires the *Access metadata properties* of the layers to have been filled.

9.2.16 Map Layers

This group contains a list of the available layers in the current project and, for each layer, their fields (stored in the dataset, virtual or auxiliary ones as well as from joins). The fields can be interacted the same way as mentioned in *Fields and Values*, except that a double-click will add the name as a string (single quoted) to the expression instead of as a field reference given that they do not belong to the active layer. This offers a convenient way to write expressions referring to different layers, such as when performing *aggregates*, *attribute* or *spatial* queries.

It also provides some convenient functions to manipulate layers.

decode_uri

Takes a layer and decodes the uri of the underlying data provider. It depends on the dataprovider, which data is available.

语法	decode_uri(layer, [part]) [] 标记可选参数
参数	• **layer** - The layer for which the uri should be decoded. • **part** - The part of the uri to return. If unspecified, a map with all uri parts will be returned.
Examples	• `decode_uri(@layer)` → {'layerId': '0', 'layerName': '', 'path': '/home/qgis/shapefile.shp'} • `decode_uri(@layer)` → {'layerId': NULL, 'layerName': 'layer', 'path': '/home/qgis/geopackage.gpkg'} • `decode_uri(@layer, 'path')` → 'C:\my_data\qgis\shape.shp'

layer_property

Returns a matching layer property or metadata value.

语法	layer_property(layer, property)
参数	• **图层** - 字符串, 表示图层名称或图层 ID • **property** - a string corresponding to the property to return. Valid options are: – name: layer name – id: layer ID – title: metadata title string – abstract: metadata abstract string – keywords: metadata keywords – data_url: metadata URL – attribution: metadata attribution string – attribution_url: metadata attribution URL – source: layer source – min_scale: minimum display scale for layer – max_scale: maximum display scale for layer – is_editable: if layer is in edit mode – crs: layer CRS – crs_definition: layer CRS full definition – crs_description: layer CRS description – crs_ellipsoid: acronym of the layer CRS ellipsoid – extent: layer extent (as a geometry object) – distance_units: layer distance units – type: layer type, e.g., Vector or Raster – storage_type: storage format (vector layers only) – geometry_type: geometry type, e.g., Point (vector layers only) – feature_count: approximate feature count for layer (vector layers only) – path: File path to the layer data source. Only available for file based layers.
Examples	• `layer_property('streets','title')` → 'Basemap Streets' • `layer_property('airports','feature_count')` → 120 • `layer_property('landsat','crs')` → 'EPSG:4326'

Further reading: *vector*, *raster* and *mesh* layer properties

load_layer

Loads a layer by source URI and provider name.

语法	load_layer(uri, provider)
参数	• **uri** - layer source URI string • **provider** - layer data provider name
Examples	• `layer_property(load_layer('c:/data/roads.shp', 'ogr'), 'feature_count')` → count of features from the c:/data/roads.shp vector layer

9.2.17 Maps Functions

This group contains functions to create or manipulate keys and values of map data structures (also known as dictionary objects, key-value pairs, or associative arrays). Unlike the *list data structure* where values order matters, the order of the key-value pairs in the map object is not relevant and values are identified by their keys.

from_json

Loads a JSON formatted string.

语法	from_json(string)
参数	• **string** - JSON string
Examples	• `from_json('{"1":"one","2":"two"}')` → { '1': 'one', '2': 'two' } • `from_json('[1,2,3]')` → [1,2,3]

hstore_to_map

Creates a map from a hstore-formatted string.

语法	hstore_to_map(string)
参数	• **string** - the input string
Examples	• `hstore_to_map('qgis=>rocks')` → { 'qgis': 'rocks' }

map

Returns a map containing all the keys and values passed as pair of parameters.

语法	map(key1, value1, key2, value2, ...)
参数	• **key** - a key (string) • **value** - a value
Examples	• `map('1','one','2', 'two')` → { '1': 'one', '2': 'two' } • `map('1','one','2', 'two')['1']` → 'one'

map_akeys

Returns all the keys of a map as an array.

语法	map_akeys(map)
参数	• **map** - a map
Examples	• `map_akeys(map('1','one','2','two'))` → ['1', '2']

map_avals

Returns all the values of a map as an array.

语法	map_avals(map)
参数	• **map** - a map
Examples	• `map_avals(map('1','one','2','two'))` → ['one', 'two']

map_concat

Returns a map containing all the entries of the given maps. If two maps contain the same key, the value of the second map is taken.

语法	map_concat(map1, map2, ...)
参数	• **map** - a map
Examples	• `map_concat(map('1','one', '2','overridden'),map('2','two', '3','three'))` → { '1': 'one', '2': 'two', '3': 'three' }

map_delete

Returns a map with the given key and its corresponding value deleted.

语法	map_delete(map, key)
参数	• **map** - a map • **key** - the key to delete
Examples	• `map_delete(map('1','one','2','two'),'2')` → { '1': 'one' }

map_exist

Returns TRUE if the given key exists in the map.

语法	map_exist(map, key)
参数	• **map** - a map • **key** - the key to lookup
Examples	• `map_exist(map('1','one','2','two'),'3')` → FALSE

map_get

Returns the value of a map, given its key. Returns NULL if the key does not exist.

语法	map_get(map, key)
参数	• **map** - a map • **key** - the key to lookup
Examples	• `map_get(map('1','one','2','two'),'2')` → 'two' • `map_get( item_variables('Map 0'), 'map_scale')` → scale of the item 'Map 0' (if it exists) in the current print layout

Hint

You can also use the *index operator ([])* to get a value from a map.

map_insert

Returns a map with an added key/value. If the key already exists, its value is overridden.

语法	map_insert(map, key, value)
参数	• **map** - a map • **key** - the key to add • **value** - the value to add
Examples	• `map_insert(map('1','one'),'3','three')` → { '1': 'one', '3': 'three' } • `map_insert(map('1','one','2','overridden'),'2','two')` → { '1': 'one', '2': 'two' }

map_prefix_keys

Returns a map with all keys prefixed by a given string.

语法	map_prefix_keys(map, prefix)
参数	• **map** - a map • **prefix** - a string
Examples	• `map_prefix_keys(map('1','one','2','two'), 'prefix-')` → { 'prefix-1': 'one', 'prefix-2': 'two' }

map_to_hstore

Merge map elements into a hstore-formatted string.

语法	map_to_hstore(map)
参数	• **map** - the input map
Examples	• `map_to_hstore(map('1','one','2','two'))` → '"1"=>"one","2"=>"two"'

map_to_html_dl

Merge map elements into a HTML definition list string.

语法	map_to_html_dl(map)
参数	• **map** - the input map
Examples	• `map_to_html_dl(map('1','one','2','two'))` → <dl><dt>1</dt><dd>one</dd><dt>2</dt><dd>two</dd></dl>

map_to_html_table

Merge map elements into a HTML table string.

语法	map_to_html_table(map)
参数	• **map** - the input map
Examples	• `map_to_html_table(map('1','one','2','two'))` → <table><thead><tr><th>1</th><th>2</th></tr></thead><tbody><tr><td>one</td><td>two</td></tr></tbody

to_json

Create a JSON formatted string from a map, array or other value.

语法	to_json(value)
参数	• **value** - The input value
Examples	• `to_json(map('1','one','2','two'))` → {"1":"one","2":"two"} • `to_json(array(1,2,3))` → [1,2,3]

url_encode

Returns an URL encoded string from a map. Transforms all characters in their properly-encoded form producing a fully-compliant query string.

Note that the plus sign '+' is not converted.

语法	url_encode(map)
参数	• **map** - a map.
Examples	• `url_encode(map('a&+b', 'a and plus b', 'a=b', 'a equals b'))` → 'a%26+b=a%20and%20plus%20b&a%3Db=a%20equals%20b'

9.2.18 Mathematical Functions

This group contains math functions (e.g., square root, sin and cos).

abs

Returns the absolute value of a number.

语法	abs(value)
参数	• **value** - a number
Examples	• `abs(-2)` → 2

acos

Returns the inverse cosine of a value in radians.

语法	acos(value)
参数	• **value** - cosine of an angle in radians
Examples	• `acos(0.5)` → 1.0471975511966

asin

Returns the inverse sine of a value in radians.

语法	asin(value)
参数	• **value** - sine of an angle in radians
Examples	• `asin(1.0)` → 1.5707963267949

atan

Returns the inverse tangent of a value in radians.

语法	atan(value)
参数	• **value** - tan of an angle in radians
Examples	• `atan(0.5)` → 0.463647609000806

atan2

Returns the inverse tangent of dy/dx by using the signs of the two arguments to determine the quadrant of the result.

语法	atan2(dy, dx)
参数	• **dy** - y coordinate difference • **dx** - x coordinate difference
Examples	• `atan2(1.0, 1.732)` → 0.523611477769969

ceil

Rounds a number upwards.

语法	ceil(value)
参数	• **value** - a number
Examples	• `ceil(4.9)` → 5 • `ceil(-4.9)` → -4

clamp

Restricts an input value to a specified range.

语法	clamp(minimum, input, maximum)
参数	• **minimum** - the smallest value *input* is allowed to take. • **input** - a value which will be restricted to the range specified by *minimum* and *maximum* • **maximum** - the largest value *input* is allowed to take
Examples	• `clamp(1,5,10)` → 5 *input* is between 1 and 10 so is returned unchanged • `clamp(1,0,10)` → 1 *input* is less than minimum value of 1, so function returns 1 • `clamp(1,11,10)` → 10 *input* is greater than maximum value of 10, so function returns 10

cos

Returns cosine of an angle.

语法	cos(angle)
参数	• **angle** - angle in radians
Examples	• `cos(1.571)` → 0.000796326710733263

degrees

Converts from radians to degrees.

语法	degrees(radians)
参数	• **radians** - numeric value
Examples	• `degrees(3.14159)` → 180 • `degrees(1)` → 57.2958

exp

Returns exponential of an value.

语法	exp(value)
参数	• **value** - number to return exponent of
Examples	• `exp(1.0)` → 2.71828182845905

floor

Rounds a number downwards.

语法	floor(value)
参数	• **value** - a number
Examples	• `floor(4.9)` → 4 • `floor(-4.9)` → -5

ln

Returns the natural logarithm of a value.

语法	ln(value)
参数	• **value** - numeric value
Examples	• `ln(1)` → 0 • `ln(2.7182818284590452354)` → 1

log

Returns the value of the logarithm of the passed value and base.

语法	log(base, value)
参数	• **base** - any positive number • **value** - any positive number
Examples	• `log(2, 32)` → 5 • `log(0.5, 32)` → -5

log10

Returns the value of the base 10 logarithm of the passed expression.

语法	log10(value)
参数	• **value** - any positive number
Examples	• `log10(1)` → 0 • `log10(100)` → 2

max

Returns the largest value in a set of values.

语法	max(value1, value2, ...)
参数	• **value** - a number
Examples	• `max(2,10.2,5.5)` → 10.2 • `max(20.5,NULL,6.2)` → 20.5

min

Returns the smallest value in a set of values.

语法	min(value1, value2, ...)
参数	• **value** - a number
Examples	• `min(20.5,10,6.2)` → 6.2 • `min(2,-10.3,NULL)` → -10.3

pi

Returns value of pi for calculations.

语法	pi()
Examples	• `pi()` → 3.14159265358979

radians

Converts from degrees to radians.

语法	radians(degrees)
参数	• **degrees** - numeric value
Examples	• `radians(180)` → 3.14159 • `radians(57.2958)` → 1

rand

Returns a random integer within the range specified by the minimum and maximum argument (inclusive). If a seed is provided, the returned will always be the same, depending on the seed.

语法	rand(min, max, [seed=NULL]) [] 标记可选参数
参数	• **min** - an integer representing the smallest possible random number desired • **max** - an integer representing the largest possible random number desired • **seed** - any value to use as seed
Examples	• `rand(1, 10)` → 8

randf

Returns a random float within the range specified by the minimum and maximum argument (inclusive). If a seed is provided, the returned will always be the same, depending on the seed.

语法	randf([min=0.0], [max=1.0], [seed=NULL]) [] 标记可选参数
参数	• **min** - an float representing the smallest possible random number desired • **max** - an float representing the largest possible random number desired • **seed** - any value to use as seed
Examples	• `randf(1, 10)` → 4.59258286403147

round

Rounds a number to number of decimal places.

语法	round(value, [places=0]) [] 标记可选参数
参数	• **value** - decimal number to be rounded • **places** - Optional integer representing number of places to round decimals to. Can be negative.
Examples	• `round(1234.567, 2)` → 1234.57 • `round(1234.567)` → 1235 • `round(1234.567, -1)` → 1230

scale_exponential

Transforms a given value from an input domain to an output range using an exponential curve. This function can be used to ease values in or out of the specified output range.

语法	scale_exponential(value, domain_min, domain_max, range_min, range_max, exponent)
参数	• **value** - A value in the input domain. The function will return a corresponding scaled value in the output range. • **domain_min** - Specifies the minimum value in the input domain, the smallest value the input value should take. • **domain_max** - Specifies the maximum value in the input domain, the largest value the input value should take. • **range_min** - Specifies the minimum value in the output range, the smallest value which should be output by the function. • **range_max** - Specifies the maximum value in the output range, the largest value which should be output by the function. • **exponent** - A positive value (greater than 0), which dictates the way input values are mapped to the output range. Large exponents will cause the output values to 'ease in', starting slowly before accelerating as the input values approach the domain maximum. Smaller exponents (less than 1) will cause output values to 'ease out', where the mapping starts quickly but slows as it approaches the domain maximum.
Examples	• `scale_exponential(5,0,10,0,100,2)` → 3.030 easing in, using an exponent of 2 • `scale_exponential(3,0,10,0,100,0.5)` → 87.585 easing out, using an exponent of 0.5

scale_linear

Transforms a given value from an input domain to an output range using linear interpolation.

语法	scale_linear(value, domain_min, domain_max, range_min, range_max)
参数	• **value** - A value in the input domain. The function will return a corresponding scaled value in the output range. • **domain_min** - Specifies the minimum value in the input domain, the smallest value the input value should take. • **domain_max** - Specifies the maximum value in the input domain, the largest value the input value should take. • **range_min** - Specifies the minimum value in the output range, the smallest value which should be output by the function. • **range_max** - Specifies the maximum value in the output range, the largest value which should be output by the function.
Examples	• `scale_linear(5,0,10,0,100)` → 50 • `scale_linear(0.2,0,1,0,360)` → 72 scaling a value between 0 and 1 to an angle between 0 and 360 • `scale_linear(1500,1000,10000,9,20)` → 9.6111111 scaling a population which varies between 1000 and 10000 to a font size between 9 and 20

scale_polynomial

Transforms a given value from an input domain to an output range using a polynomial curve. This function can be used to ease values in or out of the specified output range.

语法	scale_polynomial(value, domain_min, domain_max, range_min, range_max, exponent)
参数	• **value** - A value in the input domain. The function will return a corresponding scaled value in the output range. • **domain_min** - Specifies the minimum value in the input domain, the smallest value the input value should take. • **domain_max** - Specifies the maximum value in the input domain, the largest value the input value should take. • **range_min** - Specifies the minimum value in the output range, the smallest value which should be output by the function. • **range_max** - Specifies the maximum value in the output range, the largest value which should be output by the function. • **exponent** - A positive value (greater than 0), which dictates the way input values are mapped to the output range. Large exponents will cause the output values to 'ease in', starting slowly before accelerating as the input values approach the domain maximum. Smaller exponents (less than 1) will cause output values to 'ease out', where the mapping starts quickly but slows as it approaches the domain maximum.
Examples	• `scale_polynomial(5,0,10,0,100,2)` → 25 easing in, using an exponent of 2 • `scale_polynomial(3,0,10,0,100,0.5)` → 54.772 easing out, using an exponent of 0.5

sin

Returns the sine of an angle.

语法	sin(angle)
参数	• **angle** - angle in radians
Examples	• `sin(1.571)` → 0.999999682931835

sqrt

Returns square root of a value.

语法	sqrt(value)
参数	• **value** - a number
Examples	• `sqrt(9)` → 3

tan

Returns the tangent of an angle.

语法	tan(angle)
参数	• **angle** - angle in radians
Examples	• `tan(1.0)` → 1.5574077246549

9.2.19 Meshes Functions

This group contains functions which calculate or return mesh related values.

$face_area

Returns the area of the current mesh face. The area calculated by this function respects both the current project's ellipsoid setting and area unit settings. For example, if an ellipsoid has been set for the project then the calculated area will be ellipsoidal, and if no ellipsoid is set then the calculated area will be planimetric.

语法	$face_area
Examples	• `$face_area` → 42

$face_index

Returns the index of the current mesh face.

语法	$face_index
Examples	• `$face_index` → 4581

$vertex_as_point

Returns the current vertex as a point geometry.

语法	$vertex_as_point
Examples	• `geom_to_wkt( $vertex_as_point )` → 'POINT(800 1500 41)'

$vertex_index

Returns the index of the current mesh vertex.

语法	$vertex_index
Examples	• `$vertex_index` → 9874

$vertex_x

Returns the X coordinate of the current mesh vertex.

语法	$vertex_x
Examples	• `$vertex_x` → 42.12

$vertex_y

Returns the Y coordinate of the current mesh vertex.

语法	$vertex_y
Examples	• `$vertex_y` → 12.24

$vertex_z

Returns the Z value of the current mesh vertex.

语法	$vertex_z
Examples	• `$vertex_z` → 42

9.2.20 Operators

This group contains operators (e.g., +, -, *). Note that for most of the mathematical functions below, if one of the inputs is NULL then the result is NULL.

%

Remainder of division. Takes the sign of the dividend.

语法	a % b
参数	• **a** - value • **b** - value
Examples	• `9 % 2` → 1 • `9 % -2` → 1 • `-9 % 2` → -1 • `5 % NULL` → NULL

*

Multiplication of two values

语法	a * b
参数	• **a** - value • **b** - value
Examples	• `5 * 4` → 20 • `5 * NULL` → NULL

+

Addition of two values. If one of the values is NULL the result will be NULL.

语法	a + b
参数	• **a** - value • **b** - value
Examples	• `5 + 4` → 9 • `5 + NULL` → NULL • `'QGIS ' + 'ROCKS'` → 'QGIS ROCKS' • `to_datetime('2020-08-01 12:00:00') + '1 day 2 hours'` → 2020-08-02T14:00:00

Further reading: *concat*, ||

-

Subtraction of two values. If one of the values is NULL the result will be NULL.

语法	a - b
参数	• **a** - value • **b** - value
Examples	• `5 - 4` → 1 • `5 - NULL` → NULL • `to_datetime('2012-05-05 12:00:00') - to_interval('1 day 2 hours')` → 2012-05-04T10:00:00

/

Division of two values

语法	a / b
参数	• **a** - value • **b** - value
Examples	• `5 / 4` → 1.25 • `5 / NULL` → NULL

//

Floor division of two values

语法	a // b
参数	• **a** - value • **b** - value
Examples	• `9 // 2` → 4

<

Compares two values and evaluates to 1 if the left value is less than the right value.

语法	a < b
参数	• **a** - value • **b** - value
Examples	• `5 < 4` → FALSE • `5 < 5` → FALSE • `4 < 5` → TRUE

<=

Compares two values and evaluates to 1 if the left value is less or equal than the right value.

语法	a <= b
参数	• **a** - value • **b** - value
Examples	• `5 <= 4` → FALSE • `5 <= 5` → TRUE • `4 <= 5` → TRUE

<>

Compares two values and evaluates to 1 if they are not equal.

语法	a <> b
参数	• **a** - value • **b** - value
Examples	• `5 <> 4` → TRUE • `4 <> 4` → FALSE • `5 <> NULL` → NULL • `NULL <> NULL` → NULL

=

Compares two values and evaluates to 1 if they are equal.

语法	a = b
参数	• **a** - value • **b** - value
Examples	• `5 = 4` → FALSE • `4 = 4` → TRUE • `5 = NULL` → NULL • `NULL = NULL` → NULL

>

Compares two values and evaluates to 1 if the left value is greater than the right value.

语法	a > b
参数	• **a** - value • **b** - value
Examples	• `5 > 4` → TRUE • `5 > 5` → FALSE • `4 > 5` → FALSE

>=

Compares two values and evaluates to 1 if the left value is greater or equal than the right value.

语法	a >= b
参数	• **a** - value • **b** - value
Examples	• `5 >= 4` → TRUE • `5 >= 5` → TRUE • `4 >= 5` → FALSE

AND

Returns TRUE when conditions a and b are true.

语法	a AND b
参数	• **a** - condition • **b** - condition
Examples	• `TRUE AND TRUE` → TRUE • `TRUE AND FALSE` → FALSE • `4 = 2+2 AND 1 = 1` → TRUE • `4 = 2+2 AND 1 = 2` → FALSE

BETWEEN

Returns TRUE if value is within the specified range. The range is considered inclusive of the bounds. To test for exclusion NOT BETWEEN can be used.

语法	value BETWEEN lower_bound AND higher_bound
参数	• **value** - the value to compare with a range. It can be a string, a number or a date. • **lower_bound AND higher_bound** - range bounds
Examples	• `'B' BETWEEN 'A' AND 'C'` → TRUE • `2 BETWEEN 1 AND 3` → TRUE • `2 BETWEEN 2 AND 3` → TRUE • `'B' BETWEEN 'a' AND 'c'` → FALSE • `lower('B') BETWEEN 'a' AND 'b'` → TRUE

Note

value BETWEEN lower_bound AND higher_bound is the same as "*value >= lower_bound AND value <= higher_bound*".

Further reading: *NOT BETWEEN*

ILIKE

Returns TRUE if the first parameter matches case-insensitive the supplied pattern. LIKE can be used instead of ILIKE to make the match case-sensitive. Works with numbers also.

语法	string/number ILIKE pattern
参数	• **string/number** - string to search • **pattern** - pattern to find, you can use '%' as a wildcard, '_' as a single char and '\\' to escape these special characters.
Examples	• `'A' ILIKE 'A'` → TRUE • `'A' ILIKE 'a'` → TRUE • `'A' ILIKE 'B'` → FALSE • `'ABC' ILIKE 'b'` → FALSE • `'ABC' ILIKE 'B'` → FALSE • `'ABC' ILIKE '_b_'` → TRUE • `'ABC' ILIKE '_B_'` → TRUE • `'ABCD' ILIKE '_b_'` → FALSE • `'ABCD' ILIKE '_B_'` → FALSE • `'ABCD' ILIKE '_b%'` → TRUE • `'ABCD' ILIKE '_B%'` → TRUE • `'ABCD' ILIKE '%b%'` → TRUE • `'ABCD' ILIKE '%B%'` → TRUE • `'ABCD%' ILIKE 'abcd\\%'` → TRUE • `'ABCD' ILIKE '%B\\%'` → FALSE

IN

Returns TRUE if value is found within a list of values.

语法	a IN b
参数	• **a** - value • **b** - list of values
Examples	• `'A' IN ('A','B')` → TRUE • `'A' IN ('C','B')` → FALSE

IS

Returns TRUE if a is the same as b.

语法	a IS b
参数	• **a** - any value • **b** - any value
Examples	• `'A' IS 'A'` → TRUE • `'A' IS 'a'` → FALSE • `4 IS 4` → TRUE • `4 IS 2+2` → TRUE • `4 IS 2` → FALSE • `@geometry IS NULL` → 0, if your geometry is not NULL

IS NOT

Returns TRUE if a is not the same as b.

语法	a IS NOT b
参数	• **a** - value • **b** - value
Examples	• `'a' IS NOT 'b'` → TRUE • `'a' IS NOT 'a'` → FALSE • `4 IS NOT 2+2` → FALSE

LIKE

Returns TRUE if the first parameter matches the supplied pattern. Works with numbers also.

语法	string/number LIKE pattern
参数	• **string/number** - value • **pattern** - pattern to compare value with, you can use '%' as a wildcard, '_' as a single char and '\\' to escape these special characters.
Examples	• `'A' LIKE 'A'` → TRUE • `'A' LIKE 'a'` → FALSE • `'A' LIKE 'B'` → FALSE • `'ABC' LIKE 'B'` → FALSE • `'ABC' LIKE '_B_'` → TRUE • `'ABCD' LIKE '_B_'` → FALSE • `'ABCD' LIKE '_B%'` → TRUE • `'ABCD' LIKE '%B%'` → TRUE • `'1%' LIKE '1\\%'` → TRUE • `'1_' LIKE '1\\%'` → FALSE

NOT

Negates a condition.

语法	NOT a
参数	• **a** - condition
Examples	• `NOT 1` → FALSE • `NOT 0` → TRUE

NOT BETWEEN

Returns TRUE if value is not within the specified range. The range is considered inclusive of the bounds.

语法	value NOT BETWEEN lower_bound AND higher_bound
参数	• **value** - the value to compare with a range. It can be a string, a number or a date. • **lower_bound AND higher_bound** - range bounds
Examples	• `'B' NOT BETWEEN 'A' AND 'C'` → FALSE • `1.0 NOT BETWEEN 1.1 AND 1.2` → TRUE • `2 NOT BETWEEN 2 AND 3` → FALSE • `'B' NOT BETWEEN 'a' AND 'c'` → TRUE • `lower('B') NOT BETWEEN 'a' AND 'b'` → FALSE

Note

value NOT BETWEEN lower_bound AND higher_bound is the same as "*value* < *lower_bound OR value* > *higher_bound*".

Further reading: *BETWEEN*

OR

Returns TRUE when condition a or b is true.

语法	a OR b
参数	• **a** - condition • **b** - condition
Examples	• `4 = 2+2 OR 1 = 1` → TRUE • `4 = 2+2 OR 1 = 2` → TRUE • `4 = 2 OR 1 = 2` → FALSE

[]

Index operator. Returns an element from an array or map value.

语法	[index]
参数	• **index** - array index or map key value
Examples	• `array(1,2,3)[0]` → 1 • `array(1,2,3)[2]` → 3 • `array(1,2,3)[-1]` → 3 • `map('a',1,'b',2)['a']` → 1 • `map('a',1,'b',2)['b']` → 2

Further reading: *array_get*, *map_get*

^

Power of two values.

语法	a ^ b
参数	• **a** - value • **b** - value
Examples	• `5 ^ 4` → 625 • `5 ^ NULL` → NULL

||

Joins two values together into a string.

If one of the values is NULL the result will be NULL. See the CONCAT function for a different behavior.

语法	a || b
参数	• **a** - value • **b** - value
Examples	• `'Here' || ' and ' || 'there'` → 'Here and there' • `'Nothing' || NULL` → NULL • `'Dia: ' || "Diameter"` → 'Dia: 25' • `1 || 2` → '12'

Further reading: *concat*, *+*

~

Performs a regular expression match on a string value. Backslash characters must be double escaped (e.g., "\\s" to match a white space character).

语法	string ~ regex
参数	• **string** - A string value • **regex** - A regular expression. Slashes must be escaped, eg \\d.
Examples	• `'hello' ~ 'll'` → TRUE • `'hello' ~ '^ll'` → FALSE • `'hello' ~ 'llo$'` → TRUE • `'abc123' ~ '\\d+'` → TRUE

Further reading: *regexp_match*

9.2.21 Processing Functions

This group contains functions that operate on processing algorithms.

parameter

Returns the value of a processing algorithm input parameter.

语法	parameter(name)
参数	• **name** - name of the corresponding input parameter
Examples	• `parameter('BUFFER_SIZE')` → 5.6

9.2.22 Rasters Functions

This group contains functions to operate on raster layer.

raster_attributes

Returns a map with the fields names as keys and the raster attribute table values as values from the attribute table entry that matches the given raster value.

语法	raster_attributes(layer, band, value)
参数	• **layer** - the name or id of a raster layer • **band** - the band number for the associated attribute table lookup. • **value** - raster value
Examples	• `raster_attributes('vegetation', 1, raster_value('vegetation', 1, make_point(1,1)))` → {'class': 'Vegetated', 'subclass': 'Trees'} • `raster_attributes('vegetation', 1, raster_value('vegetation', 1, @layer_cursor_point))` → {'class': 'Vegetated', 'subclass': 'Trees'}

raster_statistic

Returns statistics from a raster layer.

语法	raster_statistic(layer, band, property)
参数	• **layer** - a string, representing either a raster layer name or layer ID • **band** - integer representing the band number from the raster layer, starting at 1 • **property** - a string corresponding to the property to return. Valid options are: – min: minimum value – max: maximum value – avg: average (mean) value – stdev: standard deviation of values – range: range of values (max - min) – sum: sum of all values from raster
Examples	• `raster_statistic('lc',1,'avg')` → Average value from band 1 from 'lc' raster layer • `raster_statistic('ac2010',3,'min')` → Minimum value from band 3 from 'ac2010' raster layer

raster_value

Returns the raster value found at the provided point.

语法	raster_value(layer, band, point)
参数	• **layer** - the name or id of a raster layer • **band** - the band number to sample the value from. • **point** - point geometry (for multipart geometries having more than one part, a NULL value will be returned)
Examples	• `raster_value('dem', 1, make_point(1,1))` → 25 • `raster_value('ndvi', 2, @layer_cursor_point)` → 25

9.2.23 Record and Attributes Functions

This group contains functions that operate on record identifiers.

attribute

Returns an attribute from a feature.

Variant 1

Returns the value of an attribute from the current feature.

语法	attribute(attribute_name)
参数	• **attribute_name** - name of attribute to be returned
Examples	• `attribute( 'name' )` → value stored in 'name' attribute for the current feature

Variant 2

Allows the target feature and attribute name to be specified.

语法	attribute(feature, attribute_name)
参数	• **feature** - a feature • **attribute_name** - name of attribute to be returned
Examples	• `attribute( @atlas_feature, 'name' )` → value stored in 'name' attribute for the current atlas feature

attributes

Returns a map containing all attributes from a feature, with field names as map keys.

Variant 1

Returns a map of all attributes from the current feature.

语法	attributes()
Examples	• `attributes()['name']` → value stored in 'name' attribute for the current feature

Variant 2

Allows the target feature to be specified.

语法	attributes(feature)
参数	• **feature** - a feature
Examples	• `attributes( @atlas_feature )['name']` → value stored in 'name' attribute for the current atlas feature

Further reading: *Maps Functions*

$currentfeature

Returns the current feature being evaluated. This can be used with the 'attribute' function to evaluate attribute values from the current feature. **WARNING: This function is deprecated. It is recommended to use the replacement @feature variable instead.**

语法	$currentfeature
Examples	• `attribute( $currentfeature, 'name' )` → value stored in 'name' attribute for the current feature

display_expression

Returns the display expression for a given feature in a layer. The expression is evaluated by default. Can be used with zero, one or more arguments, see below for details.

No parameters

If called with no parameters, the function will evaluate the display expression of the current feature in the current layer.

语法	display_expression()
Examples	• `display_expression()` → The display expression of the current feature in the current layer.

One 'feature' parameter

If called with a 'feature' parameter only, the function will evaluate the specified feature from the current layer.

语法	display_expression(feature)
参数	• **feature** - The feature which should be evaluated.
Examples	• `display_expression(@atlas_feature)` → The display expression of the current atlas feature.

Layer and feature parameters

If the function is called with both a layer and a feature, it will evaluate the specified feature from the specified layer.

语法	display_expression(layer, feature, [evaluate=true]) [] 标记可选参数
参数	• **layer** - The layer (or its ID or name) • **feature** - The feature which should be evaluated. • **evaluate** - If the expression must be evaluated. If false, the expression will be returned as a string literal only (which could potentially be later evaluated using the 'eval' function).
Examples	• `display_expression( 'streets', get_feature_by_id('streets', 1))` → The display expression of the feature with the ID 1 on the layer 'streets'. • `display_expression('a_layer_id', @feature, 'False')` → The display expression of the given feature not evaluated.

feature_id

Returns a feature's unique ID, or NULL if the feature is not valid.

语法	feature_id(feature)
参数	• **feature** - a feature object
Examples	• `feature_id( @feature )` → the ID of the current feature

Further reading: *get_feature_by_id*

get_feature

Returns the first feature of a layer matching a given attribute value.

Single value variant

Along with the layer ID, a single column and value are specified.

语法	get_feature(layer, attribute, value)
参数	• **layer** - layer name or ID • **attribute** - attribute name to use for the match • **value** - attribute value to match
Examples	• `get_feature('streets','name','main st')` → first feature found in "streets" layer with "main st" value in the "name" field

Map variant

Along with the layer ID, a map containing the columns (key) and their respective value to be used.

语法	get_feature(layer, attribute)
参数	• **layer** - layer name or ID • **attribute** - Map containing the column and value pairs to use
Examples	• `get_feature('streets',map('name','main st','lane_num', '4'))` → first feature found in "streets" layer with "main st" value in the "name" field and "4" value in the "lane_num" field

get_feature_by_id

Returns the feature with an id on a layer.

语法	get_feature_by_id(layer, feature_id)
参数	• **layer** - layer, layer name or layer id • **feature_id** - the id of the feature which should be returned
Examples	• `get_feature_by_id('streets', 1)` → the feature with the id 1 on the layer "streets"

Further reading: *feature_id*

$id

Returns the feature id of the current row. **WARNING: This function is deprecated. It is recommended to use the replacement @id variable instead.**

语法	$id
Examples	• `$id` → 42

Further reading: *feature_id*, *get_feature_by_id*

is_attribute_valid

Returns TRUE if a specific feature attribute meets all constraints.

语法	is_attribute_valid(attribute, [feature], [layer], [strength]) [] 标记可选参数
参数	• **attribute** - an attribute name • **feature** - A feature. If not set, the current feature will be used. • **layer** - A vector layer. If not set, the current layer will be used. • **strength** - Set to 'hard' or 'soft' to narrow down to a specific constraint type. If not set, the function will return FALSE if either a hard or a soft constraint fails.
Examples	• `is_attribute_valid('HECTARES')` → TRUE if the current feature's value in the "HECTARES" field meets all constraints. • `is_attribute_valid('HOUSES',get_feature('my_layer', 'FID', 10), 'my_layer')` → FALSE if the value in the "HOUSES" field from the feature with "FID"=10 in 'my_layer' fails to meet all constraints.

Further reading: *Constraints*

is_feature_valid

Returns TRUE if a feature meets all field constraints.

语法	is_feature_valid([feature], [layer], [strength]) [] 标记可选参数
参数	• **feature** - A feature. If not set, the current feature will be used. • **layer** - A vector layer. If not set, the current layer will be used. • **strength** - Set to 'hard' or 'soft' to narrow down to a specific constraint type. If not set, the function will return FALSE if either a hard or a soft constraint fails.
Examples	• `is_feature_valid(strength:='hard')` → TRUE if all fields from the current feature meet their hard constraints. • `is_feature_valid(get_feature('my_layer', 'FID', 10), 'my_layer')` → FALSE if all fields from feature with "FID"=10 in 'my_layer' fails to meet all constraints.

Further reading: *Constraints*

is_selected

Returns TRUE if a feature is selected. Can be used with zero, one or two arguments, see below for details.

No parameters

If called with no parameters, the function will return TRUE if the current feature in the current layer is selected.

语法	is_selected()
Examples	• `is_selected()` → TRUE if the current feature in the current layer is selected.

One 'feature' parameter

If called with a 'feature' parameter only, the function returns TRUE if the specified feature from the current layer is selected.

语法	is_selected(feature)
参数	• **feature** - The feature which should be checked for selection.
Examples	• `is_selected(@atlas_feature)` → TRUE if the current atlas feature is selected. • `is_selected(get_feature('streets', 'name', 'Main St.'))` → TRUE if the unique named "Main St." feature on the active "streets" layer is selected. • `is_selected(get_feature_by_id('streets', 1))` → TRUE if the feature with the id 1 on the active "streets" layer is selected.

Two parameters

If the function is called with both a layer and a feature, it will return TRUE if the specified feature from the specified layer is selected.

语法	is_selected(layer, feature)
参数	• **layer** - The layer (its ID or name) on which the selection will be checked. • **feature** - The feature which should be checked for selection.
Examples	• `is_selected( 'streets', get_feature('streets', 'name', "street_name"))` → TRUE if the current building's street is selected (assuming the building layer has a field named 'street_name' and the 'streets' layer has a field called 'name' with unique values). • `is_selected( 'streets', get_feature_by_id('streets', 1))` → TRUE if the feature with the id 1 on the "streets" layer is selected.

maptip

Returns the maptip for a given feature in a layer. The expression is evaluated by default. Can be used with zero, one or more arguments, see below for details.

No parameters

If called with no parameters, the function will evaluate the maptip of the current feature in the current layer.

语法	maptip()
Examples	• `maptip()` → The maptip of the current feature in the current layer.

One 'feature' parameter

If called with a 'feature' parameter only, the function will evaluate the specified feature from the current layer.

语法	maptip(feature)
参数	• **feature** - The feature which should be evaluated.
Examples	• `maptip(@atlas_feature)` → The maptip of the current atlas feature.

Layer and feature parameters

If the function is called with both a layer and a feature, it will evaluate the specified feature from the specified layer.

语法	maptip(layer, feature, [evaluate=true]) [] 标记可选参数
参数	• **layer** - The layer (or its ID or name) • **feature** - The feature which should be evaluated. • **evaluate** - If the expression must be evaluated. If false, the expression will be returned as a string literal only (which could potentially be later evaluated using the 'eval_template' function).
Examples	• `maptip('streets', get_feature_by_id('streets', 1))` → The maptip of the feature with the ID 1 on the layer 'streets'. • `maptip('a_layer_id', @feature, 'False')` → The maptip of the given feature not evaluated.

num_selected

Returns the number of selected features on a given layer. By default works on the layer on which the expression is evaluated.

语法	num_selected([layer=current layer]) [] 标记可选参数
参数	• **layer** - The layer (or its id or name) on which the selection will be checked.
Examples	• `num_selected()` → The number of selected features on the current layer. • `num_selected('streets')` → The number of selected features on the layer streets

represent_attributes

Returns a map with the attribute names as keys and the configured representation values as values. The representation value for the attributes depends on the configured widget type for each attribute. Can be used with zero, one or more arguments, see below for details.

No parameters

If called with no parameters, the function will return the representation of the attributes of the current feature in the current layer.

语法	represent_attributes()
Examples	• `represent_attributes()` → The representation of the attributes for the current feature.

One 'feature' parameter

If called with a 'feature' parameter only, the function will return the representation of the attributes of the specified feature from the current layer.

语法	represent_attributes(feature)
参数	• **feature** - The feature which should be evaluated.
Examples	• `represent_attributes(@atlas_feature)` → The representation of the attributes for the specified feature from the current layer.

Layer and feature parameters

If called with a 'layer' and a 'feature' parameter, the function will return the representation of the attributes of the specified feature from the specified layer.

语法	represent_attributes(layer, feature)
参数	• **layer** - The layer (or its ID or name). • **feature** - The feature which should be evaluated.
Examples	• `represent_attributes('atlas_layer', @atlas_feature)` → The representation of the attributes for the specified feature from the specified layer.

Further reading: *represent_value*

represent_value

Returns the configured representation value for a field value. It depends on the configured widget type. Often, this is useful for 'Value Map' widgets.

语法	represent_value(value, [fieldName]) [] 标记可选参数
参数	• **value** - The value which should be resolved. Most likely a field. • **fieldName** - The field name for which the widget configuration should be loaded.
Examples	• `represent_value("field_with_value_map")` → Description for value • `represent_value('static value', 'field_name')` → Description for static value

Further reading: *widget types*, *represent_attributes*

sqlite_fetch_and_increment

Manage autoincrementing values in sqlite databases.

SQlite default values can only be applied on insert and not prefetched.

This makes it impossible to acquire an incremented primary key via AUTO_INCREMENT before creating the row in the database. Sidenote: with postgres, this works via the option *evaluate default values*.

When adding new features with relations, it is really nice to be able to already add children for a parent, while the parents form is still open and hence the parent feature uncommitted.

To get around this limitation, this function can be used to manage sequence values in a separate table on sqlite based formats like gpkg.

The sequence table will be filtered for a sequence id (filter_attribute and filter_value) and the current value of the id_field will be incremented by 1 and the incremented value returned.

If additional columns require values to be specified, the default_values map can be used for this purpose.

Note

This function modifies the target sqlite table. It is intended for usage with default value configurations for attributes.

When the database parameter is a layer and the layer is in transaction mode, the value will only be retrieved once during the lifetime of a transaction and cached and incremented. This makes it unsafe to work on the same database from several processes in parallel.

语法	sqlite_fetch_and_increment(database, table, id_field, filter_attribute, filter_value, [default_values]) [] 标记可选参数
参数	• **database** - Path to the sqlite file or geopackage layer • **table** - Name of the table that manages the sequences • **id_field** - Name of the field that contains the current value • **filter_attribute** - Name the field that contains a unique identifier for this sequence. Must have a UNIQUE index. • **filter_value** - Name of the sequence to use. • **default_values** - Map with default values for additional columns on the table. The values need to be fully quoted. Functions are allowed.
Examples	• `sqlite_fetch_and_increment(@layer, 'sequence_table', 'last_unique_id', 'sequence_id', 'global', map('last_change', 'date(''now'')', 'user', '''' || @user_account_name || ''''))` → 0 • `sqlite_fetch_and_increment(layer_property(@layer, 'path'), 'sequence_table', 'last_unique_id', 'sequence_id', 'global', map('last_change', 'date(''now'')', 'user', '''' || @user_account_name || ''''))` → 0

Further reading: *数据源属性*, *Setting relations between multiple layers*

uuid

Generates a Universally Unique Identifier (UUID) for each row using the Qt QUuid::createUuid method.

语法	uuid([format='WithBraces']) [] 标记可选参数
参数	• **format** - The format, as the UUID will be formatted. 'WithBraces', 'WithoutBraces' or 'Id128'.
Examples	• `uuid()` → '{0bd2f60f-f157-4a6d-96af-d4ba4cb366a1}' • `uuid('WithoutBraces')` → '0bd2f60f-f157-4a6d-96af-d4ba4cb366a1' • `uuid('Id128')` → '0bd2f60ff1574a6d96afd4ba4cb366a1'

9.2.24 Relations

This group contains the list of the *relations* available in the current project, with their description. It provides a quick access to the relation ID for writing an expression (with e.g. the *relation_aggregate* function) or customizing a form.

9.2.25 Sensors Functions

This group contains functions to interact with sensors.

sensor_data

Returns the last captured value (or values as a map for sensors which report multiple values) from a registered sensor.

语法	sensor_data(name, [expiration]) [] 标记可选参数
参数	• **name** - the sensor name • **expiration** - maximum millisecond since last captured value allowed
Examples	• `sensor_data('geiger_1')` → '2000'

9.2.26 String Functions

This group contains functions that operate on strings (e.g., that replace, convert to upper case).

ascii

Returns the unicode code associated with the first character of a string.

语法	ascii(string)
参数	• **string** - the string to convert to unicode code
Examples	• `ascii('Q')` → 81

char

Returns the character associated with a unicode code.

语法	char(code)
参数	• **code** - a unicode code number
Examples	• `char(81)` → 'Q'

concat

Concatenates several strings to one. NULL values are converted to empty strings. Other values (like numbers) are converted to strings.

语法	concat(string1, string2, ...)
参数	• **string** - a string value
Examples	• `concat('sun', 'set')` → 'sunset' • `concat('a','b','c','d','e')` → 'abcde' • `concat('Anno ', 1984)` → 'Anno 1984' • `concat('The Wall', NULL)` → 'The Wall'

About fields concatenation

You can also concatenate strings or field values using either `||` or + operators, with some special characteristics:

- The + operator also means sum up expression, so if you have an integer (field or numeric value) operand, this can be error prone and you better use the others:

```
'My feature id is: ' + "gid" => triggers an error as gid returns an integer
```

- When any of the arguments is a NULL value, either `||` or + will return a NULL value. To return the other arguments regardless the NULL value, you may want to use the `concat` or `array_to_string` function:

```
'My feature id is: ' + NULL ==> NULL
'My feature id is: ' || NULL => NULL
concat('My feature id is: ', NULL) => 'My feature id is: '
array_to_string( array('My feature id is: ', NULL) ) => 'My feature id is: '
```

further reading: *array_to_string*, ||, +

format

Format a string using supplied arguments.

语法	format(string, arg1, arg2, ...)
参数	• **string** - A string with placeholders %1, %2, etc., for the arguments. Placeholders can be repeated. The lowest numbered placeholder is replaced by arg1, the next by arg2, etc. • **arg** - any type. Any number of arguments.
Examples	• `format('This %1 a %2','is', 'test')` → 'This is a test' • `format('This is %2','a bit unexpected but 2 is lowest number in string','normal')` → 'This is a bit unexpected but 2 is lowest number in string'

format_date

Formats a date type or string into a custom string format. Uses Qt date/time format strings. See QDateTime::toString.

语法

format_date(datetime, format, [language])
[] 标记可选参数

参数

- **datetime** - date, time or datetime value
- **format** - String template used to format the string.

Expression	Output
d	the day as number without a leading zero (1 to 31)
dd	the day as number with a leading zero (01 to 31)
ddd	the abbreviated localized day name (e.g. 'Mon' to 'Sun')
dddd	the long localized day name (e.g. 'Monday' to 'Sunday')
M	the month as number without a leading zero (1-12)
MM	the month as number with a leading zero (01-12)
MMM	the abbreviated localized month name (e.g. 'Jan' to 'Dec')
MMMM	the long localized month name (e.g. 'January' to 'December')
yy	the year as two digit number (00-99)
yyyy	the year as four digit number

These expressions may be used for the time part of the format string:

Expression	Output
h	the hour without a leading zero (0 to 23 or 1 to 12 if AM/PM display)
hh	the hour with a leading zero (00 to 23 or 01 to 12 if AM/PM display)
H	the hour without a leading zero (0 to 23, even with AM/PM display)
HH	the hour with a leading zero (00 to 23, even with AM/PM display)
m	the minute without a leading zero (0 to 59)
mm	the minute with a leading zero (00 to 59)
s	the second without a leading zero (0 to 59)
ss	the second with a leading zero (00 to 59)
z	the milliseconds without trailing zeroes (0 to 999)
zzz	the milliseconds with trailing zeroes (000 to 999)
AP or A	interpret as an AM/PM time. *AP* must be either 'AM' or 'PM'.
ap or a	Interpret as an AM/PM time. *ap* must be either 'am' or 'pm'.

- **language** - language (lowercase, two- or three-letter, ISO 639 language code) used to format the date into a custom string. By default the current QGIS user locale is used.

Examples

- `format_date('2012-05-15','dd.MM.yyyy')` → '15.05.2012'
- `format_date('2012-05-15','d MMMM yyyy','fr')` → '15 mai 2012'
- `format_date('2012-05-15','dddd')` → 'Tuesday', if the current locale is an English variant
- `format_date('2012-05-15 13:54:20','dd.MM.yy')` → '15.05.12'
- `format_date('13:54:20','hh:mm AP')` → '01:54 PM'

format_number

Returns a number formatted with the locale separator for thousands. By default the current QGIS user locale is used. Also truncates the decimal places to the number of supplied places.

语法	format_number(number, [places=0], [language], [omit_group_separators=false], [trim_trailing_zeroes=false]) [] 标记可选参数
参数	• **number** - number to be formatted • **places** - integer representing the number of decimal places to truncate the string to. • **language** - language (lowercase, two- or three-letter, ISO 639 language code) used to format the number into a string. By default the current QGIS user locale is used. • **omit_group_separators** - if set to true then group separators will not be included in the string • **trim_trailing_zeroes** - if set to true then trailing zeros following the decimal point will be trimmed from the string
Examples	• `format_number(10000000.332,2)` → '10,000,000.33' if e.g. the current locale is an English variant • `format_number(10000000.332,2,'fr')` → '10 000 000,33'

left

Returns a substring that contains the *n* leftmost characters of the string.

语法	left(string, length)
参数	• **string** - a string • **length** - integer. The number of characters from the left of the string to return.
Examples	• `left('Hello World',5)` → 'Hello'

Further reading: *right*

length

Returns the number of characters in a string or the length of a geometry linestring.

String variant

Returns the number of characters in a string.

语法	length(string)
参数	• **string** - string to count length of
Examples	• `length('hello')` → 5

Geometry variant

Calculate the length of a geometry line object. Calculations are always planimetric in the Spatial Reference System (SRS) of this geometry, and the units of the returned length will match the units for the SRS. This differs from the calculations performed by the $length function, which will perform ellipsoidal calculations based on the project's ellipsoid and distance unit settings.

语法	length(geometry)
参数	• **geometry** - line geometry object
Examples	• `length(geom_from_wkt('LINESTRING(0 0, 4 0)'))` → 4.0

lower

Converts a string to lower case letters.

语法	lower(string)
参数	• **string** - the string to convert to lower case
Examples	• `lower('HELLO World')` → 'hello world'

Further reading: *upper*, *title*

lpad

Returns a string padded on the left to the specified width, using a fill character. If the target width is smaller than the string's length, the string is truncated.

语法	lpad(string, width, fill)
参数	• **string** - string to pad • **width** - length of new string • **fill** - character to pad the remaining space with
Examples	• `lpad('Hello', 10, 'x')` → 'xxxxxHello' • `lpad('Hello', 3, 'x')` → 'Hel'

Further reading: *rpad*

ltrim

Removes the longest string containing only the specified characters (a space by default) from the start of string.

语法	ltrim(string, [characters=' ']) [] 标记可选参数
参数	• **string** - string to trim • **characters** - characters to trim
Examples	• `ltrim(' hello world ')` → 'hello world ' • `ltrim('zzzytest', 'xyz')` → 'test'

Further reading: *rtrim*, *trim*

regexp_match

Return the first matching position matching a regular expression within an unicode string, or 0 if the substring is not found.

语法	regexp_match(input_string, regex)
参数	• **input_string** - the string to test against the regular expression • **regex** - The regular expression to test against. Backslash characters must be double escaped (e.g., "\\s" to match a white space character or "\\b" to match a word boundary).
Examples	• `regexp_match('QGIS ROCKS','\\sROCKS')` → 5 • `regexp_match('Budač','udač\\b')` → 2

Further reading: *strpos*

regexp_replace

Returns a string with the supplied regular expression replaced.

语法	regexp_replace(input_string, regex, replacement)
参数	• **input_string** - the string to replace matches in • **regex** - The regular expression to replace. Backslash characters must be double escaped (e.g., "\\s" to match a white space character). • **replacement** - The string that will replace any matching occurrences of the supplied regular expression. Captured groups can be inserted into the replacement string using \\1, \\2, etc.
Examples	• `regexp_replace('QGIS SHOULD ROCK','\\sSHOULD\\s',' DOES ')` → 'QGIS DOES ROCK' • `regexp_replace('ABC123','\\d+','')` → 'ABC' • `regexp_replace('my name is John','(.*) is (.*)','\\2 is \\1')` → 'John is my name'

Further reading: *replace*, *array_replace*

regexp_substr

Returns the portion of a string which matches a supplied regular expression.

语法	regexp_substr(input_string, regex)
参数	• **input_string** - the string to find matches in • **regex** - The regular expression to match against. Backslash characters must be double escaped (e.g., "\\s" to match a white space character).
Examples	• `regexp_substr('abc123','(\\d+)')` → '123'

Further reading: *substr*, *regexp_matches*

repeat

Repeats a string a specified number of times.

语法	repeat(string, number)
参数	• **string** - string to repeat • **number** - number of times to repeat string
Examples	• `repeat('Hello', 3)` → 'HelloHelloHello'

replace

Returns a string with the supplied string, array, or map of strings replaced.

String & array variant

Returns a string with the supplied string or array of strings replaced by a string or an array of strings.

语法	replace(string, before, after)
参数	• **string** - the input string • **before** - the string or array of strings to replace • **after** - the string or array of strings to use as a replacement
Examples	• `replace('QGIS SHOULD ROCK','SHOULD','DOES')` → 'QGIS DOES ROCK' • `replace('QGIS ABC',array('A','B','C'),array('X','Y','Z'))` → 'QGIS XYZ' • `replace('QGIS',array('Q','S'),'')` → 'GI'

Map variant

Returns a string with the supplied map keys replaced by paired values. Longer map keys are evaluated first.

语法	replace(string, map)
参数	• **string** - the input string • **map** - the map containing keys and values
Examples	• `replace('APP SHOULD ROCK',map('APP','QGIS','SHOULD', 'DOES'))` → 'QGIS DOES ROCK' • `replace('forty two',map('for','4','two','2','forty two', '42'))` → '42'

Further reading: *regexp_replace*, *array_replace*

reverse

Reverses the direction of a line string or reverses a string of text.

String variant

Reverses the order of characters in a string.

语法	reverse(string)
参数	• **string** - string to reverse
Examples	• `reverse('hello')` → 'olleh'

Geometry variant

Reverses the direction of a line string by reversing the order of its vertices.

语法	reverse(geometry)
参数	• **geometry** - a geometry
Examples	• `geom_to_wkt(reverse(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)')))` → 'LINESTRING(2 2, 1 1, 0 0)'

right

Returns a substring that contains the *n* rightmost characters of the string.

语法	right(string, length)
参数	• **string** - a string • **length** - integer. The number of characters from the right of the string to return.
Examples	• `right('Hello World',5)` → 'World'

Further reading: *left*

rpad

Returns a string padded on the right to the specified width, using a fill character. If the target width is smaller than the string's length, the string is truncated.

语法	rpad(string, width, fill)
参数	• **string** - string to pad • **width** - length of new string • **fill** - character to pad the remaining space with
Examples	• `rpad('Hello', 10, 'x')` → 'Helloxxxxx' • `rpad('Hello', 3, 'x')` → 'Hel'

Further reading: *lpad*

rtrim

Removes the longest string containing only the specified characters (a space by default) from the end of string.

语法	rtrim(string, [characters=' ']) [] 标记可选参数
参数	• **string** - string to trim • **characters** - characters to trim
Examples	• `rtrim(' hello world ')` → ' hello world' • `rtrim('testxxzx', 'xyz')` → 'test'

Further reading: *ltrim*, *trim*

strpos

Return the first matching position of a substring within another string, or 0 if the substring is not found.

语法	strpos(haystack, needle)
参数	• **haystack** - string that is to be searched • **needle** - string to search for
Examples	• `strpos('HELLO WORLD','WORLD')` → 7 • `strpos('HELLO WORLD','GOODBYE')` → 0

Further reading: *regexp_match*

substr

Returns a part of a string.

语法	substr(string, start, [length]) [] 标记可选参数
参数	• **string** - the full input string • **start** - integer representing start position to extract beginning with 1; if start is negative, the return string will begin at the end of the string minus the start value • **length** - integer representing length of string to extract; if length is negative, the return string will omit the given length of characters from the end of the string
Examples	• `substr('HELLO WORLD',3,5)` → 'LLO W' • `substr('HELLO WORLD',6)` → ' WORLD' • `substr('HELLO WORLD',-5)` → 'WORLD' • `substr('HELLO',3,-1)` → 'LL' • `substr('HELLO WORLD',-5,2)` → 'WO' • `substr('HELLO WORLD',-5,-1)` → 'WORL'

Further reading: *regexp_substr*, *regexp_matches*

title

Converts all words of a string to title case (all words lower case with leading capital letter).

语法	title(string)
参数	• **string** - the string to convert to title case
Examples	• `title('hello WOrld')` → 'Hello World'

Further reading: *lower*, *upper*

to_string

Converts a number to string. The conversion is not locale-aware, see 'format_number' for a locale-aware alternative.

语法	to_string(number)
参数	• **number** - Integer or real value. The number to convert to string.
Examples	• `to_string(1.23)` → '1.23'

Further reading: *format_number*

trim

Removes all leading and trailing whitespace (spaces, tabs, etc) from a string.

语法	trim(string)
参数	• **string** - string to trim
Examples	• `trim(' hello world ')` → 'hello world'

Further reading: *ltrim*, *rtrim*

upper

Converts a string to upper case letters.

语法	upper(string)
参数	• **string** - the string to convert to upper case
Examples	• `upper('hello WOrld')` → 'HELLO WORLD'

Further reading: *lower*, *title*

wordwrap

Returns a string wrapped to a maximum/minimum number of characters.

语法	wordwrap(string, wrap_length, [delimiter_string]) [] 标记可选参数
参数	• **string** - the string to be wrapped • **wrap_length** - an integer. If wrap_length is positive the number represents the ideal maximum number of characters to wrap; if negative, the number represents the minimum number of characters to wrap. • **delimiter_string** - Optional delimiter string to wrap to a new line.
Examples	• `wordwrap('UNIVERSITY OF QGIS',13)` → 'UNIVERSITY OF QGIS' • `wordwrap('UNIVERSITY OF QGIS',-3)` → 'UNIVERSITY OF QGIS'

9.2.27 User Expressions

This group contains the expressions saved as *user expressions*.

9.2.28 Variables

This group contains dynamic variables related to the application, the project file and other settings. The availability of variables depends on the context:

- from the Select by expression dialog
- from the Field calculator dialog
- from the layer properties dialog
- from the print layout

To use these variables in an expression, they should be preceded by the `@` character (e.g, `@row_number`).

Variable	描述
algorithm_id	The unique ID of an algorithm
animation_end_time	End of the animation's overall temporal time range (as a datetime value)
animation_interval	Duration of the animation's overall temporal time range (as an interval value)
animation_start_time	Start of the animation's overall temporal time range (as a datetime value)
atlas_feature	The current atlas feature (as feature object)
atlas_featureid	The current atlas feature ID
atlas_featurenumber	The current atlas feature number in the layout
atlas_filename	The current atlas file name
atlas_geometry	The current atlas feature geometry
atlas_layerid	The current atlas coverage layer ID
atlas_layername	The current atlas coverage layer name
atlas_pagename	The current atlas page name
atlas_totalfeatures	The total number of features in atlas
band	The number of the band in the raster layer
band_description	The description of the band in the raster layer
band_name	The name of the band in the raster layer
canvas_cursor_point	The last cursor position on the canvas in the project's geographical coordinates
cluster_color	The color of symbols within a cluster, or NULL if symbols have mixed colors
cluster_size	The number of symbols contained within a cluster
current_feature	The feature currently being edited in the attribute form or table row
current_geometry	The geometry of the feature currently being edited in the form or the table row
current_parent_feature	represents the feature currently being edited in the parent form. Only usable in an embedded form context.
current_parent_geometry	represents the geometry of the feature currently being edited in the parent form. Only usable in an embedded form context.
form_mode	What the form is used for, like AddFeatureMode, SingleEditMode, MultiEditMode, SearchMode, AggregateSearchMode or IdentifyMode as string.
feature	The current feature being evaluated. This can be used with the 'attribute' function to evaluate attribute values from the current feature.
frame_duration	Temporal duration of each animation frame (as an interval value)
frame_number	Current frame number during animation playback
frame_rate	Number of frames per second during animation playback
fullextent_maxx	Maximum x value from full canvas extent (including all layers)
fullextent_maxy	Maximum y value from full canvas extent (including all layers)
fullextent_minx	Minimum x value from full canvas extent (including all layers)
fullextent_miny	Minimum y value from full canvas extent (including all layers)
geometry	The geometry of the current feature being evaluated

continues on next page

Table 9.1 – continued from previous page

Variable	描述
geometry_part_count	The number of parts in rendered feature's geometry
geometry_part_num	The current geometry part number for feature being rendered
geometry_point_count	The number of points in the rendered geometry's part
geometry_point_num	The current point number in the rendered geometry's part
geometry_ring_num	Current geometry ring number for feature being rendered (for polygon features only). The exterior ring has a value of 0.
grid_axis	The current grid annotation axis (eg, 'x' for longitude, 'y' for latitude)
grid_number	The current grid annotation value
id	The ID of the current feature being evaluated
item_id	The layout item user ID (not necessarily unique)
item_uuid	The layout item unique ID
layer	The current layer
layer_crs	The Coordinate Reference System Authority ID of the current layer
layer_crs_ellipsoid	The ellipsoid Authority ID of the current layer CRS
layer_cursor_point	Point geometry under the mouse position in map canvas (or the GetFeatureInfo position in context of QGIS Server), in active layer's CRS
layer_id	The ID of current layer
layer_ids	The IDs of all the map layers in the current project as a list
layer_name	The name of current layer
layer_vertical_crs	The Identifier for the vertical coordinate reference system of the layer (e.g., 'EPSG:5703')
layer_vertical_crs_definition	The full definition of the vertical Coordinate reference system of the layer
layer_vertical_crs_description	The name of the vertical Coordinate reference system of the layer
layer_vertical_crs_wkt	The WKT definition of the vertical Coordinate reference system of the current layer
layers	All the map layers in the current project as a list
layout_dpi	The composition resolution (DPI)
layout_name	The layout name
layout_numpages	The number of pages in the layout
layout_page	The page number of the current item in the layout
layout_pageheight	The active page height in the layout (in mm for standard paper sizes, or whatever unit was used for custom paper size)
layout_pageoffsets	Array of Y coordinate of the top of each page. Allows to dynamically position items on pages in a context where page sizes may change
layout_pagewidth	The active page width in the layout (in mm for standard paper sizes, or whatever unit was used for custom paper size)
legend_column_count	The number of columns in the legend
legend_filter_by_map	Indicates if the content of the legend is filtered by the map
legend_filter_out_atlas	Indicates if the atlas is filtered out of the legend
legend_split_layers	Indicates if layers can be split in the legend
legend_title	The title of the legend
legend_wrap_string	The character(s) used to wrap the legend text
map_crs	The Coordinate reference system of the current map
map_crs_acronym	The acronym of the Coordinate reference system of the current map
map_crs_definition	The full definition of the Coordinate reference system of the current map
map_crs_description	The name of the Coordinate reference system of the current map
map_crs_ellipsoid	The acronym of the ellipsoid of the Coordinate reference system of the current map
map_crs_proj4	The Proj4 definition of the Coordinate reference system of the current map
map_crs_projection	The descriptive name of the projection method used by the Coordinate reference system of the map (e.g. 'Albers Equal Area')
map_crs_wkt	The WKT definition of the Coordinate reference system of the current map
map_end_time	The end of the map's temporal time range (as a datetime value)
map_extent	The geometry representing the current extent of the map
map_extent_center	The point feature at the center of the map

continues on next page

Table 9.1 – continued from previous page

Variable	描述
map_extent_height	The current height of the map
map_extent_width	The current width of the map
map_id	The ID of current map destination. This will be 'canvas' for canvas renders, and the item ID for layout map renders
map_interval	The duration of the map's temporal time range (as an interval value)
map_layer_ids	The list of map layer IDs visible in the map
map_layers	The list of map layers visible in the map
map_rotation	The current rotation of the map
map_scale	The current scale of the map
map_start_time	The start of the map's temporal time range (as a datetime value)
map_units	The units of map measurements
map_z_range_lower	Lower elevation of the map's elevation range
map_z_range_upper	Upper elevation of the map's elevation range
model_path	Full path (including file name) of current model (or project path if model is embedded in a project).
model_folder	Folder containing current model (or project folder if model is embedded in a project).
model_name	Name of current model
model_group	Group for current model
notification_message	Content of the notification message sent by the provider (available only for actions triggered by provider notifications).
parent	Refers to the current feature in the parent layer, providing access to its attributes and geometry when filtering an *aggregate* function
plot_axis	The associated plot axis, e.g. 'x' or 'y'.
plot_axis_value	The current value for the plot axis.
project_abstract	The project abstract, taken from project metadata
project_area_units	The area unit for the current project, used when calculating areas of geometries
project_author	The project author, taken from project metadata
project_basename	The basename of current project's filename (without path and extension)
project_creation_date	The project creation date, taken from project metadata
project_crs	Identifier for the coordinate reference system of the project (e.g., 'EPSG:4326')
project_crs_arconym	The acronym of the Coordinate reference system of the project
project_crs_definition	The full definition of the Coordinate reference system of the project
project_crs_description	The description of the Coordinate reference system of the project
project_crs_ellipsoid	The ellipsoid of the Coordinate reference system of the project
project_crs_proj4	The Proj4 representation of the Coordinate reference system of the project
project_crs_wkt	The WKT (well known text) representation of the coordinate reference system of the project
project_distance_units	The distance unit for the current project, used when calculating lengths of geometries and distances
project_ellipsoid	The name of the ellipsoid of the current project, used when calculating geodetic areas or lengths of geometries
project_filename	The filename of the current project
project_folder	The folder of the current project
project_home	The home path of the current project
project_identifier	The project identifier, taken from the project's metadata
project_keywords	The project keywords, taken from the project's metadata
project_last_saved	Date/time when project was last saved.
project_path	The full path (including file name) of the current project
project_title	The title of current project
project_units	The units of the project's CRS
project_vertical_crs	Identifier for the vertical Coordinate reference system of the project (e.g., 'EPSG:5703')
project_vertical_crs_definition	The full definition of the vertical coordinate reference system of the project
project_vertical_crs_description	The description of the vertical coordinate reference system of the project

continues on next page

Table 9.1 – continued from previous page

Variable	描述
project_vertical_crs_wkt	The WKT (well known text) representation of the vertical coordinate reference system of the project
qgis_locale	The current language of QGIS
qgis_os_name	The current Operating system name, eg 'windows', 'linux' or 'osx'
qgis_platform	The QGIS platform, eg 'desktop' or 'server'
qgis_release_name	The current QGIS release name
qgis_short_version	The current QGIS version short string
qgis_version	The current QGIS version string
qgis_version_no	The current QGIS version number
row_number	Stores the number of the current row
snapping_results	Gives access to snapping results while digitizing a feature (only available in add feature)
scale_value	The current scale bar distance value
selected_file_path	Selected file path from file widget selector when uploading a file with an external storage system
symbol_angle	The angle of the symbol used to render the feature (valid for marker symbols only)
symbol_color	The color of the symbol used to render the feature
symbol_count	The number of features represented by the symbol (in the layout legend)
symbol_frame	The frame number (for animated symbols only)
symbol_id	The Internal ID of the symbol (in the layout legend)
symbol_label	The label for the symbol (either a user defined label or the default autogenerated label - in the layout legend)
symbol_layer_count	Total number of symbol layers in the symbol
symbol_layer_index	Current symbol layer index
symbol_marker_column	Column number for marker (valid for point pattern fills only).
symbol_marker_row	Row number for marker (valid for point pattern fills only).
user_account_name	The current user's operating system account name
user_full_name	The current user's operating system user name
value	The current value
vector_tile_zoom	Exact vector tile zoom level of the map that is being rendered (derived from the current map scale). Normally in interval [0, 20]. Unlike @zoom_level, this variable is a floating point value which can be used to interpolate values between two integer zoom levels.
with_variable	Allows setting a variable for usage within an expression and avoid recalculating the same value repeatedly
zoom_level	Vector tile zoom level of the map that is being rendered (derived from the current map scale). Normally in interval [0, 20].

Some examples:

- Return the X coordinate of a map item center in layout:

```
x( map_get( item_variables( 'map1'), 'map_extent_center' ) )
```

- Return, for each feature in the current layer, the number of overlapping airport features:

```
aggregate( layer:='airport', aggregate:='count', expression:="code",
               filter:=intersects( $geometry, geometry( @parent ) ) )
```

- Get the object_id of the first snapped point of a line:

```
with_variable(
  'first_snapped_point',
  array_first( @snapping_results ),
  attribute(
```

(continues on next page)

(continued from previous page)

```
    get_feature_by_id(
      map_get( @first_snapped_point, 'layer' ),
      map_get( @first_snapped_point, 'feature_id' )
    ),
    'object_id'
  )
)
```

9.2.29 Recent Functions

This group contains recently used functions. Depending on the context of its usage (feature selection, field calculator, generic), recently applied expressions are added to the corresponding list (up to ten expressions), sorted from more to less recent. This makes it easy to quickly retrieve and reapply previously used expressions.

CHAPTER 10

样式库

10.1 样式管理器

10.1.1 样式管理器对话框

样式管理器是管理和创建常规样式项的地方。这些符号、颜色渐变、文本格式或标注设置，可用于对要素、图层或打印布局进行符号化。它们存储在当前用户配置 的 `symbology-style.db` 数据库中，并与使用该配置文件打开的所有工程文件共享。由于 样式管理器对话框具有导出/导入功能，样式项也可以与其他人共享。

您可以打开该非模态对话框：

- 从 设置 ► 样式管理器... 菜单
- 使用工程工具栏中的 样式管理器 按钮
- 或者使用矢量 图层属性 ► 菜单中的 样式管理器 按钮 (在配置符号 或格式化文本) 。

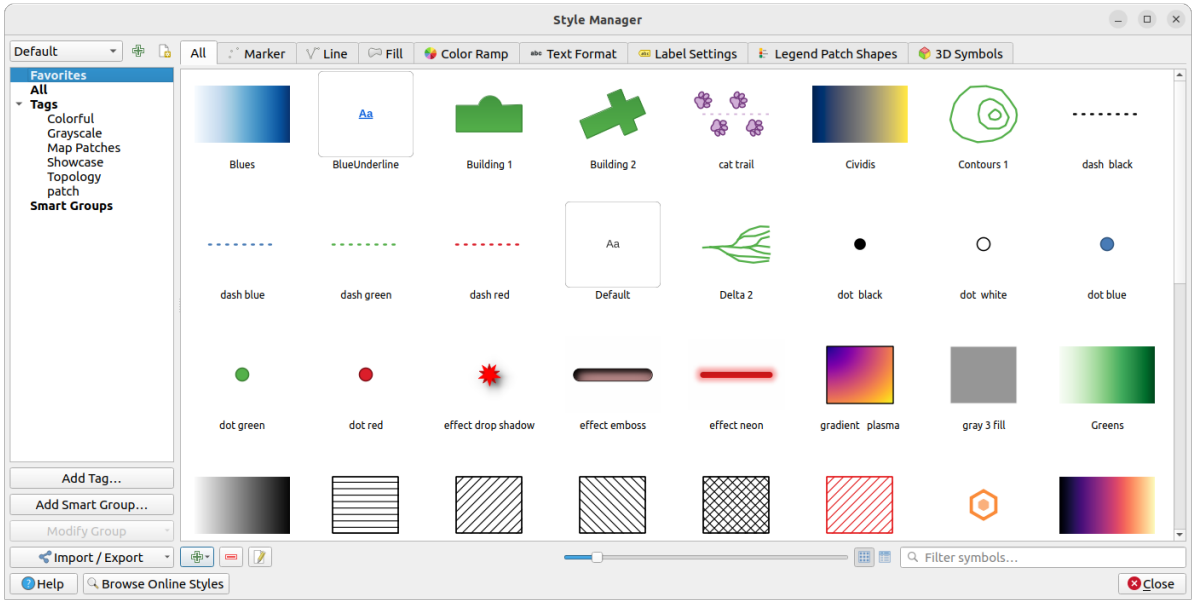


Fig. 10.1: 样式管理器

组织样式项

In the upper left corner of the *Style Manager* dialog, within the drop-down menu, you can choose the style database you want to connect to. If you choose *Default* you will be connected to the default style database where you can find all available default style items and the ones that you saved in this database. Choosing *Project Styles* will connect you to the Project Style database where you can find only symbols that you saved in this database. If you created more style databases, they will be listed in the drop-down menu. There are also options to Add existing style database to project and Create new style database (see more *Style database*).

For each style database, you can organize the elements into different categories, listed in the panel on the left:

- **收藏夹**: 配置项时默认显示，它显示一组可扩展的项；
- **全部**: 列出活动类型的所有可用项；
- **标签**: 显示了可用于识别项的标注列表。一项可以多次标记。在列表中选择一个标签，选项卡将更新为仅显示属于该标签的项。若要创建新标签，稍后可以附着到一组项，请使用 *添加标签...* 按钮或从任何标签上下文菜单中选择 *添加标签...* ；
- **智能组**: 智能组根据设置的条件动态获取其符号 (参见 Fig. 10.2) 。单击 *添加智能组...* 按钮创建智能组。该对话框允许输入表达式以过滤要选择的项 (具有特定标签、名称包含字符串等)。满足输入条件的任何符号、颜色渐变、文本格式或标注设置都会自动添加到智能组中。

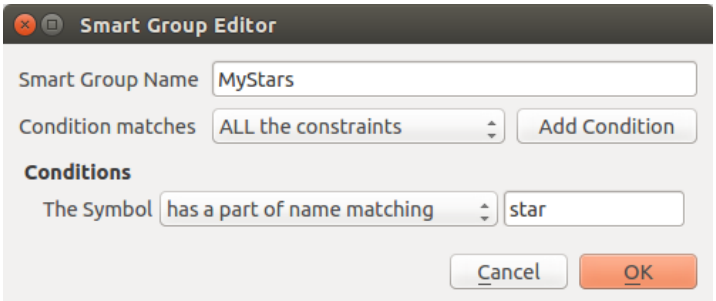


Fig. 10.2: 创建智能组

标签和智能组并不互斥：它们只是组织样式元素的两种不同方式。与根据输入约束自动获取其所属项的智能组不同，标签由用户填写。要编辑其中任何类别，您可以：

- 选择项，右键单击并选择 添加到标签 ►，然后选择标签名称或创建新标签；
- 选择标签然后按 修改组... ► 将所选标签附加到符号。每项旁边都会出现一个复选框，帮助您选择或取消选择该项。选择完成后，按 修改组... ► 完成附加标签。
- 选择智能组，按 修改组... ► 编辑智能组... 并在 智能组编辑器对话框中配置一组新的约束。此选项在智能组的上下文菜单中也可用。

要移除标签或智能组，请右键单击然后选择 移除按钮。请注意，这不会删除在类别中分组的项。

样式管理器对话框的中心显示一个框架，其中包含按选项卡组织的预览项：

- 全部包含点、线和面符号及标注设置的完整集合，以及预定义的颜色渐变和文本格式；
- 标记仅用于点符号；
- 线仅用于线符号；
- 填充仅用于面符号；
- 颜色渐变；
- abc 文本格式用于管理文本格式，储存文本的字体、颜色、描边、阴影和背景 (即标注设置的所有格式化部分，例如可以在布局中使用)；
- 标注设置用于管理标注设置，其中包括文本格式和一些特定于图层类型的设置，如标注放置、优先级、标注和渲染等；
- 图例图斑形状用于管理自定义图例图斑形状，包括 标记、线和 填充几何图形。
- *3D* 符号用于配置在*3D* 地图视图 中渲染的要素具有*3D* 属性 (拉伸、晕渲、海拔等) 的符号。

您可以在右下角的 图标视图或 列表视图中排列样式。在这两个视图中，工具提示显示了样式的较大实例。图标左侧的缩略图大小滑块可以帮助调整对话框中实际缩略图大小，以便更好地预览符号！

添加、编辑或移除项

如前所述，样式元素列在不同的选项卡中，这些选项卡的内容取决于活动类别 (标签、智能组、收藏夹等)。启用选项卡后，您可以：

- 添加新项: 点击 添加项 按钮，按照符号, 颜色渐变 或文本格式和标注 构建器描述配置项。
- 修改现有项: 选择某项并点击 编辑项 按钮，如上所述进行配置。
- 删除现有项: 要删除不再需要的元素，请选中该元素，然后单击 移除项 (也可通过右键单击获得)。该项将从本地数据库中删除。

请注意，全部标签为每种类型的项提供了对这些选项的访问权限。

右键单击所选项还可以：

- 添加到收藏夹；
- 从收藏夹移除；
- 添加到标签 ► 并选择合适的标签或创建新的标签来使用；检查当前分配的标签；
- 清除标签: 从任何标签分离符号；
- 移除项；

- 编辑项: 适用于右键单击的项;
- 复制项;
- 粘贴项...: 粘贴到样式管理器的某个类别中或 QGIS 中的其他位置 (符号或颜色按钮)
- 将选中的符号导出为 *PNG* 文件... (仅适用于符号);
- 将选中的符号导出为 *SVG* 文件... (仅适用于符号);

共享样式项

样式管理器对话框左下角的 导入/导出工具提供了一些选项，可以轻松地与他人共享符号、颜色渐变、文本格式和标注设置。这些选项也可以通过右键单击项来使用。

导出项

您可以将一组项导出到 .XML 文件中:

1. 展开 导入/导出下拉菜单，然后选择 导出项...
2. 选择要集成的项。可以使用鼠标或先前设置的标签或组进行选择。
3. 准备好后点击 导出按钮。系统将提示您选择保存文件的位置。XML 格式生成包含所有选定项的单个文件。然后可以将此文件导入到其他用户的样式库中。

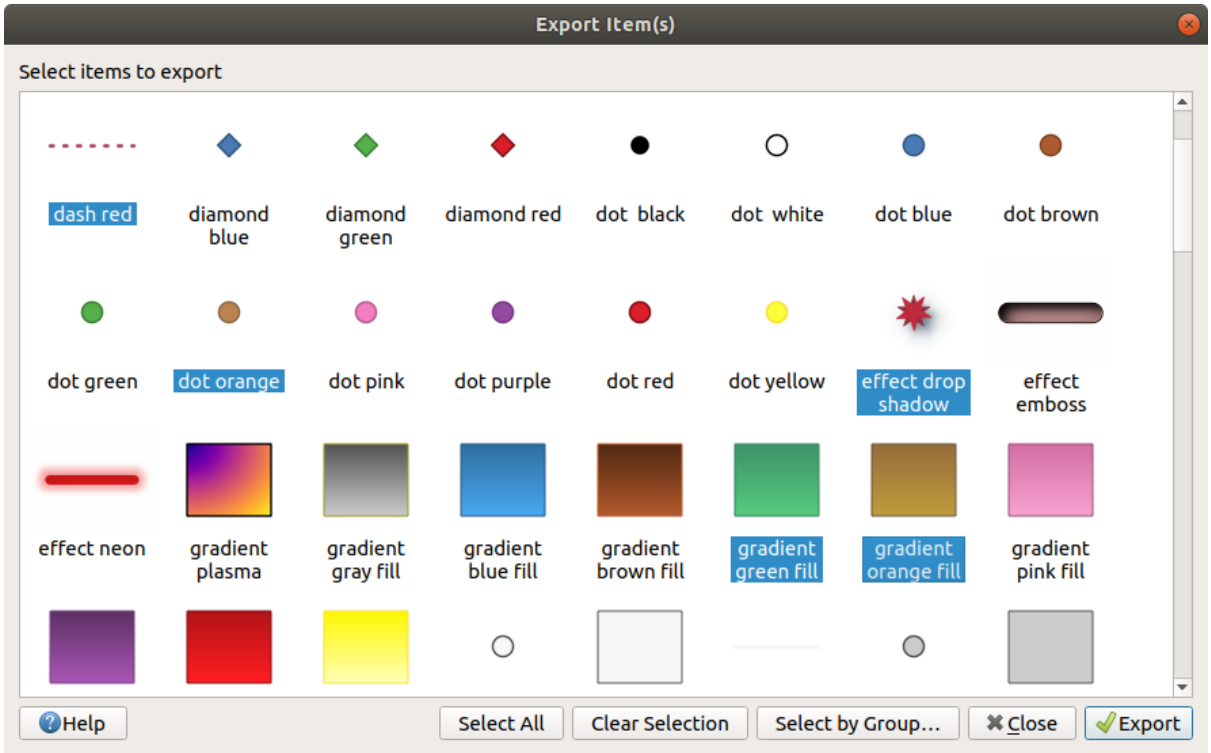


Fig. 10.3: 导出样式项

选择符号后，还可以将其导出为 .PNG 或 .SVG 文件。导出为 .PNG 或 .SVG (两者都不适用于其他样式项类型) 文件时，会为每个选定符号在给定文件夹中创建一个文件。SVG 文件夹可以在 设置 ► 选项 ► 系统菜单中添加到 *SVG* 路径中，允许其他用户直接访问所有这些符号。

导入项

可以通过导入新项来扩展样式库：

1. 展开 导入/导出下拉菜单，然后选择对话框左下角的 导入项。
2. 在新对话框中，指示样式项的来源 (可以是磁盘或 URL 的 `.xml` 文件)。
3. 设置是否勾选将导入的项 添加到收藏夹。
4. 勾选 不导入嵌入的标签，避免导入与要导入的项相关联的标签。
5. 给出要应用于新项的任何 附加标签的名称。
6. 从预览中选择要添加到库的符号。
7. 点击 导入按钮。

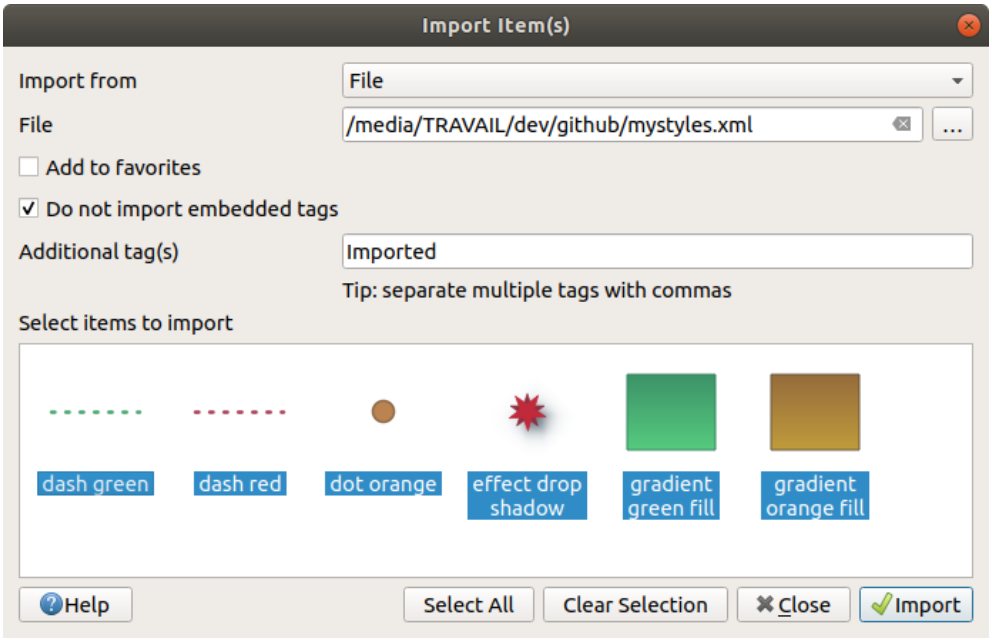


Fig. 10.4: 导入样式项

使用浏览器面板

也可以直接从 浏览器面板将样式项导入当前用户配置文件样式数据库：

1. 在浏览器中选择样式 `.xml` 文件
2. 将其拖放到地图画布上，或右键单击并选择 导入样式...
3. 在导入项 后填充 导入项对话框
4. 点击 导入按钮，选定的样式项将添加到样式数据库

在浏览器中双击样式文件将打开 样式管理器对话框，其中显示文件中的样式项。您可以对其进行选择，然后点击 复制到默认样式... 按钮将它们导入到活动样式数据库。标签可以指定给项，也可通过右键单击 打开样式... 命令。

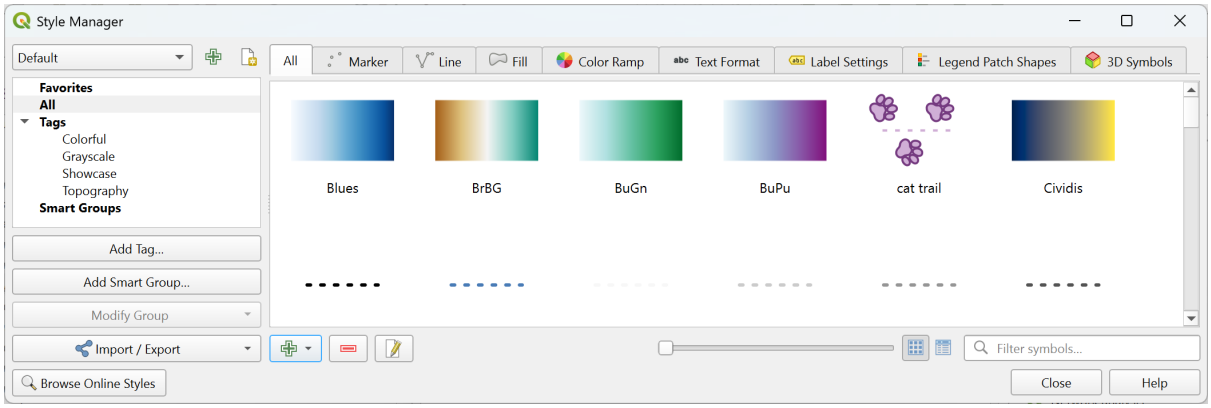


Fig. 10.5: 打开样式项文件

该对话框还允许将单个符号导出为 `.PNG` 或 `.SVG` 文件。

使用在线资源库

The QGIS project maintains a repository with a collection of styles shared by QGIS users. This is available at https://hub.qgis.org/styles and can be accessed from the *Style Manager* dialog, pressing the *Browse Online Styles* button at the bottom.

在该资源库，您可以：

1. 根据类型或名称浏览和搜索任何样式项
2. 下载样式文件并解压缩
3. 使用上述提到的任何导入方法，将基于 `.xml` 的文件加载到 QGIS 样式数据库。

10.1.2 设置颜色渐变

样式管理器对话框中的颜色渐变选项卡可以帮助您根据左侧面板中选定的类别预览不同的颜色渐变。

要创建自定义颜色渐变，请激活颜色渐变选项卡，然后单击 添加项 按钮。该按钮显示一个下拉列表，用于选择渐变类型：

- *梯度*: 给定开始和结束颜色，生成 **连续**或 **离散**的颜色渐变。双击渐变预览，可以根据需要添加任意数量的中间色标。单击色标指示器，在 *梯度色标*中，您可以：
  - 从颜色渐变开始调整其 *相对位置*。也可以用鼠标拖动指示器，或按方向键 (与 `Shift` 键组合使用可进行更大距离的移动)
  - 指定在颜色之间进行插值时要使用的颜色模型：可以是 *RGB* 、*HSL* 或 *HSV* 。在某些情况下，此选项可以帮助避免非饱和色的中间色调，从而产生更赏心悦目的梯度。
  - 为 *HSL* 或 *HSV* 颜色规格的 **色彩**成份设置插值应遵循的方向，可以是 *顺时针方向*或 *逆时针方向*。
  - 设置*颜色属性*
  - 点击 *删除色标*或按 `DEL` 键移除色标

  *绘图*组提供了另一种设计颜色渐变的图形方式，可以更改色标的位置或不透明度和 HSL 分量。

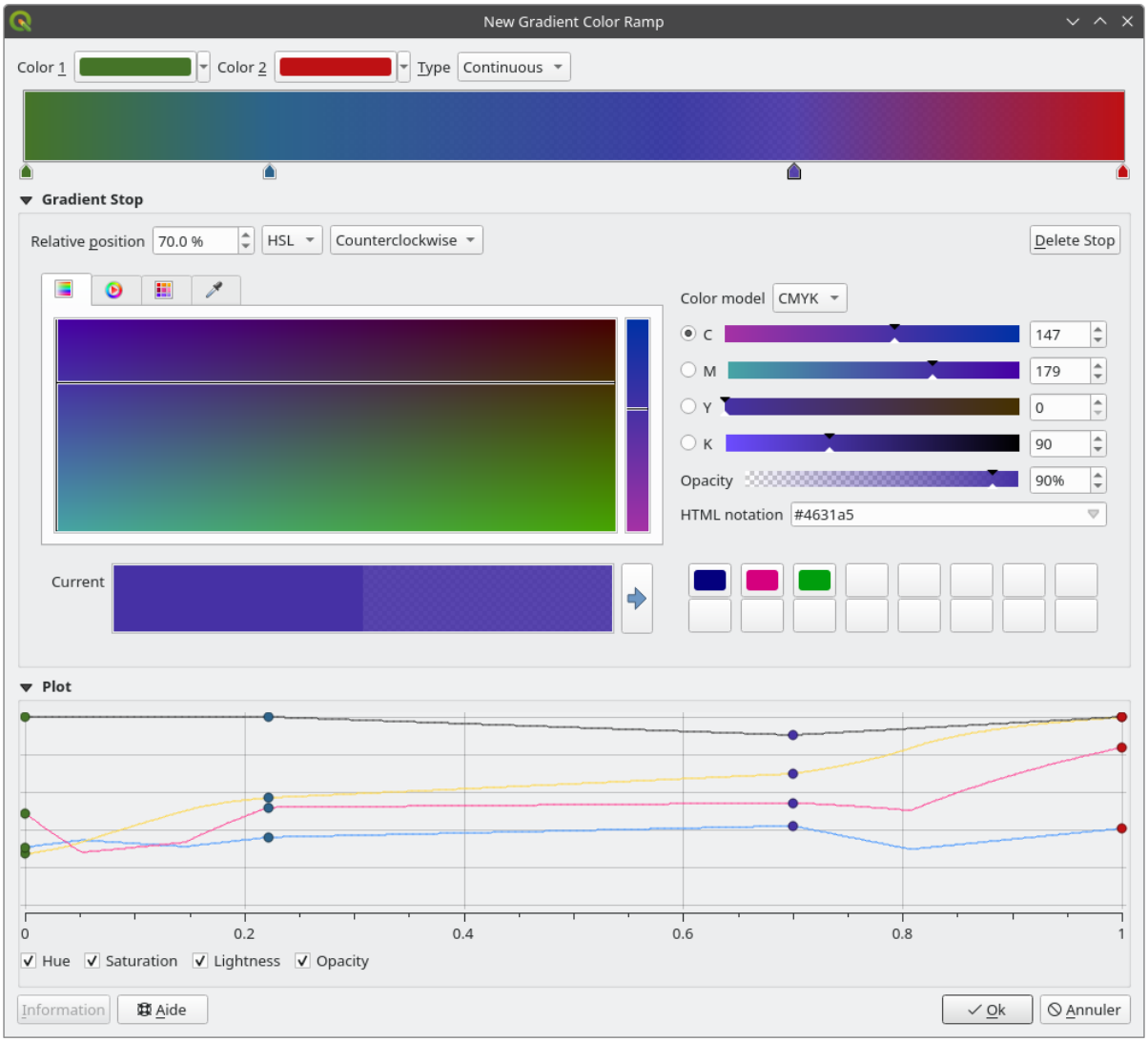


Fig. 10.6: 具有多个色标的自定义梯度颜色渐变示例

Hint

将颜色从色斑拖放到梯度渐变预览可添加新色标。

- 颜色预设: 允许创建由用户选择的颜色列表组成的颜色渐变;
- 随机: 基于 色相、饱和度、值和 不透明度的值的范围以及多种颜色 (类别) 创建一组随机颜色;
- 目录分类: *ColorBrewer*: 一组预定义的离散颜色渐变，可以自定义渐变中的颜色数量;
- 或 目录分类: *cpt-city*: 访问颜色渐变的完整目录到本地 另存为标准梯度。cpt-city 选项打开新的对话框，其中包括数百个" 开箱即用" 的主题。

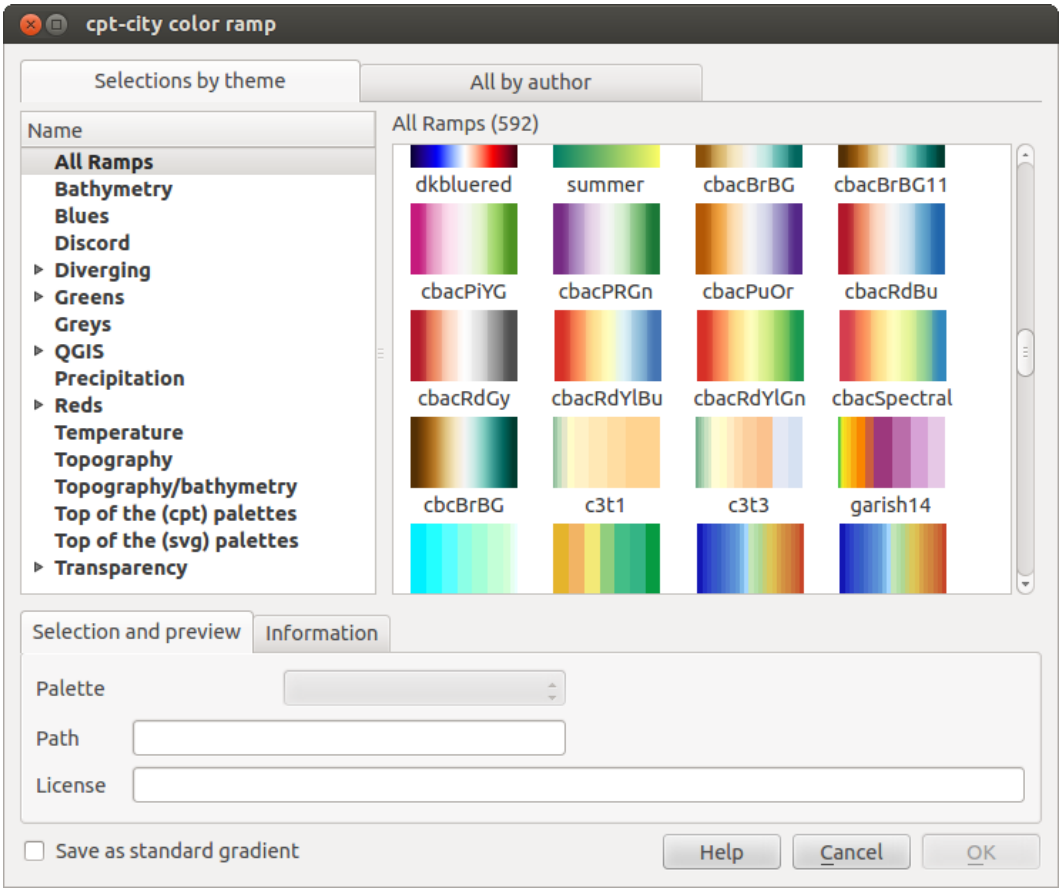


Fig. 10.7: 具有数百个颜色渐变的 cpt-city 对话框

10.1.3 创建图例图斑形状

若要创建新的图例图斑形状，请激活 *图例图斑形状*选项卡，然后单击 添加项 按钮。该按钮显示一个下拉列表，用于选择几何图形类型：

- *标记图例图斑形状...*: 用于点几何图形。
- *线图例图斑形状...*: 用于线几何图形。
- *填充图例图斑形状...*: 用于多边形几何图形。

所有三个选项将显示相同的对话框。

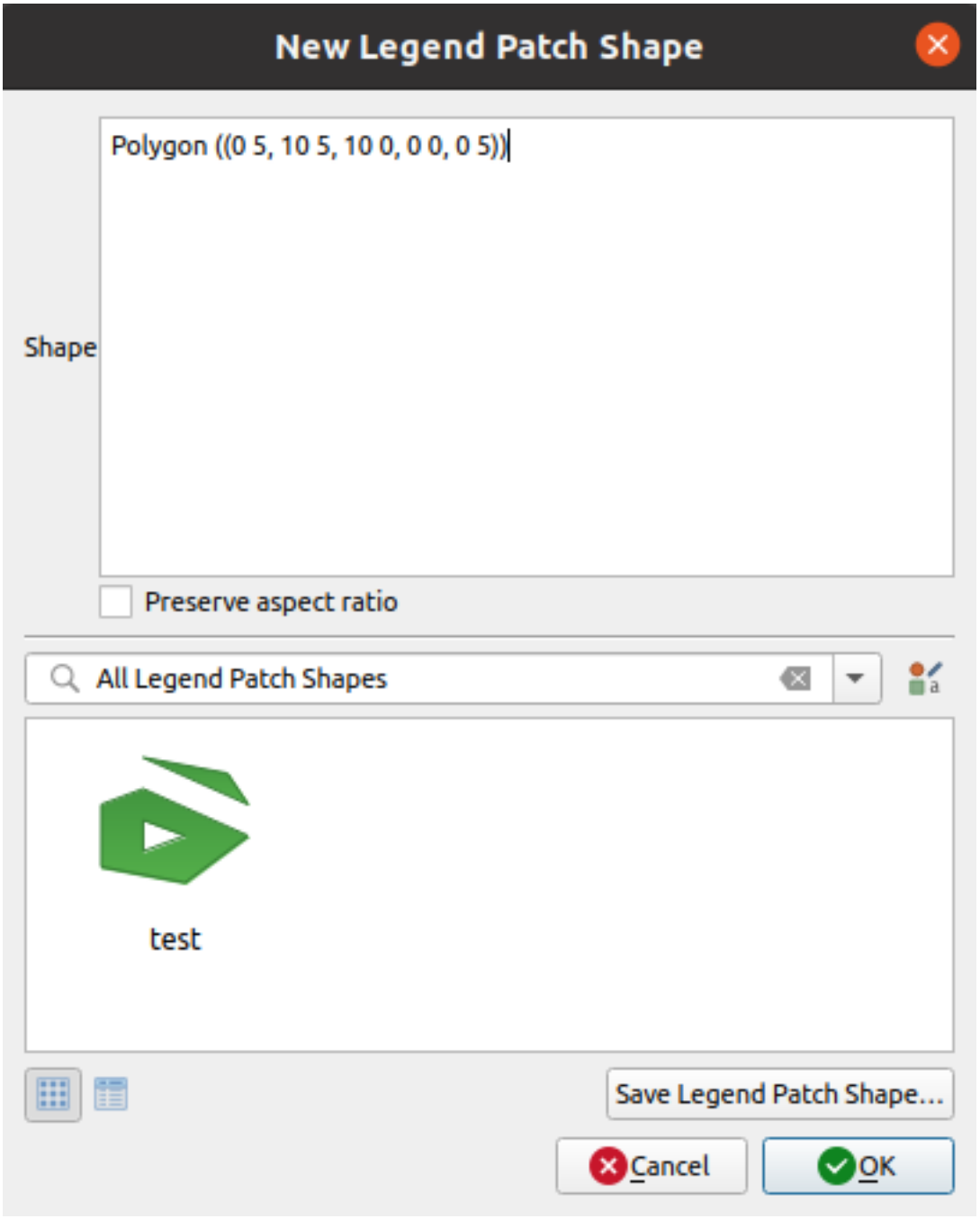


Fig. 10.8: 创建新的图例图斑形状

仅形状类型和显示的图例图斑形状会因所选几何图形类型而有所不同。将提供以下选项：

- 形状: 将图例图斑形状的形状定义为 WKT 字符串。可以使用单个和多部件几何图形，但不能使用几何图形集合。
- 保持纵横比
- 图标视图或 列表视图可用于图例图斑形状显示，按标签过滤。

定义新形状后，可以 保存图例图斑形状... 或按 确定 键，这两种操作都将显示相同的对话框。

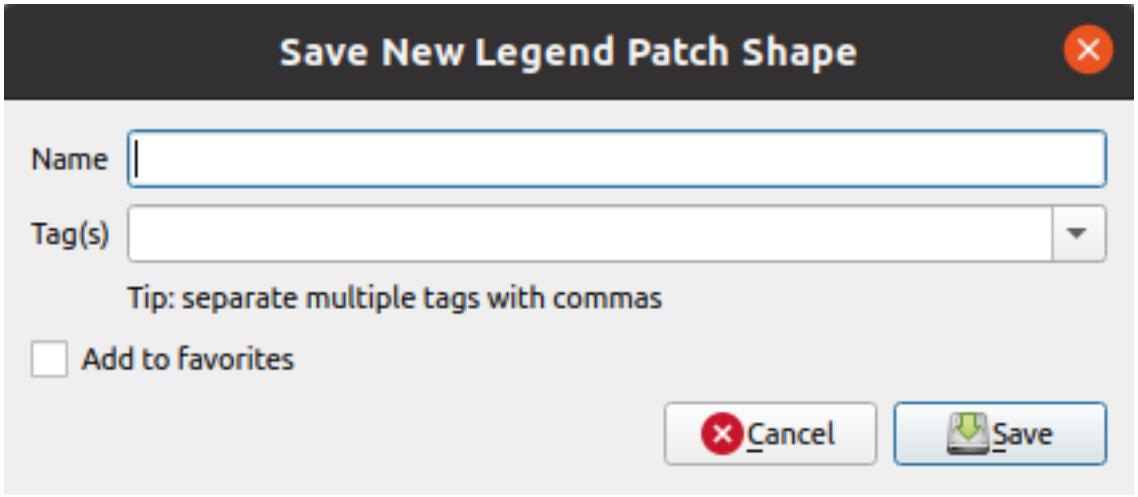


Fig. 10.9: 保存新的图例图斑形状

在此必须输入名称、描述形状的标签，以及是否应该将其添加到收藏夹。

如果点击 保存... 按钮，该形状将添加到列表中，并且返回 新建图例图斑形状对话框以继续创建新形状。

10.2 符号选择器

符号选择器是设计符号的主对话框。可以创建或编辑标记、线或填充符号。

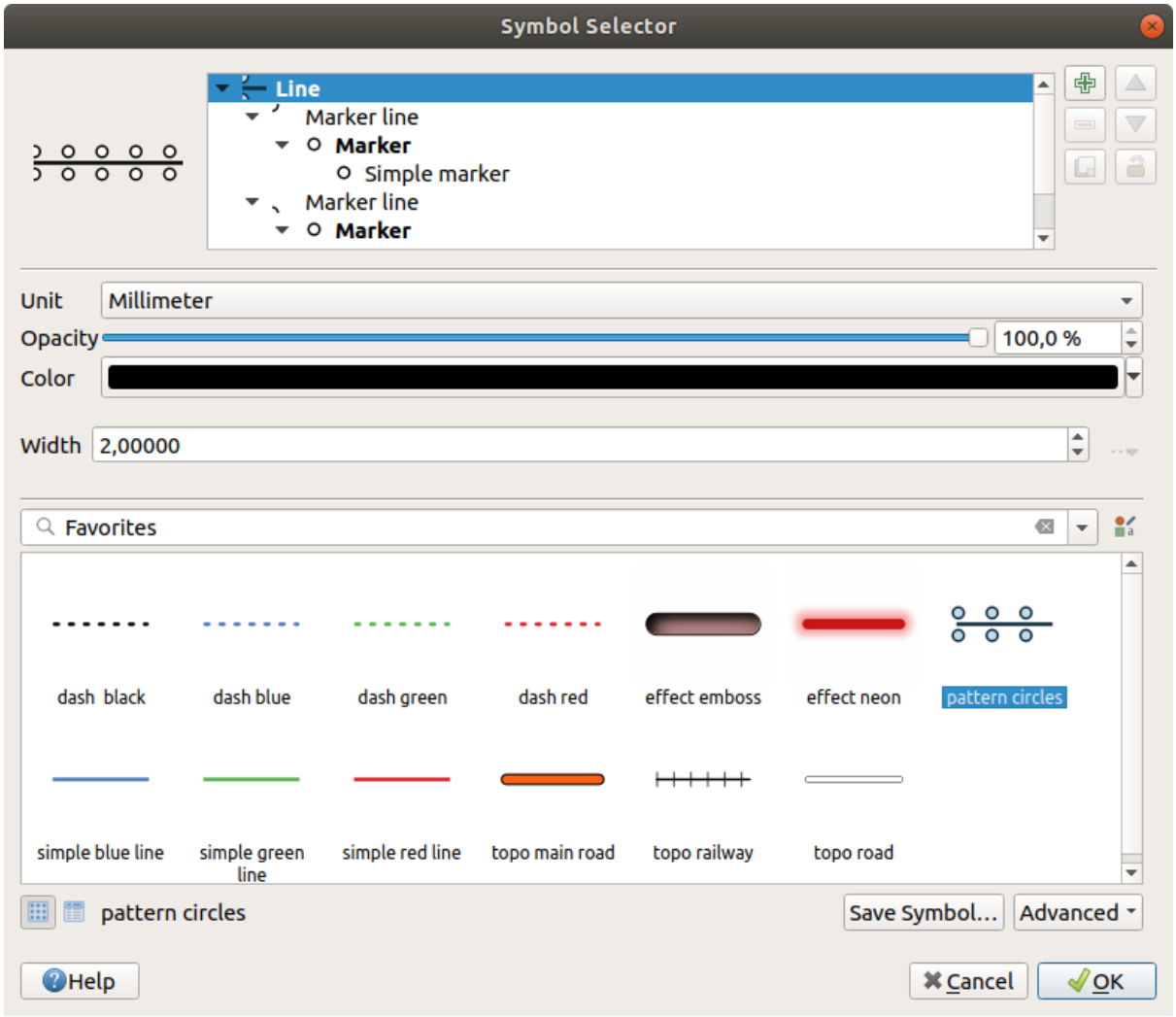


Fig. 10.10: 设计线符号

符号选择器对话框由两个主要组件构成：

- 符号树，显示组合后形状新全局符号的符号图层
- 以及用于配置树中所选符号图层的设置。

10.2.1 符号图层树

符号可以由多个 符号图层组成。符号树显示了这些符号图层的叠加层次，并被组合形成新的全局符号。此外，一旦符号属性发生更改，就会更新动态符号表示。

根据在符号树条目中选择的级别，提供了各种工具来帮助您管理树：

- 添加新符号图层: 可以根据需要堆栈任意数量的符号
- 移除所选符号图层
- 锁定符号图层颜色: 当用户在全局 (或更高级) 符号级别更改颜色时， 锁定的颜色保持不变
- 复制 (一组) 符号图层
- 向上或向下移动符号图层

10.2.2 配置符号

在 QGIS 中，配置符号分为两步：符号然后是符号图层。

符号

在树的顶层，取决于图层几何图形，可以是 **标记**、**线**或 **填充**类型。每个符号可以嵌入一个或多个符号 (包括任何其他类型的符号) 或符号图层。

您可以设置一些应用于全局符号的参数：

- 单位: 可以是 **毫米**、**点**、**像素**、**比例米**、**地图单位**或 **英寸** (更多信息请参阅*Unit Selector*)
- 不透明度
- 颜色: 当用户更改此参数时，其值将覆盖所有未锁定的子符号颜色
- 大小和 旋转用于标记符号
- 宽度用于线符号

> **Tip**
>
> 在符号级别使用 大小 (用于标记符号) 或 宽度 (用于线符号) 属性按比例调整其所有嵌入符号图层 的尺寸。

> **Note**
>
> 在样式管理器对话框设置符号时，宽度、大小或旋转参数旁边的覆盖自定义数据 按钮处于非活动状态。当符号连接到地图图层时，此按钮可帮助您创建比例或多变量分析 渲染。

- 符号库进行过滤。还可以使用 样式管理器 按钮更新符号列表，然后打开同名对话框。在那里，可以使用 矢量样式管理器部件中显示的任何功能。

  符号显示为：

  - 在图标列表中 (带有缩略图、名称和相关标签)，使用框架下方的 列表视图 按钮；
  - 或者使用 图标视图 按钮作为图标预览。

- Press the *Save Symbol* button to open the *Save New Symbol* dialog. Here, you can choose the *Destination* where you want to add the symbol being edited, give it a *Name* and add *Tag(s)*. You also have the option to Add to favorites your new symbol.

- 使用 高级 选项，您可以：
  - 对于线和填充符号，按画布范围裁剪要素。
  - 对于填充符号, 强制右手定则朝向: 允许强制渲染的填充符号遵循环方向的标准” 右手定则” (即外部环为顺时针方向，内部环均为逆时针方向的多边形)。

    仅在渲染时应用方向固定，并且原始要素几何图形保持不变。无论渲染的数据集和单个要素的环方向如何，这允许创建具有一致外观的填充符号。
  - for marker symbols, the *Buffer settings...* enables addition of a halo effect around the marker symbol in order to make it more readable against different backgrounds. The buffer is calculated and drawn using the shape of ALL the symbol layers in the marker. You can adjust the *Size*, *Join style* and *Symbol* of display.
  - for all symbol types, enable and configure a distance for the map canvas *Extent buffer*, or the *Animation settings*.

  - Depending on the *symbology* of the layer the symbol is being applied to, additional settings are available in the *Advanced* menu:
    * 符号层级... 定义符号渲染的顺序
    * 尺寸由数据定义的图例
    * 匹配已保存符号... 和 匹配来自文件的符号... 自动将符号分配给类

符号图层

在树的较低层级，可以自定义符号图层。可用的符号图层类型取决于上面的符号类型。可以在符号图层应用 绘制效果 以增强其渲染效果。

因为无法描述所有符号图层类型的所有选项，所以下面仅提及特定且重要的选项。

常用参数

一些常用选项和控件可用于构建符号图层，无论是标记、线还是填充子类型：

- 颜色选择器 控件可简化颜色操作
- 单位: 可以是 **毫米**、**点**、**像素**、**比例米**、**地图单位**或 **英寸** (更多信息请参阅*Unit Selector*)
- 数据定义覆盖 控件几乎接近所有选项，扩展了自定义每个符号的功能 (更多信息请参阅*Data defined override setup*)
- 启用符号图层选项控制符号图层的可见性。渲染符号时不会绘制禁用的符号图层，而是将其保存在符号中。在寻找符号的最佳设计时，能够隐藏符号图层很方便，因为无需为测试而移除任何符号图层。然后，通过数据定义重写，可以基于表达式 (例如，使用要素属性) 隐藏或显示不同的符号图层。
- 用于效果渲染 的 绘制效果按钮。

> **Note**
>
> 虽然以下描述假设符号图层类型绑定到要素几何图形，但请记住，可以将符号图层嵌入到其他图层。在这种情况下，较低层级的符号图层参数 (放置、偏移等) 可能绑定到较高层级的符号，而不是绑定到要素几何图形本身。

标记符号

标记符号具有多种 符号图层类型，适用于点几何图形要素：

Table 10.1: List of marker symbol layer types

Simple marker	*Animated marker*	*Ellipse marker*
Filled marker	*Font marker*	*Geometry generator*
Mask	*Raster image marker*	*Vector Field marker*
SVG marker		

- **简单标记** (默认)

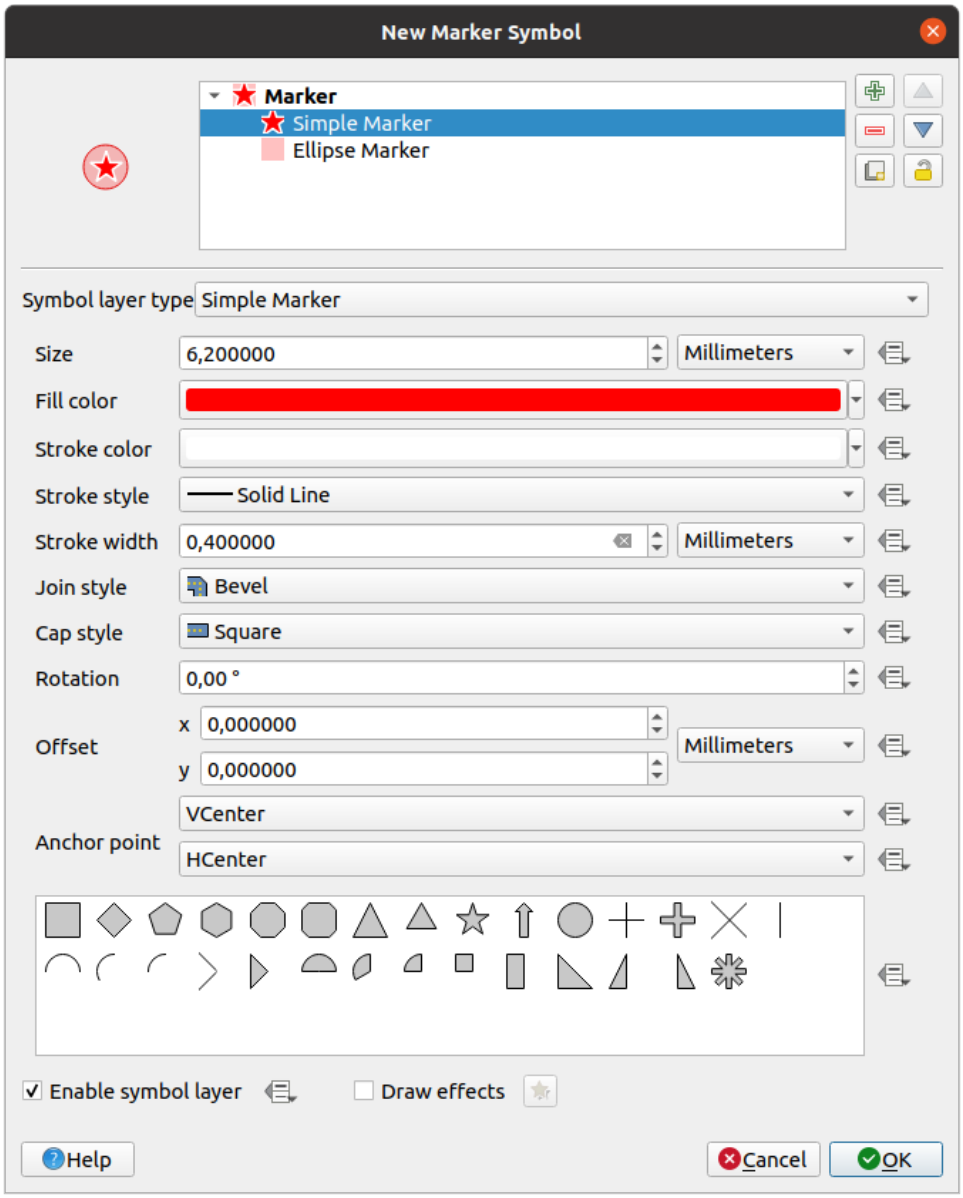


Fig. 10.11: 设计简单标记符号

简单标记符号图层类型具有以下属性：

- 在各种支持单位中的 大小
- 填充颜色
- 预定义列表和 描边大小中的 描边颜色、描边样式
- 连接样式: 可以是 **斜角**、**尖角**或 **圆角**
- 端点样式: 可以是 **方角**、**扁平**或 **圆角**
- 旋转角度
- 从要素沿 *X* 和 *Y* 方向 偏移
- 锚点: 将符号的象限点定义为放置原点。这是应用 偏移的点。

- **Animated marker** (see *Animated marker*)
- **椭圆标记**: 简单标记符号图层，具有可自定义的宽度和高度

- **实心标记**: 类似于简单标记符号图层，不同之处在于使用填充子符号 渲染标记。这允许使用所有现有的 QGIS 填充 (和描边) 样式渲染标记，例如梯度或形状突发填充。
- **文字符号标记**: 类似于简单标记符号图层，只是使用已安装的字体渲染标记。其附加属性为:
  - 字体族
  - 字体样式
  - 字符, 表示要显示为符号的文本。可以在字体字符集控件中键入或选择，您可以使用所选设置实时 预览。
- **几何图形生成器** (请参阅几何图形生成器)
- **掩膜**: 其子符号定义了掩膜形状，颜色属性将被忽略，并且仅使用不透明度。当标注或其他颜色相近的标记符号重叠时辨认困难，这很方便。更多信息请访问*Masks Properties* 。
- **栅格图像标记**: 使用图像 (PNG 、JPG 、BMP ...) 作为标记符号。图像可以是磁盘上的文件、远程 URL、嵌入样式数据库 (更多信息) ，也可以是编码为 base64 的字符串。图像的宽度和高度可以单独设置，也可以使用 锁定纵横比 。可以使用任何常用单位 设置大小，也可以设置图像为原始大小的百分比 (按宽度缩放)。
- **矢量字段标记** (请参阅矢量字段标记)
- **SVG 标记**: 为您提供来自 SVG 路径的图像 (在 设置 ► 选项... ► 系统菜单中设置) 渲染为标记符号。符号的宽度和高度可以单独设置，也可以使用 锁定纵横比 。每个 SVG 文件的颜色和描边可以进行调整。图像可以是磁盘上的文件、远程 URL、嵌入样式数据库 (更多信息) ，也可以是编码为 base64 的字符串。

  该符号也可以使用 *动态 SVG 参数*进行设置。请参阅参数化 *SVG* 章节的参数化 SVG 符号。

> **Note**
>
> SVG 版本要求
>
> QGIS 渲染是遵循 SVG Tiny 1.2 配置 的 SVG 文件，旨在从手机和 PDA 到笔记本和台式机等一系列设备范围实现相同的效果，因此包括 SVG 1.1 Full 中包含的功能子集，以及扩展 SVG 功能的新功能。
>
> 这些规范中未包含的某些功能可能无法在 QGIS 正确渲染。

线符号

适用于线几何图形要素的线符号具有以下符号图层类型:

Table 10.2: List of line symbol layer types

Simple line	*Arrow*	*Filled line*
Geometry generator	*Hashed line*	*Interpolated line*
Linear referencing	*Lineburst*	*Marker line*
Raster line		

- **简单线** (默认)

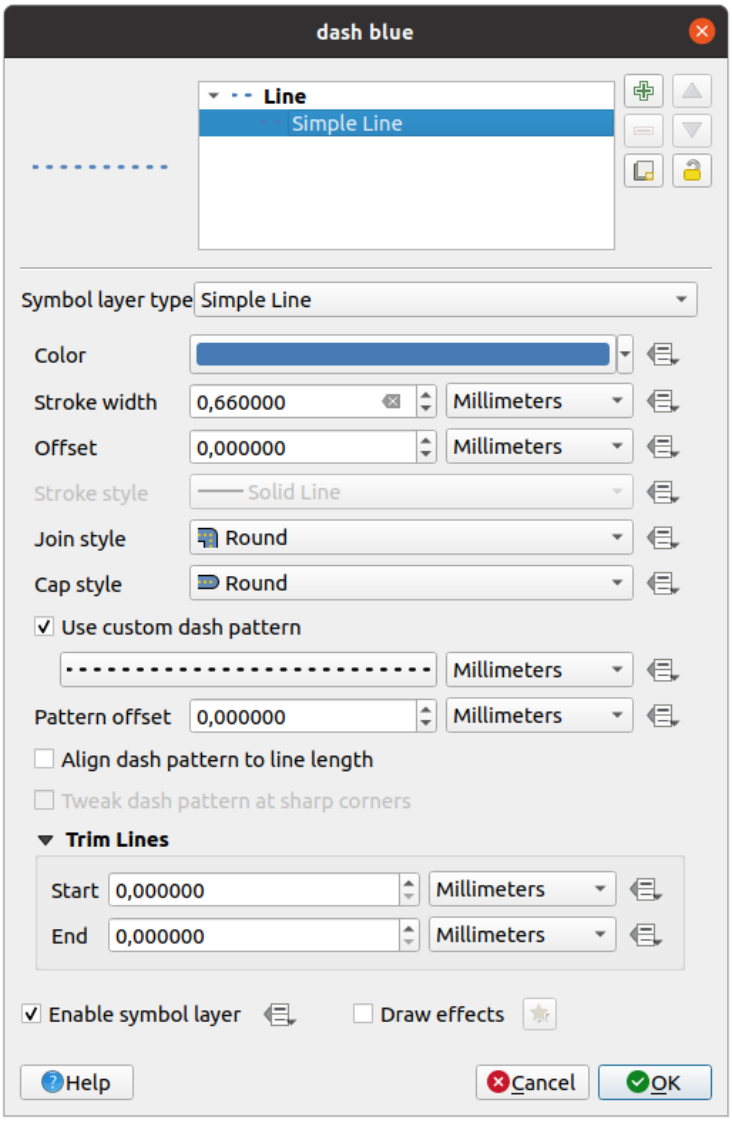


Fig. 10.12: 设计简单线符号

简单线符号图层类型具有许多与简单标记符号 相同的属性，此外：

- 使用自定义虚线图型：使用自定义虚线代替 描边样式设置。您需要以选定的单位定义整形模型的连续虚线和空格的长度。图型的总长度显示在对话框的底部。
- 图型偏移：线中虚线/空格的位置可以调整，这样就可以放置在更好的位置，以解释线中的转角 (也可能用于" 对齐" 邻接的虚线图型边界)
- 将虚线图型与线长对齐：将调整虚线图型的长度，使线以完整的虚线元素结束，而不是有间隙。
- 调节尖拐角的虚线图型：动态调整虚线图型的布置，以便尖拐角由进出尖拐角的完整虚线元素表示。这依赖于 将虚线图型与线长对齐。
- 从 起点和/或 终点 裁剪线：允许线渲染在绘制线时从实际线串裁剪开始 x 毫米和结束 y 毫米。它支持一系列单位 ，包括总线长度的百分比，并且可以为附加控制而定义的数据。例如，当创建复杂符号时，可以使用开始/结束裁剪距离，其中线图层不应与放置在线起点和终点的标记符号图层重叠。

- **箭头**：将线绘制为具有可配置 (和数据定义) 的单头或双头的弯曲 (或非) 箭头：
  - 头部类型

  - 箭头类型
  - 箭头宽度
  - 箭头起点宽度
  - 箭头头部长度
  - 箭头头部厚度
  - 偏移量

  可以创建 弯曲箭头 (线要素必须至少有三个顶点) 和 每条线段重复箭头。还可以使用填充符号 (如梯度或形状突发填充) 渲染箭头主体。结合几何图形生成器，这种图层符号的类型可帮助表示流型图。

- **几何图形生成器** (请参阅几何图形生成器)
- **内插线**: 允许渲染的 描边宽度和/或 颜色可以不变 (给定 固定宽度和 单色参数) 或沿几何图形变化的线。当改变时，必要的输入是:
  - 起始值和 最终值: 用于在要素几何图形的末端进行插值的值，可以是固定值、要素的属性或基于表达式。
  - 最小值和 最大值: 在其间执行插值的值。按 加载 按钮可根据应用于图层的最小和最大起始/最终值自动填充。
  - 仅适用于描边选项:
    * 最小宽度和 最大宽度: 定义可变宽度的范围。最小宽度指定给 最小值，最大宽度指定给 最大值。可以关联单位 。
    * 使用绝对值: 仅考虑插值的绝对值 (负值用作正值)。
    * 忽略超出范围: 默认情况下，当要素的 [起始值 – 最终值] 范围不包含在 [最小值 – 最大值] 的范围内时，要素的几何图形界外部分将以最小或最大宽度渲染。勾选此选项则完全不渲染。
  - 对于可变颜色，可以使用颜色渐变类别 的任何插值方法

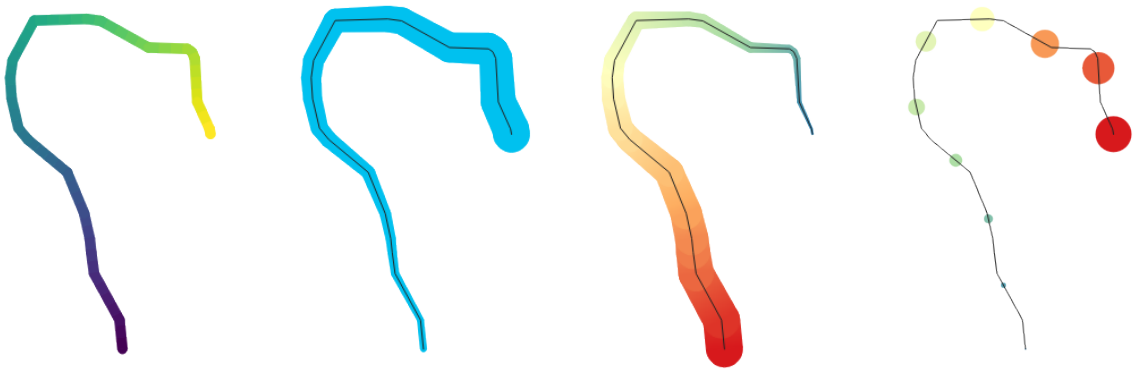


Fig. 10.13: 插值线示例

- **标记线**: 在线长度之上重复标记符号 。
  - 标记位置可以使用规则间隔设置 (从第一个顶点开始) 和/或线几何图形 (在第一个或最后一个顶点、内部顶点、线或每条线段的中心点或每条曲线点上) 进行设置。
  - 启用第一个或最后一个顶点放置时， 放置在每个部件末端选项将使标记也在多部件几何图形的每个部件的第一个或最后一个顶点处渲染。

  - 沿线偏移: 标记的放置也可以在您选择的单位 中沿线给出偏移量 (毫米、点、地图单位、比例米、百分比等):
    * 正值将使标记符号在线方向 (使用 在第一个顶点上和 间隔分布放置) 和向后 (使用 在最后一个顶点上放置) 偏移。
    * 非闭合线的负值将导致符号没有偏移 (对于 在第一个顶点上和 在最后一个顶点上的放置) 或向后偏移 (从最后一个顶点开始)。
    * 对于闭合环，QGIS 将偏移视为继续绕环循环 (向前或向后)。例如，将偏移设置为 150% (分别为 -10% 或 -110%) 将导致偏移被视为闭合环长度的 50% (分别为 90%)。
  - 旋转标记以遵循线方向选项设置每个标记符号的方向是否应相对于线方向定向。

    因为线通常是由一系列不同方向的线段组成，所以标记的旋转是通过沿该线对指定距离取平均值来计算的。例如，将 超过平均角度属性设置为 4mm 意味着沿线在符号位置前后 2mm 的两个点用于计算该标记符号的线角度。这样可以平滑 (或消除) 与整个线方向的任何微小局部偏差，从而使标记线符号具有更好的视觉方向。
  - 线偏移: 标记符号也可以从线要素偏移。

- **散列线**: 在线符号的长度上重复线段 (散列)，并使用线子符号渲染每条单独的线段。也就是说，散列线类似于标记线，其中标记符号被线段取代。因此，散列线与标记线符号具有相同的属性 ，以及:
  - 散列线长度
  - 散列线旋转

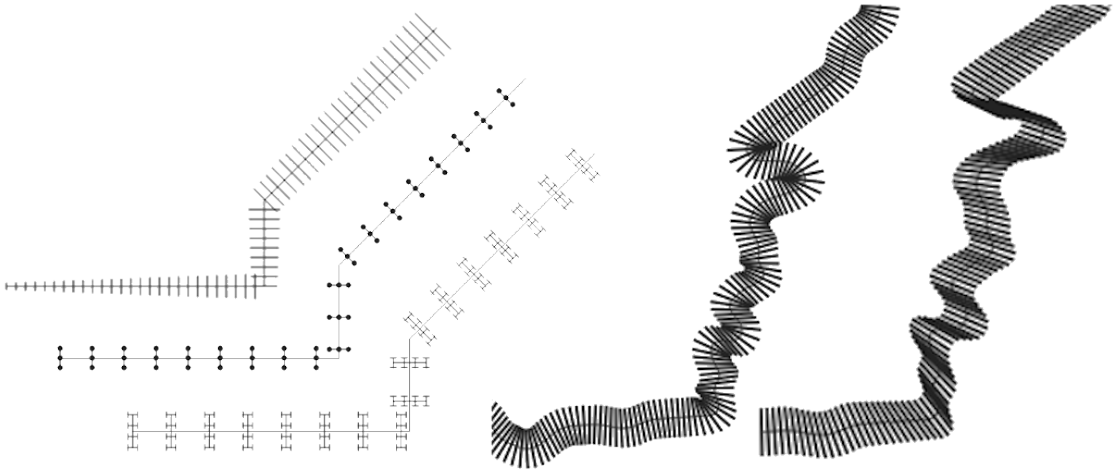


Fig. 10.14: 散列线示例

- **栅格线**: 按照线要素形状的长度渲染并重复栅格图像。可以调整 描边宽度、偏移量、连接样式、端点样式和 不透明度。

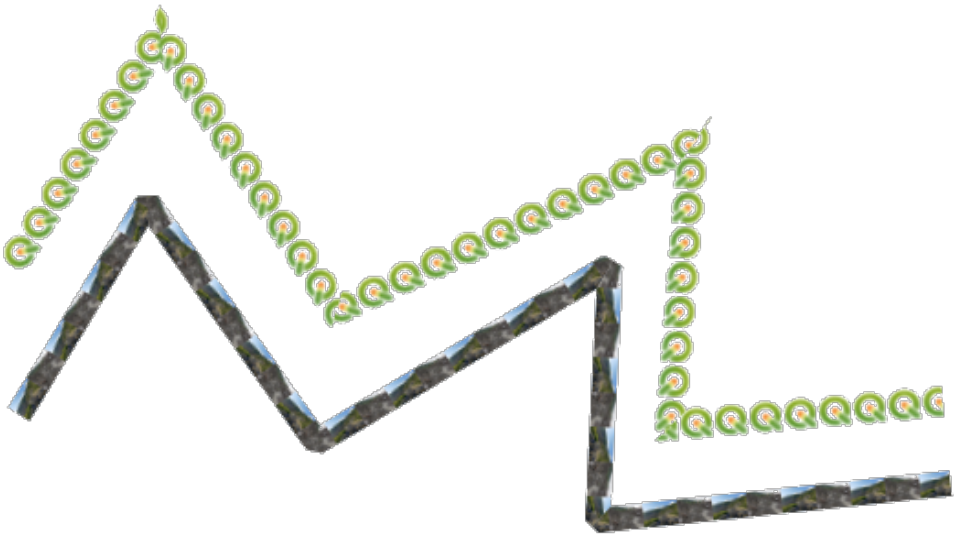


Fig. 10.15: 栅格线示例

- **Linear referencing**: allows placing text labels at regular intervals along a line, or at positions corresponding to existing vertices. Positions can be calculated using Cartesian distances, or interpolated from z/m values. Parameters include:
  - *Measure placement*: Labels can be placed at regular cartesian 2d distances, at regular linearly interpolated spacing calculated using the Z or M values in geometries, or at existing vertices. A fixed or data-defined *Interval* is given.
  - *Quantity*: Label displayed at each position represent the running distance along the line, or the linearly interpolated Z or M value. Labels are rendered using the full range of *text* and *number* formatting functionalities available in QGIS.

> **Note**
>
> When using the distance-based placement or labels, the distances are calculated using 2D only. Cartesian calculations are based on the original layer CRS.

  - *Skip multiples of*: If set, labels which are a multiple of this value will be skipped over. This allows construction of complex referencing labels, e.g. where a symbol has two linear referencing symbol layers, one set to label every 100m in a small font, skipping multiples of 1000, and a second set to label every 1000m in a big bold font.
  - *Average angle over*: Labels are rendered using an angle calculated by averaging the linestring, so sharp tiny jaggies don't result in unslightly label rotation.
  - *Show marker symbols*, at referenced points in the line feature, using a full QGIS marker symbol. This allows e.g. showing a cross-hatch at the labeled point, for a "ruler" style line.

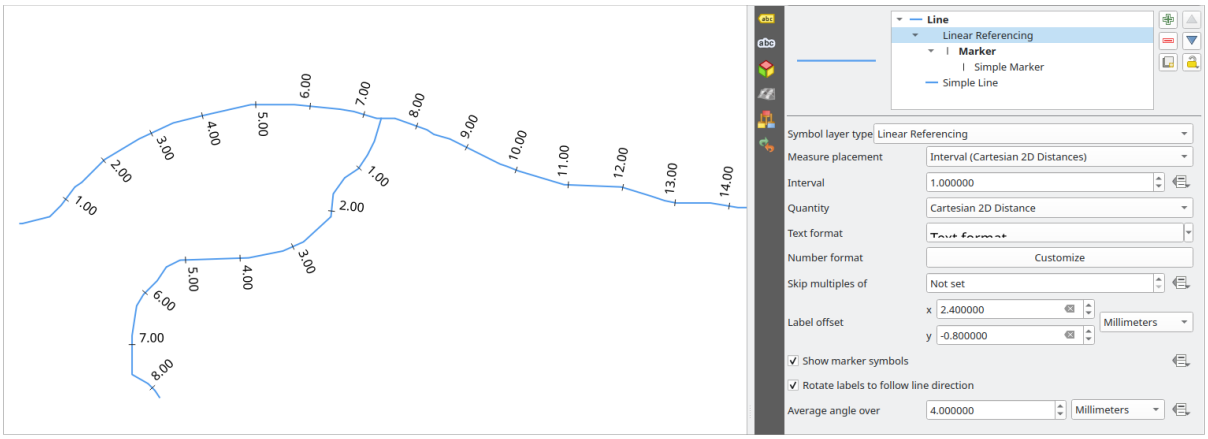


Fig. 10.16: Drawing linear labels along line feature

- **线突发**: 沿线宽渲染渐变。可以在 双色或 颜色渐变和 描边宽度之间进行选择，并且可以调整 描边宽度、偏移量、连接样式和 端点样式。

Fig. 10.17: 线突发示例

- **Filled line**: renders the interior of the lines using a fill symbol allowing for lines filled with gradients, line hatches, etc. The *Stroke width*, *Offset*, *Join style*, *Cap style* can be adjusted.

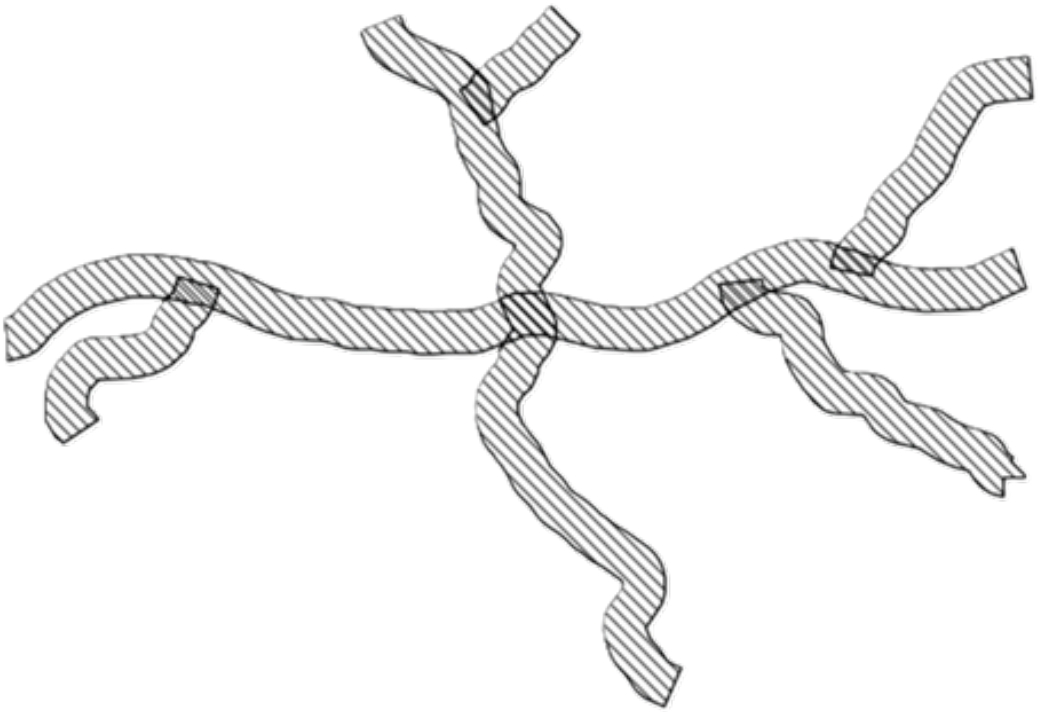


Fig. 10.18: Examples of filled lines

填充符号

适用于多边形几何图形要素，填充符号也有几种符号图层类型：

Table 10.3: List of fill symbol layer types

For polygon interior	For polygon ring
Simple fill	*Outline: Arrow*
Centroid fill	*Outline: Filled line*
Geometry generator	*Outline: Hashed line*
Gradient fill	*Outline: Interpolated line*
Line pattern fill	*Outline: Linear referencing*
Point pattern fill line	*Outline: Lineburst*
Random marker fill	*Outline: Marker line*
Raster image	*Outline: Raster line*
SVG fill	*Outline: Simple line*
Shapeburst fill	

- **简单填充** (默认): 用相同颜色填充多边形

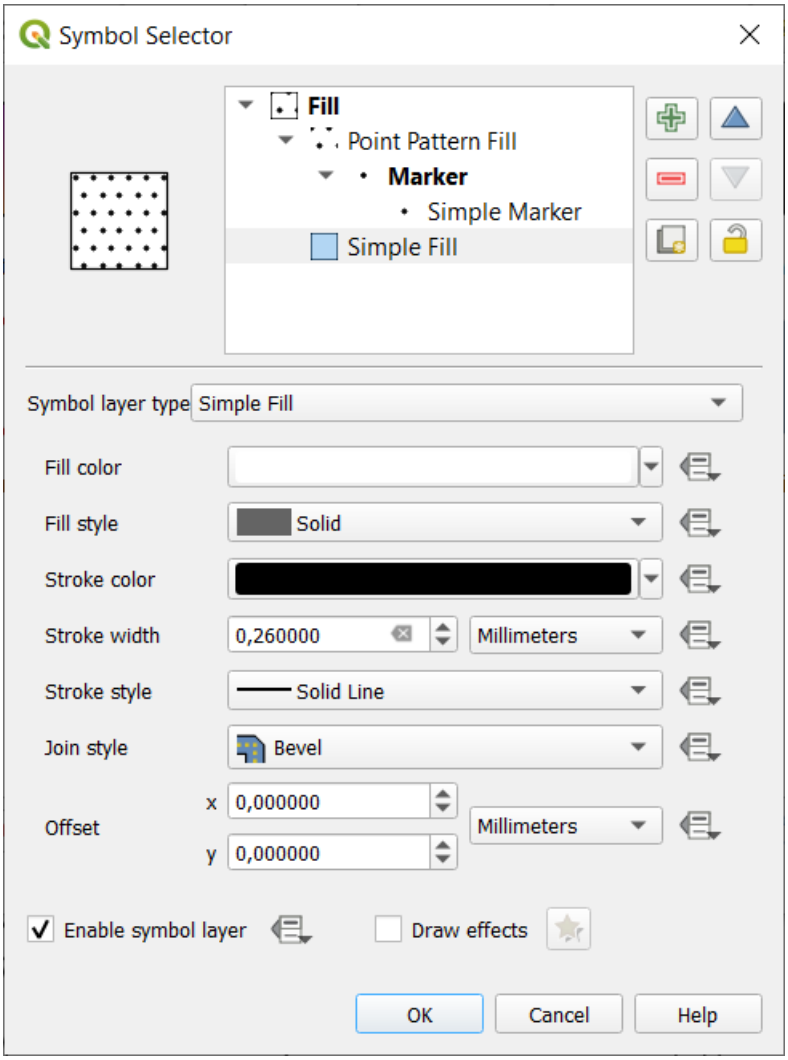


Fig. 10.19: 设计简单填充符号

- **质心填充**: 在可见要素的质心处放置标记符号 。标记的位置可能不是要素的真实质心，因为计算会考虑裁剪到地图画布中可见区域的多边形进行渲染，并忽略孔。如果需要精确的质心，请使用几何图形生成器符号 。

  您可以:

  - 强制在多边形内放置标记
  - 在多部件要素的每个部件绘制标记或仅将点放在其最大的部件中
  - 显示全部或部件标记符号，保持部件与当前要素几何图形 (将标记裁剪到多边形边界) 或符号所属几何图形部件重叠 (将标记仅裁剪到当前部件边界)

- **几何图形生成器** (请参阅几何图形生成器)
- **渐变填充**: 基于简单的两个颜色渐变或预定义梯度颜色渐变 ，使用放射、线性或锥形渐变填充多边形。梯度可以在单个要素的基础上或整个地图范围内旋转和应用，也可以通过坐标或使用 (要素或地图的) 质心设置起点和终点。可以定义数据定义的偏移量。
- **线图案填充**: 使用线符号图层 的阴影图案填充多边形。您可以设置:
  - 对齐: 定义图案相对于要素的定位方式:
    - 对齐图案到要素: 在每个要素中渲染线
    - 对齐图案到地图范围: 图案在整个地图范围内渲染，允许线在要素之间完美对齐

  - 线的 旋转为逆时针方向
  - 间距: 连续线之间的距离
  - 线与要素边界的 偏移距离
  - 裁剪: 允许控制如何将填充中的线裁剪为多边形形状。选项包括:
    * 仅在渲染期间裁剪: 将创建覆盖要素的整个边界框的线，然后在绘制时进行剪裁。线末端 (起点和终点) 将不可见。
    * 渲染前裁剪线: 在渲染之前，线被剪裁为多边形的准确形状。线末端 (包括端点样式、起点/终点标记线对象等) 将可见，有时可能会延伸到多边形之外 (取决于线符号设置)。
    * 无裁剪: 不进行剪裁 - 线将覆盖要素的整个边界框

> **Note**
>
> When exporting maps to vector formats such as PDF or SVG, or when the line symbol in the pattern has data-defined properties, QGIS renders the line pattern fill using a line-by-line vector-based approach. This results in smaller output file sizes, higher quality exports, and improved editability of the pattern lines in external vector graphics applications.

- **点图案填充**: 用标记符号 的网格图案填充多边形。您可以设置:
  - 对齐: 定义图案相对于要素的定位方式:
    * 对齐图案到要素: 标记线在每个要素内渲染
    * 对齐图案到地图范围: 图案在整个地图范围内渲染，允许标记在要素之间完美对齐

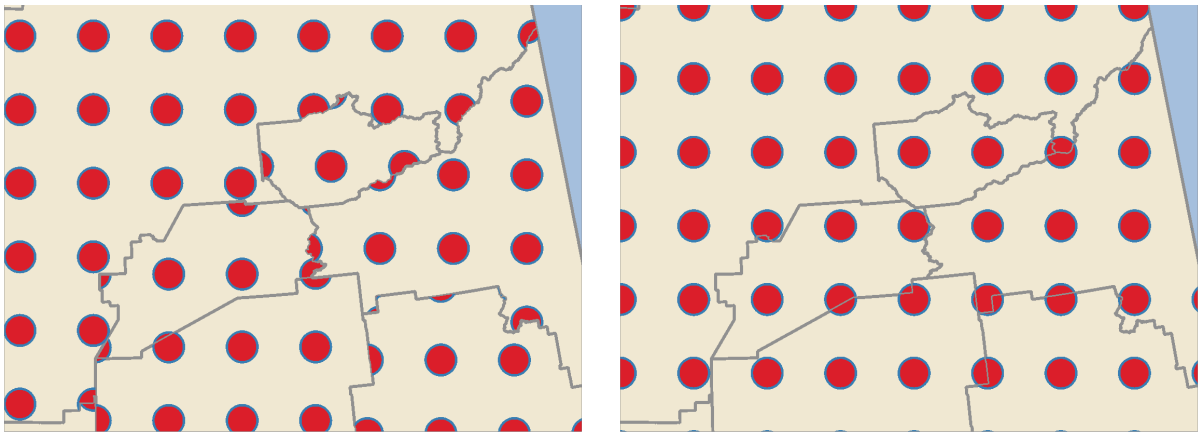


Fig. 10.20: 将点图案与要素 (左) 和地图范围 (右) 对齐

  - 距离: 连续标记之间的 水平和 垂直距离
  - 位移: 一列 (或一行) 中的连续标记之间对齐的 水平 (或 垂直) 偏移
  - 偏移: 距要素边界的 水平和 垂直距离
  - 裁剪: 允许控制如何将填充中的标记裁剪为多边形形状。选项包括:
    * 裁剪到形状: 标记被剪裁，以便仅显示多边形内的部分
    * 形状内标记质心: 只绘制标记中心位于多边形内的标记，但这些标记不会剪裁到多边形外部
    * 标记完全在形状内: 仅显示完全位于多边形内的标记

* 无裁剪: 任何与多边形相交的标记都将被完全渲染 (严格地说，它" 与标记的边界框相交")

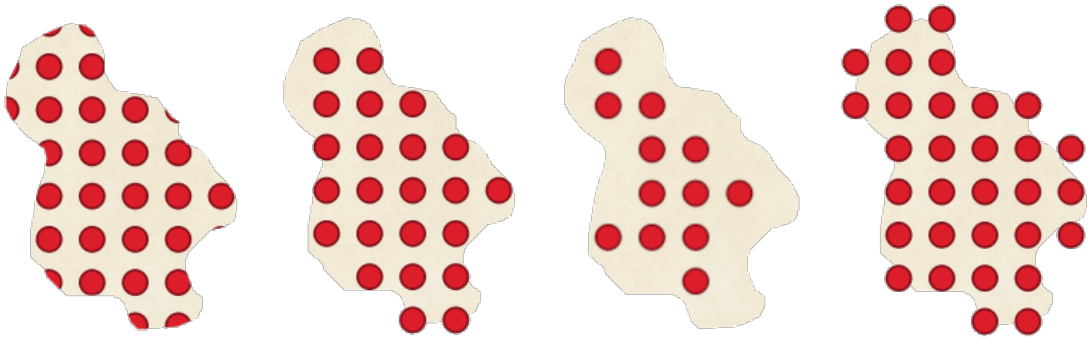


Fig. 10.21: 裁剪填充中的标记 - 从左到右：裁剪到形状、形状内标记质心、标记完全在形状内、无裁剪

  - 整个图案顺时针方向 旋转
  - 随机图案组设置允许点图案填充中的每个点在 水平或 垂直方向随机移动到指定的最大距离。可以以任何支持的单位指定最大偏移，例如毫米、点、地图单位，甚至" 百分比" (其中百分比是相对于图案的宽度或高度)。

    可以设置一个可选的随机数种子，避免符号图案在地图刷新时" 跳跃" 。还支持数据定义的覆写。

    Note

    随机图案和随机标记填充 符号类型之间的主要区别在于，点图案的随机偏移允许标记准" 规则" 放置–因为图案中的点被有效地约束在网格中，这允许创建没有空白区域或重叠标记的半随机填充。(与随机标记填充相反，总是完全随机地放置点…有时会导致点的视觉聚类或不需要空白区域)。

- **随机标记填充**: 使用放置在多边形边界内随机位置的标记符号 填充多边形。可以设置:
  - 计算方法: 要渲染的标记符号数量被视为绝对计数还是基于密度
  - 点计数: 要渲染的标记符号的数量，
  - 可选的随机数 种子，以提供一致的位置
  - 密度面积: 在基于密度的计数方法的情况下，确保每当刷新地图时，标记的填充密度在不同比例/缩放级别的标记保持不变 (还允许随机放置与 QGIS Server 和基于瓦片的渲染配合使用)
  - 将标记裁剪到多边形边界: 在多边形边缘附近渲染的标记是否应剪裁到多边形边界
- **Raster image fill**: fills the polygon with tiles from a raster image (`PNG` `JPG`, `BMP` ...). The image can be a file on the disk, a remote URL or an embedded file encoded as a string (*more details*). Options include (data defined) opacity, size, coordinate mode (object or viewport), rotation and offset. Under the option Size you can independently adjust the width and height of the fill pattern, enabling stretched raster fills in either the horizontal or vertical directions. The image width and height can be set using any of the *common units* or as a percentage of the original size.
- **SVG 填充**: 使用给定大小 (纹理宽度) 的*SVG 标记* 填充多边形。
- **形状突发填充**: 缓冲梯度填充，其中梯度从多边形的边界向多边形的中心绘制。可配置的参数包括从边界到阴影的距离、使用颜色渐变或简单的双色梯度、可选填充和偏移模糊处理。
- **Outline: Arrow**: uses a line *arrow symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The settings for the outline arrow are the same as for arrow line symbols.

- **Outline: Filled line**: uses a *filled line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The settings are the same as for filled line symbols.
- **Outline: Hashed line**: uses a *hash line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings are the same as for hashed line symbols.
- **Outline: Interpolated line**: uses an *interpolated line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings are the same as for interpolated line symbols.
- **Outline: Linear referencing**: uses a *linear referencing symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings are the same as for linear referencing symbols.
- **Outline: Lineburst**: uses a *lineburst symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings are the same as for lineburst symbols.
- **Outline: Marker line**: uses a *marker line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings for the outline marker line are the same as for marker line symbols.
- **Outline: Raster line**: uses a *raster line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings for the outline hashed line are the same as for hashed line symbols.
- **Outline: Simple line**: uses a *simple line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The *Draw line only inside polygon* option displays the polygon borders inside the polygon and can be useful to clearly represent adjacent polygon boundaries. The other settings for the outline simple line are the same as for simple line symbols.

Note

当几何图形类型为多边形时，可以选择禁用线/多边形自动裁剪到画布范围。在某些情况下，这种裁剪会导致不适宜的符号 (例如质心填充，其中质心必须始终是实际要素的质心)。

参数化 SVG

You have the possibility to change the colors of a *SVG marker*. You have to add the placeholders `param(fill)` for fill color, `param(fill-opacity)` for fill opacity, `param(outline)` and `param(outline-opacity)` for stroke color and opacity respectively, and `param(outline-width)` for stroke width. These placeholders can optionally be followed by a default value, e.g.:

```
<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>
```

一般来说，SVG 可以使用 `param(param_name)` 自由参数化。此参数可以用作属性值或节点文本：

```
<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>
```

然后可以将参数定义为 *动态 SVG 参数*表中的表达式。

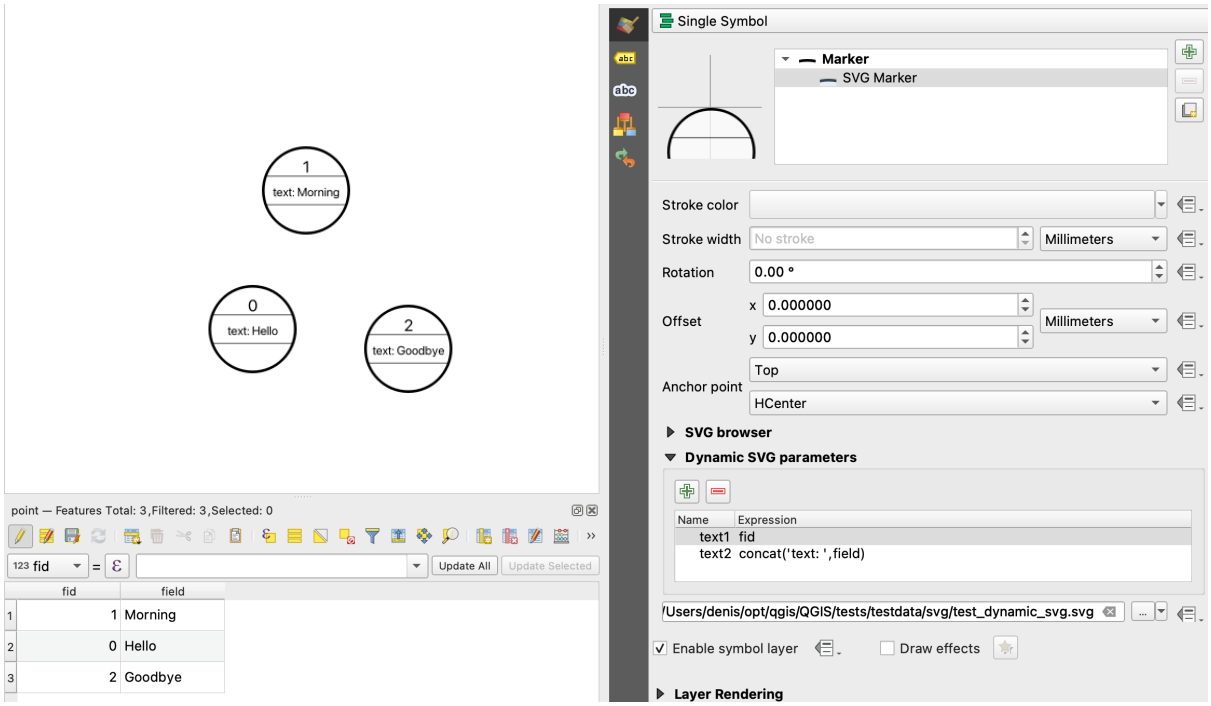


Fig. 10.22: 动态 SVG 参数表

Note

QGIS is looking for a complete SVG node. So if your parameter is within a more complex node, you need to inject the complete node with the expression. For instance, `transform="rotate(param(angle)"` will not work. Instead, you need to do `transform="param(rotation)"` and `rotation` parameter will be defined with the expression `'rotate(' || coalesce(my_field, 0) || ')'`.

几何图形生成器

Available with all types of symbols, the *geometry generator* symbol layer allows to use *expression syntax* to generate a geometry on the fly during the rendering process. The resulting geometry does not have to match with the original *Geometry type* and you can add several differently modified symbol layers on top of each other.

可以设置 单位属性：当几何图形生成器符号未应用于图层时 (例如，用于布局项)，这允许对生成的输出进行更多控制。

一些例子：

- Render symbol as the centroid of a feature

```
centroid( $geometry )
```

- Visually overlap features within a 100 map units distance from a point feature, i.e generate a 100m buffer around the point

```
buffer( $geometry, 100 )
```

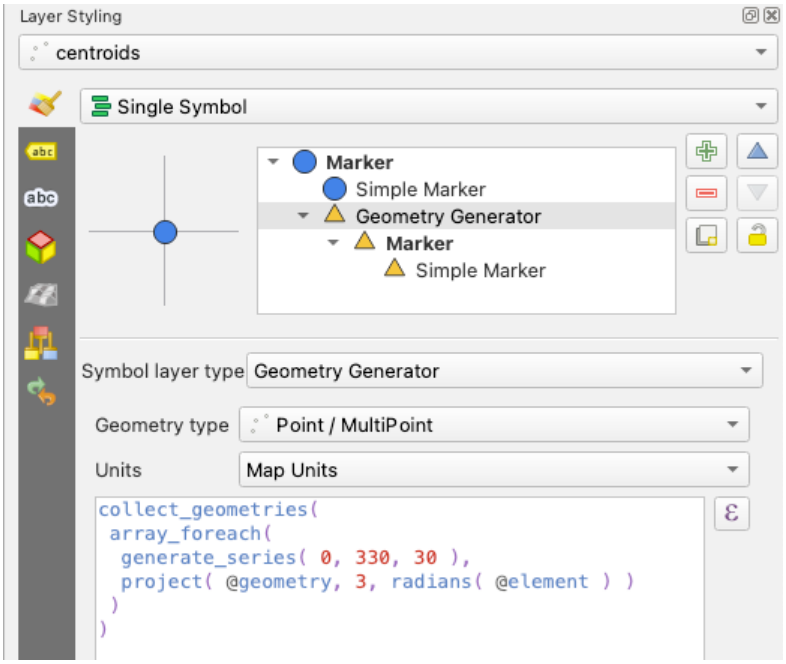
- Create a radial effect of points surrounding the central feature point when used as a MultiPoint geometry generator

```
collect_geometries(
  array_foreach(
```

(continues on next page)

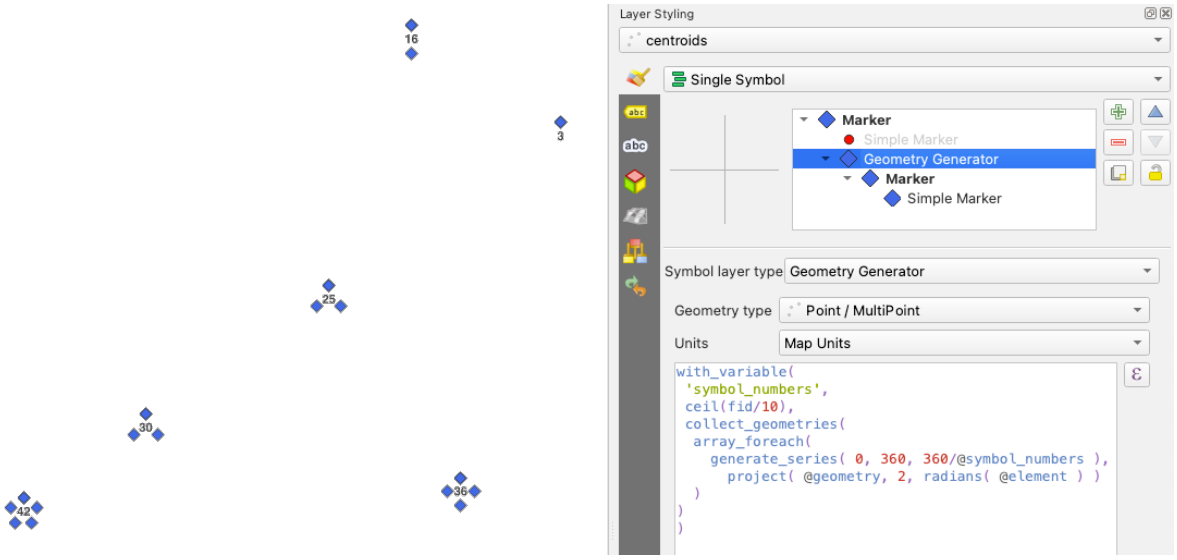
(continued from previous page)

```
    generate_series( 0, 330, 30 ),
    project( $geometry, 3, radians( @element ) )
  )
)
```



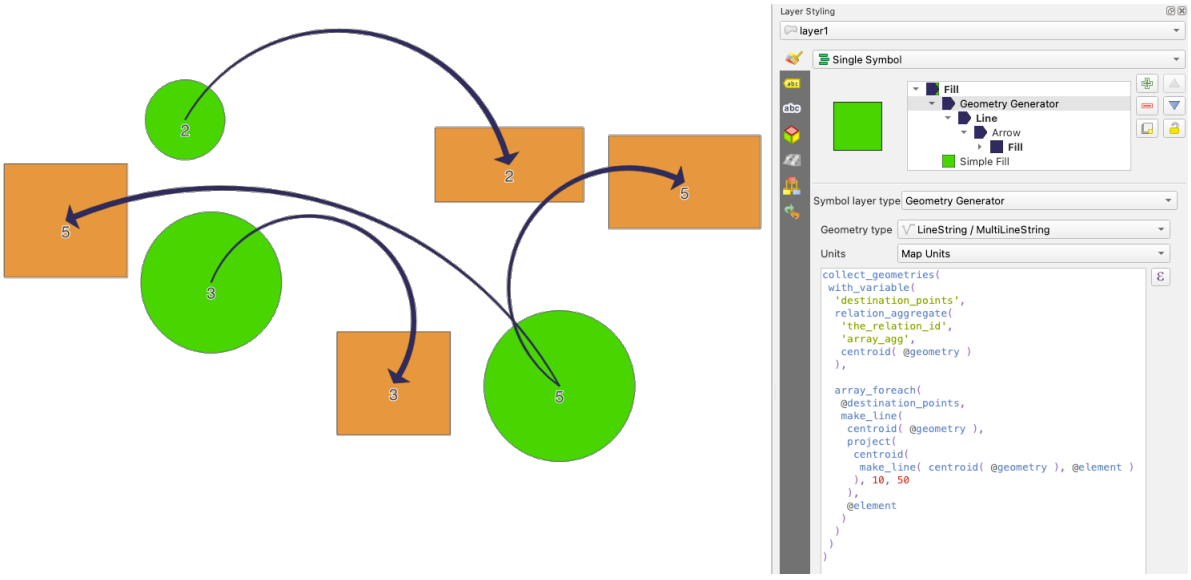
- Create a radial effect of points surrounding the central feature point. The number of points varies based on a field.

```
with_variable(
  'symbol_numbers',
  ceil(fid/10),
  collect_geometries(
    array_foreach(
      generate_series( 0, 360, 360/@symbol_numbers ),
      project( $geometry, 2, radians( @element ) )
    )
  )
)
```



- Create a curved arrow line connecting features of two layers based on their *relation*

```
collect_geometries(
  with_variable(
    'destination_points',
    relation_aggregate( 'the_relation_id', 'array_agg', centroid( $geometry ) ),
    array_foreach(
      @destination_points,
      make_line(
        centroid( @geometry ),
        project(
          centroid( make_line( centroid( @geometry ), @element ) ),
          10,
          50
        ),
        @element
      )
    )
  )
)
```



矢量字段标记

矢量字段标记用于显示矢量字段数据，例如地球变形、潮汐流等。它将矢量显示为线 (最好是箭头)，这些线根据数据点的选定属性进行缩放和定向。只能用于渲染点数据，线和多边形图层不是由该符号绘制的。

矢量字段由数据中的属性定义，这些属性可以通过以下方式表示字段：

- **笛卡尔坐标**分量 (字段的分量 `x` 和 `y`)
- 或 **极**坐标：在此情况下，属性定义 长度和 角度。角度可以从北开始顺时针方向测量，也可以从东逆时针方向测量，并且可以以度或弧度为单位。
- 或作为 **仅高度**数据，显示使用数据的属性缩放的垂直箭头。例如，这适用于显示变形的垂直分量。

字段的大小可以放大或缩小到适合的尺寸以查看字段。

Animated marker

Animated marker symbol type allows you to use a `.GIF`, `.WebP`, `.MNG`, etc. animation file to represent points on your map. You can specify:

- *File* path,
- *Frame rate*: number of steps that are shown per second, indicating how fast the animation is played,
- *Size* in any *supported unit*,
- *Opacity*,
- *Rotation*,
- *Offset* in *x* and *y* directions from the marker position,
- *Anchor point*

There are two ways to handle animated symbols:

- **When your map is not configured as an animation** (i.e. it's a standard QGIS project without animations), the frame for the animated markers will be determined solely by the current timestamp.
- **When your map is** *configured as an animation*, the animated markers will sync with the animation's timeline. This means that animated markers will pause when the animation is paused, progress with the animation, and so forth. The map will also be redrawn according to the frame rate established for temporal animation. This mode is also applied when exporting an animation using the temporal controller.

10.3 设置标注

Labels are textual information you can display on maps. They add details you could not necessarily represent using symbols, and may refer to vector features, raster cells, mesh elements, or simple annotations on the map... Two types of text-related items are available in QGIS:

- 文本格式: 定义文本的外观，包括字体、大小、颜色 、阴影 、背景 、描边 等等

  它们可用于地图文本 (布局/地图标题、装饰、比例尺等) 的渲染，通常是通过字体 控件进行操作。

  要创建 文本格式项，请执行以下操作：

  1. 打开 样式管理器对话框
  2. 激活 文本格式选项卡

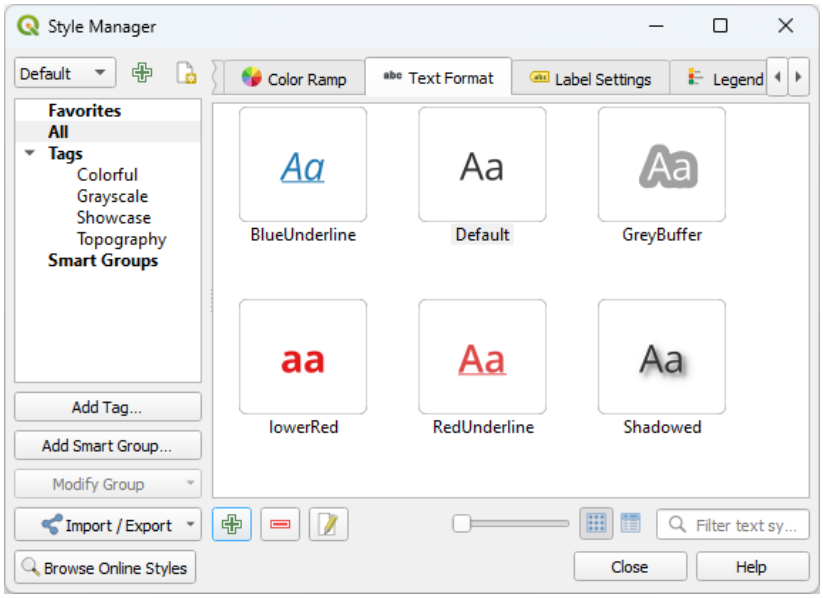


Fig. 10.23: 样式管理器对话框中的文字格式

3. 点击 添加项 按钮。将打开 文本格式对话框进行配置 。通常，这些属性是数据可定义 的。

- *Label Settings*: extend the text format settings with properties related to the location or the interaction with other texts or features (*callouts*, *placement*, *overlay, scale visibility*, *mask* ...).

  They are used to configure smart labelling for vector and mesh layers through the *Labels* tab of the vector or mesh *Layer Properties* dialog or *Layer Styling* panel or using the Layer Labeling Options button of the *Label toolbar*.

  要创建 标注设置项，请执行以下操作：

  1. 打开 样式管理器对话框
  2. 激活 标注设置选项卡

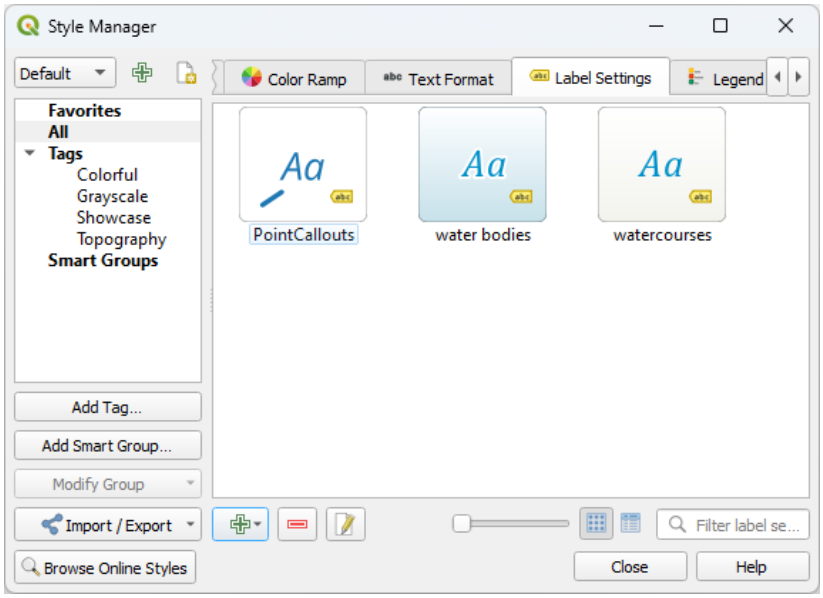


Fig. 10.24: 样式管理器对话框中的标注设置

3. 点击 添加项 菜单，然后选择与要标注的要素的几何图形类型相对应的条目。

此时打开 标注设置对话框，其中包含以下属性。通常，这些属性是数据可定义 的。

10.3.1 格式化标注文本

Whether you are configuring a *Text Format* or *Label Settings* item, and depending on the type of layer you are configuring, you will be given the following options:

Table 10.4: Label properties by data type

属性选项卡	文本格式	标注设置	Vector or mesh layer	raster layer
文本	☑	☑	☑	☑
格式	☑	☑	☑	☑
描边	☑	☑	☑	☑
掩膜		☑	☑	
背景	☑	☑	☑	☑
阴影	☑	☑	☑	☑
牵引线		☑	☑	
位置		☑	☑	☑
渲染		☑	☑	☑

Attention

While for legibility, ”feature” is the name used below to indicate the item being labeled, depending on the underlying layer type, it can be replaced by ”pixel”, ”face” or ”vertex”.

文本选项卡

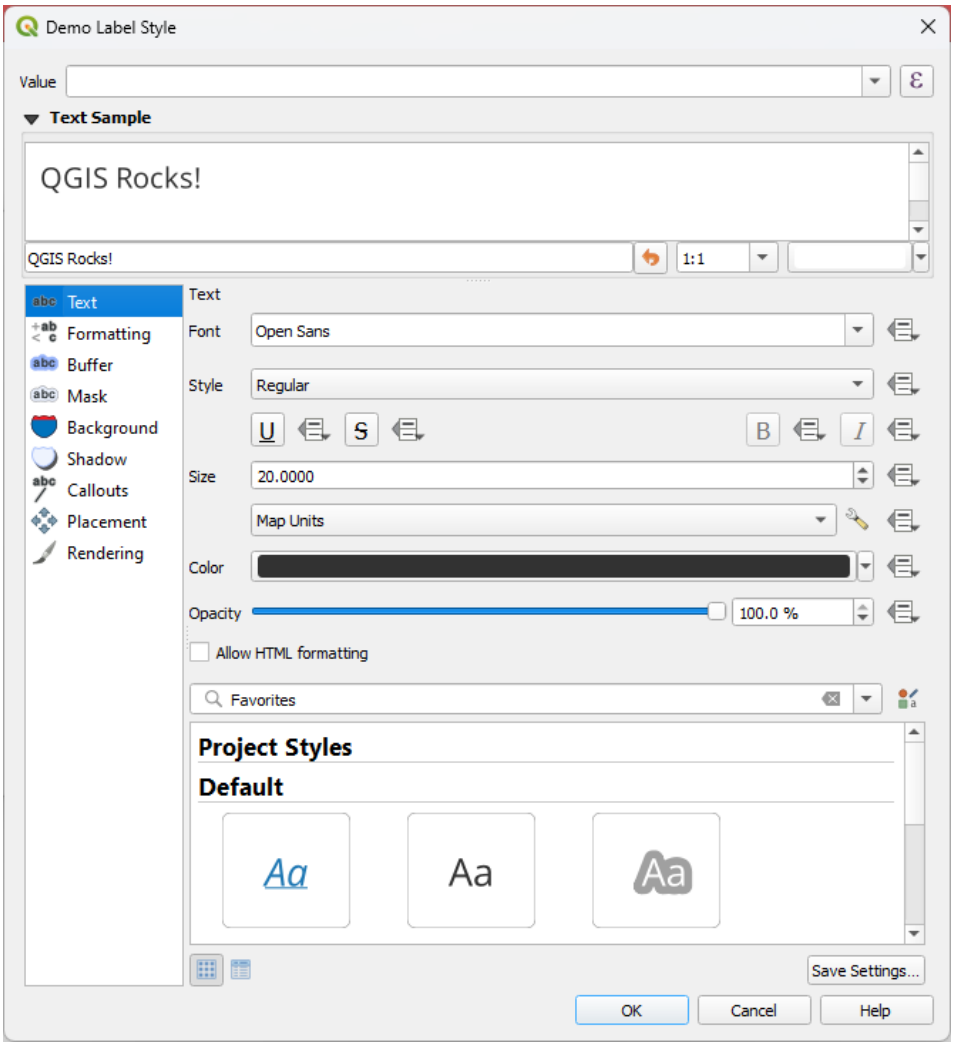


Fig. 10.25: 标注设置 - 文本选项卡

在 abc 文本选项卡，可以设置：

- 字体: 显示操作系统已安装的字体
- 样式: 除了常见的字体样式外，还可以设置文本是否添加下划线或删除线
- *Size*: 所有支持的单位
- 颜色
- 不透明度
- and *Allow HTML Formatting* enables the use of a subset of HTML tags and CSS rules to customize the label.

在选项卡的底部，预览控件显示了存储在样式管理器数据库 中的兼容项的可滤过列表。这允许您根据现有文本格式或标注设置轻松进行配置，还可以将新项保存到样式数据库：点击 保存格式... 或 保存设置... 按钮，并提供名称和标签。

Note

配置 标注设置项时，文本格式项也可在此控件中使用。选择一个可快速盖写当前标注的文本属性 。同样，可以从那里创建/盖写文本格式。

Allow HTML Formatting

With *Allow HTML Formatting* enabled, you need to provide the HTML code in the *Value* field. Use whitespaces instead of tabs for any kind of indentation. The expression is parsed and any supported HTML tag overrides its corresponding setting in the labels properties. Because it is impossible to list and detail every HTML tag and CSS property that QGIS currently supports, we invite you to explore and test in your labels the ones supported by the underlying Qt library.

Examples of supported HTML tags:

- Text formatting, such as italic or bold, e.g.:

```
<i>QGIS</i> <b>rocks!</b>
```

- Superscript and subscript, where the text will be vertically $^{\text{super}}$ or $_{\text{sub}}$ aligned and automatically sized to 2/3 of the parent font size. You can also set a fixed font size for the superscript/subscript by including css rules, e.g.:

```
<sup style="font-size:33pt">my superscript text</sup>
```

- Text horizontal alignment, using either HTML the `align="xxx"` attribute or `center` tag. For HTML5 compatibility, prefer using the CSS property `text-align`.

> **Attention**
>
> Horizontal alignment can not be used for curved labels.

- Structuring a block of text using header tags (such as `h1`, `h2`, ...), or paragraphs (with `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>
```

- Image insertion: any image format readable by QGIS can be used in HTML label content. It can be served from local file paths, HTTP links, or base64 embedded content, using the `src="xxx"` attribute. Image sizes can be specified via the `width="##"` and `height="##"` attributes, in `points` unit. If width or height is not specified it will automatically be calculated from the original image size. Images are placed inline only, not as floating images, and not on curved text labels.

```
<img src="qgis.png" width=40 height=60>
```

Examples of supported CSS properties:

- Font properties (`color`, `font-family`, `font-size`, `font-weight`, `font-style`, `word-spacing`). Note that `word-spacing` will always use unit points.
- Text decorations such as underline, overline and line-through (`text-decoration`)
- Text alignment (`vertical-align`, `text-align`). Horizontal alignment can not be used for curved labels.
- Line height, in `points` or `percent` unit, e.g. ”line-height: 40pt” or ”line-height: 40%”
- Background properties such as `background-color` and `background-image`. They are supported for block type items (e.g. `div`) or inline items (e.g. `span`). For images, the CSS should be formatted as `background-image: url(xx)` and supports local file paths, HTTP links, or base64 embedded content.

> **Attention**
>
> Backgrounds are not supported for curved text and are always rendered above any background shape for the label, and below drop shadows/buffers properties.

- Margin properties, available for block type items only, such as `div`, `p`, `h1`,... They can be specified in either the longhand or shorthand way, in `points` unit only. Negative values can be set for the bottom margin.

```
<div class="myDiv">
  <h2 style="margin-left: 5pt; margin-right: 10pt">QGIS still rocks...</h2>
  <p style="margin: 5pt 0pt -10pt 0pt">Thanks to you!!</p>
</div>
```

CSS properties can be set on HTML tags with the `style` attribute. The HTML tag `span` does not apply any formatting to text by itself and is ideal if you just want to apply CSS styling. A CSS property name and its value are separated by a colon (`:`). Multiple CSS properties are separated by semicolon (`;`), e.g.:

```
<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>
```

下面是基于 HTML 的表达式和渲染的示例 (对同一标注应用不同的颜色和下划线):

```
format(
  '<span style="color:blue">%1</span> ( <span style="color:red"><u>%2 ft</u></span>
↪ )',
  title( lower( "Name" ) ),
  round($length)
)
```

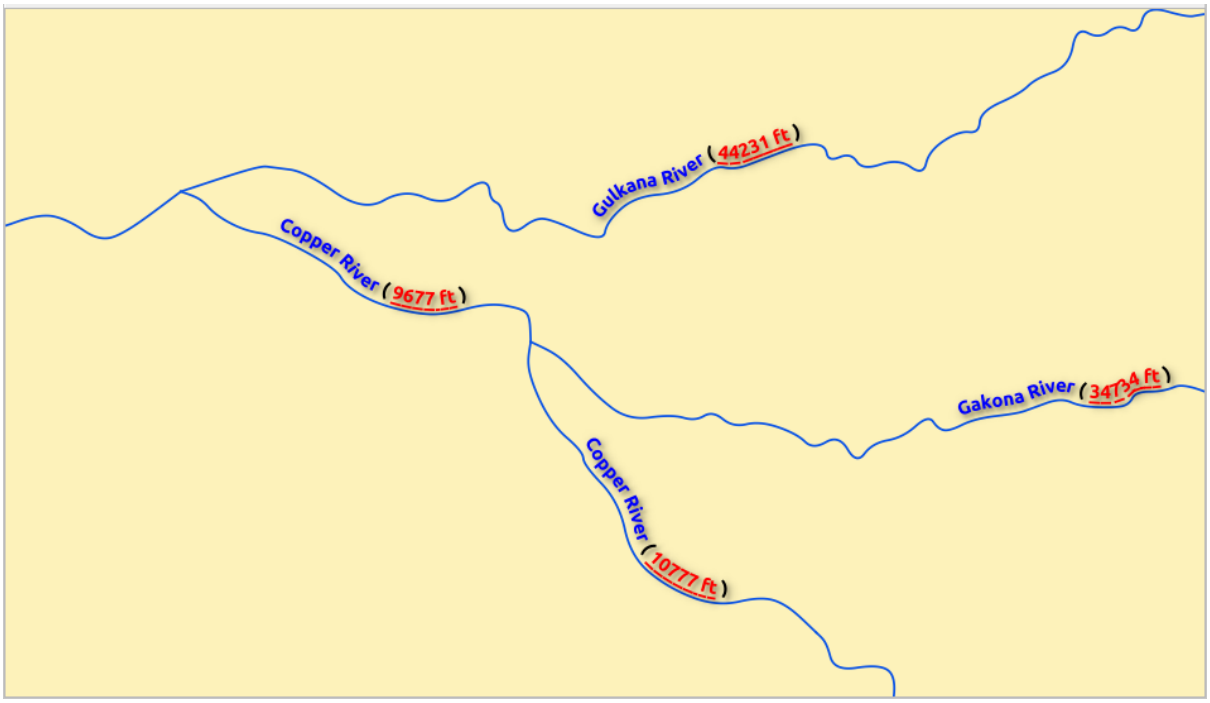


Fig. 10.26: 启用 HTML 格式的标注

格式选项卡

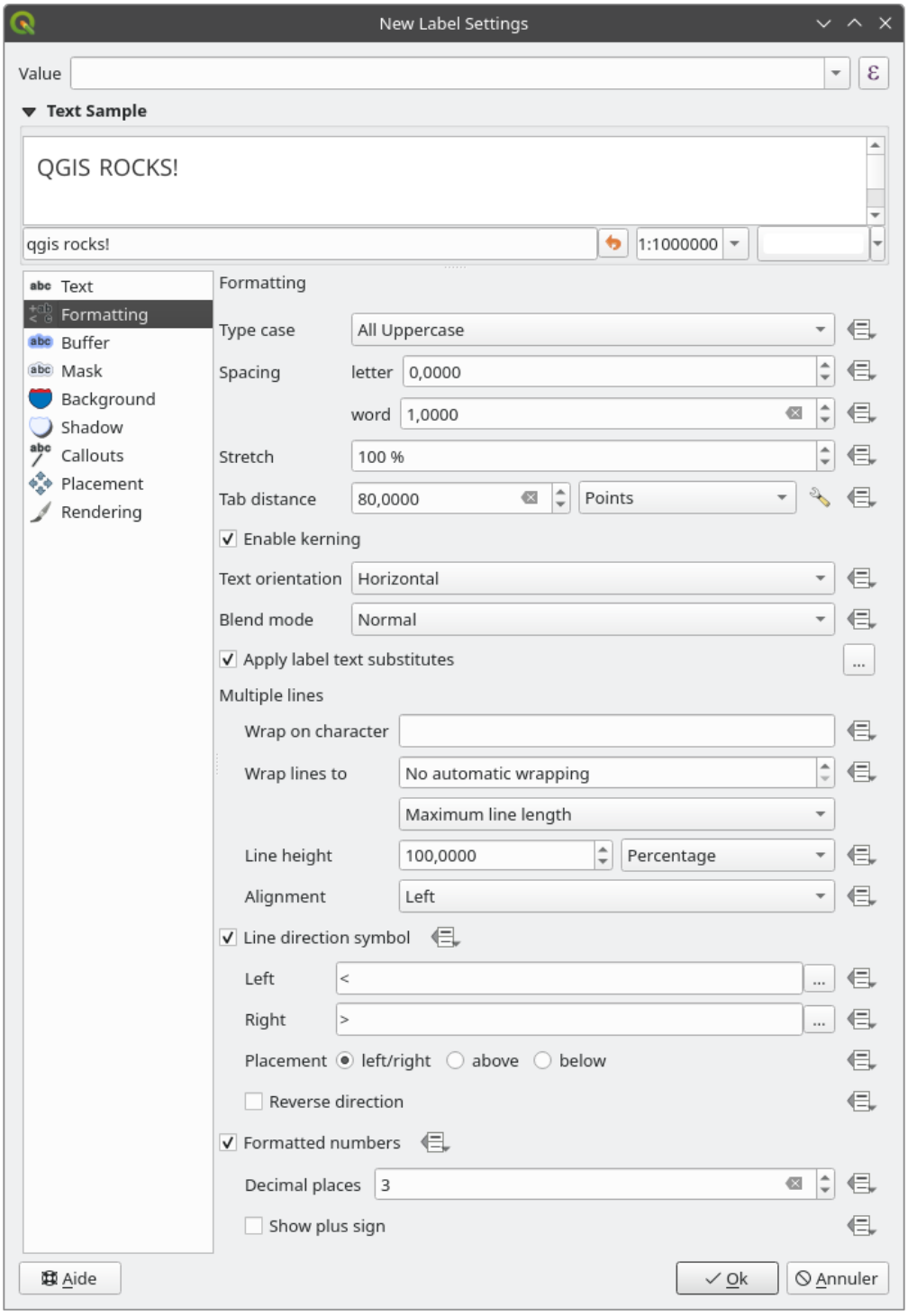


Fig. 10.27: 标注设置 - 格式选项卡

在 格式选项卡，您可以：

- 使用 类型大小写选项可以更改文本的大小写样式。可以将文本渲染为：
  - 未更改
  - 全部大写

  - 全部小写
  - *Small Caps*: renders lowercase characters as small caps.
  - *All Small Caps*: renders all characters as small caps regardless of their original case.
  - 首字母大写: 将每个单词的首字母修改为大写。如果原文使用的是单类型大小写，则将其他字母改为小写；如果文本中是混合类型大小写，则其他字母保持不变。
  - 强制首字母大写: 将每个单词的第一个字母改为大写，并使文本中的其他字母保持不变。
- 在 间距中，更改单词之间和单个字母之间的间距。
- With *Tab distance* you can adjust the replacement spacing for `Tab` characters in the label, in any supported unit. This allows e.g. to properly separate or vertically align elements of a layout legend text.

  Pressing the Configure tab stops button on the right, you can add a list of tab positions, instead of a single distance. This allows e.g. the creation of table-like multiline labels, where texts are split over different "columns" and lined up nicely, based on the `Tab` characters.
- 拉伸比率：允许文本按因子水平拉伸或压缩。方便调整字体宽度，以便在标注中添加一些额外的文本。
- 文本字体的 启用字距调整
- 设置 文本方向，可以是 水平或 垂直方向。设置标注时也可以 基于旋转 (例如，在平行 放置模式下正确标注线要素)。
- 使用 混合模式选项确定标注将如何与下方的地图要素混合 (更多信息请参阅*Blending Modes*) 。
- 应用标注文本替换选项允许指定要替换要素标注中文本的文本列表 (例如，缩写的街道类型)。在地图上显示标注时使用替换文本。用户还可以导出和导入替换列表，便于重复使用和共享。
- 配置 多行:
  - 使用 按该字符换行选项设置一个将强制在文本中换行的字符
  - 使用 换行到选项设置自动换行的理想线大小。大小可以表示 最长线长度或 最短线长度。
  - 决定 行高: 可以将值设置为 毫米、点、像素、百分比或 英寸。当行高设置为百分比时，它是该字体族的默认文本行距的百分比。通常是文本大小的 1.2 到 1.5 倍。
  - 对齐格式: 可用的标准值是 左对齐、右对齐、两端对齐和 水平居中。

    设置点标注属性时，文字对齐方式也可以是 跟随标注位置。在这种情况下，对齐将取决于标注相对于点的最终位置。例如，如果标注放置在点的左侧，标注是右对齐；而如果将其放置在右侧，则为左对齐。

  Note

  基于曲线的标注位置 尚不支持 多行格式设置，这些选项将无效。

- 对于线标注，可以包含 线方向符号以帮助确定线方向，其中的符号用于标示 左或 右。当与 位置选项卡中的 弯曲或 平行位置选项一起使用时，效果特别好。有些选项可以设置符号位置，也可以 反向。
- 使用 格式化数字选项来格式化数字文本。您可以设置 小数位数。默认情况下，将使用小数点后 3 位。如果要为正数显示加号，请勾选 显示加号复选框。

描边选项卡

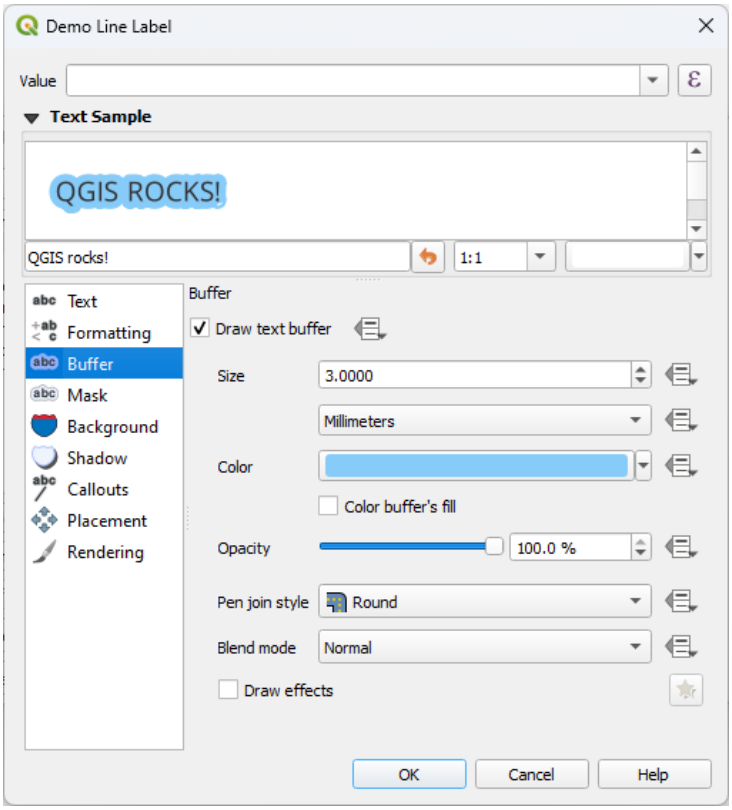


Fig. 10.28: 标注设置 - 描边选项卡

要在标注周围创建描边，请激活 abc 描边选项卡中的 绘制文本描边复选框。然后，您可以：

- 以任何支持单位 设置描边的 大小
- 选择描边的 颜色
- 使用描边颜色填充文本颜色: 描边从标注轮廓展开，因此，如果该选项被激活，标注的内部将被填充。当使用部分透明标注或使用非正常混合模式时，将允许看到标注文本后面的填充。取消勾选该选项 (使用完全透明的标注时) 将允许创建轮廓文本标注。
- 定义描边 不透明度
- 应用 画笔连接样式: 可以是 圆角、尖角或 斜角
- 使用 混合模式选项确定标注的描边与下面地图组件将如何搭配 (更多信息，请参阅*Blending Modes*) 。
- 为提高文本可读性，可勾选 绘制效果以添加高级 绘图效果 ，例如外发光和模糊等。

背景选项卡

背景选项卡允许配置位于每个标注下方的形状。若要添加背景，请激活 绘制背景复选框，然后选择 形状类型。可以是：

- 使用填充符号 的所有属性的规则形状，如 矩形, 正方形, 圆形或 椭圆
- 来自文件、URL、嵌入工程或样式数据库的 *SVG* 符号 (更多信息)
- 或 标记符号，您可以创建或从符号库 选择。

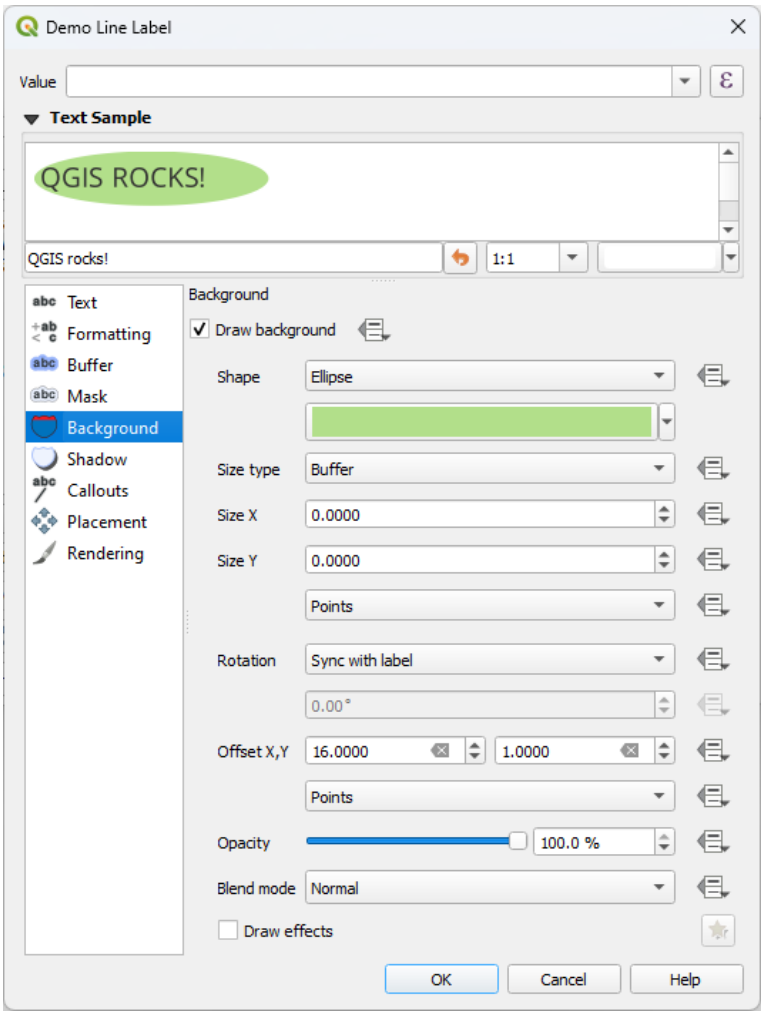


Fig. 10.29: 标注设置 - 背景选项卡

根据所选形状，您需要配置以下某些属性：

- 框架的 大小类型，可以是：
  - 固定: 对所有标注使用相同的尺寸，而不考虑文本的大小
  - 或文本边界框上的 描边
- 框架在 X 和 Y 方向的 大小，可以使用任何支持的单位
- 背景的 旋转角度，可以在 与标注同步、标注偏移和 固定之间选择。最后两个是以度为单位的角度。
- 偏移 *X,Y* 是在 X 和/或 Y 方向移动背景项
- 半径 *X,Y* 用于使背景形状的角转为圆角 (仅适用于矩形和正方形)
- 背景的 不透明度

- 混合模式用于将背景与渲染中的其他项混合 (请参阅*Blending Modes*) 。
- 对于 SVG 符号，可以使用其默认属性 (加载符号参数) 或设置自定义 填充颜色、描边颜色和 描边宽度。
- 绘制效果添加高级 绘图效果 ，可提高文本可读性，例如外发光和模糊等。

阴影选项卡

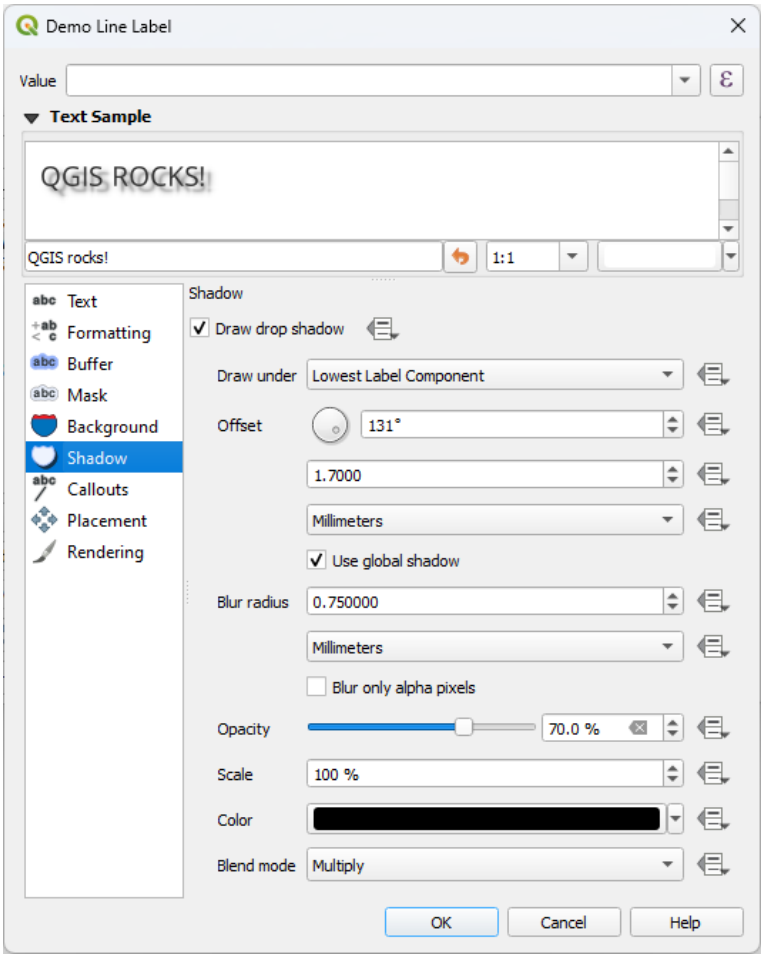


Fig. 10.30: 标注设置 - 阴影选项卡

若要向文本添加阴影，请启用 阴影选项卡，并激活 绘制阴影。然后可以：

- 指出在 绘制于用于生成阴影的项。可以是 最低标注构件或特定构件，如 文本本身、描边或 背景。
- Set the shadow's *Offset* from the item being shadowed, ie:
  - 角度：顺时针方向，取决于基础项的方向
  - The distance of offset from the item being shadowed
  - 偏移的单位

  如果勾选 使用全局阴影复选框，则角度的零点始终指向北，而不是依赖标注项的方向。
- 使用 模糊半径影响阴影的外观。数字越高，阴影越柔和。以您选择的单位为单位。
- 定义阴影的 不透明度
- 使用 比例因子重缩放阴影的大小

- 选择阴影的 颜色
- 使用 混合模式选项可以确定标注的阴影与其下面的地图构件如何混合 (更多信息，请参阅*Blending Modes*) 。

10.3.2 配置与标注的交互

除了上述文本格式设置外，还可以设置标注之间或与要素的交互方式。

掩膜选项卡

The *Mask* tab allows you to define a mask area around the labels. This feature is very useful when you have overlapping symbols and labels with similar colors, and you want to make the labels visible. A label mask prevents specified features from drawing within the boundary set for the mask. For example, you could set a label mask so that a specified layer does not draw within 2mm of the label, but allow features from another layer to still show. Label masks are similar to label buffers in that they allow control of the legibility of labels that cover other features. The label buffer draws on top of any underlying features, while the label mask selectively stops other layers from drawing.

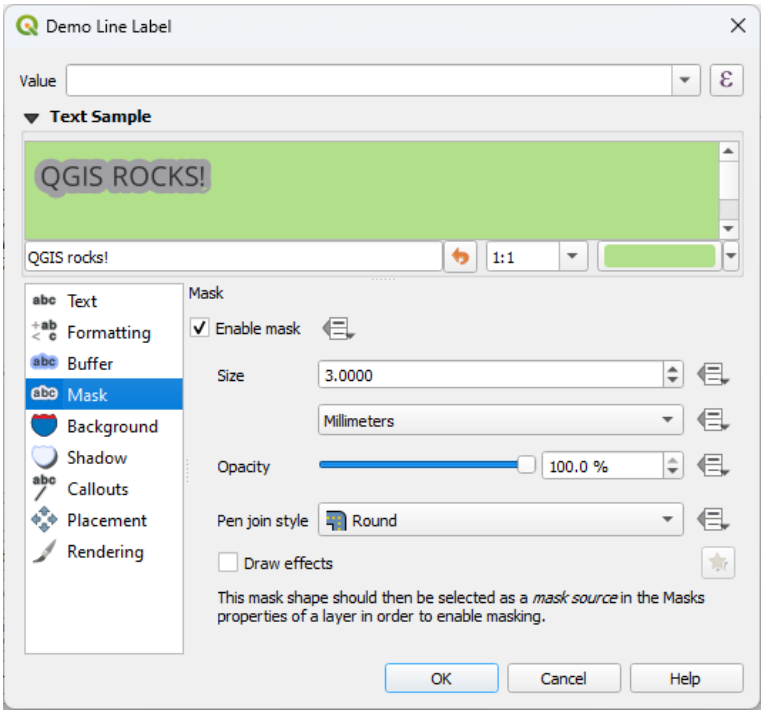


Fig. 10.31: 标注设置 - 掩膜选项卡 (文本示例显示绿色背景，表示被排除的另一个图层)

要在标注上创建掩膜效果，请执行以下操作：

1. 激活 选项卡中的 启用掩膜复选框。
2. 然后可以设置：
   - 以支持的单位 表示掩膜的 大小
   - 标注周围掩膜区域的 不透明度
   - 画笔连接样式
   - 通过 绘制效果复选框实现绘图效果 。
3. 在重叠图层属性 掩膜选项卡中选择此掩膜形状作为掩膜源 (请参阅*Masks Properties*) 。

牵引线选项卡

在拥挤的地图放置标注时常见的做法是使用 **牵引线** - 放置在其关联要素外部 (或从置换) 的标注，是由连接标注和要素的动态线标识组成。如果移动两个端点中的一个 (标注或要素)，则会重新计算连接线的形状。

Fig. 10.32: 各种牵引线设置的标注

若要向标注添加牵引线，请启用 abc 牵引线选项卡，并激活 绘制牵引线。然后您可以：

1. 选择连接的 样式，其中之一是：
   - 简单线: 一条直线，最短的路径
   - 曼哈顿样式: 90° 的虚线
   - 曲线: 一条曲线
   - 泡泡式标注: 围绕标注并指向要素的对话泡泡，可以有圆角。
2. Set the properties of the callout. The following table shows the different properties, with description and compatible connector style.

Table 10.5: Label callout properties

Property	Style of callout	描述
Fill style	Balloons	A *fill symbol* with full display capabilities, including layer effects, data-defined settings, ... for drawing the balloon shape.
Corner radius		Corner radius of the speech bubble
Wedge width		Sets how large the bubble speech connection with feature's pointer should be
Margins		Margins around the label's text, in the unit of your choice
Line style	All but balloons	A *line symbol* with full display capabilities, including layer effects, data-defined settings, ... for drawing the connector line.
Curvature	Curved lines	The percentage of curvature of the connection line

continues on next page

Table 10.5 – continued from previous page

Property	Style of callout	描述
Orientation		Orientation, starting from the label to the feature. It can be *Clockwise*, *Counter-clockwise*, or *Automatic* (determining an optimal orientation for each label).
Minimum length	All but balloons	Minimum length of the connector line
Draw lines to all feature parts		In case of a multi-part feature, indicates whether a connector line should be drawn from the label to each geometry part.
Label anchor point		Controls where the connector line should join to the label text. Available options: • 最近点 • 质心 • 边缘固定位置 (左上角、顶部居中、右上角, 中间靠左、中间靠右、左下角、底部居中和 右下角)。
Offset from label area		Controls the distance from the label anchor point (where the callout line ends). This avoids drawing lines right up against the text.
Offset from feature	All	Controls the distance from the feature (or its anchor point if a polygon) where callout lines end. E.g., this avoids drawing lines right up against the edges of the features.
Feature anchor point		Where the connector line ends on the (polygon) feature. Available options: • 难抵极点 • 外部点 • 表面上的点 • 质心
End point marker	Balloons	A *marker symbol* with full display capabilities including layer effects, data-defined, ... for rendering a marker symbol below the endpoint of the balloon callout.
Blend mode	All	Controls the *blending* of the callout.

在 数据定义位置组中，可以控制牵引线 原点 (标注侧) 和/或 终点 (要素侧) 的坐标。也可以使用标注工具栏 中的 移动标注、图表或牵引线 工具手动控制牵引线。可以通过这种方式移动每条牵引线的起点和终点。当鼠标指针在附近时，节点应高亮显示。如果需要，可以在移动过程中按住 Shift 键。这将以两条牵引线点之间的角度增加 15 度的方式捕捉点。

位置选项卡

Choose the *Placement* tab for configuring label placement and labeling priority. Note that the placement options differ according to the type of vector or mesh layer, namely point, line or polygon, and are affected by the global *PAL setting*.

点图层的位置

可用的点标注位置模式有：

- 制图：按照理想的制图位置规则，生成的点标注与点要素具有良好的视觉关系。标注可以放置在：
  - 从点要素自身或从用于表示要素的符号的边界 (在 距离偏移自 中设置)，使用支持的单位 设置 距离。后一个选项在符号大小不固定时特别有用，例如，如果是由数据定义的大小或在类别 渲染器中使用不同符号时的设置。
  - within a *Maximum Distance* from the feature, which is an optional setting that allows you to control how far a label can be placed from the feature it's labeling. This works alongside the *Distance* setting to create a range for label placement, adding flexibility to position labels more effectively, especially on busy maps, ensuring they fit neatly around their corresponding features.
  - using the *Prioritize Placement* option, which decides what's more important when placing labels. There are two options:
    - *Prefer closer labels*: By default, labels are kept close to the feature.
    - *Prefer position ordering*: The label will try to stay in a specific position (like top left or top right), even if it's a bit farther away from the feature. The label only moves to other positions if there's no room within the maximum distance at your preferred position.
  - following a *Position priority* which dictates placement candidates for anchoring labels around and (centered) over the point feature, and the order in which the positions are tested. The default order, based on guidelines from Krygier and Wood (2011) and other cartographic textbooks, is as follows:
    1. 右上角
    2. 左上角
    3. 右下角
    4. 左下角
    5. 中间靠右
    6. 中间靠左
    7. 顶部, 稍右
    8. 底部, 稍左。

    Using the Data-defined override button, you can provide a comma separated list of placements in order of priority. This also allows only certain placements to be used, for certain features only, so e.g., for coastal features you can prevent labels being placed over the land.
- *Around Point*: labels are placed in a circle around the feature with an equal radius set in *Distance*. Additionally you can set *Maximum Distance* from the feature, to control how far a label can be placed from the feature it's labeling. The placement priority is clockwise from the "top right". The position can be constrained using the data-defined *Quadrant* option.
- 从点偏移：标注可以使用不同单位放置在距点要素的 偏移 *X,Y* 距离处，或者最好放置在要素上方。可以使用数据定义的 象限来约束放置，也可以为标注指定 旋转角度。

线图层的位置

线图层的标注模式包括：

- 平行：绘制与表示要素的广义线平行的标注，优先放置在线的较直部分上。您可以定义：
  - 允许的位置：线上方、线中间、线下方和 位置依赖于线朝向 (将标注放置在线的左侧或右侧)。可以同时选择多个选项，在此情况下，QGIS 将寻找最佳标注位置。
  - 距离：标注和线之间的距离
- 弯曲：沿线要素的曲率绘制标注。除了 平行模式提供的参数外，还可以设置 弯曲字符之间的最大角度 (里面或外面)。
- 水平：沿线要素的长度水平绘制标注。

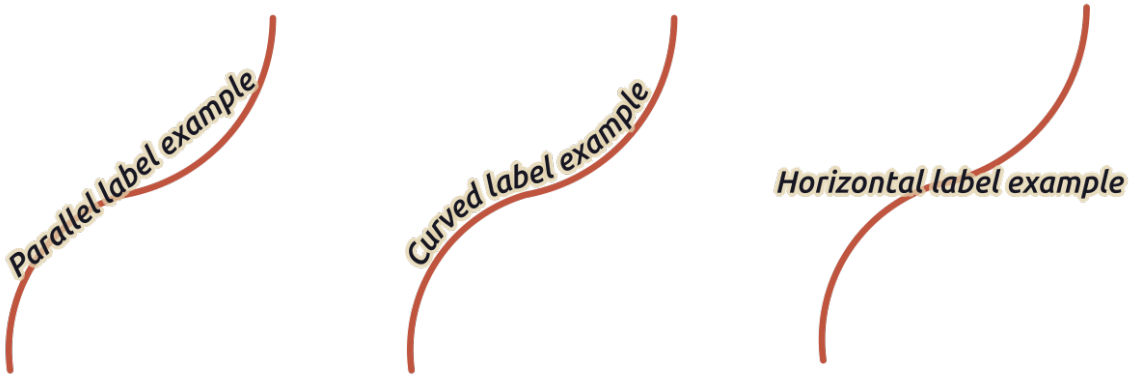


Fig. 10.33: 线的标注放置示例

在放置模式的其他地方，可以设置：

- 重复标注 距离在要素长度上显示重复标注。距离可以是 毫米、点、像素、比例米、地图单位和 英寸。
- 标注溢出 距离 (不适用于水平模式)：指定标注可能经过线要素的终点 (或起点) 的最大允许距离。增加此值可以允许为较短线要素显示标注。
- 标注定位：控制标注沿其所参照的线要素的位置。单击 设置... 并选择：
  - 标注将靠近的沿线位置 (比值表示)。可以是数据定义的，值为：
    - 线中间
    - 线开始端
    - 线末端
    - 或者 自定义... 。
  - 裁剪：确定如何计算标注在线上的位置。默认只使用线的可见范围，但可以使用整个范围以获得更一致的结果。
  - 文本锚点：控制文本的哪一部分 (开头、中心或结尾) 与锚点对齐。使用 自动锚点意味着：
    - 对于定位在线起点附近 (0-25%) 的标注，锚位置将是标注文本的 **开头**
    - 对于定位在线末端附近 (75-100%) 的标注，锚位置将是标注文本的 **结尾**
    - 对于定位在线中心附近 (25-75%) 的标注，锚位置将是标注文本的 **中心**
  - 放置行为：使用 首选位置提示仅将标注锚视为标注位置的提示。通过选择 精确，标注将精准地放置在标注锚上。

多边形图层的位置

可以选择以下模式之一来放置多边形的标注：

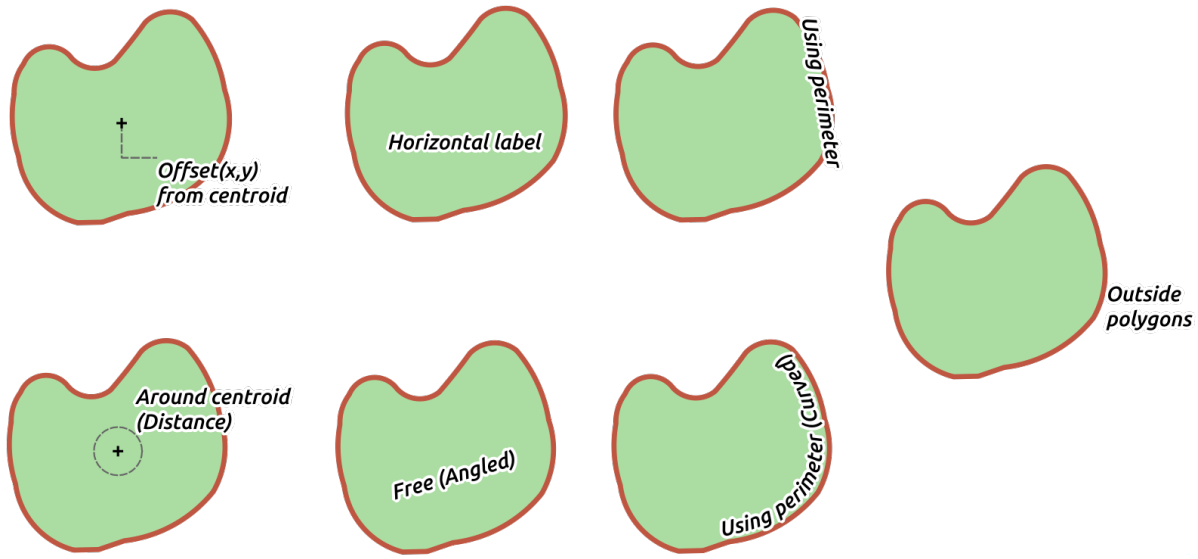


Fig. 10.34: 多边形的标注放置示例

- 从质心偏移: 标注放置在要素质心上或与质心固定的 偏移 *X,Y* 距离 (支持单位) 。无论是否可见，参照质心可以根据地图画布中渲染的部分多边形 (可见多边形) 或 整个多边形来确定。您还可以：
  - 强制质心点位于其多边形内部
  - 将标注放置在指定的象限内
  - 指定旋转角度
  - 当无法将标注放置在多边形内部时，允许在多边形之外放置标注。由于具有数据定义的属性，因此可以允许外部标注、阻止外部标注，或逐个要素强制外部标注。
- 质心周围: 将标注放置在质心周围的预设距离内，首选是直接放置在质心上。同样，可以定义质心是 可见多边形还是 整个多边形，以及是否强制质心点位于多边形内部。
- 水平: 在多边形内部的最佳位置放置水平标注。首选位置是远离多边形的边缘。可以 允许在多边形之外放置标注。
- 自由 (斜置): 将旋转后的标注放置在多边形内的最佳位置。旋转遵循多边形的方向，并且首选位置是远离多边形的边缘。可以 允许在多边形之外放置标注。
- 使用边界线: 绘制与表示多边形边界线的广义线平行的标注，优先选择边界线较直的部分。您可以定义：
  - 允许的位置: 线上方、线中间、线下方和 位置依赖于线朝向 (将标注放置在多边形边界的左侧或右侧)。可以同时选择多个选项，在此情况下，QGIS 将寻找最佳标注位置。
  - 标注和多边形轮廓之间的 距离
  - 重复标注的 距离可在周长范围内多次显示标注。
- 使用边界线 (弯曲): 根据多边形边界的曲率绘制标注。除了 使用边界线模式中可用的参数外，还可以设置 弯曲字符多边形之间的最大角度 (里面或外面)。
- 多边形外侧: 始终以设置的 距离将标注放置在多边形外部。

常规位置设置

某些标注位置的设置可用于所有图层几何图形类型：

Label Spacing

The *Label Spacing* option allows you to set the margin around the label, preventing all other labels from being placed within that area. You can dynamically control *Margin around label* using the Data-defined override button. Choose unit from the unit selector, and there you can also *adjust scaling range*.

Attention

This option is only available on QGIS installed with GEOS >= 3.10 (see *Help* ► *About* menu).

Duplicate Labels

Use *Avoid duplicate labels* to improve label appearance in situations like road networks, where features such as dual carriageways or service roads can result in many identical labels appearing close together. When active, it prevents placing labels with the exact same text (case-sensitive) closer than the minimum distance defined in the *Settings...* option. The setting applies across layers, so duplicate text from any layer will be considered.

Attention

This option is only available on QGIS installed with GEOS >= 3.10 (see *Help* ► *About* menu).

几何图形生成器

The *Geometry Generator* section allows a user to alter the underlying geometry used to place and render the label, by using *expressions*. This can be useful to perform displacement of the geometry dynamically or to convert it to another geometry (type).

要使用几何图形生成器：

1. 勾选 几何图形生成器选项
2. 输入生成要依赖的几何图形的表达式
3. 如果必要的话，请选择表达式输出的几何图形类型：标注基于几何图形的设置 (如位置或渲染) 将更新，以匹配新的几何图形类型的功能。

一些使用案例：

- 使用保存在另一字段”label_position” 中的几何图形
- 使用符号中生成的几何图形 进行标注
- 使用 @map_scale 变量计算与缩放层级无关的距离/大小。
- 结合曲线放置模式，可以围绕点要素创建环形标注:

```
exterior_ring(make_circle($geometry, 20))
```

- 在线要素的起点和终点添加标注:

```
collect_geometries( start_point($geometry), end_point($geometry) )
```

- 依赖平滑的河流线获取更多标注放置的空间:

```
smooth( $geometry, iterations:=30, offset:=0.25, min_length:=10 )
```

由数据定义

数据定义组可在逐个要素的基础上直接控制标注的位置。依赖于其属性或表达式来设置:

- *X* 和 *Y* 坐标
- 以上设置的自定义位置上的文本对齐方式为:
  - 水平: 可以是 **左边**、**中间**或 **右边**
  - 文本 垂直: 可以是 **Bottom** 、**Base** 、**Half** 、**Cap** 或 **Top**
- the text *Rotation*. Rotation is defined as clockwise angle with 0° pointing in the direction of East for `Horizontal` oriented text and with 0° pointing in North direction for `Vertical` oriented text. Different units can be defined for the labeling rotation (e.g. `degrees`, `minutes of arc`, `turns`). QGIS prioritizes screen display rotation angles by default and always rotates text to be most readable on screen, so "upside-down" rotation angles are overridden while rendering. To change this behavior and force QGIS to render exact data-defined rotation angles there is an option in *Rendering* tab, under *Show upside-down labels*. Check the *Preserve data rotation values* entry if you want to keep the rotation value in the associated field and apply it to the label, whether the label is pinned or not. If unchecked, unpinning the label rotation is reset and its value cleared from the attribute table.

  > **Note**
  >
  > 目前仅 质心周围位置模式支持使用多边形要素的数据定义旋转。

> **Note**
>
> 表达式不能与标注地图工具 (即 旋转标注和 移动标注工具) 结合使用来数据定义 标注的位置。控件将被重置到相应的附加存储字段 。

优先级

在 优先级部件，您可以定义每个标注的位置优先级等级，即如果同一位置有不同的候选图表或标注，则会显示优先级较高的项，而其他项可能会被忽略。

优先级等级还用于评估标注是否由于加权障碍要素 较大而被省略。

障碍

在某些情况下 (例如，高密度标注、重叠要素等)，标注的位置可能导致标注被放置在不相关的要素上。

障碍是 QGIS 避免放置其他要素标注或图表的要素。可以通过 障碍部件进行控制:

1. 激活 要素充当障碍选项，以确定图层的要素应充当任何标注和图表 (包括同一图层中其他要素的项) 的障碍。

   您可以使用选项旁边的 数据定义重写 控件来选择要用作障碍的要素子集，而不是整个图层。
2. 使用 设置按钮可以调整障碍的权重。

- 对于每个潜在的障碍要素，您可以指配 *障碍权重*: 任何位置优先级等级高于该值的标注 或图表 都可以放置。如果无法进行其他放置，将忽略级别较低的标注或图表。

  此权重也可以由数据定义，因此在同一图层中，某些要素比其他要素更有可能被覆盖。

- 对于多边形图层，您可以选择要素的障碍类型：
  - **越过要素内部**: 避免在多边形内部放置标注 (最好将标注完全放置在多边形外部或只是略微位于多边形内部)
  - 或 **越过要素边界**: 避免在多边形边界上放置标注 (最好在多边形外部或完全在多边形内部放置标注)。这对于要素覆盖整个区域的图层 (行政单位、分类覆盖范围等) 非常有用。在此情况下，无法避免在这些要素中放置标注，并且避免将标注放置在要素之间的边界上时看起来会更好。

渲染选项卡

在 *渲染* 选项卡，可以调整何时渲染标注，及其与其他标注和要素的交互。

标注选项

在 *标注选项* 下：

- 您可以找到基于比例 和 基于像素大小的可见性设置。
- *标注 Z 索引* 确定了标注的渲染顺序，以及与图层中其他要素标注的关系 (使用数据定义覆写表达式)，就像与其他图层中的标注一样。具有较高 Z 索引的标注渲染在较低 Z 索引的 (来自任何图层) 标注之上。

  此外，还对逻辑进行了调整，如果两个标注具有匹配的 Z 索引，则：
  - 如果来自同一图层，则较小的标注将绘制在较大的标注之上
  - 如果来自不同的图层，标注将按照与图层本身相同的顺序绘制 (即遵循地图图例中设置的顺序)。

  > **Note**
  >
  > 此设置不会使标注绘制在其他图层的要素之下，而只是控制在所有图层的要素之上绘制标注的顺序。

- *允许下级回退位置*: 默认 QGIS 会根据您的设置，尝试以最佳位置渲染标注。如果没有其他选择，请勾选此模式允许要素回退到较次的位置选项，例如，当线太短而无法容纳弯曲的标注文本时，标注可能会水平放置在要素的中心点之上。
- 使用 *显示标注* 和 *始终显示* 中的数据定义表达式，可以微调应渲染的标注。
- Allow to *Show upside-down labels*: alternatives are **never**, **when rotation defined** or **always**.
  - **never** - default setting, screen readability is prioritized,
  - **when rotation defined** - label rotation should be defined under *Placement* tab, within the *Data Defined* group
  - **always** - upside-down labels are allowed

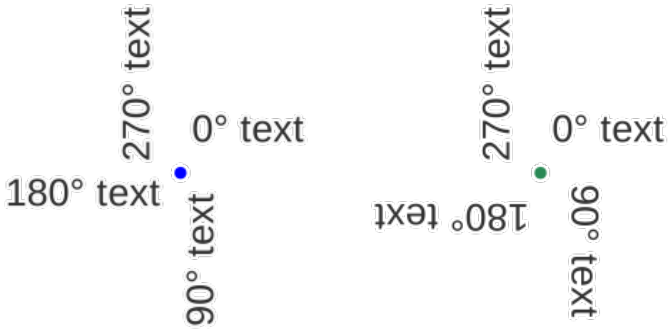


Fig. 10.35: Data defined label rotation with show upside-down labels option set to: ”never” (left), ”when rotation defined” (right)

- 使用 *重叠标注* 组可以控制图层中的要素是否允许重叠标注，以及如何处理这些标注：
  - *绝不重叠*: 绝不为图层放置重叠的标注，即使某些标注将会丢失。
  - *如果需要允许重叠*: 如果无法以其他方式放置标注，那么就绘制重叠标注。如有可能，此模式将导致标注移动到不太理想的位置，例如，标注将远离线或多边形的中心，这样做的目的是避免标注重叠。但是，如果没有其他理想的位置，那么就绘制重叠的标注。
  - *允许重叠而不丢失*: 标注是否与其他标注或障碍重叠并不重要，只要这样做就可以了，并且应该始终使用最佳位置 (例如，最中心的位置)，即使处于较远的位置，从而完全避免重叠。

  允许重叠标注和回退位置选项将确保图层中的所有要素都被标记... 不一定是最佳渲染！

要素选项

在 *要素选项* 下：

- 可以选择 *标注多部件要素的每个部件* 和 *限制要标注的要素数量为*。
- 线和多边形图层都提供了使用 *小于如下尺寸的要素禁止标注* 的选项，目的是为要标注的要素设置最小尺寸。
- 对于多边形要素，还可以根据标注是否完全适合其要素，过滤显示的标注。
- 对于线要素，可以选择 *合并相连的线以防止重复标注*，并结合*位置* 选项卡中的 *距离* 或 *重复* 选项来渲染非常优美的地图。

10.4 创建 3D 符号

样式管理器可帮助您在*3D 地图视图* 中渲染各种几何图形类型来创建和储存 3D 符号。

对于其他项，启用 *3D 符号* 选项卡并展开 按钮菜单以创建：

- *3D 点符号*
- *3D 线符号*
- *3D 多边形符号*

10.4.1 点图层

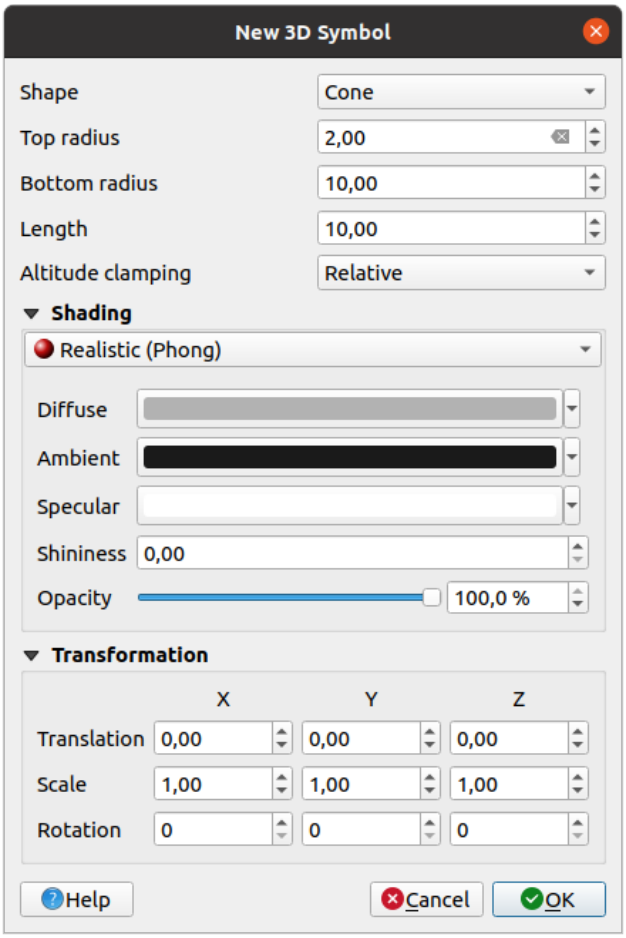


Fig. 10.36: 3D 点符号的属性

- 可以定义不同 3D 类型的 形状用于点符号。主要是由它们的维度来定义，其单位是指工程的 CRS。可用类型包括：
  - 球体由 半径定义
  - 圆柱由 半径和 长度定义
  - 立方形由 大小定义
  - 圆锥由 顶部半径、底部半径和 长度定义
  - 平面由 大小定义
  - 圆环由 半径和 短半径定义
  - *3D Model*, using a 3D model file: supported formats include wavefront `.obj`, `.glTF` and `.fbx`. Models can be a file on disk, a remote URL or *embedded in the project*. Community-created models are shared on the QGIS Hub.
  - *Billboard* 由 *Billboard* 高度和 *Billboard* 符号 (通常基于标记符号) 定义。该符号将具有稳定的尺寸，便于可视化 3D 点云形状。
- 海拔限制可以设置为 绝对、相对或 地形。当 3D 矢量的高度值被提供为从 0 开始的绝对测量时，可以使用 绝对设置。相对和 地形将给定的高程值添加到基础地形高程中。
- 可以定义晕渲 属性。
- 在 变换框架下，可以对符号应用仿射变换：

- 平移可以在 x、y 和 z 轴移动对象。
- 比例可以调整 3D 形状的大小
- 旋转角度围绕 x、y 和 z 轴旋转。

10.4.2 线图层

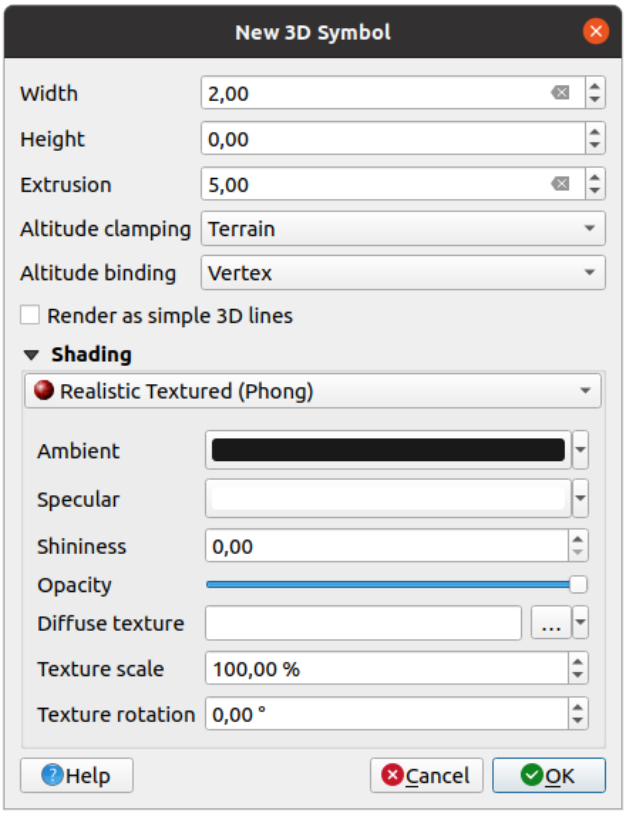


Fig. 10.37: 3D 线符号的属性

- 在 宽度和 高度设置之下，可以定义矢量线的 拉伸。如果线没有 Z 值，可以使用此设置定义 3D 体积。
- 如果已包含栅格高程或其他 3D 矢量数据，可以使用 海拔限制定义 3D 线相对于基础地形表面的位置。
- 海拔约束用于定义如何将要素限制到地形。要素的每个 顶点都将被限制到地形，或者是 质心。
- 可以 渲染为简单 *3D* 线。
- 可以定义晕渲 属性。

10.4.3 多边形图层

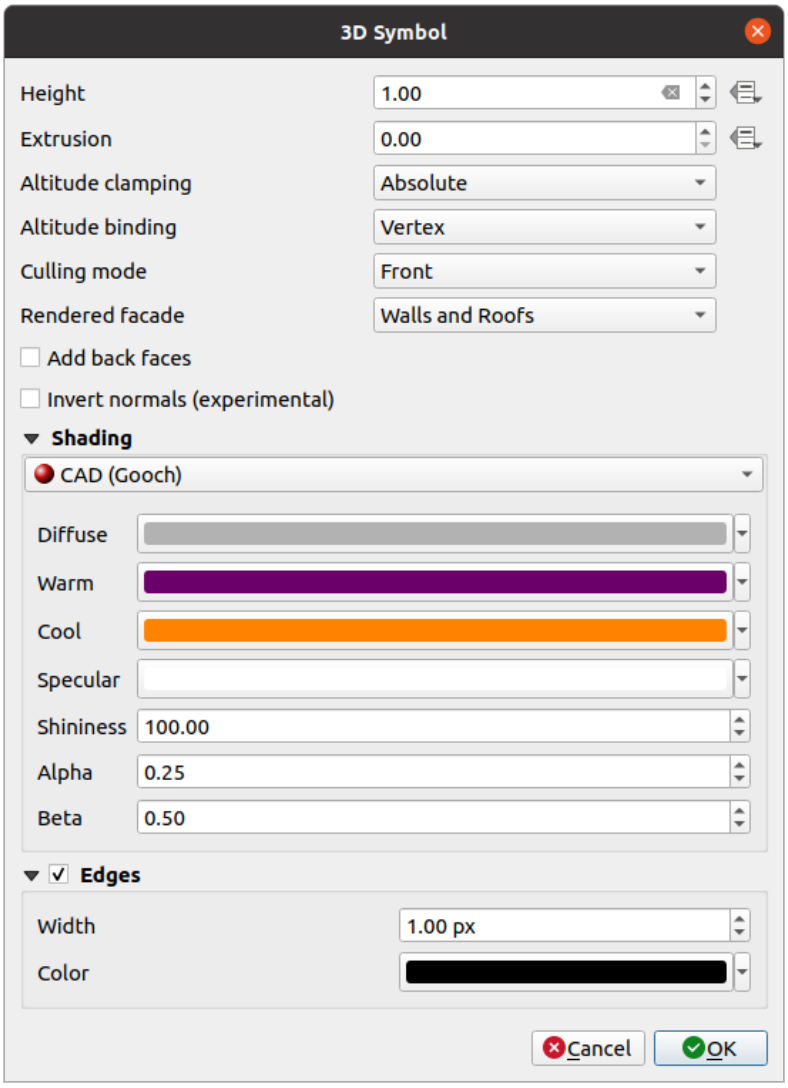


Fig. 10.38: 3D 多边形符号的属性

- 高度使用 CRS 单位进行定义。还可以使用 按钮用自定义表达式、变量或属性表条目盖写值
- 同样，缺少 Z 值时可能会出现 拉伸选项。此外，对于拉伸，您可以点击 按钮来使用矢量图层的值，为每个多边形获得不同的结果：

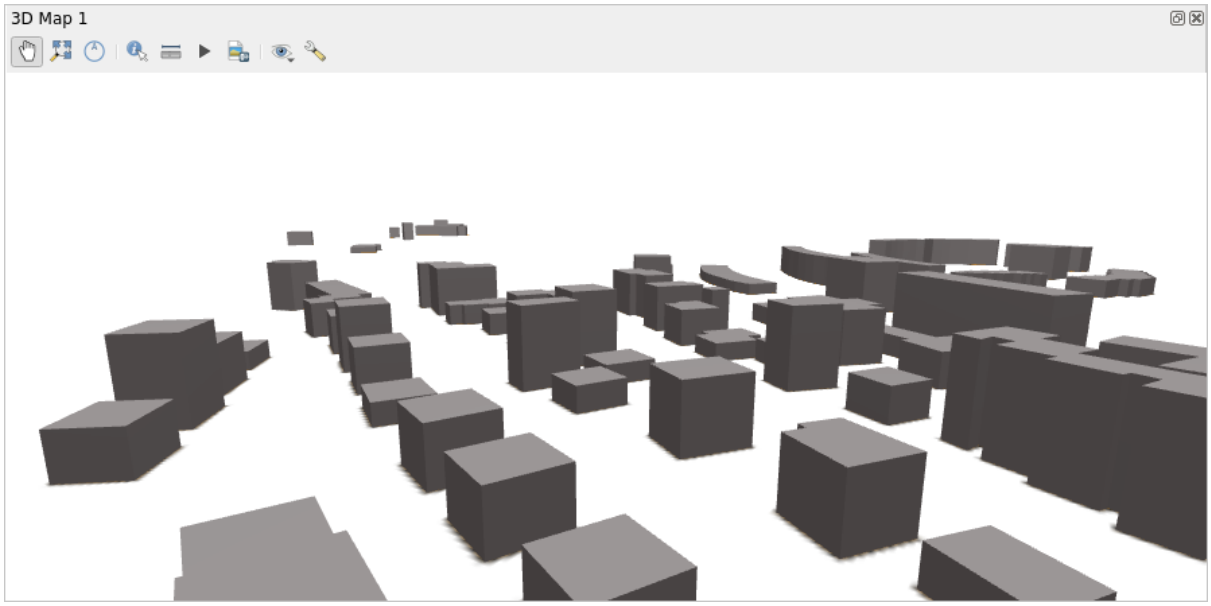


Fig. 10.39: 数据定义拉伸

- 海拔限制和 海拔约束可以如上所述进行定义。
- 应用于符号的 剔除模式，可以是：
  - 未剔除: 有助于避免在 polygonZ/multipatch 数据的顶点排序不一致时 (例如，所有顺时针或逆时针方向)，看似缺失的表面
  - 正面
  - 或 后面
- 渲染外观确定要显示的面。可使用的值是 无立面、立面、顶部或 立面和顶部
- ☑ 添加背面: 对于每个三角形，创建具有正确法线的正面和背面 - 以增加顶点数据数量为代价。此选项可用于解决晕渲问题 (例如，数据顶点顺序不一致)。
- ☑ 反转法线 (实验): 可用于修复顺时针方向/逆时针方向的面的顶点顺序
- 可以定义晕渲 属性。
- 可以启用符号的 ☑ 边缘是否显示，并指定 宽度和 颜色。

Hint

组合设置可实现 3D 数据的最佳渲染效果

剔除模式、添加背面和 反转法线都是为了在 3D 数据看起来不正确的情况下修复其外观。通常，在加载某些数据时，最好先尝试 剔除模式 = 后面和 添加背面 = 禁用 - 这是最有效的设置。如果渲染看起来不正确，请尝试 添加背面 = 启用，并保持 剔除模式 = 未剔除。其他更高级的组合设置，仅在基于输入数据集的混合程度的某些场景中有效。

10.4.4 晕渲纹理

晕渲有助于显示对象的 3D 细节，否则可能会因场景的照明而被隐藏。最终，它是一种更容易使用的材质，因为不需要担心设置是否为恰当的场景照明而影响可视化要素。

QGIS 使用了各种晕渲技术，其可用性取决于符号的几何图形类型：

- *Realistic (Phong)*: describes the way a surface reflects light as a combination of the *Diffuse* reflection of rough surfaces with the *Specular* reflection of shiny surfaces (*Shininess*). It also includes an *Ambient* option to account for the small amount of light that is scattered about the entire scene. Use the *Opacity* slider to render semi-transparent objects in 3D. Read more at Phong reflection description.
- 写实纹理 (材质): 与 写实 (材质) 相同，不同之处在于将图像用作 漫反射纹理。图像可以是磁盘的文件、远程 URL 或嵌入到工程中 。纹理缩放和 纹理旋转角度是必需项。使用 不透明度滑块以 3D 形式渲染半透明对象。
- *CAD (Gooch)*: this technique allows shading to occur only in mid-tones so that edge lines and highlights remain visually prominent. Along with the *Diffuse*, *Specular*, *Shininess* options, you need to provide a *Warm* color (for surface facing toward the light) and a *Cool* color (for the ones facing away). Also, the relative contributions to the cool and warm colors by the diffuse color are controlled by *Alpha* and *Beta* properties respectively. See also Gooch shading.
- *Metal Roughness*: a physically based rendering material that provides an accurate representation of how light interacts with surfaces. Options are available for setting the material *Base color*, *Metalness* and *Roughness*.
- 具有 3D 模型形状的 嵌入式纹理

10.4.5 应用示例

要完成上述设置，您可以查看 https://app.merginmaps.com/projects/saber/luxembourg/tree 。

CHAPTER 11

管理数据源

11.1 打开数据

作为开源软件生态系统的一部分，QGIS 构建在不同的库基础之上，这些库与自己的提供程序相结合，提供读取和编辑多种格式的功能：

- Vector data formats include GeoPackage, GML, GeoJSON, GPX, KML, Comma Separated Values, ESRI formats (Shapefile, Geodatabase...), MapInfo and MicroStation file formats, AutoCAD DWG/DXF, GRASS and many more... Read the complete list of supported vector formats.
- Raster data formats include GeoTIFF, JPEG, ASCII Gridded XYZ, MBTiles, R or Idrisi rasters, GDAL Virtual, SRTM, Sentinel Data, ERDAS IMAGINE, ArcInfo Binary Grid, ArcInfo ASCII Grid, and many more... Read the complete list of supported raster formats.
- Database formats include PostgreSQL, SQLite/SpatiaLite, Oracle, MS SQL Server, SAP HANA, MySQL...
- Web 地图和数据服务 (WM(T)S、WFS、WCS、CSW、XYZ 瓦片、ArcGIS 服务等) 也由 QGIS 提供程序处理。有关内容的更多信息，请参阅 *Working with OGC / ISO protocols* 。
- 您可以从存档文件夹读取支持的文件，并使用 QGIS 本地格式，如 QML 文件 (*QML - QGIS 样式文件格式*) 以及虚拟和存储图层。

More than 80 vector and 140 raster formats are supported by GDAL and QGIS native providers.

> **Note**
>
> 由于各种原因，并非所有列出的格式都可以在 QGIS 中使用。例如，有些需要外部专有库，或者您的操作系统的 GDAL/OGR 安装程序可能无法支持想要使用的格式。要查看可用格式的列表，请在命令行运行 `ogrinfo --formats` (矢量) 和 `gdalinfo --formats` (栅格)，或在 QGIS 菜单 *设置* ► *选项* ► *GDAL* 中查验。

在 QGIS 中，不同的数据格式使用不同的工具打开数据集，主要在 *图层* ► *添加图层* ► 菜单或从 *图层管理工具栏* (通过 *视图* ► *工具栏*菜单激活) 中获取。但是，所有这些工具都指向同一对话框，即 *数据源管理器*对话框，您可以使用 *数据源管理器工具栏*上的 打开数据源管理器 按钮或按 `Ctrl+L` 打开该对话框。数

据源管理器对话框 (Fig. 11.1) 提供了统一的界面来打开基于文件的数据以及 QGIS 支持的数据库或 Web 服务。

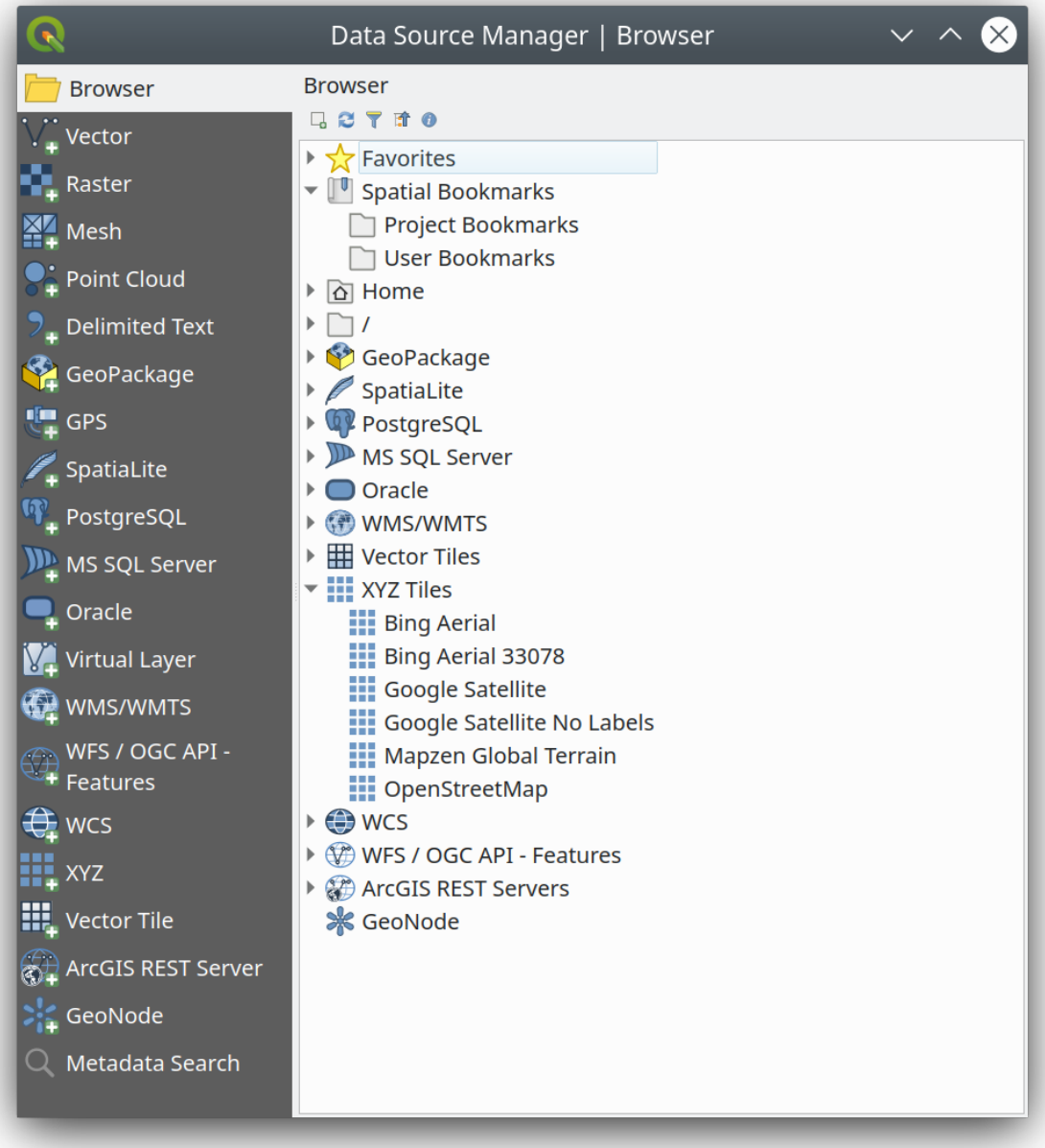


Fig. 11.1: QGIS 数据源管理器对话框

除了这个主要入口点之外，还可以使用 数据库管理器插件，它提供了分析和操作连接数据库的高级功能。有关数据库管理器功能的更多信息，请参阅*DB Manager Plugin* 。

还有许多其他工具 (本地或第三方插件)，可以帮助您打开各种数据格式。

本章仅介绍 QGIS 中默认提供的用于加载数据的工具。主要聚焦 *数据源管理器*对话框，但除了介绍每个选项卡之外，还将探索基于数据提供程序或格式特性的工具。

11.1.1 浏览器面板

浏览器是快速方便地将数据添加到工程中的主要方式之一。它有以下几种途径：

- 数据源管理器选项卡，可按 打开数据源管理器 按钮 (Ctrl+L) 激活；
- 作为 QGIS 面板，可以从菜单 视图 ► 面板 (或 设置 ► 面板) 或按 Ctrl+2 打开。

在这两种情况下，无论图层类型 (栅格、矢量、表) 或数据源格式 (普通或压缩文件、数据库、Web 服务) 如何，浏览器都可以帮助您在文件系统中导航并管理地理数据。

探索界面

在浏览器面板的顶部，可以找到一些按钮来帮助您执行以下操作：

- 添加选中的图层：可以通过从图层的上下文菜单选择 **添加选中的图层**，将数据添加到地图画布；
- 刷新 浏览器树；
- 过滤浏览器 搜索指定数据。输入搜索关键词或通配符，浏览器将过滤树，只显示匹配数据库表、文件名或文件夹的路径 -- 其他数据或文件夹不会显示。请参阅 Fig. 11.2 中的浏览器面板 (2) 示例。可以区分或不区分大小写，也可以设置为：
  - 正常：显示包含搜索文本的项
  - 通配符：使用 ? 和/或 * 字符精确搜索指定搜索文本的位置
  - 正则表达式
- 全部折叠 整个树；
- 启用/禁用属性控件：打开后，将在面板底部添加一个新控件，显示所选项的元数据 (如果可用)。

浏览器面板中的条目是按层级结构组织的，并有几个顶级条目：

1. 收藏夹可以在其中放置指向常用位置的快捷方式
2. 空间书签可以在其中储存常用的地图范围 (请参阅*Bookmarking extents on the map*)
3. 工程主目录：快速访问存储与工程相关 (大部分) 数据的文件夹。默认值是工程文件所在的目录。
4. 文件系统和文件系统根目录中的 主目录。
5. 连接本地或网络驱动器
6. 之后是容器/数据库类型和服务协议，具体取决于您的平台和底层库：
   - 地理包
   - *SpatiaLite*
   - *PostgreSQL*
   - *SAP HANA*
   - *MS SQL* 服务器
   - *Oracle*
   - *WMS/WMTS*
   - 矢量瓦片

- *XYZ 瓦片*
- *WCS*
- *WFS/OGC API-Features*
- *ArcGIS REST Server*

与浏览器项交互

浏览器支持在浏览器中拖放，从浏览器拖放到画布和 图层面板，从 图层面板拖放到浏览器中的图层容器 (例如地理包)。

可以展开浏览器中的工程文件项，显示该工程中包含的完整图层树 (包括组)。工程项的处理方式与浏览器中的任何其他项相同，因此可以在浏览器中拖放 (例如，将图层项复制到地理包文件中)，也可以通过拖放或双击将其添加到当前工程中。

在 浏览器面板中的某个元素上单击鼠标右键，即可打开该元素的上下文菜单。

对于文件系统目录条目，上下文菜单提供以下内容：

- 新建 ► 要在所选条目中创建:
  - 目录...
  - 地理包...
  - 形状文件...
- 添加到收藏夹: 收藏夹文件夹可以随时重命名 (重命名收藏条目...) 或移除 (移除收藏条目) 。
- 从浏览器中隐藏: 隐藏文件夹可以从 设置 ► 选项 ► 数据源 ► 隐藏浏览器路径设置切换为可见
- 快速扫描此目录
- 打开目录
- 在终端打开
- 工程属性...
- 目录属性...

对于可以在工程中充当图层的子条目，上下文菜单将支持条目。例如，对于非数据库、非基于服务的矢量、栅格和网孔数据源：

- 导出图层 ► 到文件...
- 添加图层到工程
- 图层属性
- 打开数据源管理器...
- 管理 ► 重命名"*<name of file>*"... 或 删除"*<name of file>*"...
- 在文件资源管理器中显示
- 文件属性

在 图层属性条目中，您将发现 (类似于将图层添加到工程后在矢量 和栅格 图层属性中发现的内容)：

- 图层的 元数据。元数据组: 来自数据源的信息 (如果可以，路径将是到源的超链接)、识别、范围、访问、字段 (用于矢量图层) 、波段 (用于栅格图层) 、联系人、链接 (用于矢量图层) 、参考 (用于栅格图层) 、历史。
- 预览面板
- 矢量源的属性表 (在 属性面板中)。

Use *Open with Data Source Manager...* to directly open and configure the data source in the *Data Source Manager* using the URI of your data source. This simplifies the process of adding a layer from the *Browser* by allowing you to set specific opening options for the data source. It is currently available for vector (including the dedicated GeoPackage entry), raster, and SpatiaLite data sources.

要使用 浏览器将图层添加到工程中，请执行以下操作：

1. 如上所述启用 浏览器后，将显示包含文件系统、数据库和 Web 服务的浏览器树。在数据库和 Web 服务呈现之前，您可能需要进行连接 (请参阅专用章节)。
2. 在列表中查找图层。
3. 使用上下文菜单、双击其名称，或将其拖放到地图画布 中。您的图层现在已添加到图层面板 ，并且可以在地图画布上查看。

> **Tip**
>
> **直接从浏览器打开 QGIS 工程**
>
> 也可以双击其名称或拖放到地图画布，直接从浏览器面板打开 QGIS 工程。

加载文件后，可以使用地图导航工具对其进行缩放。要更改图层的样式，请双击图层名称或右键单击图例中的名称，然后从上下文菜单选择 属性，打开 图层属性对话框。有关为矢量图层设置符号的详细信息，请参阅*Symbology Properties* 章节。

右键单击浏览器树中的项可以帮助您：

- 对于文件或表，显示其元数据或在您的工程中打开。表甚至可以被重命名、删除或清空。
- 对于文件夹，可以将其标记到您的收藏夹，或从浏览器树中隐藏。隐藏文件夹可以在 设置 ► 选项 ► 数据源选项卡进行管理。
- 管理空间书签: 书签可以作为 XML 文件创建、导出和导入。
- 创建与数据库或 Web 服务的连接。
- 刷新、重命名或删除架构。

还可以通过简单拖放将文件导入数据库，或将表从一个架构/数据库复制到另一个。第二个浏览器面板可用于避免拖动时长时间滚动，只需选择文件并从一个面板拖放到另一个面板即可。

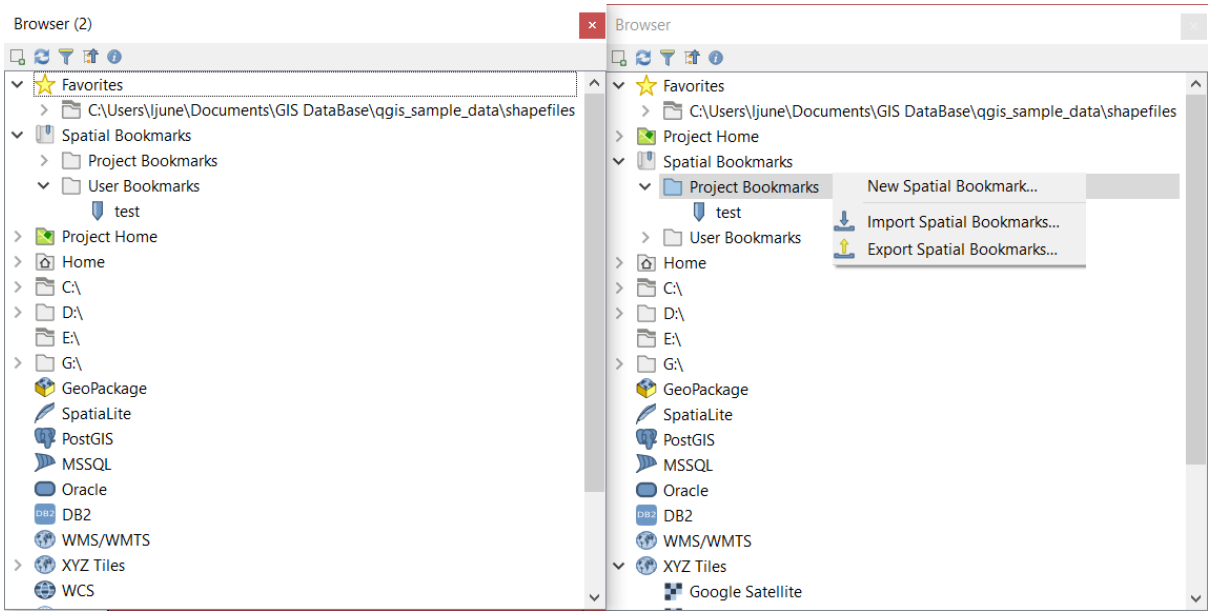


Fig. 11.2: QGIS 浏览器面板并排显示

> **Tip**
>
> **从操作系统文件浏览器中通过简单拖放将图层添加到 QGIS**
>
> 您还可以通过将文件从操作系统文件浏览器拖放到 图层面板或地图画布来将文件添加到工程中。

11.1.2 数据库管理器

The *DB Manager* Plugin is another tool for integrating and managing spatial database formats supported by QGIS (PostgreSQL, SpatiaLite, GeoPackage, Oracle Spatial, MS SQL Server, Virtual layers). It can be activated from the *Plugins* ► *Manage and Install Plugins...* menu.

数据库管理器 插件提供了几个功能：

- 连接到数据库并显示其结构和内容
- 预览数据库表
- 通过双击或拖放将图层添加到地图画布
- 从 QGIS 浏览器或其他数据库向数据库添加图层
- 创建 SQL 查询并将其输出添加到地图画布
- 创建虚拟图层

有关数据库管理器功能的更多信息，请参阅*DB Manager Plugin* 。

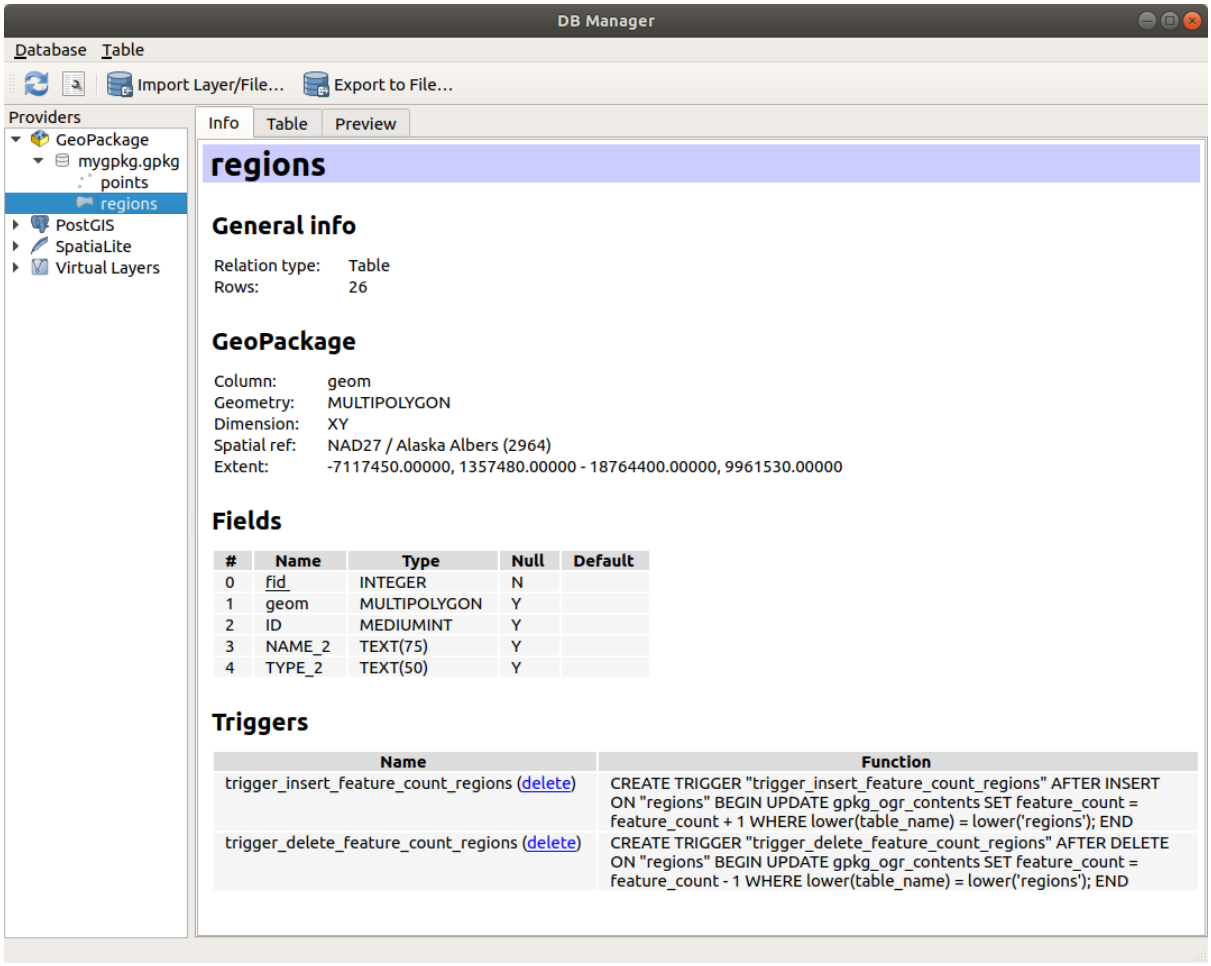


Fig. 11.3: 数据库管理器对话框

11.1.3 基于提供程序加载工具

除了浏览器面板和数据库管理器 (QGIS 提供的用于添加图层的主要工具) 之外，您还可以找到特定于数据提供程序的工具。

> **Note**
>
> 一些外部插件 还提供了在 QGIS 中打开特定文件格式的工具。

从文件加载图层

要从文件加载图层，请执行以下操作：

1. 在 *数据源管理器*对话框中打开图层类型选项卡，即单击 打开数据源管理器 按钮 (或按 `Ctrl+L`) 并启动目标选项卡或：
   - 对于矢量数据 (如 GML、ESRI 形状文件、Mapinfo 和 DXF 图层): 按 `Ctrl+Shift+V`, 或选择 *图层* ► *添加图层* ► 添加矢量图层菜单选项，或单击 添加矢量图层 工具栏按钮。

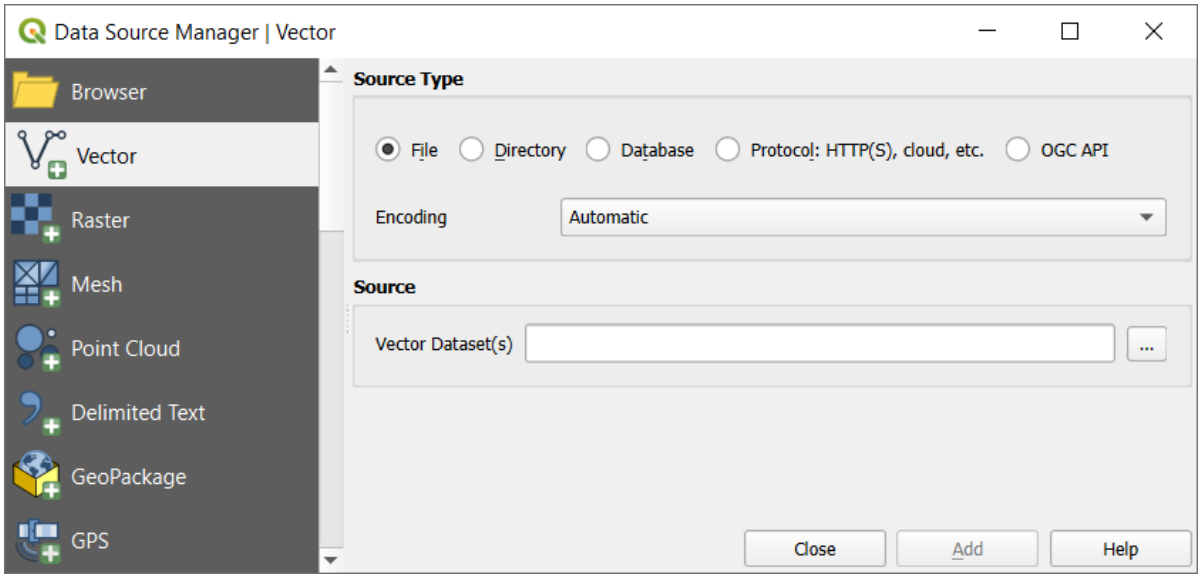


Fig. 11.4: 添加矢量图层对话框

- 对于栅格数据 (如 GeoTiff、MBTiles、GRIded 二进制和 DWG 图层): 按 `Ctrl+Shift+R`, 或选择 图层 ► 添加图层 ► 添加栅格图层菜单选项，或单击 添加栅格图层 工具栏按钮。

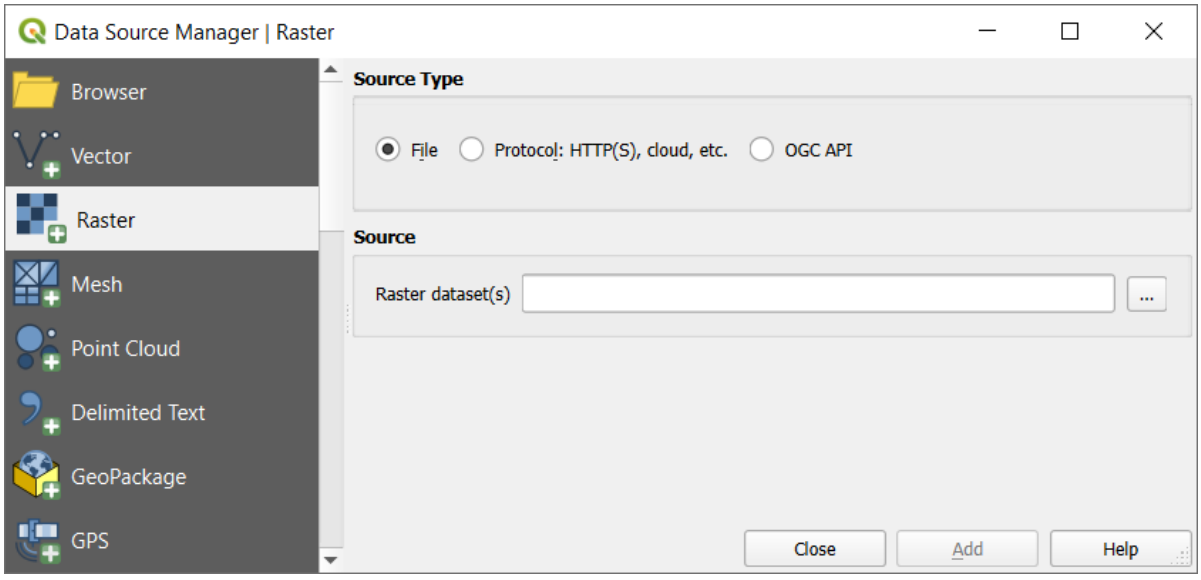


Fig. 11.5: 添加栅格图层对话框

2. 勾选 文件源类型
3. 单击 ... 浏览 按钮
4. 导航到文件系统并加载支持的数据源。按住 `Ctrl` 键并单击对话框中的多个项目，或者按住 `Shift` 键单击范围中的第一个和最后一个项目来选择项目范围，这样可以同时加载多个图层。只有受支持的格式才会显示在格式过滤器中。可以通过选择 所有文件 (下拉菜单中的顶级项) 加载其他格式。
5. Press *Open* to load the selected file into *Data Source Manager* dialog.

   Depending on the selected layer type, additional *Options* (encoding, geometry type, table filtering, file locking, data formatting ...) are available for configuring. These options are described in detail in the specific GDAL vector or raster driver documentation. At the top of the options, a text with hyperlink will directly lead to the documentation of the appopriate driver for the selected file format.

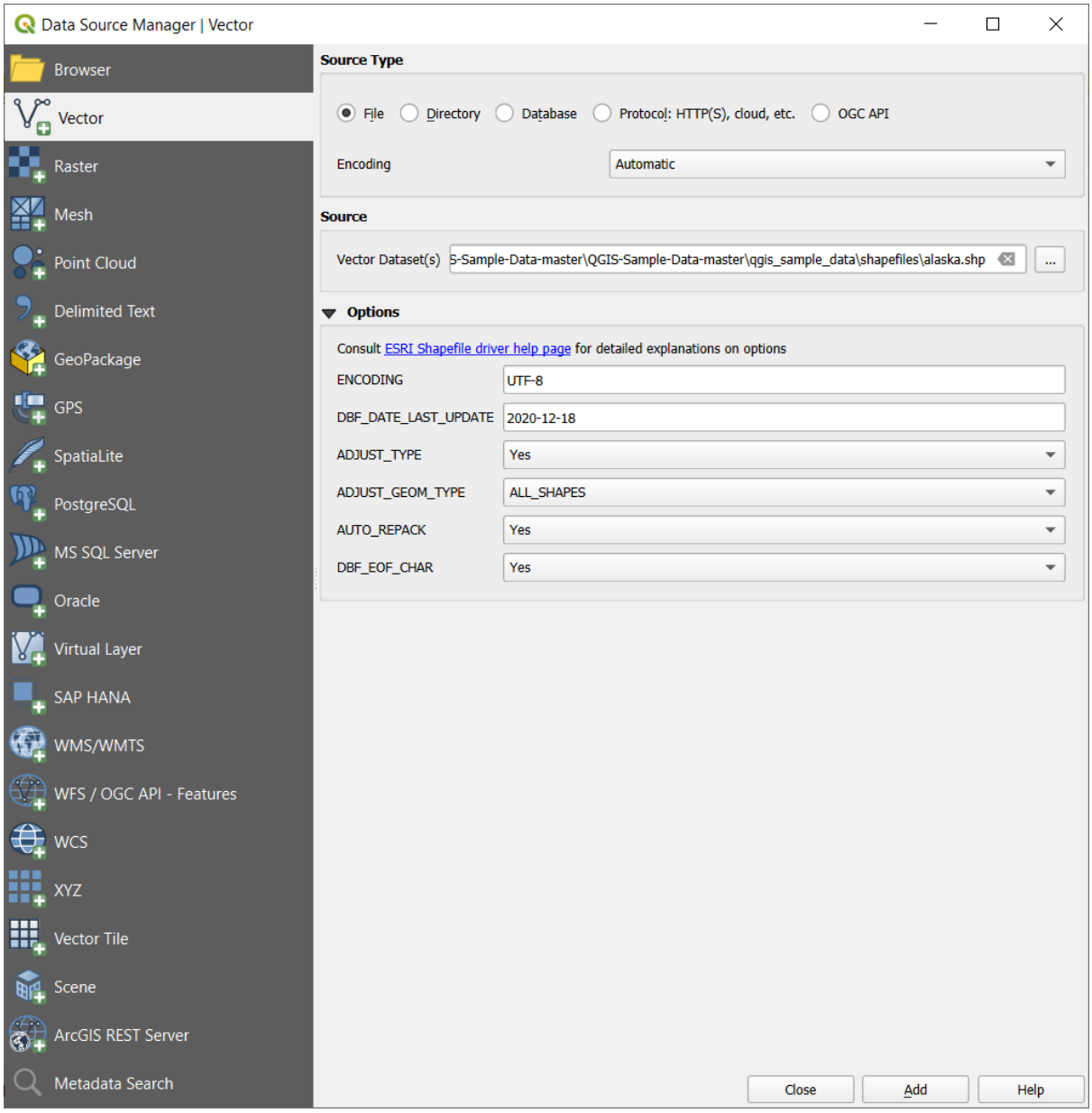


Fig. 11.6: 加载具有打开选项的形状文件

6. Press *Add* to load the file in QGIS and display them in the map view. When adding vector datasets containing multiple layers, the *Select Items to Add* dialog will appear. In this dialog, you can choose the specific layers from your dataset that you want to add. Also, under *Options* you can choose to:

   - *Add layers to a group*
   - *Show system and internal tables*
   - *Show empty vector layers.*

   Fig. 11.7 shows QGIS after loading the `alaska.shp` file.

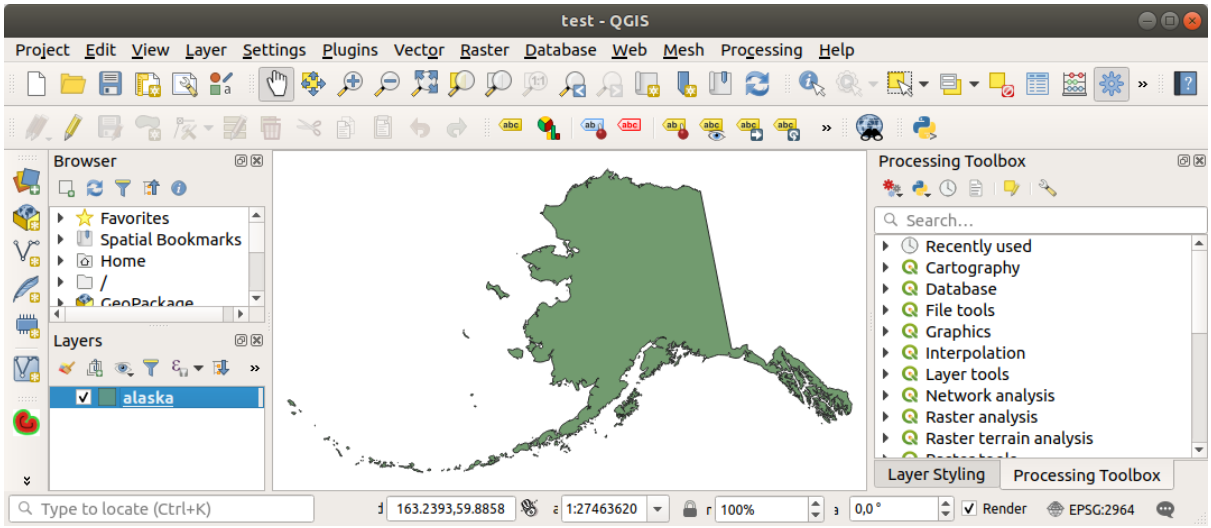


Fig. 11.7: 加载 Alaska 形状文件的 QGIS

Note

由于某些格式如 MapInfo (`.tab`) 或 Autocad (`.dxf`) 允许在单个文件中混合不同类型的几何图形，因此在加载此类数据集会打开一个对话框，选择要使用的几何图形，以便每个图层有一个几何图形类型。

添加矢量图层 和 添加栅格图层 选项卡允许从 文件其他源类型加载图层：

- 您可以加载特定的矢量格式，如 `ArcInfo Binary Coverage`、`UK. National Transfer Format`, 以及 `US Census Bureau` 或 `OpenfileGDB` 的原始 TIGER 格式。要执行此操作，请选择 目录作为 源类型。在这种情况下，点击 ... 浏览 后可以在对话框中选择目录。
- 使用 数据库源类型，可以选择现有的数据库连接或创建一个所选数据库类型的连接。一些支持的数据库类型是 `ODBC`, `Esri 个人地理数据库`, `MS SQL Server` 以及 `PostgreSQL` 或 `MySQL`。

  Pressing the *New* button opens the *Create a New OGR Database Connection* dialog whose parameters are among the ones you can find in 创建存储的连接. Pressing *Open* lets you select from the available tables.

- 协议: *HTTP*、云及其他源类型可以打开存储在本地或网络的数据，可以是公共访问或商业云存储服务的私有存储桶中的数据。支持的协议类型包括：
  - `HTTP/HTTPS/FTP`, 带有 *URI*，如果需要，还可以进行认证。
  - Cloud storage such as `AWS S3`, `Google Cloud Storage`, `Microsoft Azure Blob`, `Microsoft Azure Data Lake Storage`, `Alibaba OSS Cloud`, and `Open Stack Swift Storage` supports direct control over VSI *Credential Options* when adding OGR vector or GDAL raster layers. You need to fill in the *Bucket or container* and the *Object key* first. After that, you can add the necessary *Credential Options*.

    When adding OGR vector or GDAL raster layers from the cloud based protocols, you can also set additional *Credential options* for that specific driver and bucket. When credential options are found in a layer's URI, they will also be automatically set. This allows different layers to use different credentials.
  - 支持 OGC `WFS 3` 服务 (仍处于实验阶段)，使用 `GeoJSON` 或 `GEOJSON - Newline Delimited` 格式或基于 `CouchDB` 数据库。*URI* 必需填写，可选认证。
  - 对于所有矢量源类型，可以定义 编码或使用 自动 ► 设置。

- The *OGC API* source type allows you to access vector and raster data from servers that implement the OGC API standards. To use this option:

  1. Select *OGC API* from the *Data Source Manager* dialog.
  2. Enter the endpoint of the OGC API service you want to connect to. Note that you don't need to prefix the endpoint with "OGCAPI:".
  3. Click *Connect* to establish a connection to the server.

加载网孔图层

网孔是一种具有时态和其他构件的非结构化网格。空间构件包含 2D 或 3D 空间中的顶点、边和面的集合。有关网孔图层的详细信息，请参阅*Working with Mesh Data* 。

要将网孔图层添加到 QGIS，请执行以下操作：

1. 打开 *数据源管理器*对话框，方法是从 *图层* ► 菜单或单击 打开数据源管理器 按钮。
2. 启用左侧面板上的 网孔选项卡
3. 点击 ... 浏览 按钮选择文件。请查看支持的*各种格式* 。
4. 选择文件，然后点击 *添加*按钮。图层添加将使用本地网孔渲染。
5. 如果选中的文件包含多个网孔图层，将显示一个对话框，提示您选择要加载的子图层。进行选择并点击 *确定*按钮，图层将使用本地网孔渲染加载。也可以将其加载到群组中。

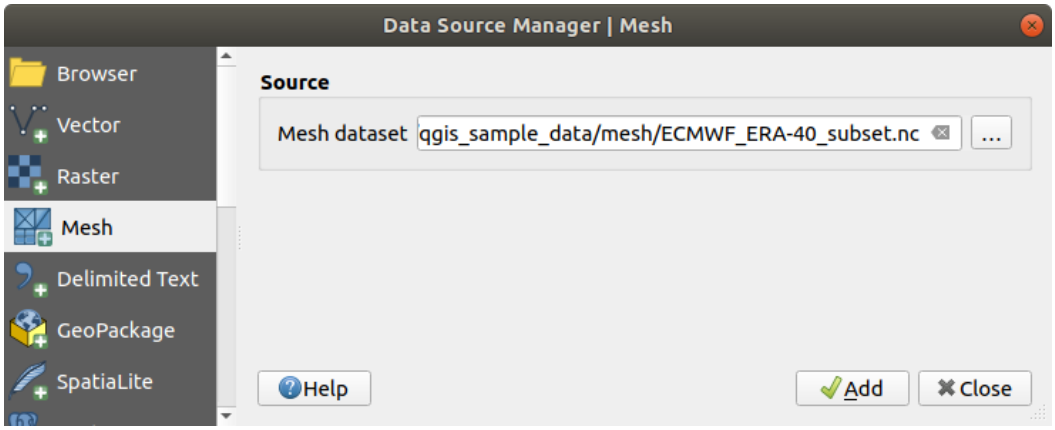


Fig. 11.8: 数据源管理器中的网孔选项卡

导入分隔文本文件

可以使用上述工具加载分隔文本文件 (例如 `.txt`, `.csv`, `.dat`, `.wkt`)。这样，它们将显示为简单的表格。有时，分隔文本文件可能包含您想要可视化的坐标/几何图形。这就是 添加分隔文本图层的设计目的。

1. 单击 打开数据源管理器 图标打开 *数据源管理器*对话框
2. 启用 分隔文本选项卡
3. 单击 ... 浏览 按钮选择要导入的分隔文本文件 (例如 `qgis_sample_data/csv/elevp.csv`) 。
4. 配置设置以满足您的数据集和需求，如下所述。

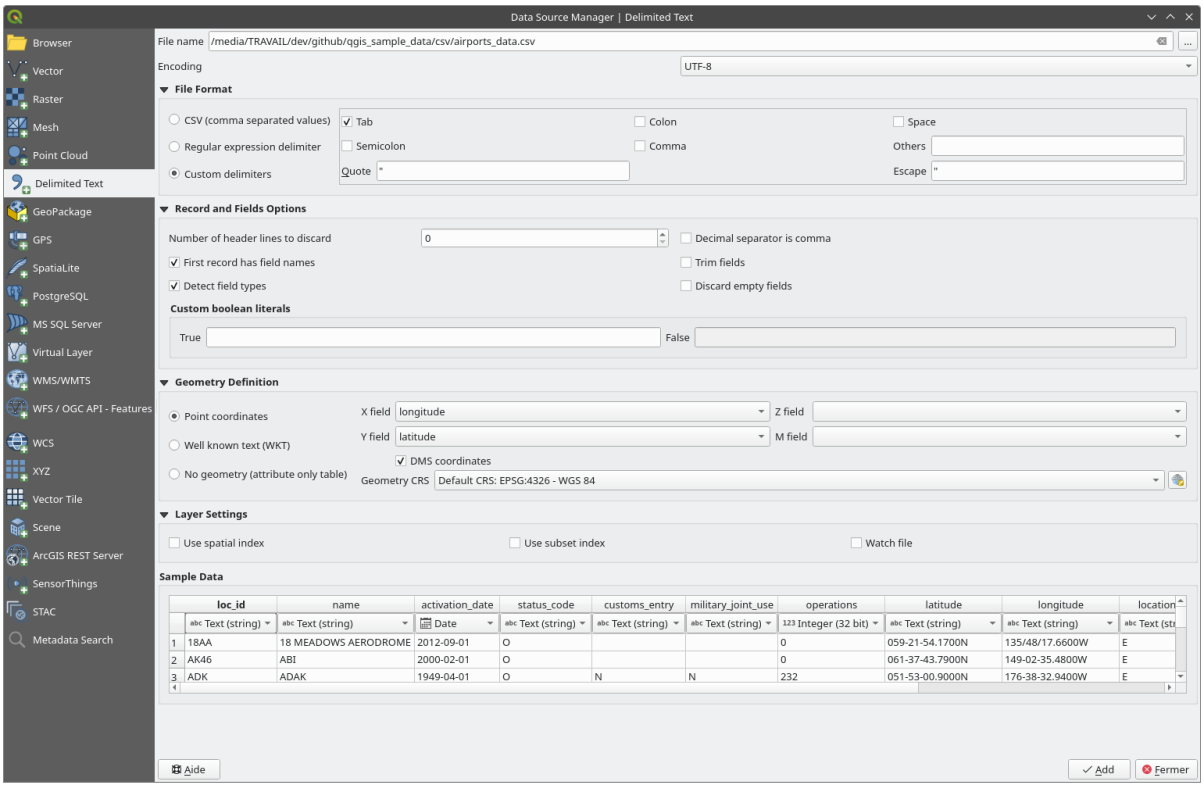


Fig. 11.9: 分隔文本对话框

文件格式

选择文件后，QGIS 就会尝试用最近使用的分隔符解析文件，识别字段和行。为了使 QGIS 能够正确解析文件，选择正确的分隔符非常重要。您可以通过以下选项指定分隔符：

- *CSV (逗号分隔值)* 以使用逗号字符。
- 正则表达式分隔符，在 表达式字段输入文本。例如，要将分隔符更改为制表符，请使用 `\t` (这在正则表达式中用于制表字符)。
- 自定义分隔符，在一些预定义分隔符中进行选择，如 逗号，空格，制表符，分号等。

记录和字段

其他一些合适的选项可用于数据识别：

- 从文件开头起忽略的行数: 当您想要避免导入文件中的第一行时非常方便，因为这些可能是空行或其他格式。
- 首行包含字段名称: 第一行中的值用作字段名称，否则 QGIS 将使用字段名 `field_1`, `field_2` 等。
- 检测字段类型: 自动识别字段类型。如果不勾选，所有属性都将被视为文本字段。
- 小数点为逗号: 可以强制小数分隔符为逗号。
- 裁剪字段前后的空格: 允许从字段中裁剪前导空格和尾随空格。
- 忽略空字段。

- 自定义布尔常量: 允许添加一对将被检测为布尔值的自定义字符串。

字段类型检测

QGIS tries to detect the field types automatically (unless *Detect field types* is not checked) by examining the content of an optional sidecar CSVT file (see GeoCSV specification) and by scanning the whole file to make sure that all values can actually be converted without errors, the fall-back field type is text.

检测到的字段类型显示在数据样例预览表中的字段名称下，如有必要可以手动更改。

支持以下字段类型：

- 布尔型被解析为布尔值的不区分大小写的文字对是 `1/0`, `true/false`, `t/f`, `yes/no`
- 非负整数 （整数）
- 非负整数 （整数 - 64 位）
- 小数: 双精度浮点数
- 日期
- 时间
- 日期和时间
- 文本

几何图形定义

解析文件后，将 *几何图形定义* 设置为

- 点坐标图层为点几何图形类型并包含此类字段: *X* 字段, *Y* 字段, *Z* 字段 (3D 数据) 和 *M* 字段 (测量尺寸)。如果坐标定义为度/分/秒，则激活 *度分秒格式坐标* 复选框。使用 选择 CRS 控件选择恰当的 *几何图形 CRS* 。
- 如果空间信息表示为 WKT，则 *熟知文本 (WKT)* 选项是：选择包含 WKT 几何图形的 *几何图形字段*，选择适当的 *几何图形类型* 或让 QGIS 自动检测。使用 选择 CRS 控件选择恰当的 *几何图形 CRS* 。
- 如果文件包含非空间数据，请激活 *无几何图形 (仅属性表)* ，将作为普通表加载。

图层设置

此外，您还可以启用：

- 使用空间索引可以提高显示和空间选择要素的性能。
- 使用子集索引可以提高子集筛选 的性能 (在图层属性中定义时)。
- 追踪文件在 QGIS 运行时追踪其他应用程序对文件的更改。

最后，单击 *添加* 将图层添加到地图。在我们的示例中，名称为 `Elevation` 的点图层被添加到工程中，其行为与 QGIS 中的任何其他地图图层类似。该图层是对 `.csv` 源文件 (因此链接到该文件) 进行查询的结果，需要保存 才能在磁盘上获得空间图层。

示例数据

在设置解析属性时，示例数据预览会根据应用的设置进行更新。

此外，在示例数据表中，可以覆盖自动确定的列类型。

导入 DXF 或 DWG 文件

DXF 和 DWG 文件可以通过浏览器面板的简单拖放添加到 QGIS 中。系统将提示您选择要添加到工程中的子图层。添加的图层具有随机样式属性。

> **Note**
>
> 对于包含多种几何图形类型 (点、线和/或多边形) 的 DXF 文件，图层名称将生成为 *<filename.dxf> entities <geometry type>* 。

为了在 QGIS 中保留 dxf/dwg 文件结构及其符号，您可能需要使用专用的 工程 ► 导入/导出 ► 从 *DWG/DXF* 文件导入图层... 工具，并允许：

1. 将元素从图形文件导入到地理包数据库。
2. 将导入的元素添加到工程中。

在 *DWG/DXF* 导入对话框，要导入图形文件内容，请执行以下操作：

1. Input the location of the *Source drawing*, i.e. the DWG/DXF drawing file to import.
2. 指定图形文件中数据的坐标参照系。
3. Input the location of the *Target package*, i.e. the GeoPackage file that will store the data. If an existing file is provided, then it will be overwritten.
4. Choose how to import `blocks` with the dedicated combobox:
   - *Expand Block Geometries*: imports the blocks in the drawing file as normal elements.
   - *Expand Block Geometries and Add Insert Points*: imports the blocks in the drawing file as normal elements and adds the insertion point as a point layer.
   - *Add Only Insert Points*: adds the blocks insertion point as a point layer.
5. 选中 使用曲线将导入的图层提升为 曲线几何图形类型。
6. Use the *Import* button to import the drawing into the destination GeoPackage file. The GeoPackage database will be automatically populated with the drawing file content. Depending on the size of the file, this can take some time.

将 `.dwg` 或 `.dxf` 数据导入地理包数据库后，对话框下半部分的框架将填充导入文件的图层列表。在此，可以选择要添加到 QGIS 工程的图层：

1. At the top, set a *Group name* to group the drawing files in the project. By default this is set to the filename of the source drawing file.
2. 检查要显示的图层：每个选中图层都添加到特设组中，该组包含图形图层的点、线、标注和面积要素的矢量图层。图层的样式类似于它们最初在 *CAD 中的外观。
3. 选择图层是否应在打开时可见。
4. 选中 合并图层选项可将所有图层放置在单个组中。
5. 点击 确定在 QGIS 中打开图层。

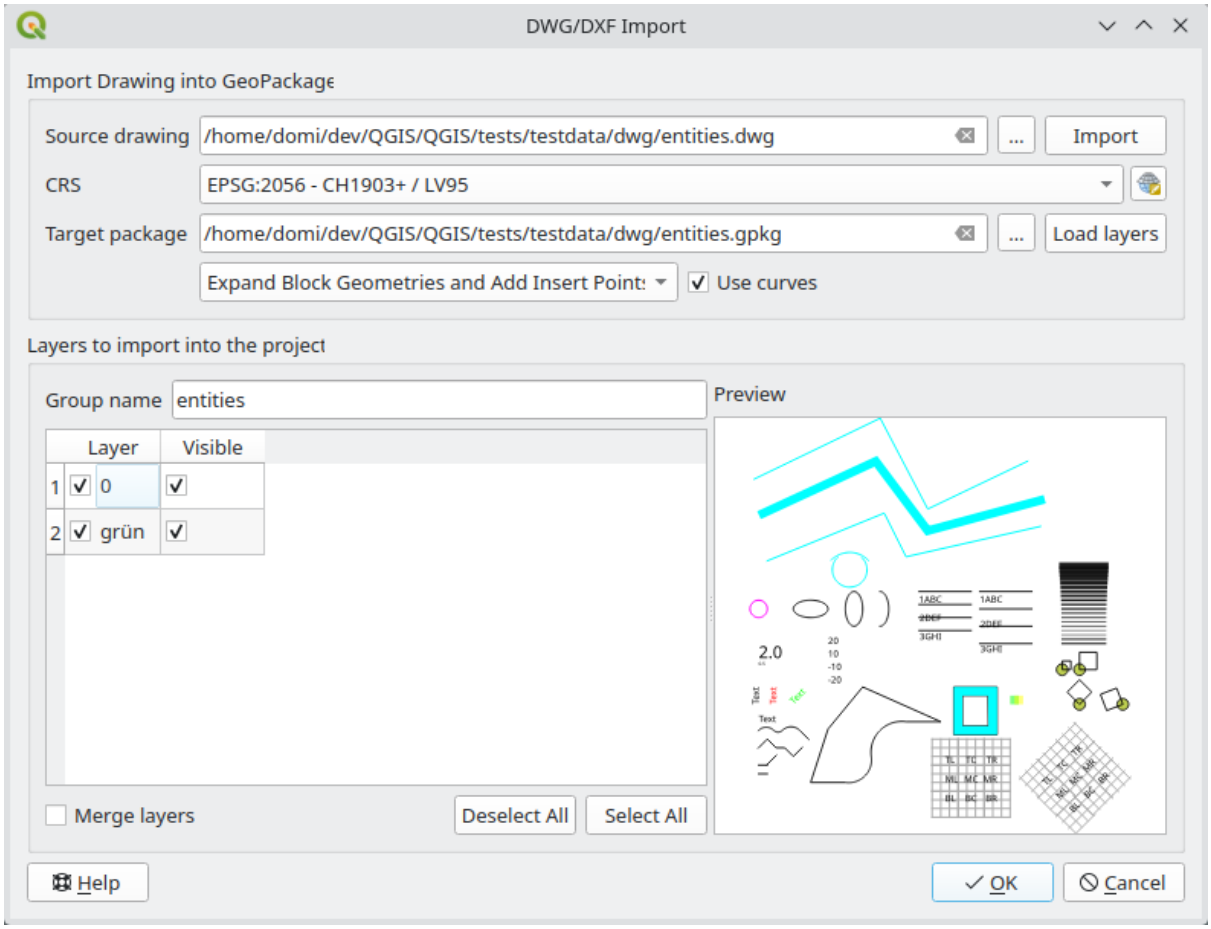


Fig. 11.10: DWG/DXF 文件的导入对话框

导入开放街道地图矢量

开放街道地图项目之所以受欢迎，是因为许多国家没有免费的地理数据，比如数字路线图。OSM 项目的目标是根据 GPS 数据、航空摄影和当地知识创建免费可编辑的世界地图。为了支持这一目标，QGIS 提供了对 OSM 数据的支持。

使用 *浏览器面板*，可以将 `.osm` 文件加载到地图画布，在此情况下，您将看到一个对话框，用于根据几何图形类型选择子图层。加载的图层将包含 `.osm` 文件中几何图形类型的所有数据，并保留 `osm` 文件的数据结构。

SpatiaLite 图层

第一次从 SpatiaLite 数据库加载数据时，首先执行以下操作：

- 单击 添加 SpatiaLite 图层 工具栏按钮
- 从 *图层* ► *添加图层* 菜单选择 *添加 SpatiaLite 图层...* 选项
- 或按 `Ctrl+Shift+L`

这将打开一个窗口，允许您连接到 QGIS 已知的 SpatiaLite 数据库 (从下拉菜单中选择)，或者定义与新数据库的新连接。要定义新连接，请单击 *新建*，然后使用文件浏览器指向 SpatiaLite 数据库，该数据库是扩展名为 `.sqlite` 的文件。

QGIS 还支持 SpatiaLite 中的可编辑视图。

GPS

有几十种不同的文件格式用于存储 GPS 数据。QGIS 使用的格式称为 GPX (GPS 交换格式)，这是一种标准的交换格式，可以在同一文件中包含任意数量的航点、路径和轨迹。

使用 ... 浏览 按钮选择 GPX 文件，然后使用复选框选择要从该 GPX 文件加载的要素类型。每个要素类型都将加载在一个单独的图层中。

有关 GPS 数据操作的更多信息，请访问*使用 GPS 数据* 。

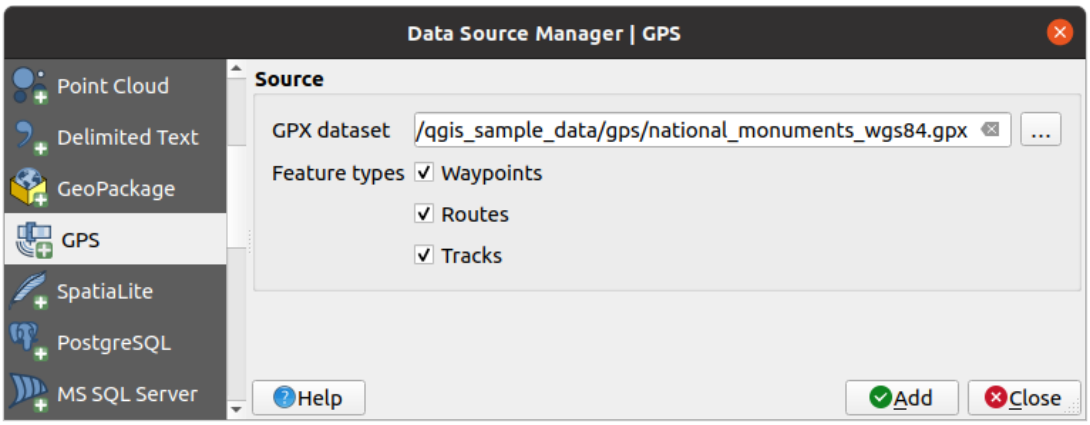


Fig. 11.11: 加载 GPS 数据对话框

GRASS

GRASS GIS Integration 章节介绍了使用 GRASS 矢量数据的方法。

数据库相关工具

创建存储的连接

为了从 QGIS 支持的数据库格式读取和写入表，必须创建与该数据库的连接。虽然*QGIS 浏览器面板* 是连接和使用数据库的最简单和推荐的方式，但 QGIS 提供了其他工具与每个数据库连接并加载其表:

- *Add PostgreSQL Layer...* or by typing `Ctrl+Shift+D`
- 添加 *Oracle Spatial* 图层... 或按 `Ctrl+Shift+O`
- 添加 *MS SQL Server* 图层
- 添加 *SAP HANA Spatial* 图层... 或按 `Ctrl+Shift+G`

这些工具可从 图层管理工具栏和 图层 ► 添加图层 ► 菜单访问。*SpatiaLite 图层* 介绍了连接到 SpatiaLite 数据库的方法。

> **Tip**
>
> **从 QGIS 浏览器面板创建与数据库的连接**
>
> 在浏览器树中选择相应的数据库格式，右键单击并选择连接，将提供数据库连接对话框。

Most of the connection dialogs follow a common structure:

- a section with credentials information to connect to the database

- a section with options to tune which data can be requested in the database

Connecting to PostgreSQL

The first time you use a PostgreSQL data source, you must create a connection to a database that contains the data. Press the appropriate button as exposed above, opening the *PostgreSQL* tab of the *Data Source Manager* dialog. To access the connection manager, click on the *New* button to display the *Create a New PostgreSQL Connection* dialog.

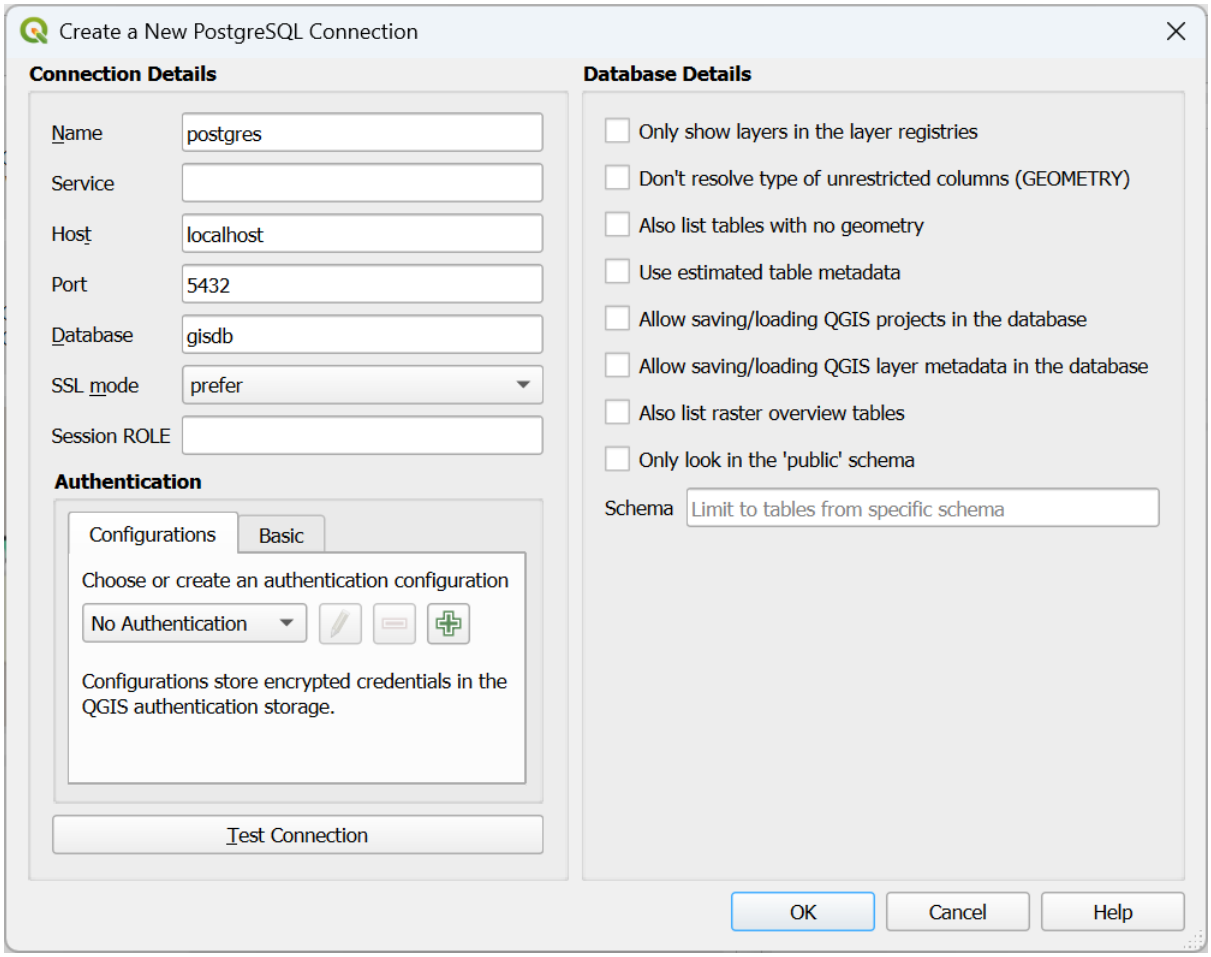


Fig. 11.12: Create a New PostgreSQL Connection Dialog

- *名称*: 此连接的名称。可以与 *数据库* 相同。
- *Service*: Service parameter to be used alternatively to hostname/port (and potentially database). This can be defined in `pg_service.conf`. Check the *PostgreSQL Service connection file* section for more details.
- *主机*: 数据库主机的名称。必须是可解析的主机名，例如用于打开 TCP/IP 连接或 Ping 主机的名称。如果数据库与 QGIS 处于同一台计算机，只需在此处输入 *localhost* 即可。
- *Port*: Port number the PostgreSQL database server listens on. The default port for PostgreSQL is `5432`.
- *数据库*: 数据库的名称。
- *SSL mode*: SSL encryption setup. The following options are available:
  - *优先* (默认): 我不关心加密，但如果服务器支持，希望支付加密的开销。
  - *需要*: 我希望我的数据被加密并接受开销。相信网络会确保始终连接到我想要的服务器。
  - *检验证书颁发机构*: 我希望我的数据加密并接受开销。希望确保连接到我信任的服务器。

  - 完全检验: 我希望我的数据加密并接受开销。我想确保连接到我信任的服务器，并且是由我指定的服务器。
  - 允许: 我不关心安全性，但如果服务器坚持加密，我会支付加密的开销。
  - 禁用: 我不关心安全性，也不想为加密支付开销。
- *Session role*: used to set the current user identifier of the current session. This is useful to automatically give the ownership of a new object (table, view, function) to the session_role group and thus share ownership and associated rights with all members of the session_role group. Read more about session role.
- *Authentication*: For general details about the authentication dialog behavior, see 认证.

根据数据库的类型，可以选择激活以下复选框:

- ☑ 只显示图层注册表中的图层
- ☑ 不解析无限制的列类型 (几何图形)
- ☑ 同时列出没有几何图形的表: 表示默认也应列出没有几何图形的表。
- ☑ *Use estimated table metadata*: When initializing layers, various queries may be needed to establish the characteristics of the geometries stored in the database table. When this option is checked, these queries examine only a sample of the rows and use the table statistics, rather than the entire table. This can drastically speed up operations on large datasets, but may result in incorrect characterization of layers (e.g. the feature count of filtered layers will not be accurately determined) and may even cause strange behaviour if columns that are supposed to be unique actually are not.
- ☑ 允许在数据库中保存/加载 *QGIS* 工程 - 更多信息在此
- ☑ 允许在数据库中保存/加载 *QGIS* 图层元数据 - 更多信息在此
- ☑ *Also list raster overview tables*
- ☑ 仅查找'公共'架构
- *Schema*: Allows to specify a single schema to limit a connection to. When set, only tables from the matching schema will be included in the browser panel and data source select for the connection. This can be used to limit the database work required to populate tables for a connection pointing to a large database store.

Once all parameters and options are set, you can test the connection by clicking the *Test Connection* button or apply it by clicking the *OK* button.

PostgreSQL Service connection file

服务连接文件允许 PostgreSQL 连接参数与单个服务名称相关联。然后，客户端可以指定该服务名称，并将使用相关的设置。

在 *nix 系统 (GNU/Linux、macOS 等) 称为 `.pg_service.conf`，在 Windows 称为 `pg_service.conf`。

服务文件可能如下所示:

```
[water_service]
host=192.168.0.45
port=5433
dbname=gisdb
user=paul
password=paulspass

[wastewater_service]
host=dbserver.com
dbname=water
user=waterpass
```

Note

There are two services in the above example: `water_service` and `wastewater_service`. You can use these to connect from QGIS, pgAdmin, etc. by specifying only the name of the service you want to connect to (without the enclosing brackets). If you want to use the service with `psql`, you can do `psql service=water_service`.

可以 在此 查找所有 PostgreSQL 参数

Note

如果不想将密码保存在服务文件中，可以使用 .pg_pass 选项。

Note

QGIS Server and service

When using a service file and QGIS Server, you must configure the service on the server side as well. You can follow the QGIS Server documentation.

在 *nix 操作系统 (GNU/Linux、macOS 等)，可以将 `.pg_service.conf` 文件保存在用户的主目录，PostgreSQL 客户端在此自动查找。例如，如果登录的用户是 `web`，应将 `.pg_service.conf` 保存在 `/home/web/` 目录中，以便直接工作 (无需指定任何其他环境变量)。

You can specify the location of the service file by creating a `PGSERVICEFILE` environment variable (e.g. run the `export PGSERVICEFILE=/home/web/.pg_service.conf` command under your *nix OS to temporarily set the `PGSERVICEFILE` variable).

还可以将 `.pg_service.conf` 文件放在 `pg_config --sysconfdir` 或添加 `PGSYSCONFDIR` 环境变量来指定包含服务文件的目录，从而使服务文件在系统范围内 (所有用户) 可用。如果用户和系统文件中存在相同名称的服务定义，则优先用户文件。

Warning

Windows 的注意事项：

- 服务文件应保存为 `pg_service.conf`，而不是 `.pg_service.conf`。
- 服务文件应该以 Unix 格式保存才能工作。一种方法是用 Notepad++ 打开并 *编辑* ► *EOL 转换* ► *UNIX 格式* ► *文件保存*。
- 可以通过各种方式添加环境变量。一个经过测试而可靠的方式是通过 *控制面板* ► *系统和安全* ► *系统* ► *高级系统设置* ► 环境变量添加 `PGSERVICEFILE` 的路径 - 例如 `C:\Users\John\pg_service.conf`
- 添加环境变量后，可能还需要重新启动计算机。

连接到 Oracle Spatial

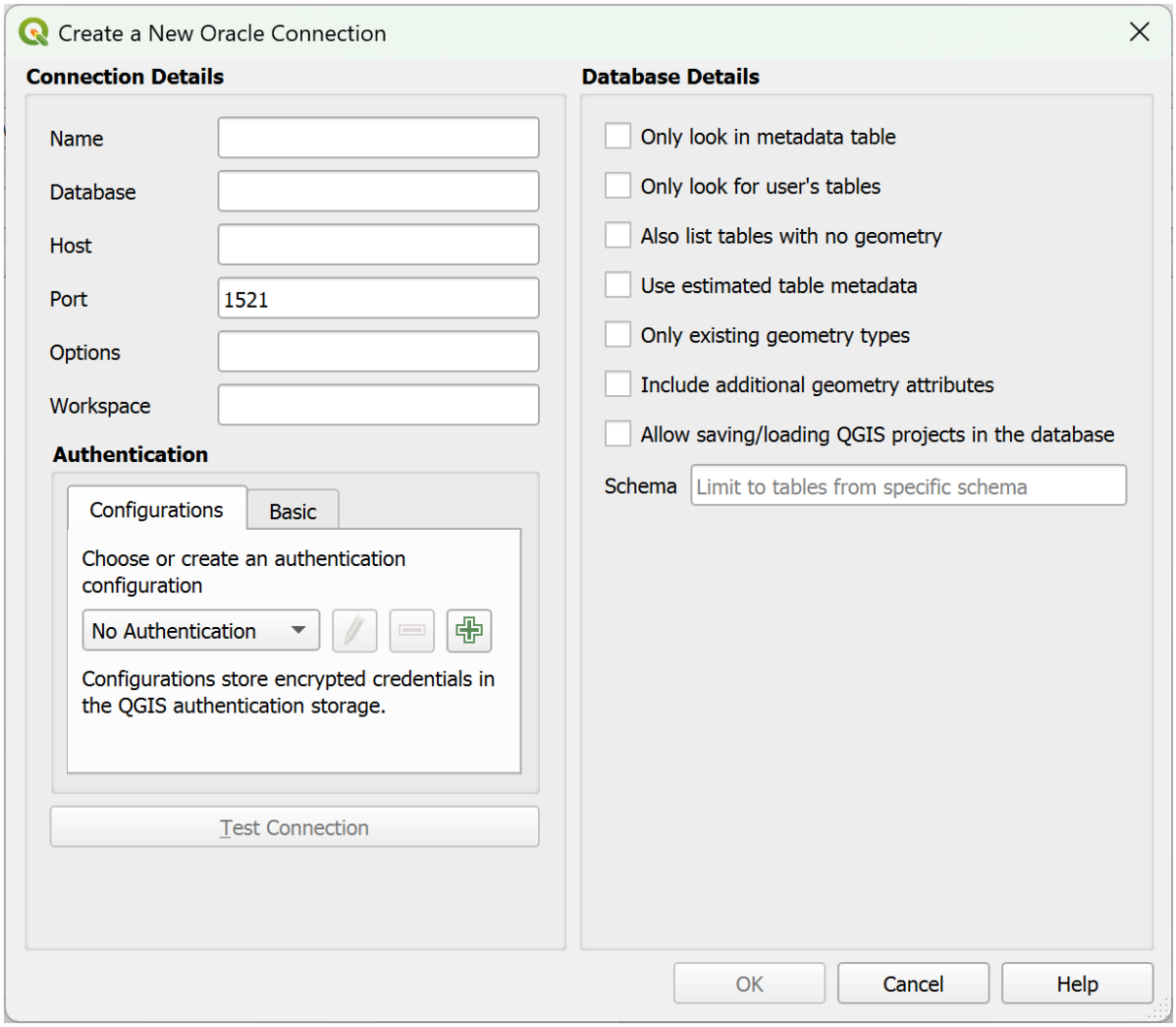


Fig. 11.13: Create a New Oracle Connection Dialog

The spatial features in Oracle Spatial aid users in managing geographic and location data in a native type within an Oracle database. The connection dialog proposes:

- *Name*: A name for this connection. It can be the same as *Database*;
- *Database*: SID or SERVICE_NAME of the Oracle instance;
- *Host*: The name of the database host;
- *Port*: Port number the Oracle database server listens on. The default port is `1521`;
- *Options*: Oracle connection specific options (e.g. OCI_ATTR_PREFETCH_ROWS, OCI_ATTR_PREFETCH_MEMORY). The format of the options string is a semicolon separated list of option names or option=value pairs;
- *Workspace*: Workspace to switch to;
- *Authentication*: For general details about the authentication dialog behavior, see 认证.

您可以激活以下复选框 (可选):

- 仅在元数据表中查找: 将显示的表限制为 `all_sdo_geom_metadata` 视图中的表。这可以加快空间表的初始显示。

- ☑ *仅查找用户表*: 在搜索空间表时，将搜索限制为用户拥有的表。
- ☑ *同时列出没有几何图形的表*: 表示默认也应列出没有几何图形的表。
- ☑ *对图层元数据使用预估表统计*: 设置图层时，Oracle 表需要各种元数据，包括表行数、几何图形类型和几何图形列中数据的空间范围等信息。如果表包含大量行，准确算出此表元数据可能非常耗时。通过激活此选项，将完成以下快速表元数据操作：行计数由 `all_tables.num_rows` 确定。表范围始终由 SDO_TUNE.EXTENTS_OF 函数确定，即使应用了图层过滤也是如此。表几何图形是根据表中的前 100 个非空几何图形行确定的。
- ☑ *仅既有几何图形类型*: 仅列出现有而不提供添加其他几何图形类型。
- ☑ *包含附加几何图形属性*。
- ☑ *允许在数据库中保存/加载 QGIS 工程* - 更多信息在此
- *Schema*: Allows to specify a single schema to limit a connection to. When set, only tables from the matching schema will be included in the browser panel and data source select for the connection. This can be used to limit the database work required to populate tables for a connection pointing to a large database store.

> **Tip**
>
> **Oracle Spatial 图层**
>
> 通常，Oracle Spatial 图层由 **USER_SDO_METADATA** 表中的条目定义。
>
> 为确保选择工具正常工作，建议您的表设置 **主键**。

连接到 MS SQL Server

As mentioned in 创建存储的连接 QGIS allows you to create MS SQL Server connection through *Data Source Manager*.

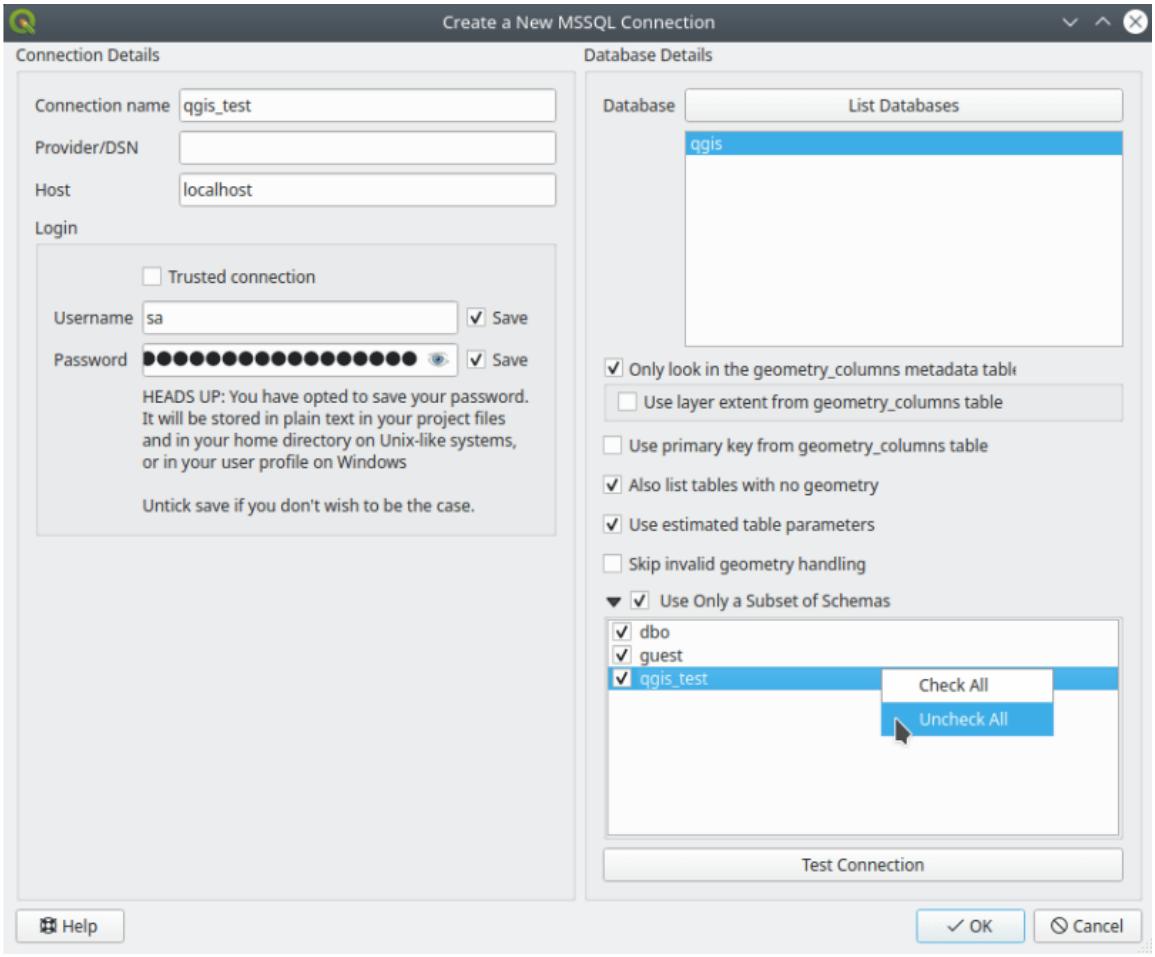


Fig. 11.14: MS SQL Server Connection

To create a new MS SQL Server connection, you need to provide some of the following information in the *Connection Details* dialog:

- *Connection name*
- *Provider/DSN*
- *Host*
- *Login* information. You can choose to *Save* your credentials.

Navigate to the *Database Details* section and click the *List Databases* button to view the available datasets. Select datasets that you want, then press *OK*. Optionally, you can also perform a *Test Connection*. Once you click *OK* the *Create a New MS SQL Server Connection* dialog will close and in the *Data Source Manager* press *Connect*, select a layer and then click *Add*.

Optionally, you can activate the following options:

- *Only look in the geometry_columns metadata table*: restricts the available tables to the ones in the `geometry_columns` metadata table when scanning for tables. This can speed up the table scanning.
- *Use layer extent from geometry_columns table*: this option, dependent on the previous one, allows QGIS to skip extent calculation when loading layers and thus lowering the amount of time needed to load them. It relies on extent manually specified using additional QGIS-specific columns (`qgis_xmin`, `qgis_xmax`, `qgis_ymin`, `qgis_ymax`) in the `geometry_columns` table.
- *Use primary key from geometry_columns table*: allows QGIS to skip primary key calculation for views when loading them, thus lowering the amount of time needed to load them. It relies on names manually filled

in a QGIS-specific `qgis_pkey` column set in the `geometry_columns` table. If more than one column is used for the primary key, they should be filled as comma separated values.

- *Also list table with no geometry*: tables without a geometry column attached will also be shown in the available table list.
- *Use estimated table parameters*: only estimated table metadata will be used. This avoids a slow table scan, but may result in incorrect layer properties such as layer extent.
- *Skip invalid geometry handling*: all handling of records with invalid geometry will be disabled. This speeds up the provider, however, if any invalid geometries are present in a table then the result is unpredictable and may include missing records. Only check this option if you are certain that all geometries present in the database are valid, and any newly added geometries or tables will also be valid.
- *Use only a Subset of Schemas* will allow you to filter schemas for MS SQL connection. If enabled, only checked schemas will be displayed. You can right-click to *Check* or *Uncheck* any schema in the list.

连接到 SAP HANA

Note

需要 SAP HANA 客户端连接到 SAP HANA 数据库。可以在 SAP Development 工具网站 下载适用于您平台的 SAP HANA 客户端。

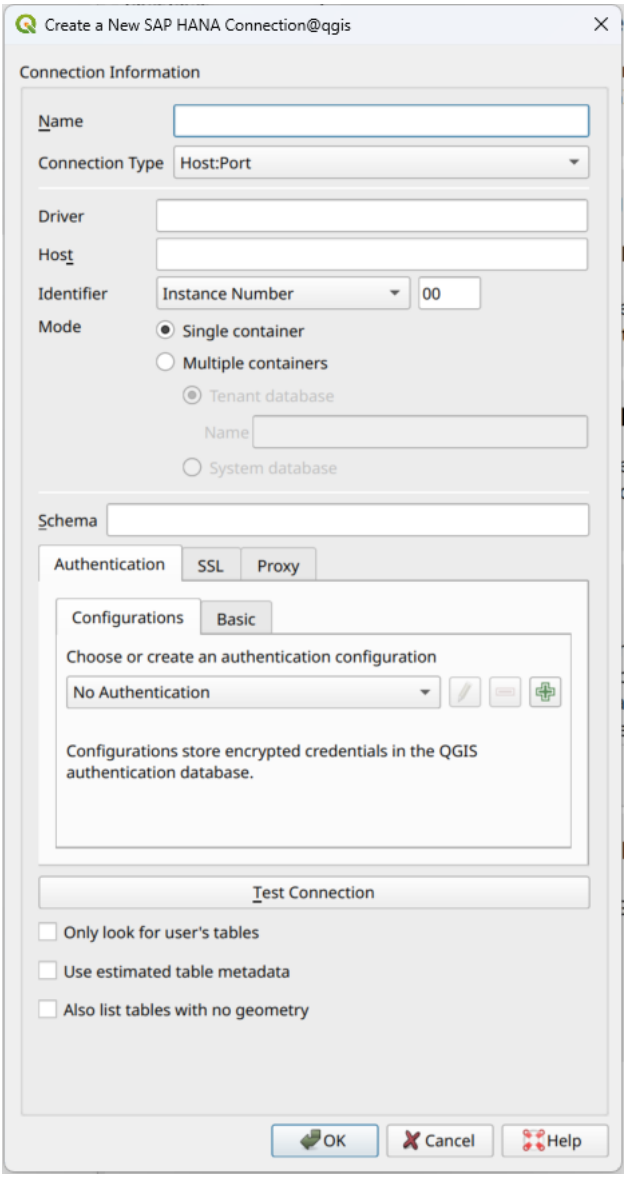


Fig. 11.15: 新建 SAP HANA 连接对话框

可以输入以下参数：

- *名称*: 此连接的名称。
- *驱动* : HANA ODBC 驱动程序的名称。如果使用 64 位 QGIS 则为 `HDBODBC` ，如果使用 32 位 QGIS 则为 `HDBODBC32` 。将自动输入相应的驱动程序名称。
- *驱动* : 在 `/etc/odbcinst.ini` 中注册 SAP HANA ODBC 驱动程序时使用的名称，或 SAP HANA ODBC 驱动程序的完整路径。默认 SAP HANA 客户端安装程序会将 ODBC 驱动程序安装到 `/usr/sap/hdbclient/libodbcHDB.so` 。
- *主机*: 数据库主机的名称。
- *标识符*: 标识主机上要连接到的实例。可以是 *实例号* 或 *端口号* 。实例号由两位数字组成，端口号在 1 到 65,535 之间。
- *模式*: 指定 SAP HANA 实例运行的模式。只有当 *标识符* 设置为 *实例号* 时，才会考虑此设置。如果数据库托管多个容器，可以使用 *Tenant 数据库* 中给定的名称连接到租用者，也可以连接到系统数据库。

- 架构: 此参数可选。如果给定了架构名称，QGIS 只搜索该架构中的数据。如果此字段为空，QGIS 将搜索所有架构中的数据。
- *Authentication*: For general details about the authentication dialog behavior, see 认证.
- *SSL* 设置
  - 启用 *TLS/SSL* 加密: 启用 TLS 1.1 - TLS1.2 加密。服务器将选择可用的最高级。
  - 提供程序: 指定用于 SSL 通信的加密库提供程序。*sapcrypto* 适用于所有平台, *openssl* 适用于 , *mscrypto* 适用于 ， *commoncrypto* 需要安装 CommonCryptoLib。
  - 验证 *SSL* 证书: 如果勾选，SSL 证书将使用 带公钥的信任储存文件中给定的信任存储库进行验证。
  - 替换证书中的主机名: 指定用于检验服务器标识的主机名。此处指定的主机名验证服务器的标识，而不是与之建立连接的主机名。如果将 * 指定为主机名，则不会验证服务器的主机名。不允许使用其他通配符。
  - 含私钥的密钥库文件: 当前已忽略。此参数允许将来通过证书进行身份验证，而不是使用用户名和密码。
  - 带公钥的信任储存文件: 如果使用 OpenSSL，则指定包含服务器公共证书的信任储存文件的路径。通常，信任储存包含根证书或对服务器的公共证书进行签名的证书颁发机构的证书。如果您正在使用加密库 CommonCryptoLib 或 msCrypto，请将此属性留空。
- 仅查找用户表: 如果勾选，QGIS 只搜索连接到数据库的用户所拥有的表和视图。
- *Use estimated table metadata*: If checked, estimated table metadata will be used if available. For large tables, this avoids slow table loads and potentially expensive computations, but may result in incorrect layer properties such as layer extent. The fast extent estimation is available starting with QRC1/2024 and SP8 in HANA Cloud and HANA On-Premise respectively.
- 同时列出没有几何图形的表: 如果勾选，QGIS 还会搜索不包含空间列的表和视图。

> **Tip**
>
> **连接到 SAP HANA 云**
>
> 如果要连接到 SAP HANA 云实例，必须将 端口号 设置为 443 并勾选 启用 *TLS/SSL* 加密。

加载数据库图层

Once you have one or more connections defined to a database (see section 创建存储的连接), you can load layers from it. Of course, this requires that data are available. See section 将数据导入 *PostgreSQL* for a discussion on importing data into a PostgreSQL database.

要从数据库加载图层，可以执行以下步骤:

1. Open the corresponding tab of the database in the *Data Source Manager* dialog.
2. Choose the connection name from the drop-down list and press *Connect*.
3. The table below will be filled with your data grouped by schema, with a number of metadata information helpful for loading.
4. 选择或取消选择 同时列出没有几何图形的表。
5. 可选，使用某些 搜索选项将表列表缩减为与搜索匹配的表。也可以在点击 连接按钮之前设置此选项，从而加快数据库获取速度。
6. 在可用图层列表中查找要添加的图层。

7. 单击即可选择。单击时按住 `Shift` 或 `Ctrl` 键可以选择多个图层。

   Layers can be selected only if they have no ⚠ warning icon at the left-hand side of their first column. This may indicate an issue to detect:

   - features geometry type: in which case you can select the appropriate one in the drop-dowmn list of the *Spatial Type* column
   - layer CRS: you can enter the correct code in the *SRID* column
   - layer's primary key, in order to unequivocally identify each feature: this can be fixed by selecting one or more attributes in the drop-down list at the corresponding *Feature id* column.

   > **Tip**
   >
   > **Use the first column to store "primary keys" for views**
   >
   > Since PostgreSQL views don't support primary keys, a unique attribute or combination of attributes should always be selected. To help users to speed-up workflows, QGIS automatically selects the first attribute in the view. Therefore, users can define their views in a way that a unique column is in the first position of the view's definition. In this way, the view will be loaded with no extra interaction and the warning icon will never appear.

8. 如果适用，请使用 *设置过滤按钮* (或双击图层) 启动 *查询构建器对话框* (请参阅查询构建器)，并定义从所选图层加载的要素。过滤表达式显示在 `sql` 列。可以在 *图层属性* ► *常规* ► *提供程序的要素过滤框架*中移除或编辑此限制。
9. 默认激活的 *按 `ID` 选择列*中的复选框将获取不带属性的要素 ID，通常会加快数据加载速度。
10. 单击 *添加*按钮将图层添加到地图。

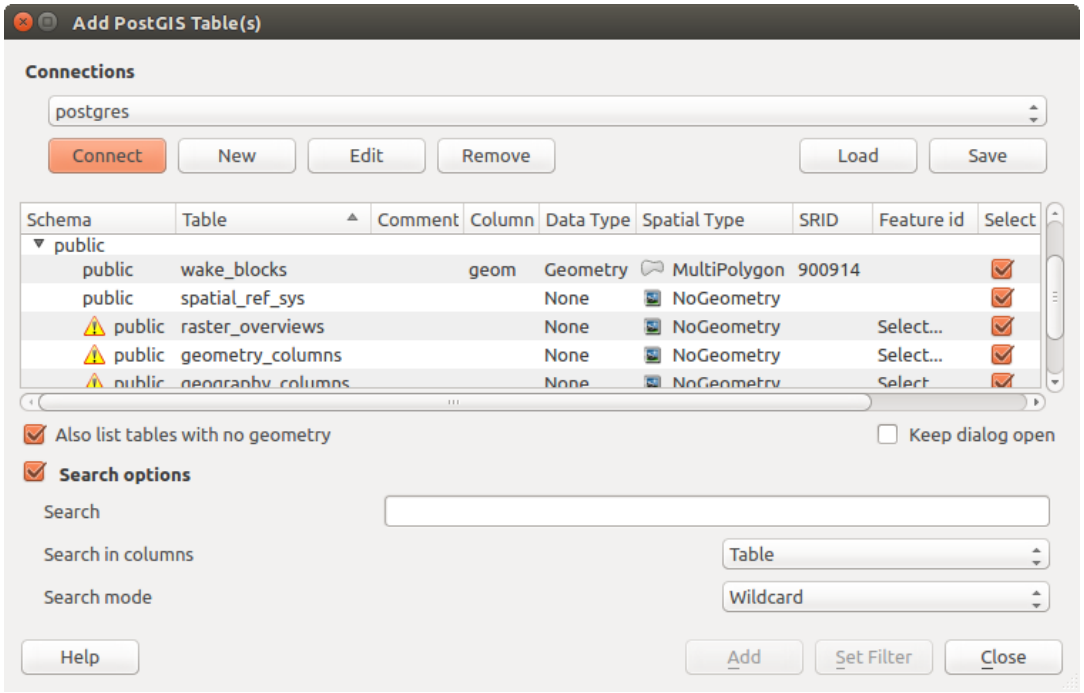


Fig. 11.16: Add PostgreSQL Table(s) Dialog

> **Tip**
>
> **使用浏览器面板加速数据库表的加载**

从 数据源管理器添加数据库表有时很耗时，这是因为 QGIS 会事先获取每个表的统计信息和属性 (例如几何图形类型和字段、CRS、要素数量)。为了避免这种情况，设置连接 后，最好使用浏览器面板 或数据库管理器 将数据库表拖放到地图画布。

11.1.4 图层元数据搜索面板

By default, QGIS can retrieve layers metadata from the connections or data providers that allow metadata storage (more details on *saving metadata to the database*). The *Metadata search* panel allows to browse the layers by their metadata and add them to the project (either with a double-click or the *Add* button). The list can be filtered:

- by text, watching a set of metadata properties (identifier, title, abstract)
- by spatial extent, using the current *project extent* or the map canvas extent
- by the layer (geometry) type

Note

元数据的来源是通过图层元数据提供程序系统实现的，该系统可以通过插件进行扩展。

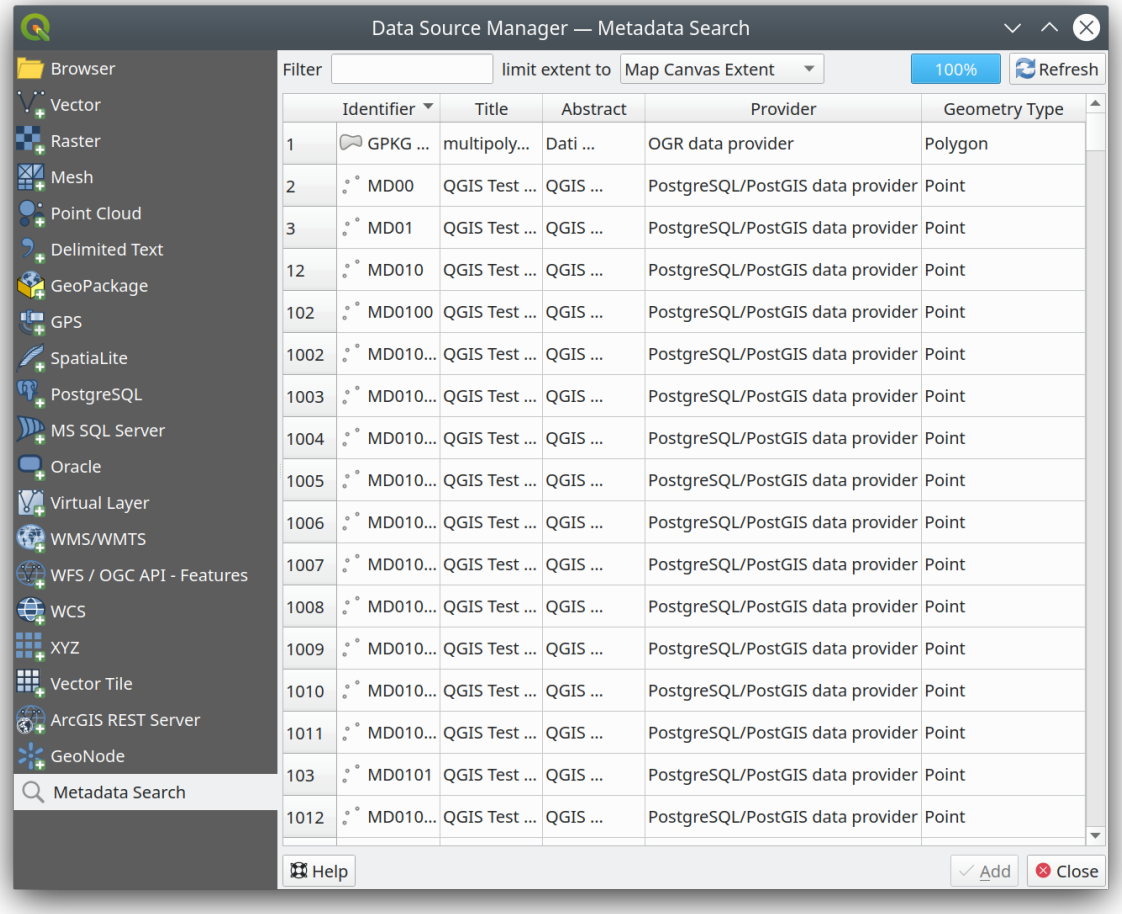


Fig. 11.17: 图层元数据搜索面板

11.1.5 QGIS 自定义格式

QGIS 提供了两种自定义格式：

- 临时图层: 绑定到工程的内存图层 (更多信息请参阅新建临时图层)
- 虚拟图层: 由对其他图层的查询而产生的图层 (更多信息请参阅创建虚拟图层)

11.1.6 QLR - QGIS 图层定义文件

可以在图层上下文菜单使用 导出 ► 另存为图层定义文件... 将图层定义保存为图层定义文件 (QLR - `.qlr`) 。

QLR 格式可以与其他 QGIS 用户共享" 完整" 的 QGIS 图层。QLR 文件包含指向数据源的链接，以及为图层设置样式所需的所有 QGIS 样式信息。

QLR 文件显示在浏览器面板中，可用于将图层 (及其保存的样式) 添加到图层面板。还可以将 QLR 文件从系统文件管理器拖放到地图画布。

Available actions for QLR files in the Browser Panel are:

- *Export Layer* ► *To file*
- 添加图层到工程
- 图层属性...

11.1.7 连接到 Web 服务

使用 QGIS，可以访问不同类型的 OGC Web 服务 (WM(T)S, WFS(-T), WCS, CSW 等)。借助 QGIS Server 还可以发布此类服务。QGIS-Server-manual 包含对这些功能的描述。

使用矢量瓦片服务

可以通过 数据源管理器对话框的 矢量瓦片选项卡或 浏览器面板中 矢量瓦片条目的上下文菜单添加矢量瓦片服务。服务可以是 新建通用连接... 或 新建 *ArcGIS* 矢量瓦片服务连接... 。

您可以通过添加以下内容来设置服务：

- 名称
- 样式 *URL*: 指向 MapBox GL JSON 样式配置的 URL。如果提供，那么每当将连接中的图层添加到 QGIS 时，就会应用该样式。对于 Arcgis 矢量瓦片服务连接，URL 会覆盖服务器配置中指定的默认样式配置。

  You can load vector tiles directly from a *Style URL*. The data source is automatically parsed from the style, and URLs with multiple sources are supported. That makes *Source URL* optional.
- the *Source URL*: of the type `http://example.com/{z}/{x}/{y}.pbf` for generic services and `http://example.com/arcgis/rest/services/Layer/VectorTileServer` for ArcGIS based services. The service must provide tiles in `.pbf` format.
- ☑ 最小缩放级别和 ☑ 最大缩放级别: 矢量瓦片具有金字塔结构。通过使用这些选项，可以从瓦片金字塔单独生成图层。然后，这些图层将用于在 QGIS 中渲染矢量瓦片。

  对于墨卡托投影 (OpenStreetMap 矢量瓦片使用)，缩放级别 0 以 1:500.000.000 的比例表示整个世界。缩放级别 14 表示的比例为 1:35.000。
- the *authentication* configuration if necessary
- 引用

Fig. 11.18 shows the dialog with the Vector Tiles service configuration.

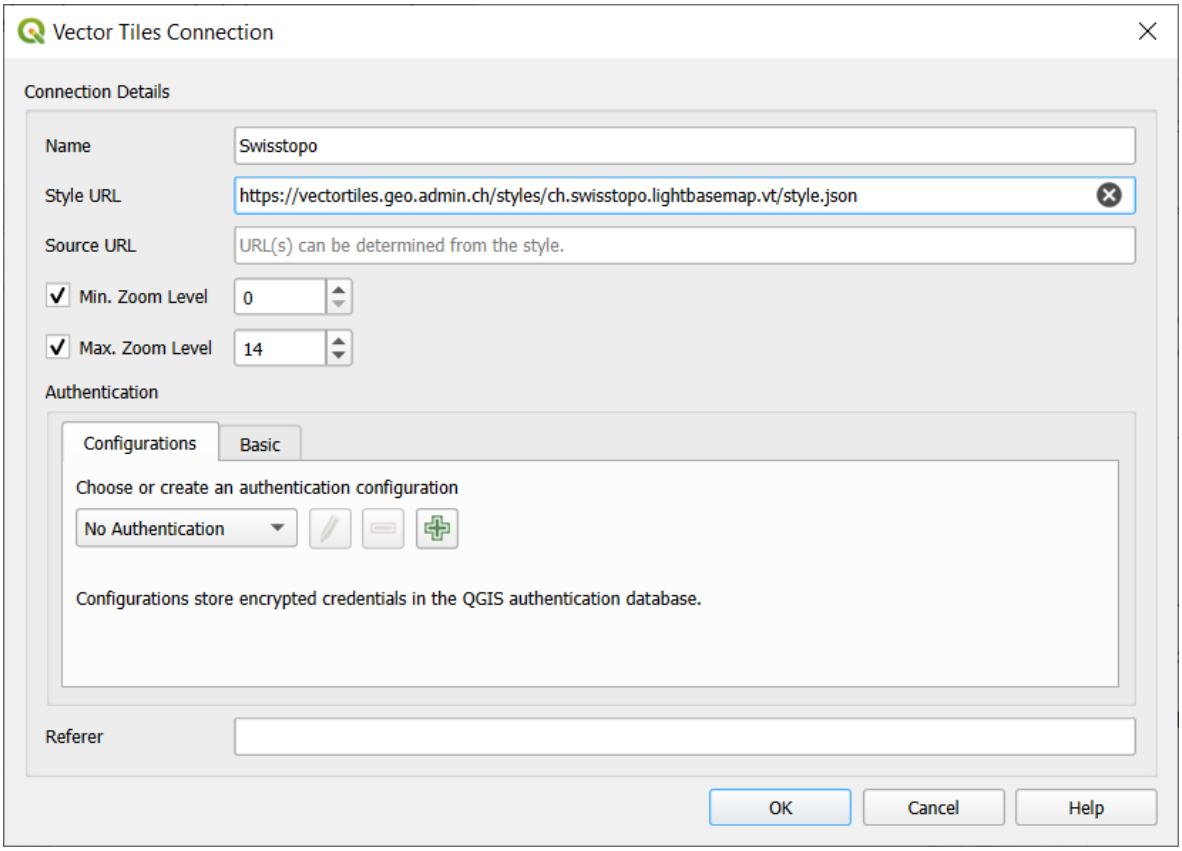


Fig. 11.18: Vector Tiles - Service configuration

可以通过 *数据源管理器*对话框中的 *矢量瓦片*条目或 *浏览器*面板中的上下文菜单将配置保存到 `.XML` 文件 (*保存连接*) 。同样，也可以从文件 (*加载连接*) 添加配置。

设置与矢量瓦片服务的连接后，可以：

- 编辑矢量瓦片连接设置
- 移除连接
- 在 *浏览器*面板的条目上右键单击，还可以：
  - *添加图层到工程*: 双击也可以添加图层
  - 查看 *图层属性*... 并访问元数据以及由服务提供的数据预览。将图层加载到工程后，可以使用更多设置。

使用 XYZ 瓦片服务

XYZ Tile services can be added via the *XYZ* tab of the *Data Source Manager* dialog or the contextual menu of the *XYZ Tiles* entry in the *Browser* panel. By default, QGIS provides some default and ready-to-use XYZ Tiles services:

- *Mapzen Global Terrain*: 允许直接访问工程的全球 DEM 源。更多信息和资源，请访问 https://registry.opendata.aws/terrain-tiles/
- *OpenStreetMap*: 访问世界 2D 地图。Fig. 11.19 显示了具有 OpenStreetMap XYZ 瓦片服务配置的对话框。

To add a new service, press *New* (respectively *New Connection* from the Browser panel) and provide:

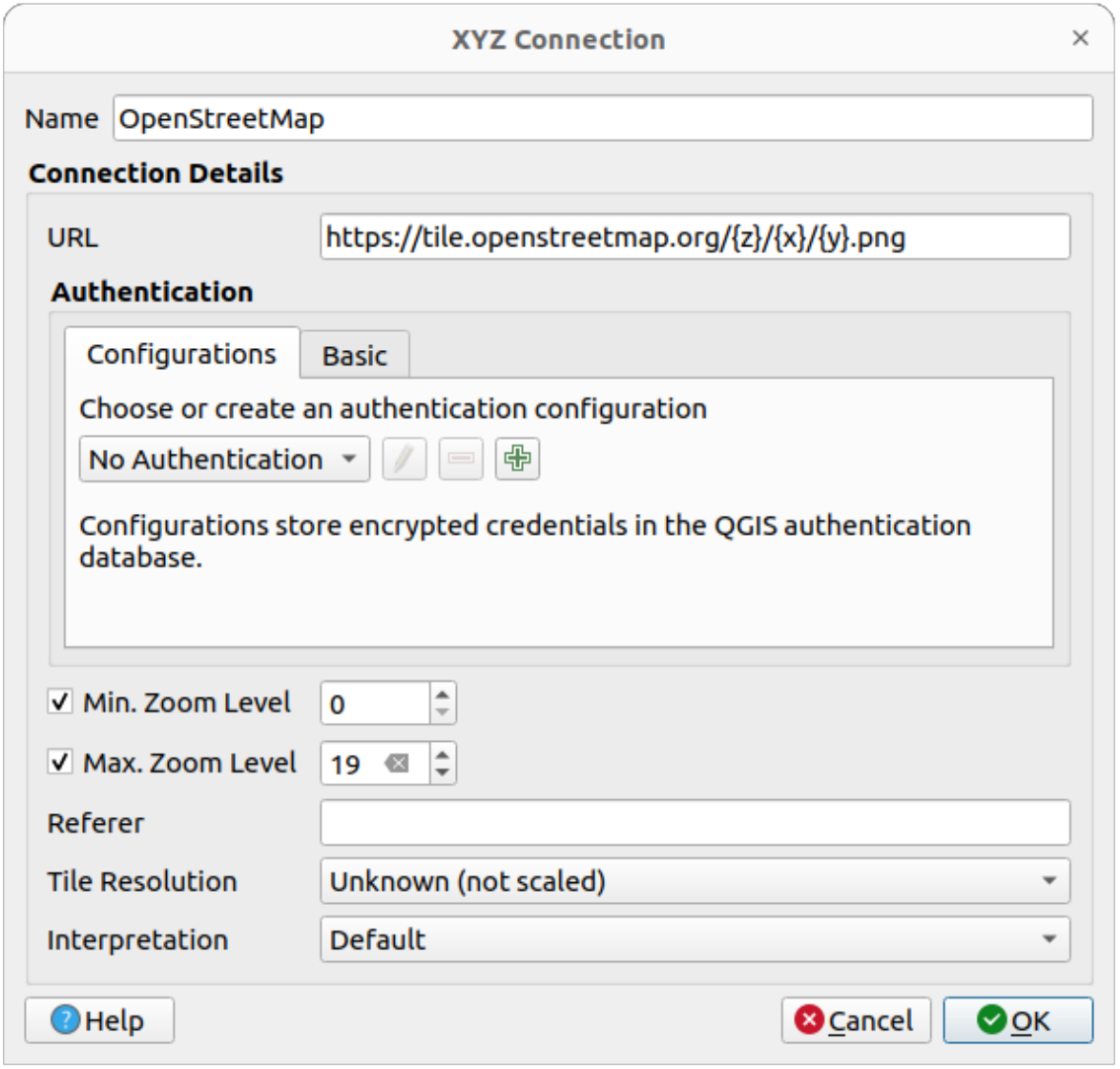


Fig. 11.19: XYZ 瓦片 - 开放街道地图配置

- 名称
- the *URL*, you can add `http://example.com/{z}/{x}/{y}.png` or `file:///local_path/{z}/{x}/{y}.png`
- the *authentication* configuration if necessary
- 最小缩放级别和 最大缩放级别
- 引用
- 瓦片分辨率: 可接受的值为 未知 *(无缩放)*, 标准 *(256x256 / 96DPI)* 和 高 *(512x512 / 192DPI)*
- 插值: 按照预定义的编码方案将 WMTS/XXYZ 栅格数据集转换为单波段浮点类型的栅格图层。支持的方案有 默认 (不进行转换), *MapTiler 地形 RGB* 和 *Terrarium 地形 RGB* 。所选转换器将把 RGB 源值转换为每个像素的浮点值。加载后，该图层将显示为单波段浮点栅格图层，可以使用 QGIS 常见的栅格渲染器 进行样式设置。

Press *OK* to establish the connection. It will then be possible to:

- *Add* the new layer to the project; it is loaded with the name given in the settings.

- 编辑 XYZ 连接设置
- 移除连接
- 在 浏览器面板的条目上右键单击，还可以：
  - 导出图层... ► 到文件, 将其另存为栅格
  - 添加图层到工程: 双击也可以添加图层
  - 查看 图层属性... 并访问元数据以及由服务提供的数据预览。将图层加载到工程后，可以使用更多设置。

Configurations can be saved to `.XML` file (*Save Connections*) through the *XYZ* entry in *Data Source Manager* dialog or its contextual menu in the *Browser* panel. Likewise, they can be added from a file (*Load Connections*).

开放街道地图的 XML 文件如下所示：

```
<!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>
```

Tip

Loading XYZ tiles without creating a connection

It is also possible to add XYZ tiles to a project without necessarily storing its connection settings in you user profile (e.g. for a dataset you may need once). In the *Data Source Manager* ► *XYZ* tab, edit any properties in the *Connection Details* group. The *Name* field above should turn into `Custom`. Press *Add* to load the layer in the project. It will be named by default `XYZ Layer`.

XYZ 瓦片服务示例：

- OpenStreetMap 单色: *URL*: `http://tiles.wmflabs.org/bw-mapnik/{z}/{x}/{y}.png`, 最小缩放级别: 0, 最大缩放级别: 19。
- 谷歌地图: *URL*: `https://mt1.google.com/vt/lyrs=m&x={x}&y={y}&z={z}`, 最小缩放级别: 0, 最大缩放级别: 19。
- Open Weather Map Temperature: *URL*: `http://tile.openweathermap.org/map/temp_new/{z}/{x}/{y}.png?appid={api_key}` 最小缩放级别: 0, 最大缩放级别: 19。

使用 ArcGIS REST 服务器

An ArcGIS REST Server can host many different types of web services (feature service, map service, image service, ...). ArcGIS REST Servers can be added via the *ArcGIS REST Server* tab of the *Data Source Manager* dialog or the contextual menu of the *ArcGIS REST Servers* entry in the *Browser* panel:

1. Press *New* (respectively *New Connection*) and provide:
   - a *Name*: A name for the connection.
   - the *URL*: Main address of the ArcGIS REST Server.
   - a *Prefix*: Used to specify the proxy prefix in the URL, which is necessary for some ArcGIS servers that use web proxy prefixes.
   - a *Community endpoint URL*: Endpoint URL for ArcGIS Online or Portal for ArcGIS, used to access content groups.
   - a *Content endpoint URL*: Endpoint URL for the content service. This is used to access items in ArcGIS Online or Portal for ArcGIS.

- the *authentication* credentials if necessary.
- a *Referer*: The referer URL to be sent in the HTTP headers when making requests to the server. This may be required by some servers for authentication purposes.

> **Note**
>
> 设置相应 Portal 终端 URLS 集的 ArcGIS Feature Service 连接后，可由浏览器面板中的内容组进行浏览。
>
> 如果连接设置了 Portal 终端，在浏览器展开连接将显示“组”和“服务”文件夹，而不是通常显示服务的完整列表。展开组文件夹将显示用户所属的所有内容组的列表，每个内容组都可以展开以显示属于该组的服务条目。

2. Then press *OK* to validate the configuration settings. These configurations can be saved to `.XML` file (*Save*). Likewise, they can be added from a file (*Load*).
3. Once a connection to an ArcGIS REST Server is set, it is possible to:
   - 编辑 ArcGIS REST Server 连接设置
   - 移除连接
   - 刷新连接
4. Press *Connect* to request the server and display its contents. They are organized in a tree structure whose nodes depend on the connection's endpoint.

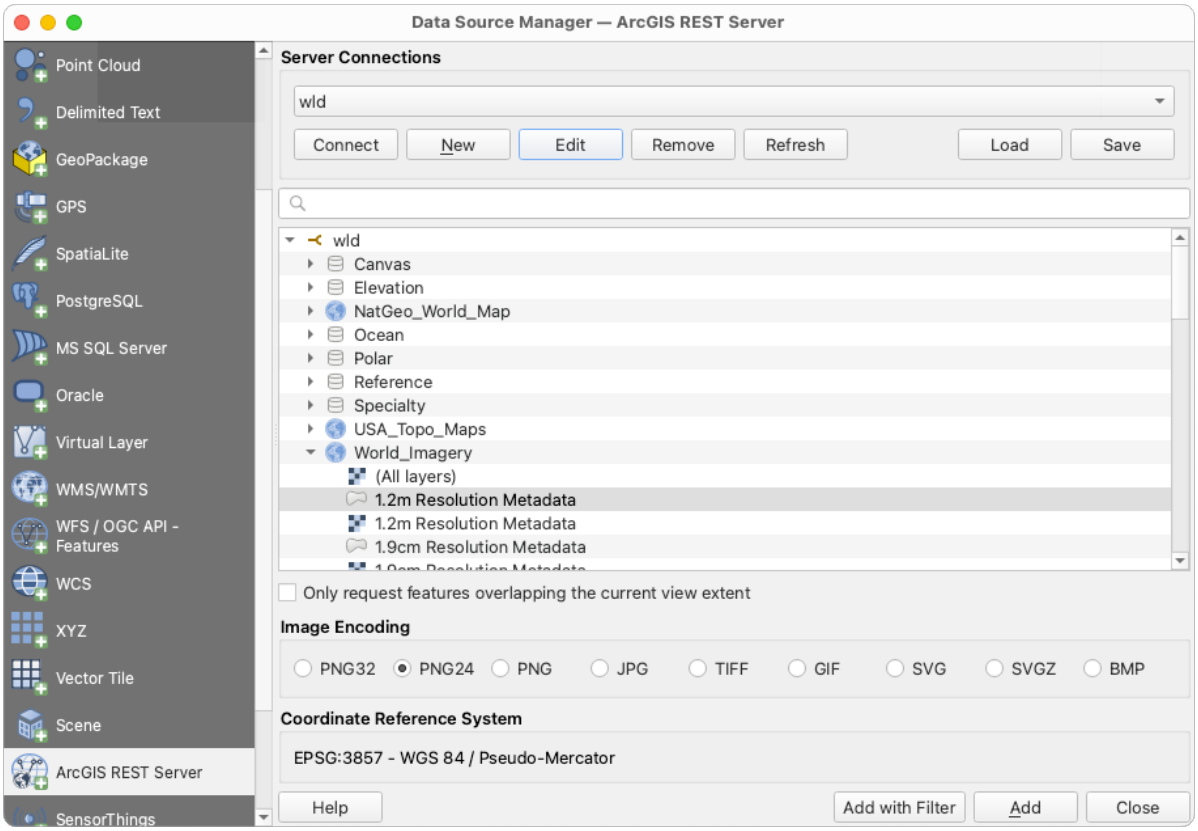


Fig. 11.20: Data Source Manager - ArcGIS REST Server

5. Expand the tree to find and select the layers of interest. Their *Coordinate Reference System* is displayed at the bottom of the dialog.

6. For raster-based layers, you can select the *Image encoding* to use among a number of image formats advertized by the target service (e.g., `PNG`, `JPG`, `GIF`, `SVG`, `SVGZ`, ...).
7. To add the selected layers to the map canvas, press *Add* button.

   Because layers can sometimes load and render slowly on the client side, applying a filter to restrict the features retrieved from the service can significantly improve performance, since only the filtered features are requested from the server. This can be done by:

   - checking *Only request features overlapping the current view extent*
   - for vector layers (feature service), pressing *Add with filter* to apply attribute-based filters to the layer with 表达式字符串构建器 functions.

     In the *Browser* panel, right-click the ArcGIS REST Server layer and select *Add Filtered Layer to Project* will also open the builder dialog.

Most of the above tools to connect and access layers on an ArcGIS REST Server are also available from the *Browser* panel, within the contextual menu of the target node item. Depending on the node, you can also:

- *View Service Info* which will open the default web browser and display information on the requested service.
- View the *Layer Properties...* and get access to the metadata and a preview of the data provided by the service. More settings are available when the layer has been loaded into the project.

Using 3D tiled scene services

QGIS supports multiple formats of 3D tiled datasets, grouped together as ”tiled scenes”. These include Cesium 3D Tiles and Quantized Mesh tiles.

To load a tiled scene dataset into QGIS, use the *Scene* tab in the *Data Source Manager* dialog.

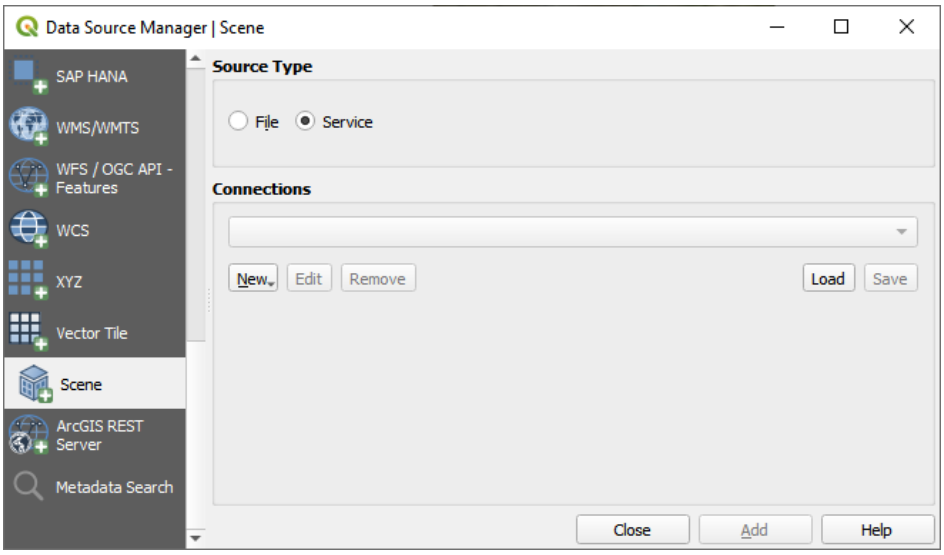


Fig. 11.21: Data Source Manager - Scene

Create a connection by clicking on *New*. You can add a *New Cesium 3D Tiles Connection* or a *New Quantized Mesh Connection*.

Choose a *Name* and set the *URL* to the URL of a layer description JSON file.

The URL may be remote (e.g. `http://example.com/tileset.json`) or local (e.g. `file:///path/to/tiles/tileset.json`).

The *authentication* configuration if necessary.

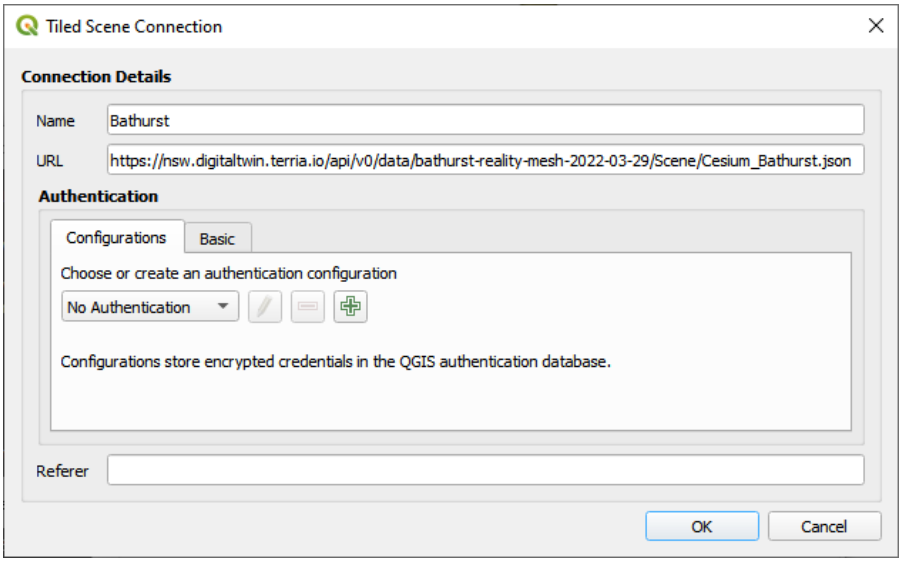


Fig. 11.22: Tiled Scene Connection

You can also add the service from *Browser Panel*.

After creating new connection you are able to *Add* the new layer to your map.

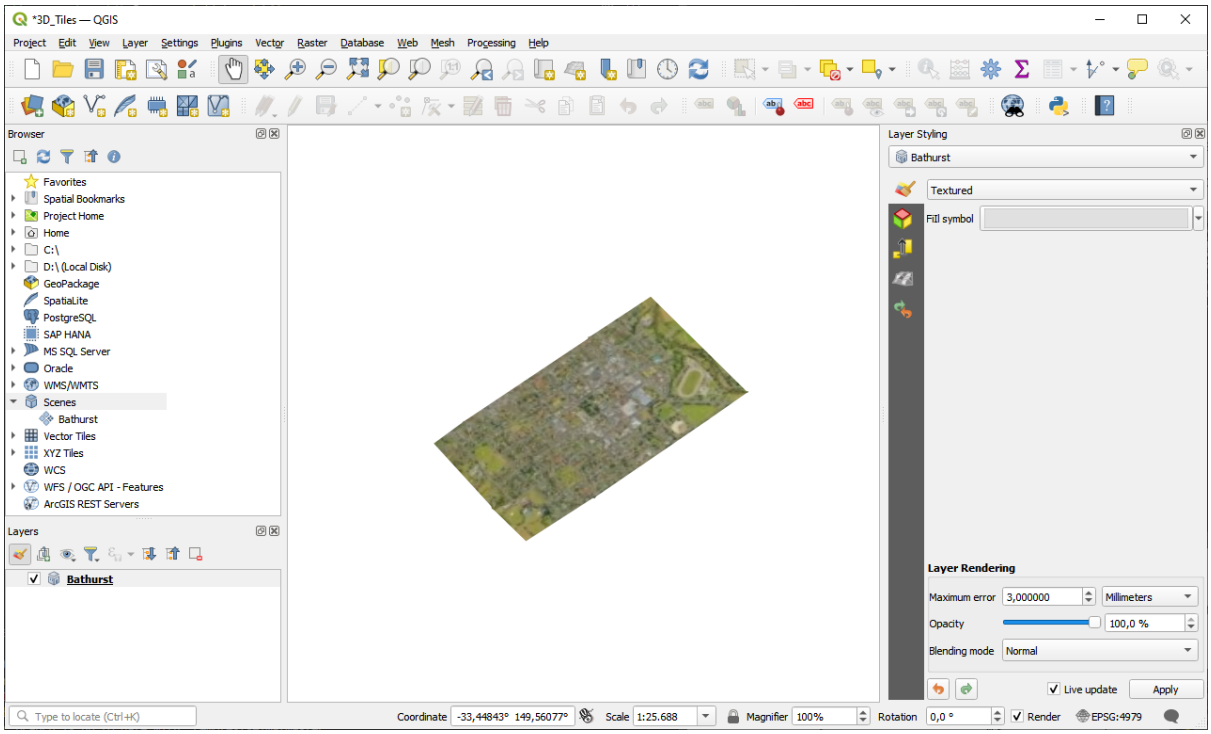


Fig. 11.23: 3D Tiles Layer - Textured

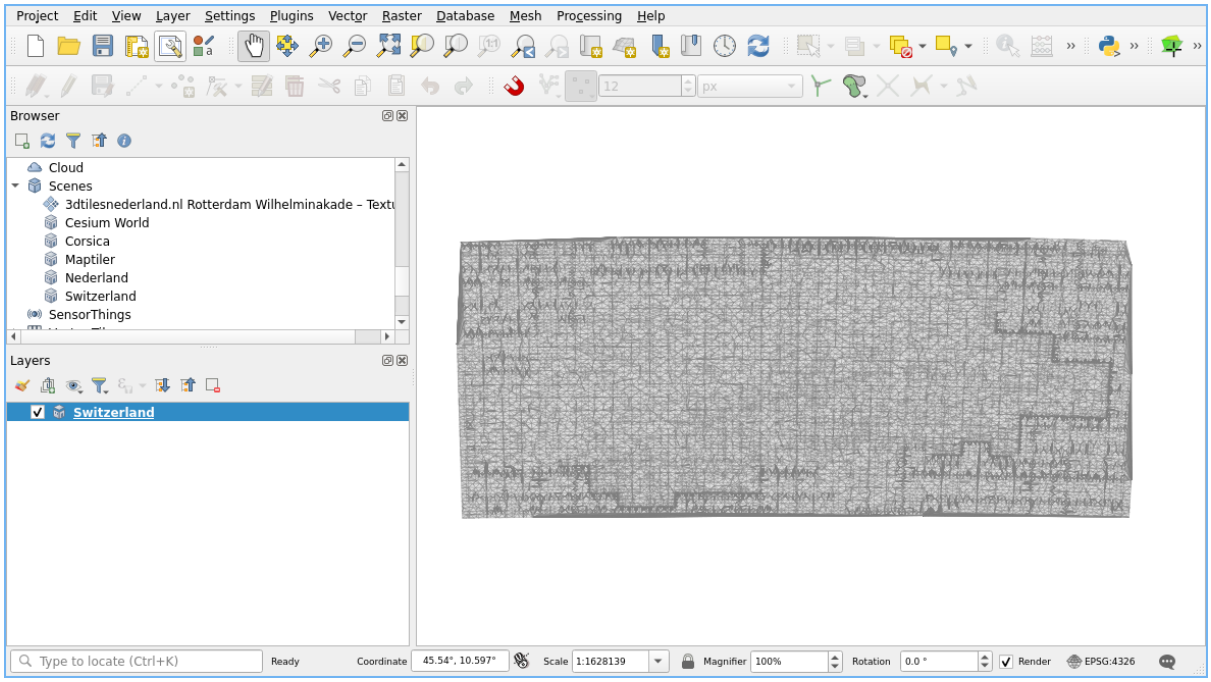


Fig. 11.24: Quantized Mesh layer

Using Cloud Connections

QGIS supports connections to cloud services like Alibaba Cloud OSS, Amazon S3, Google Cloud Storage, Microsoft Azure Blob Storage, Microsoft Azure Data Lake Storage, and OpenStack Swift Object Storage. You can load vector and raster data from these services into QGIS. Set up a new *Cloud* connection in the *Browser* panel by right-clicking on the *Cloud* entry and selecting *New Connection*. You will see a drop-down list of available cloud services. Select the service you want to connect to and fill in the required fields:

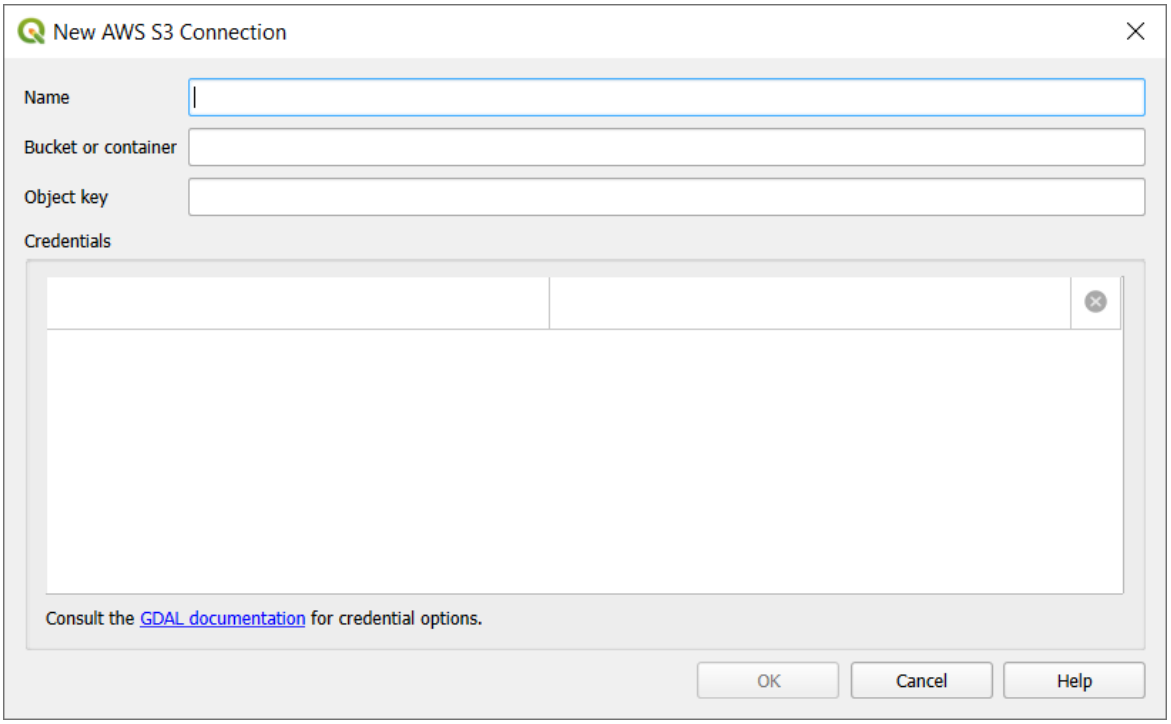


Fig. 11.25: Cloud Connection Dialog

- *Name*: A name for the connection.
- *Bucket or Container*: The name of the bucket or container in the cloud service.
- *Object Key* (optional): The key of the object in the bucket or container.
- *Credentials*: The credentials to access the cloud service.

You can also choose to *Save Connection* to an XML file or *Load Connection* from an XML file.

11.2 创建图层

可以通过多种方式创建图层，包括：

- 从头创建空图层
- 从现有图层创建图层
- 从剪贴板创建图层
- 图层作为基于一个或多个图层的类似 SQL 查询的结果 (虚拟图层)

QGIS 还提供了从/到不同格式导入/导出的工具。

11.2.1 新建矢量图层

QGIS 允许您创建不同格式的新图层。提供了用于创建地理包、形状文件、SpatiaLite、GPX 格式和临时图层 (也称为内存图层) 的工具。GRASS 插件支持创建新的 *GRASS* 图层 。

新建地理包图层

To create a new GeoPackage layer, press the *New GeoPackage Layer...* button in the *Layer* ► *Create Layer* ► menu or from the *Data Source Manager* toolbar. You can also create a new GeoPackage layer through the *Browser Panel* by selecting the *Create Database and Layer...*. The *New GeoPackage Layer* dialog will be displayed as shown in Fig. 11.26.

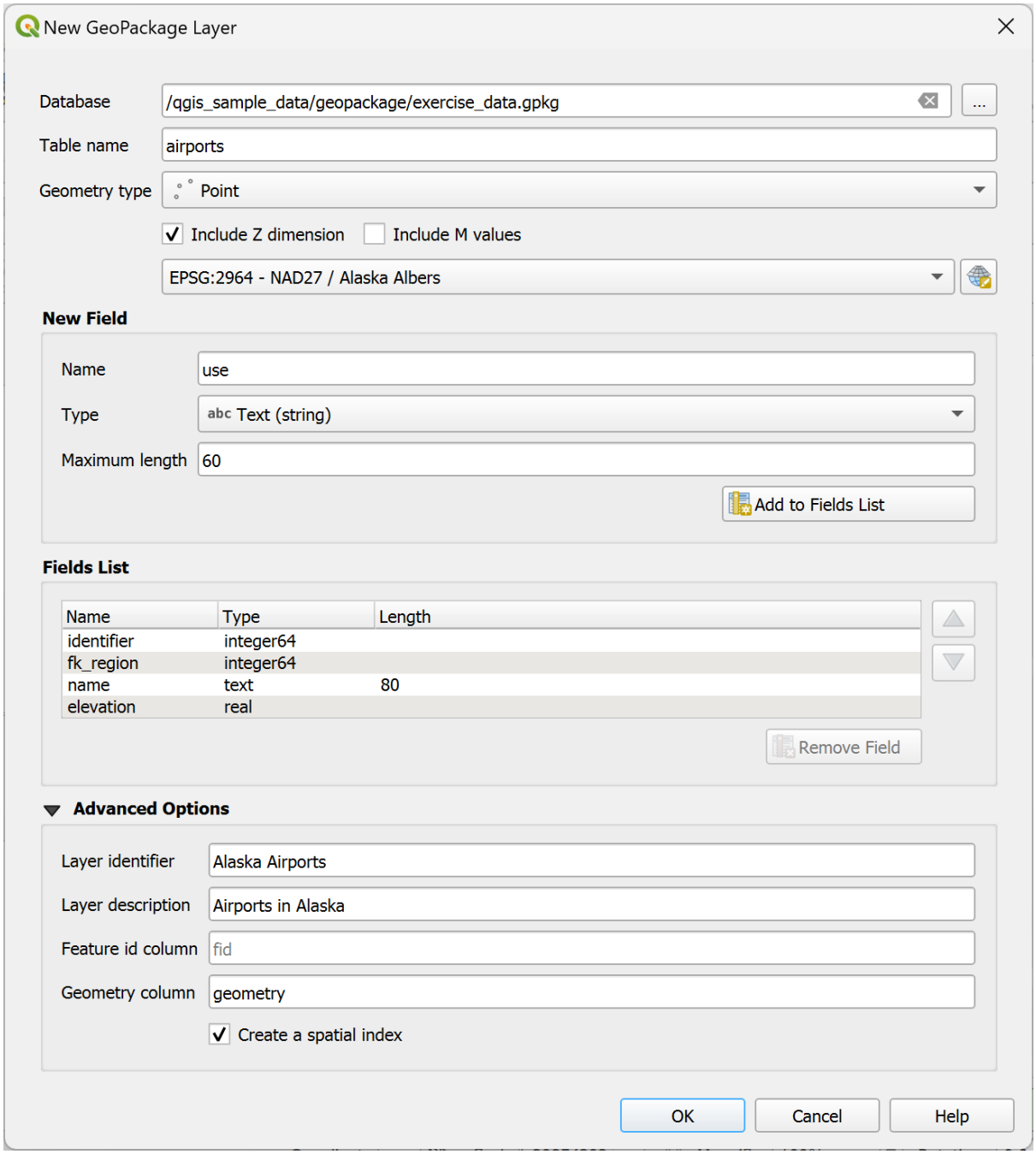


Fig. 11.26: 新建地理包图层对话框

1. 第一步是选择数据库文件的位置。可以通过点击 *数据库*字段右侧的 ... 按钮，然后选择现有或新建地理包文件。QGIS 将自动为您提供的名称添加正确的扩展名。
2. 为新图层/表命名 (*表名称*)
3. 定义 *几何图形类型*。如果不是无几何图形图层，可以指定是否应 *包括 Z 维度*和/或 *包括 M 值*。
4. 使用 按钮指定坐标参照系

要将字段添加到正在创建的图层，请执行以下操作：

1. 输入字段的 *名称*
2. 选择数据 *类型*。支持的类型有 *文本数据*, *非负整数* (整数和 64 位整数), *小数*, *日期*和 *日期和时间*, *二进制 (BLOB)* 和 *布尔型*。
3. 根据选中的数据格式，输入值的 *最大长度*。
4. 单击 *添加到字段列表*按钮
5. 为需要添加的每个字段重复上述步骤
6. You can later change the fields order using the Move Up and Move Down buttons
7. 设置满意的属性后，请单击 *确定*按钮。QGIS 将向图例添加新图层，您可以按照数字化现有图层 中的说明对其进行编辑。

默认情况下，在创建地理包图层时，QGIS 会生成一个名为 `fid` 的 *要素 ID 列*，该列充当该图层的主键。名称可以更改。几何图形字段 (如果可用) 被命名为 *几何图形*，您可以选择在其上 *创建空间索引*。这些选项与 *图层标识符* (图层的简短人类可读名称) 和 *图层描述*都位于 *高级选项*。

可以使用数据库管理器 对地理包图层进一步管理。

新建形状文件图层

要新建 ESRI 形状文件格式图层，请在 *图层* ► *创建图层* ► 菜单或从 *数据源管理器*工具栏点击 *新建形状文件图层...* 按钮。将显示 *新建形状文件图层*对话框，如 Fig. 11.27 所示。

1. 使用 *文件名*旁边的 ... 按钮选择路径和文件名。QGIS 将自动为您提供的名称添加正确的扩展名。
2. 接下来，选择数据的 *文件编码*
3. 选择图层的 *几何图形类型*：无几何图形 (生成 `.DBF` 格式文件)、点、点集、线或多边形
4. 指定几何图形是否应具有附加尺寸: *无*, *Z (+ M 值)* 或 *M 值*
5. 使用 按钮指定坐标参照系

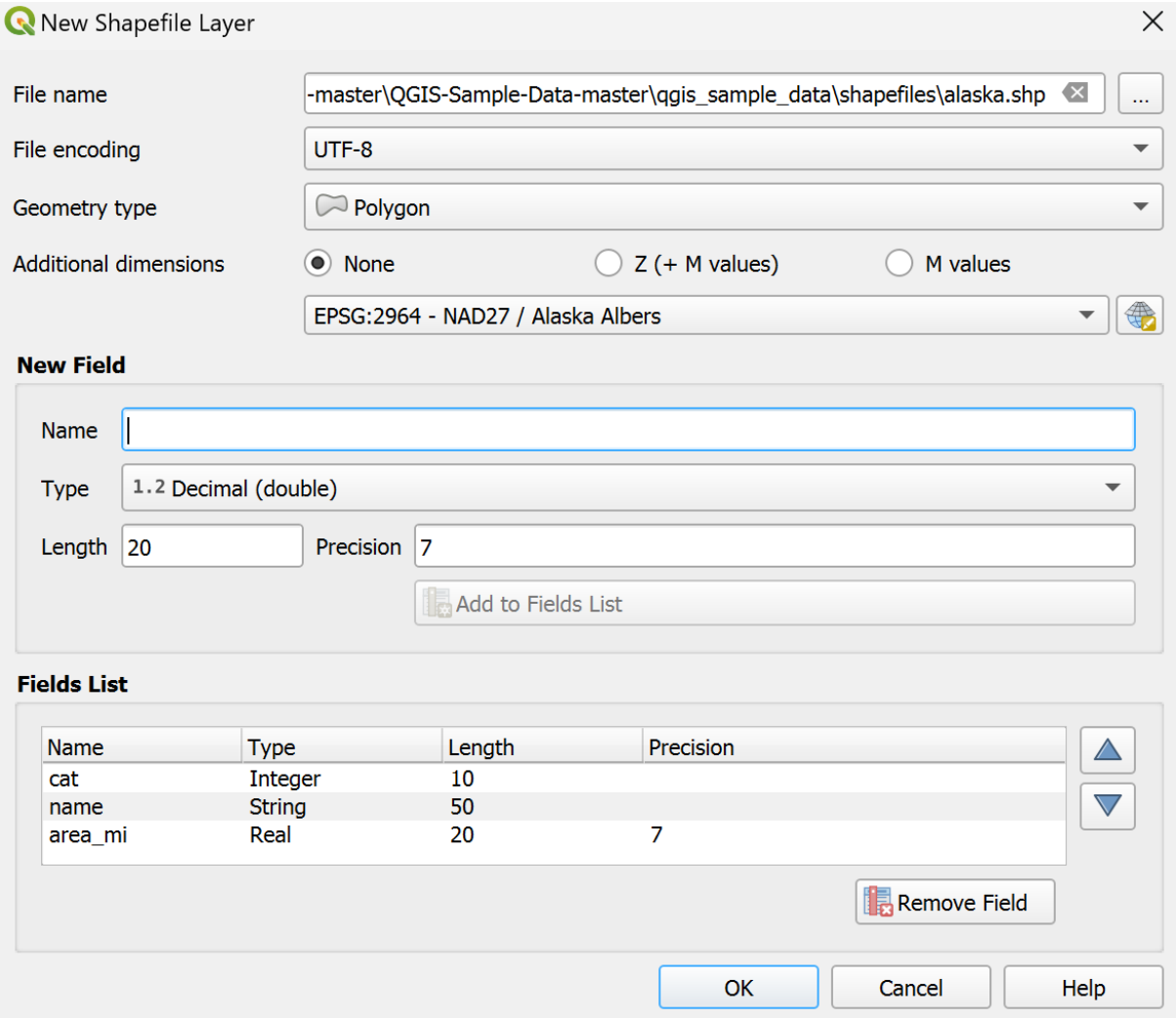


Fig. 11.27: 新建形状文件图层对话框

要将字段添加到正在创建的图层，请执行以下操作：

1. 输入字段的 *名称*
2. Select the data *Type*. Only *Decimal number*, *Whole number*, *Text data*, *Date*, and *Boolean* data types attributes are supported.
3. 根据选中的数据格式，输入 *长度* 和 *精度*。
4. 单击 添加到字段列表按钮
5. 为需要添加的每个字段重复上述步骤
6. You can later change the fields order using the Move Up and Move Down buttons
7. 设置满意的属性后，请单击 *确定* 按钮。QGIS 将向图例添加新图层，您可以按照数字化现有图层 中的说明对其进行编辑。

默认将添加第一个整数 `id` 列，但可以将其移除。

新建 SpatiaLite 图层

要新建 SpatiaLite 图层，请在 图层 ► 创建图层 ► 菜单或从 数据源管理器工具栏点击 新建 *SpatiaLite* 图层... 按钮。将显示 新建 *SpatiaLite* 图层对话框，如 Fig. 11.28 所示。

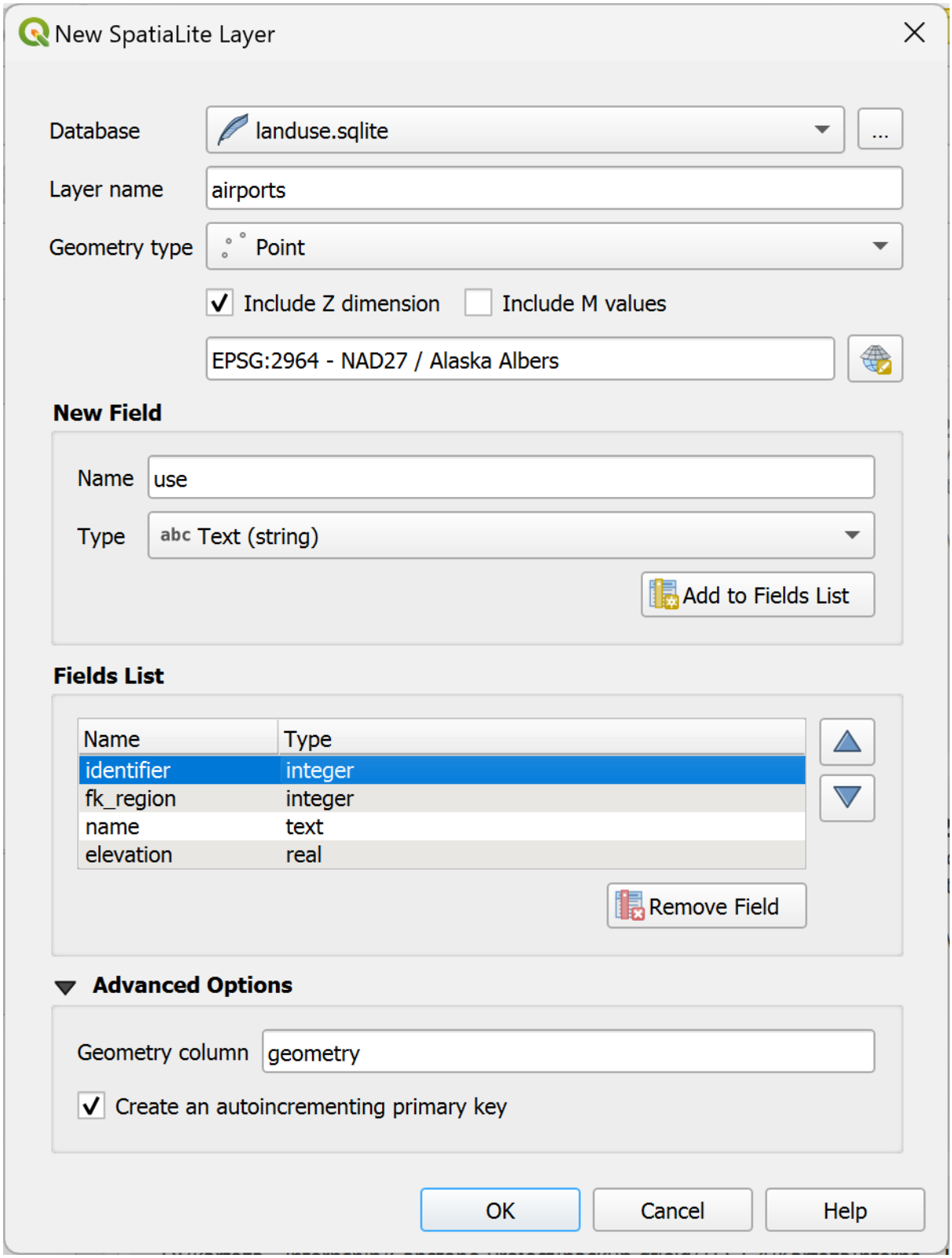


Fig. 11.28: 新建 SpatiaLite 图层对话框

1. 第一步是选择数据库文件的位置。可以通过点击 *数据库*字段右侧的 ... 按钮，然后选择现有或新建 SpatiaLite 文件。QGIS 将自动为您提供的名称添加正确的扩展名。
2. 为新建图层提供名称 (*图层名*)
3. 定义 *几何图形类型*。如果不是无几何图形图层，可以指定是否应 *包括 Z 维度*和/或 *包括 M 值*。
4. 使用 按钮指定坐标参照系。

要将字段添加到正在创建的图层，请执行以下操作：

1. 输入字段的 *名称*
2. 选择数据 *类型*。支持的类型有 *文本数据*, *整数*, *小数*, *日期*和 *日期时间*。
3. 单击 *添加到字段列表*按钮
4. 为需要添加的每个字段重复上述步骤
5. You can later change the fields order using the Move Up and Move Down buttons
6. 设置满意的属性后，请单击 *确定*按钮。QGIS 将向图例添加新图层，您可以按照*数字化现有图层* 中的说明对其进行编辑。

If desired, you can select *Create an autoincrementing primary key* under the *Advanced Options* section. You can also rename the *Geometry column* (`geometry` by default).

可以使用*数据库管理器* 对 SpatiaLite 图层进一步管理。

新建网孔图层

要新建网孔图层，请在 *图层* ► *创建图层* ► 菜单或从 *数据源管理器*工具栏点击 *新建网孔图层...* 按钮。将显示 *新建网孔图层*对话框，如 Fig. 11.29 所示。

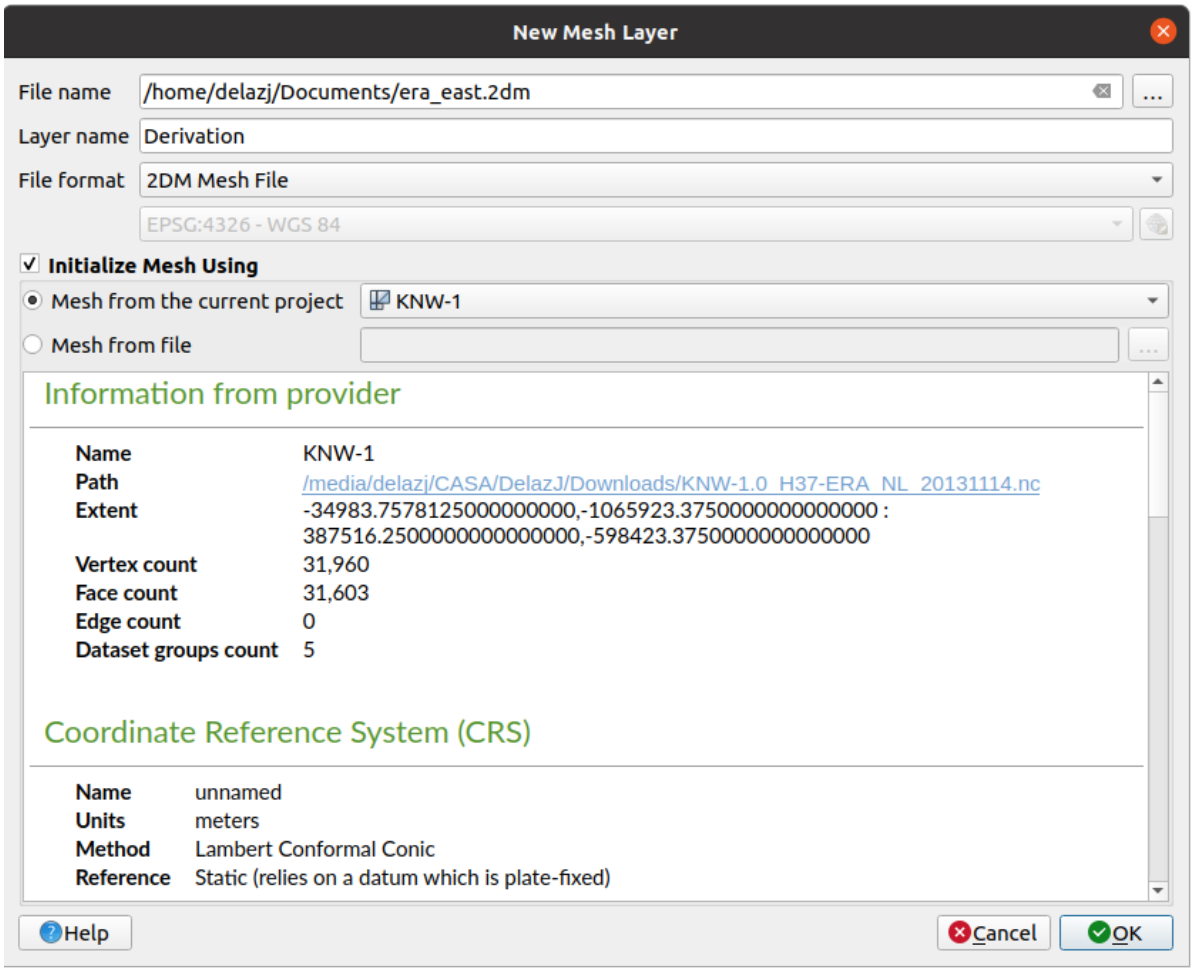


Fig. 11.29: 新建网孔图层对话框

1. 第一步是选择网孔文件的位置。可以通过点击 *文件名*字段右侧的 ... 按钮，然后选择现有或新建网孔文件。
2. 提供名称 (*文件名*), 即图层在 *图层*面板显示的名称
3. 选择 *文件格式*: 目前支持的网孔文件格式有 `2DM 网孔文件 (*.2dm)`, `Selafin 文件 (*.slf)` 和 `UGRID (*.nc)`。
4. 选择指定给数据集的*坐标参照系*
5. 上述步骤将生成一个空图层，随后可以数字化顶点并向其添加数据集组。但是，也可以使用现有网孔图层初始化该图层，即使用其他网孔图层的顶点或面填充新图层。为此：

   1. 勾选 *使用如下数据初始化网孔*
   2. 然后选择 *来自当前工程的网孔*或 *来自文件的网孔*。将显示选中网孔文件的信息以进行检查。

   请注意，只有网孔图层的框架被转移到新图层，不会复制其数据集。

新建 GPX 图层

要创建新的 GPX 文件，请执行以下操作：

1. Select *Create Layer* ► *New GPX Layer...* from the *Layer* menu.
2. 在对话框中，选择保存新文件的位置，为其命名并点击 *保存*按钮。
3. 图层面板添加了三个新图层：
   - 用于数字化位置 (`waypoints`) 的点图层，其字段存储名称、高程、注释、描述、源、URL 和 URL 名称
   - 用于数字化构成计划路径 (`routes`) 的位置序列的线图层，其字段存储名称、符号、数字、注释、描述、源、URL、URL 名称
   - 用于跟踪接收器随时间的移动 (`tracks`) 的线图层，其中字段存储名称、符号、数字、注释、描述、源、URL、URL 名称
4. 现在，可以按照数字化现有图层 所述编辑其中任一图层。

新建临时图层

临时图层是内存中的图层，其数据不会保存在磁盘上，并且 QGIS 关闭时将被丢弃。可用于存储临时需要的要素，或者在地理处理操作期间作为中间图层。

要新建临时图层，请在 *图层* ► *创建图层* ► 菜单或在 *数据源管理器*工具栏选择 *新建临时图层...* 按钮。将显示 *新建临时图层*对话框，如 Fig. 11.30 所示。然后：

1. 提供 *图层名*
2. 选择 *几何图形类型*。在此您可以创建：
   - 没有几何图形类型图层，作为简单表
   - 点或 点集图层
   - 线串 / 复合曲线或 线串集 / 曲线集图层
   - 多边形 / 曲线多边形或 多边形集 / 表面集图层
3. 对于几何类型，指定数据集的维度：检查是否应 *包括 Z 维度*和/或 *包括 M 值*
4. 使用 按钮指定坐标参照系。
5. 将字段添加到图层。请注意，与许多其他格式不同，可以在不使用任何字段的情况下创建临时图层。因此，此步骤是可选的。
   1. 输入字段的 *名称*
   2. 选择数据 *类型*: *文本*, *整数*, *小数*, *布尔*, *日期*, *时间*, *日期和时间*和 *二进制 (BLOB)*
   3. 根据选中的数据格式，输入 *长度*和 *精度*
   4. 单击 *添加到字段列表*按钮
   5. 对需要添加的每个字段重复上述步骤
   6. You can later change the fields order using the Move Up and Move Down buttons
6. 对设置感到满意后，请单击 *确定*。QGIS 将新图层添加到 *图层*面板，您可以按照数字化现有图层 中的说明对其进行编辑。

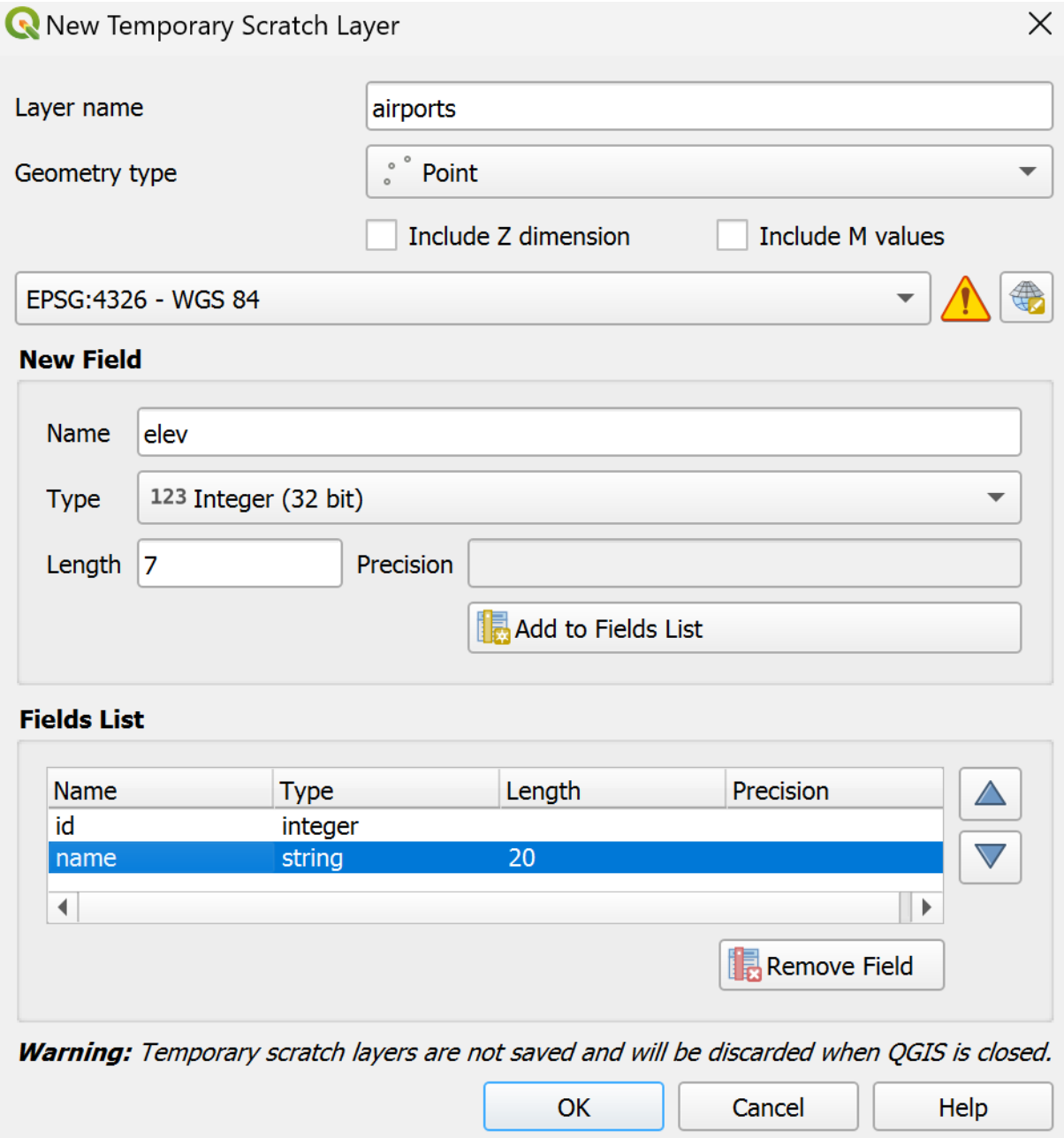


Fig. 11.30: 新建临时图层对话框

也可以使用例如剪贴板 (请参阅从剪贴板创建新图层) 或作为处理算法 的结果创建预填充的临时图层。

Tip

将内存图层永久储存在磁盘

为了避免在关闭带有临时图层的工程时丢失数据，可以将这些图层保存为 QGIS 支持的矢量格式：

- 单击图层旁边的 指示器图标；
- 在图层上下文菜单中选择 永久化条目；
- 使用上下文菜单中的 导出 ► 条目或 图层 ► 另存为... 菜单。

上述每条命令都会打开在从现有图层创建新图层 章节所描述的 矢量图层另存为对话框，并且保存的

文件将替换 图层面板中的临时文件。

11.2.2 从现有图层创建新图层

Layers (raster, vector and point cloud) can be saved in a different format and/or reprojected to a different coordinate reference system (CRS) using the *Layer* ► *Save As...* menu or right-clicking on the layer in the *Layers panel* and selecting:

- *Export* ► *Save As...* for raster and point cloud layers
- 导出 ► 要素另存为... 或 导出 ► 选中的要素另存为... 用于矢量图层。
- Drag and drop the layer from the layer tree to the PostgreSQL entry in the *Browser Panel*. Note that you must have a PostgreSQL connection in the *Browser Panel*.

常用参数

保存图层为... 对话框显示了几个参数，用于在保存图层时更改行为。栅格和矢量常见参数包括：

- 文件名: 文件在磁盘的位置。可以是输出图层或存储该图层的容器 (例如类似数据库格式，如地理包、SpatiaLite 或开放文档电子表格)。
- *CRS*: 可以更改以重投影数据
- 范围: 使用*extent_selector* 控件限制要导出的输入范围
- 将已保存的文件添加到地图中: 将新图层添加到画布

However, some parameters are specific to certain formats:

栅格特定参数

根据导出的格式，其中某些选项可能不可用：

- 输出模式 (可以是 **原始数据**或 **渲染图像**)
- 格式: 导出为 GDAL 可以写入的栅格格式，如 GeoTiff、地理包、MBTile、地理空间 PDF、SAGA GIS 二进制网格、Intergraph 栅格、ESRI .hdr Labeled 等
- 分辨率
- 创建选项: 生成文件时使用高级选项 (文件压缩、块大小、色度等)，可以与输出格式有关的从预定义创建配置文件 或设置每个参数。
- 金字塔创建
- 如果您勾选了 创建 *VRT* ，请设置 *VRT* 瓦片
- 无数据值

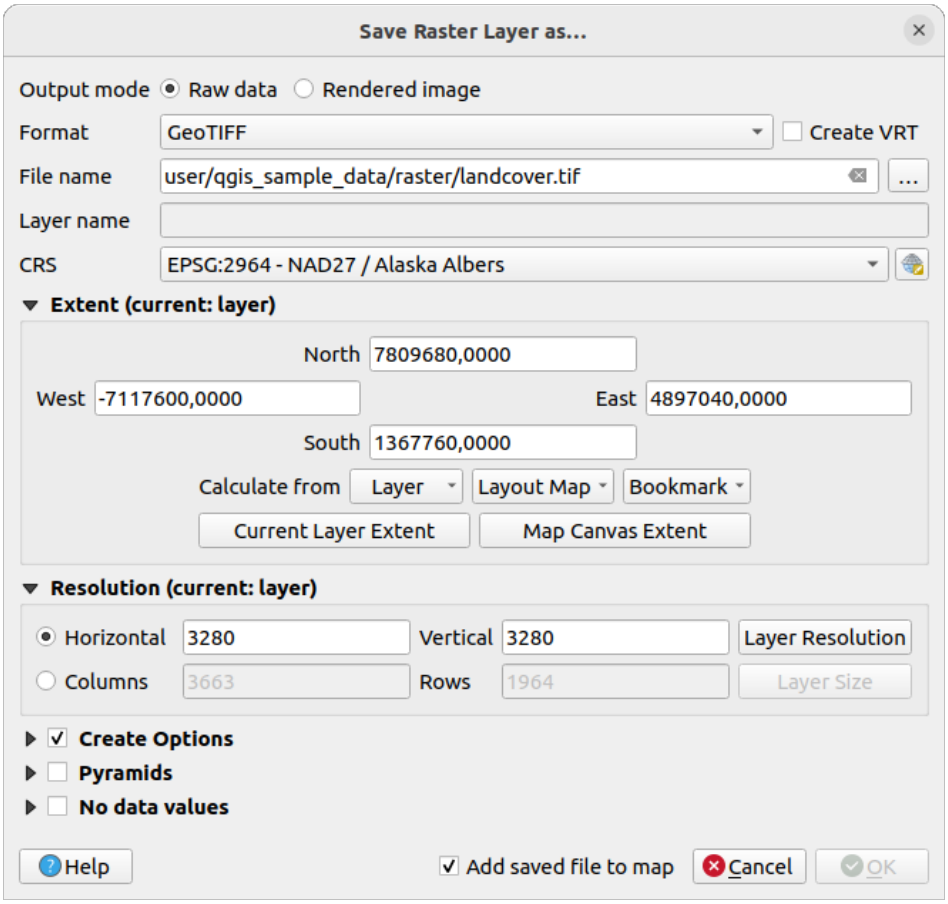


Fig. 11.31: 另存为新栅格图层

矢量特定参数

根据导出的格式，可能会提供以下一些选项：

- *格式*: 导出到 GDAL 可以写入的矢量格式，如地理包、GML、ESRI 形状文件、AutoCAD DXF、ESRI FileGDB、Mapinfo TAB 或 MIF、SpatiaLite、CSV、KML、ODS 等。
- *图层名*: 当 *文件名*涉及类似容器格式时可用，此条目表示输出图层。
- *编码*
- *仅保存选中的要素*
- *选择要导出的字段以及导出选项*: 提供了导出具有自定义名称和表单控件 设置字段的方法：
  - 勾选 *名称*列下面的行以选择要保留在输出图层中的字段，或点击 *全选*或 *取消全选*按钮
  - 切换 *为导出名称使用别名*复选框，以使用相应的字段别名填充 *导出名称*列，或重置为原始字段名称。双击单元格也可以编辑名称。
  - 根据是否使用属性表单自定义控件，可以 *以显示值取代所有选中的原始字段值*。例如，如果将 *值映射*控件应用于字段，则输出图层将包含描述值，而不是原始值。也可以在 *以显示值取代*之列中逐个字段完成替换。
- *保留图层元数据*: 确保复制并存储源图层中存在的所有图层元数据 ：
  - 在新创建的图层中，如果输出为地理包格式。
  - 作为 `.qmd` 文件以及输出图层，用于其他格式。请注意，支持多个数据集的基于文件的格式 (例如 SpatiaLite、DXF 等) 可能会出现意外行为。

- *导出符号*: 主要用于 DXF 导出，也可用于管理 OGR 要素样式 (请参阅如下备注) 的所有文档格式，如 DXF、KML，TAB 文件格式：
  - **无符号**: 读取数据所应用的默认样式
  - **要素符号**: 使用 OGR 要素样式保存样式 (请参阅如下备注)
  - **符号图层符号化**: 使用 OGR 要素样式保存 (请参阅如下备注)，但如果使用了多个符号化符号图层，则多次导出相同的几何图形
  - **比例**值可以应用于最新选项

Note

*OGR 要素样式*是将样式作为隐藏属性直接储存在数据中的方法。只有某些格式可以处理此类信息。KML、DXF 和 TAB 文件格式就是这样的格式。有关高级详细信息，请阅读 OGR 要素样式规范 文档。

- *几何图形*: 可以配置输出图层的几何图形功能
  - *几何图形类型*: 当设置为 **自动**时，保留要素的原始几何图形，否则将使用其它类型移除或覆盖。可以将空几何图形列添加到属性表，并移除空间图层的几何图形列。
  - *强制为类型集*: 强制在图层中创建几何图形集要素。
  - *包括 Z-维度至几何图形*。

Tip

通过覆盖图层几何图形类型，可以将无几何图形表 (例如 `.csv` 文件) 保存到具有任何类型几何图形 (点、线、多边形) 的形状文件中，以便使用 添加部件 工具手动将几何图形添加到行。

- *数据源选项*, *图层选项*或 *自定义选项*允许根据输出格式配置高级参数。有些在*探索数据格式与字段*中进行了描述，但有关详细信息，请参阅 GDAL 驱动文档。每个文件格式都有自己的自定义参数，例如，对于 `GeoJSON` 格式，请查看 GDAL GeoJSON 文档。

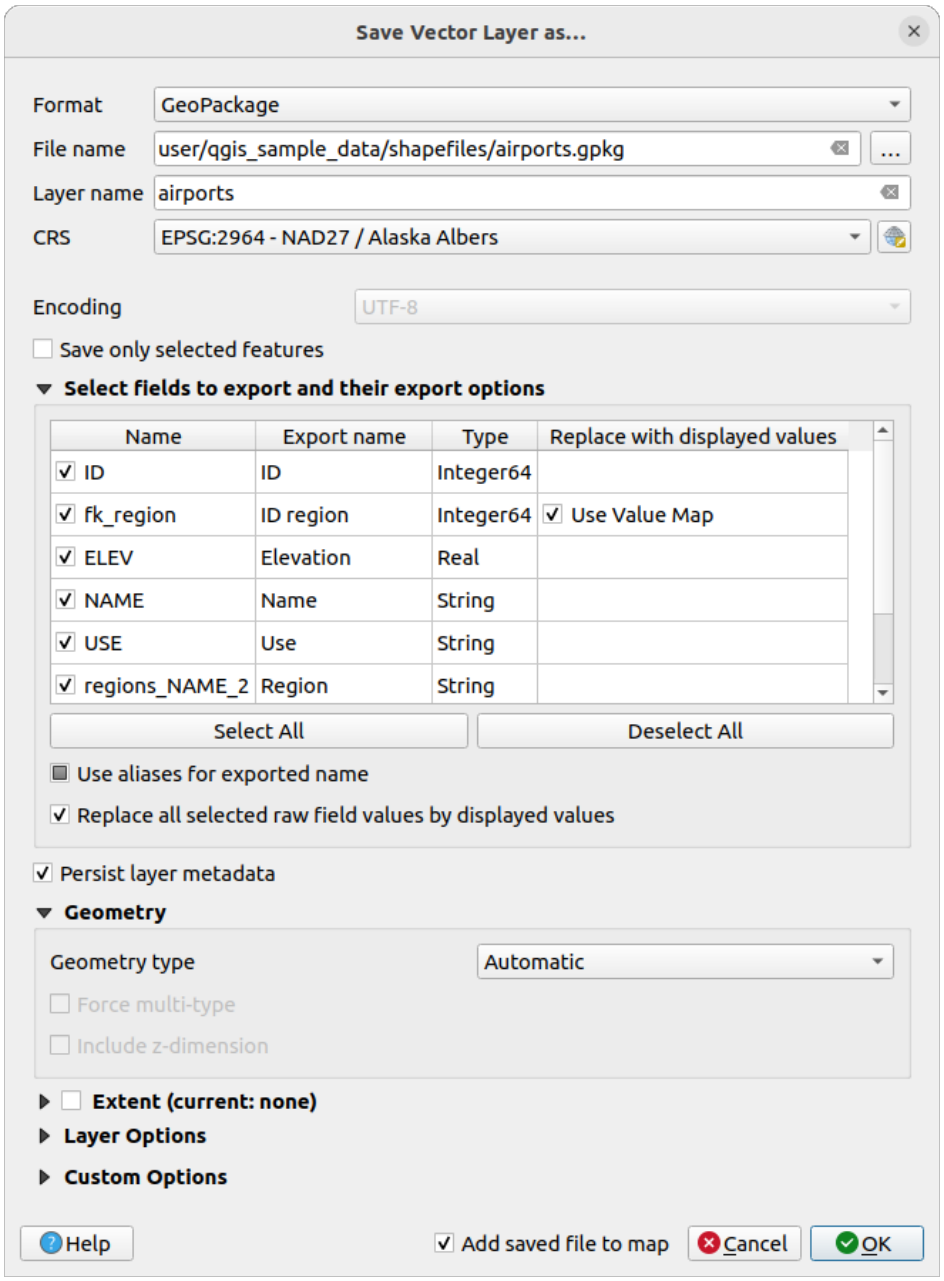


Fig. 11.32: 另存为新矢量图层

将矢量图层保存到现有文件中时，根据输出格式 (地理包、SpatiaLite、FileGDB 等) 的功能，用户可以决定是否：

- 盖写整个文件
- 仅盖写目标图层 (图层名称可配置)
- 将要素追加到现有目标图层
- 追加要素，添加新字段 (如果有的话)

对于 ESRI 形状文件、MapInfo .tab 等格式，也可以使用要素追加。

Point Cloud specific parameters

Similar to raster and vector layers, point cloud layers can be saved in a different format and/or reprojected to a different coordinate reference system (CRS). This allows you to export a point cloud layer to vector or point cloud formats. Current supported formats are: Temporary scratch (memory layer), GeoPackage, ESRI Shapefile, DXF and LAS/LAZ point cloud. In addition to the common parameters listed above, exporting point cloud layers includes the following options:

- *Filter by Polygon Layer*: Allows you to filter the point cloud data based on a polygon layer.
- *Elevation Range*: Enables filtering of the point cloud data based on a specified Z range.
- *Limit number of points*: Provides an option to limit the number of points exported from the point cloud layer.

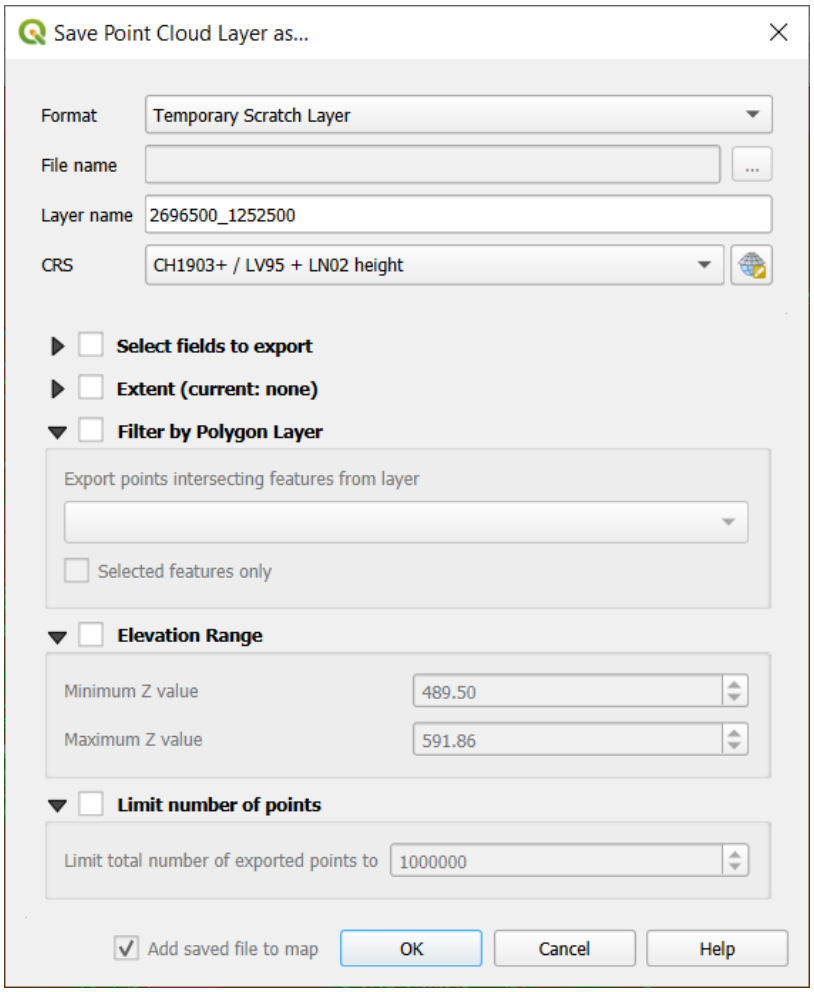


Fig. 11.33: Saving a point cloud layer as a new layer

11.2.3 新建 DXF 文件

除了 *另存为...* 对话框提供将单图层导出为其他格式 (`*.DXF`) 的选项外，QGIS 还提供了另一个将多个图层导出为单个 DXF 图层的工具。可以在 *工程* ► *导入/导出* ► *导出工程为 DXF 文件...* 菜单进行访问。

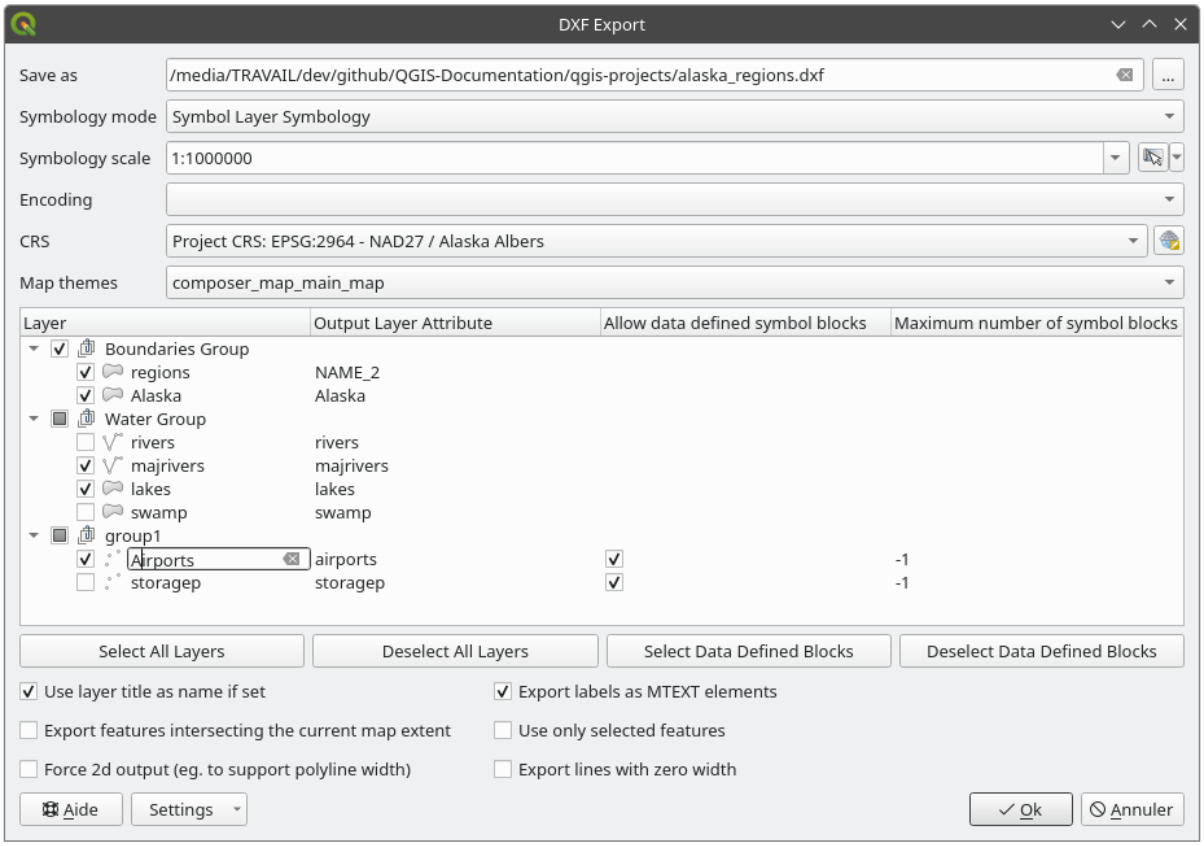


Fig. 11.34: 将工程导出到 DXF 对话框

在 *DXF 导出*对话框:

1. 提供目标文件。
2. 如果适用，请选择符号模式和比例 (请参阅*OGR 要素样式* 备注)。
3. 选择数据 *编码*。
4. 选择要应用的 *CRS*: 所选图层将被重投影到给定 CRS。
5. 选择要包含在 DXF 文件中的图层，方法是在表控件检索图层，或从现有*地图主题* 中自动选取图层。*全选*和 *取消全选*按钮有助于快速设置要导出的数据。

   For each layer, you can:

   - Override the output layer name without altering the original project layer. For this, click on the *Layer* name in the dialog and write the output name to use.
   - *Output layer attribute*: Choose whether to export all the features in a single DXF layer or rely on a field whose values are used to split the features into layers in the DXF output. In the latter case, each layer will take its name from the corresponding field value.
   - *Allow data defined symbol blocks*:
   - *Maximum number of symbol blocks*: creates symbol blocks up to the specified limit, starting with the ones containing the highest number of references. The other symbols are written as they are. `-1` means no limitation.

也可以选择：

- ☐ *Use the layer title as name if set* instead of the layer name itself: the title is taken from the *metadata* or *server* properties of the layer;
- ☐ *Export features intersecting the current map extent*;
- ☐ 强制 *2D* 输出 *(*例如*:* 为支持折线宽度*)*
- ☑ *Export label as MTEXT elements* or TEXT elements;
- ☐ *Use only selected features*;
- ☐ *Export lines with zero width*: all the lines are exported with minimal width `0` (hairline) if enabled. This helps keep the lines minimal in the file regardless of the zoom level, and can be handy for doing further CAD-editing with the exported dxf, especially if there are many features next to each other on the map.

Note

The precedence for defining the output layer name is as follows:

1. The field value from *Output layer attribute*
2. The overridden name in *Layer* column
3. The *Use the layer title as name if set* option
4. The layer name

Current settings defined in the *DXF Export* dialog may be stored in an XML file for reusing them in other sessions. For this, the *Settings* combo box has two options: *Load Settings from File...* and *Save Settings to File...*.

11.2.4 从剪贴板创建新图层

剪贴板的要素可以粘贴到新图层。为此，请选择一些要素，复制到剪贴板，然后使用 编辑 ► 粘贴要素为 ► 将其粘贴到新图层，并选择：

- 新建矢量图层...: 显示 矢量图层另存为... 对话框 (有关参数请参阅从现有图层创建新图层)
- 或 临时图层...: 需要为图层提供名称

将创建新图层，填充选中的要素及其属性 (并将其添加到地图画布)。

Note

对于在 QGIS 中选择和复制的要素以及来自其他应用程序的要素，只要它们的几何图形是使用熟知文本 (WKT) 定义的，就可以从剪贴板创建图层。

11.2.5 Creating SQL Query Layers

Beside loading an entire layer in a project or creating new layers from scratch or pasted features, you can also load layers generated on the fly from other layer(s). They are the result of a more or less advanced filter using SQL language, applied to stored layers regardless of their data provider or their availability in the active project. This means that temporary layers are not compatible with this feature. Depending on the provider, one or more layers can be used to write the query. The generated layer remains dependent on the layer(s) involved in the query and is loaded with the Filter icon next to it.

This feature is accessible:

- From the *Browser* panel, right-click on a supported data (plain layer, database connection, schema or table) and select *Execute SQL...* entry in the contextual menu.
- From the *Layers* panel, select a loaded layer, right-click and select *Execute SQL...* entry in the contextual menu.

This opens a window with a central text box widget where you can write SQL queries.

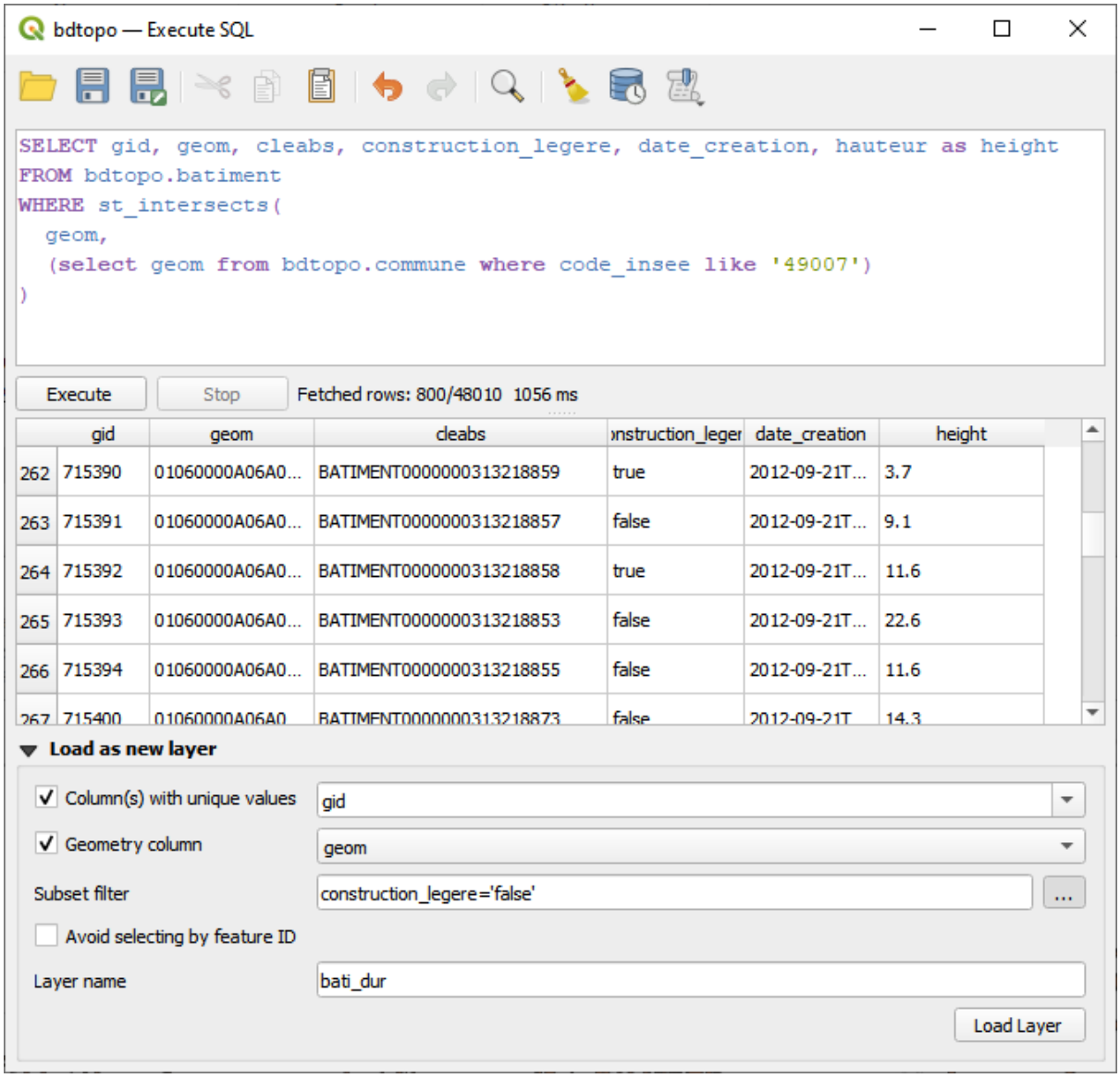


Fig. 11.35: Executing SQL queries in the Execute SQL window

At the top of the dialog, a toolbar provides a set of tools to create, store and manipulate your queries:

- Open Queries...: fills the text editor widget with contents from an existing `.sql` file
- Save Queries... and Save Queries as... help you store the written query to a `.sql` file
- Statements of the query can be adjusted using the Cut, Copy and Paste buttons. Likewise, you can Undo or Redo your changes.
- The Find & Replace enables, at the bottom of the dialog, a widget allowing to look for a particular string in your SQL code. The search can be case sensitive, affect partial or whole word, rely on a regular expression. It is then possible to navigate through the found strings, replacing them one by one or all in a row.

- Use Clear SQL Editor to wipe the text editor.
- The History button opens a dialog storing previously run queries. More at *Query History*.
- As previously mentioned, queries can be saved as an `.sql` file stored on disk. Using the Store Current Query button, they can also be stored:
  - In the active *User Profile*, in the associated `QGIS3.ini` file, thus accessible in subsequent projects
  - or as part of the *Current Project*.

  Clicking an entry from the stored queries drop-down menu inserts that query in the expression being written. The stored entry can also be deleted, from the drop-down menu.

Running and loading queries as layer

In the central part of the *Execute SQL* dialog, you build your query using the SQL syntax supported by the underlying provider (e.g., OGR, GeoPackage, PostgreSQL).

By default, if opened from a layer entry, a sample SQL query is provided. Editing tools to select, cut, copy, paste, undo and redo are available as well from the contextual menu.

> **Tip**
>
> **Finding the right SQL syntax for your dataset**
>
> To find the appropriate SQL dialect for your dataset, press the ... button next to the *Subset filter* option in the bottom part of the dialog.

When ready, pressing the *Execute* button below the text area will run the query. It is possible to highlight a portion of the SQL to only execute that portion when pressing `Ctrl+R` or clicking the *Execute selection* button. Use the *Stop* button to abort the execution.

A successful query execution will display a table at the bottom of the dialog with returned features. You can select specific cells in the result set. Use the `Ctrl+C` shortcut to copy the selected cells to the clipboard. The copied data is available as a formatted table. This allows you to paste the data into other applications, such as spreadsheet where it will show up as a table.

The returned table can be loaded in QGIS expanding the *Load as new layer* group and configuring parameters (their availability depends on the layer provider):

- *Column(s) with unique values* to indicate primary key of the data,
- *Geometry column*: check the box to load the layer as a spatial one, and indicate the geometry field name
- *Subset filter*: allows to filter the results using a `WHERE` clause. It can be written in the text box, or built using the *Query builder* after you pressed .... Make sure to use fields that are available in the SQL layer.
- *Avoid selecting by feature ID*
- *Layer name* in the project

At any time, in the *Layers panel*, you can adjust the output layer by right-clicking and selecting *Update SQL expression...*. The *Update SQL* dialog opens, prefilled with the applied query that you can edit as you wish. Once ready, press *Update layer* and the layer will be modified in-place.

Query History

The *Query History* dialog displays all the previously run queries, sorted by date and data provider type. It is accessible from the *Database* ► *Query History*··· menu, or pressing the History button from the *Execute SQL* dialog.

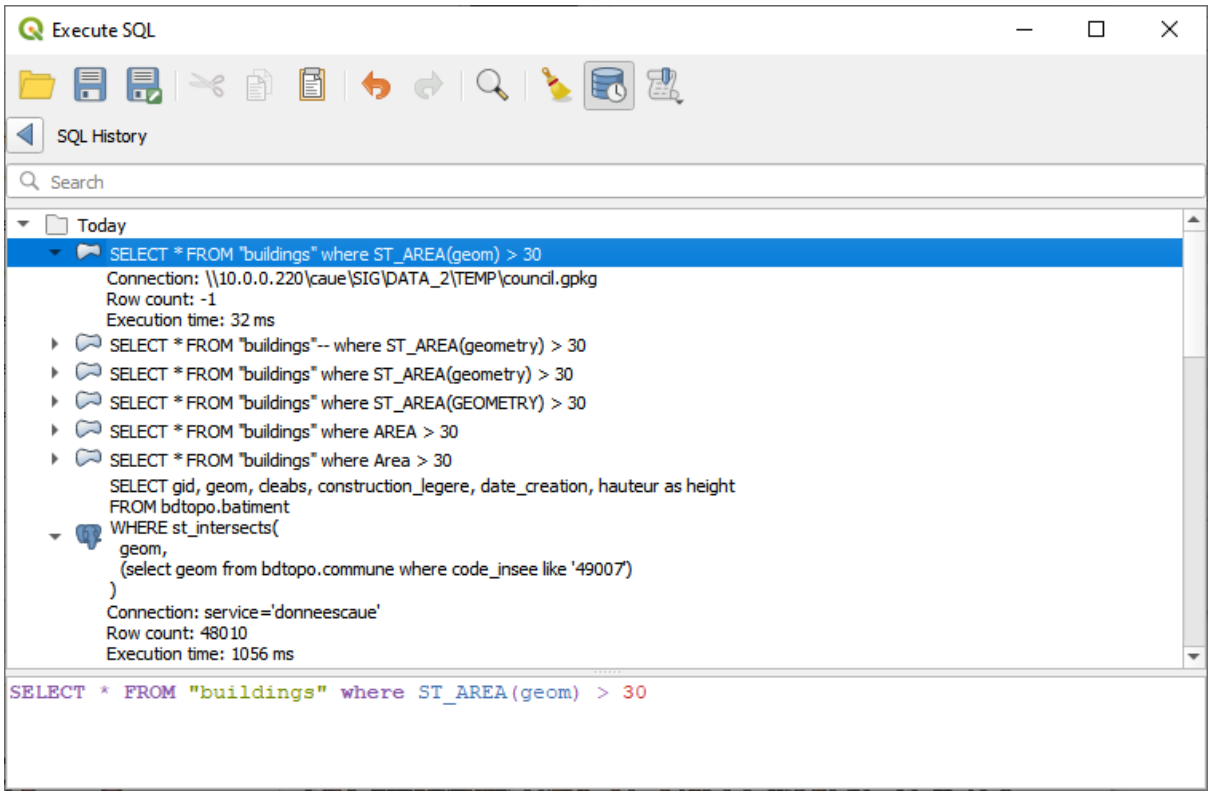


Fig. 11.36: History of the executed SQL queries

At the top of the dialog, a search box can be used to filter out the queries. The search is done through every strings displayed in the query history. If the dialog is opened from within the *Execute SQL* dialog, the search box is preceded by the *Execute SQL toolbar*.

Below, collapsible and sorted date entries list their related queries. Each query is identifiable by its SQL command, and when expanded, reveals:

- The *Connection*: the source of the data the query is based on
- The *Row count*: number of returned features
- The *Execution time*

Hover over an entry and a tooltip displays the full query. Likewise, selecting an entry also displayed it in the bottom part of the dialog. You can interact with the text, copying all or part of it.

Double-click the SQL command entry and the *Execute SQL* dialog opens, filled with the clicked query. Any query being written is overwritten. Right-click the SQL command entry and you can either:

- *Load SQL Command*···, loads the target command into the *Execute SQL* dialog, replacing any existing query. It is the same as double clicking the entry.
- *Copy SQL Command* and paste it wherever you want.

11.2.6 创建虚拟图层

虚拟图层是一种特殊的矢量图层类型。允许将图层定义为涉及 QGIS 可以打开许多其他矢量图层的 SQL 查询的结果。虚拟图层本身不携带数据，可以看作视图。

要创建虚拟图层，请通过以下方式打开虚拟图层创建对话框：

- 在 *图层* ► *添加图层* ► 菜单中选择 *添加/编辑虚拟图层*
- 在 *数据源管理器*对话框中启用 *添加虚拟图层*选项卡；
- 或使用 *数据库管理器*对话框树。

该对话框允许您指定 *图层名*和 SQL *查询*。查询可以使用加载的矢量图层的名称 (或 ID) 作为表，也可以将其字段名称用作列。

例如，如果您有一个名为 `airports` 的图层，可以使用如下 SQL 查询新建名为 `public_airports` 的虚拟图层：

```
SELECT *
FROM airports
WHERE USE = "Civilian/Public"
```

SQL 查询将被执行，而与 `airports` 图层的底层提供程序无关，即使该提供程序不直接支持 SQL 查询也是如此。

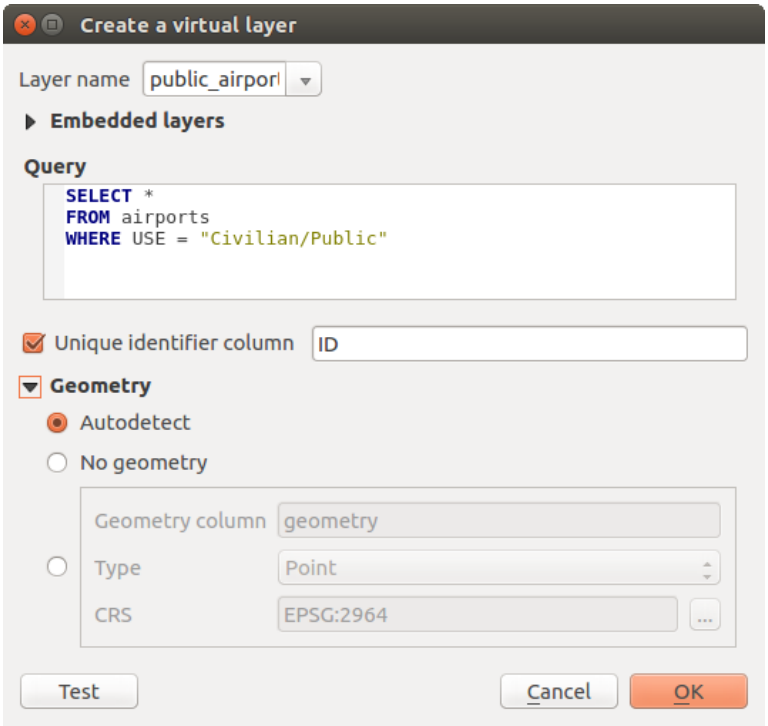


Fig. 11.37: 创建虚拟图层对话框

还可以创建连接和复杂查询，例如，连接 airports 和 country 信息：

```
SELECT airports.*, country.population
FROM airports
JOIN country
ON airports.country = country.name
```

> **Note**
>
> 也可以使用*DB Manager Plugin* 的 SQL 窗口创建虚拟图层。

嵌入用于查询的图层

除了地图画布中可用的矢量图层外，用户还可以将图层添加到 *嵌入式图层*列表，该列表可以在查询中使用，而无需将其显示在地图画布或图层面板。

若要嵌入图层，请单击 *添加*，然后提供 *本地名称*, *提供程序*, *编码*和 *源路径*。

*导入*按钮允许将地图画布中的图层添加到嵌入式图层列表。然后可以从图层面板移除这些图层，而不会中断现有查询。

支持的查询语言

底层引擎使用 SQLite 和 SpatiaLite 进行操作。

这意味着可以使用您本地安装的 SQLite 所理解的所有 SQL。

SQLite 中的函数和 SpatiaLite 中的空间函数也可以用于虚拟图层查询。例如，可以使用类似于以下内容的查询仅从属性图层创建点图层：

```
SELECT id, MakePoint(x, y, 4326) as geometry
FROM coordinates
```

Functions of QGIS expressions can also be used in a virtual layer query.

要引用图层的几何图形列，请使用名称 `geometry` 。

与纯粹的 SQL 查询相反，必须命名虚拟图层查询的所有字段。如果列是计算或函数调用的结果，请不要忘记使用 `as` 关键字命名。

性能问题

使用默认参数时，虚拟图层引擎将尝试检测查询的不同列的类型，包括几何图形列的类型 (如果存在)。

这是通过尽可能使用内省查询或作为最后手段获取查询的第一行 (LIMIT 1) 来完成的。出于性能原因，仅仅为了创建图层而获取结果的第一行或许是不可取的。

创建对话框参数：

- *唯一标识符列*: 指定查询的字段，该字段表示 QGIS 可用作行标识符的唯一整数值。默认使用自动递增的整数值。定义唯一标识符列可以加快按 ID 选择行的速度。
- *无几何图形*: 强制虚拟图层忽略任何几何图形字段。生成的图层为仅包含属性的图层。
- *几何图形 列*: 指定几何图形列的名称。
- *几何图形 类型*: 指定几何图形的类型。
- *几何图形 CRS*: 指定虚拟图层的坐标参照系。

特殊注释

虚拟图层引擎尝试确定查询每列的类型。如果失败，将提取查询的第一行以确定列类型。

通过使用一些特殊注释，可以直接在查询中指定特定列的类型。

语法如: `/*:type*/` 。必须放置在列名称之后。对于整数 类型可以是 整型，而浮点数可以是 实数或 文本。

例如:

```
SELECT id+1 as nid /*:int*/
FROM table
```

几何图形列的类型和坐标参照系也可以通过以下语法 `/*:gtype:srid*/` 的特殊注释进行设置，其中 `gtype` 是几何图形类型类型 (点, 线串, 多边形, 点集, 线串集或 多边形集) ， `srid` 是表示坐标参照系的 EPSG 代码的整数。

索引的使用

当通过虚拟图层查询图层时，源图层索引将按以下方式使用:

- 如果在图层的主键列上使用 = 谓词，将要求底层数据提供程序提供特定的 ID (FilterFid)
- 对于任何其他谓词 (>, <=, != 等) 或没有主键的列，将使用从表达式构建的请求来请求底层矢量数据提供程序。这意味着索引可以在数据库提供程序上使用 (如果存在)。

存在一种特定的语法来处理请求中的空间谓词，并触发空间索引的使用: 每个虚拟图层都存在一个名为 `_search_frame_` 的隐藏列。可以将此列与边界框进行相等比较。如下示例:

```
SELECT *
FROM vtab
WHERE _search_frame_=BuildMbr(-2.10,49.38,-1.3,49.99,4326)
```

空间二进制谓词 (如 `ST_Intersects`) 在与此空间索引语法结合使用时会显著加快速度。

11.3 Georeferencer

The Georeferencer is a tool for aligning unreferenced raster or vector layers to known coordinate systems using Ground Control Points (GCPs). It supports exporting transformed rasters (GeoTIFF) and vectors (shapefiles, geopackages, etc.) or writing accompanying world files (for raster only). The basic approach to georeferencing a layer is to locate points on it for which you can accurately determine coordinates.

Features

Table 11.1: Georeferencer Tools

Icon	目标	Icon	目标
	Open raster		Open vector
	Start georeferencing		Generate GDAL Script
	Load GCP Points		Save GCP Points As
	Transformation settings		
	Add GCP Point		Delete GCP Point
	Move GCP Point		Snapping Type
	Show Advanced Digitizing Dock		Pan
	Zoom In		Zoom Out
	Zoom To Layer		Zoom Last
	Zoom Next		Link Georeferencer to QGIS
	Link QGIS to Georeferencer		Full histogram stretch
	Local histogram stretch		

11.3.1 Georeferencing raster layer

As X and Y coordinates (DMS (dd mm ss.ss), DD (dd.dd) or projected coordinates (mmmm.mm)), which correspond with the selected point on the image, two alternative procedures can be used:

- The raster itself sometimes provides crosses with coordinates "written" on the image. In this case, you can enter the coordinates manually.
- Using already georeferenced layers. This can be either vector or raster data that contain the same objects/features that you have on the image that you want to georeference and with the projection that you want for your image. In this case, you can enter the coordinates by clicking on the reference dataset loaded in the QGIS map canvas.

The usual procedure for georeferencing an image involves selecting multiple points on the raster, specifying their coordinates, and choosing a relevant transformation type. Based on the input parameters and data, the Georeferencer will compute the world file parameters. The more coordinates you provide, the better the result will be.

The first step is to start QGIS and click on *Layer* ► *Georeferencer*, which appears in the QGIS menu bar. The Georeferencer dialog appears as shown in Fig. 11.38.

For this example, we are using a topo sheet of South Dakota from SDGS. It can later be visualized together with the data from the GRASS `spearfish60` location. You can download the topo sheet here: https://grass.osgeo.org/sampledata/spearfish_toposheet.tar.gz.

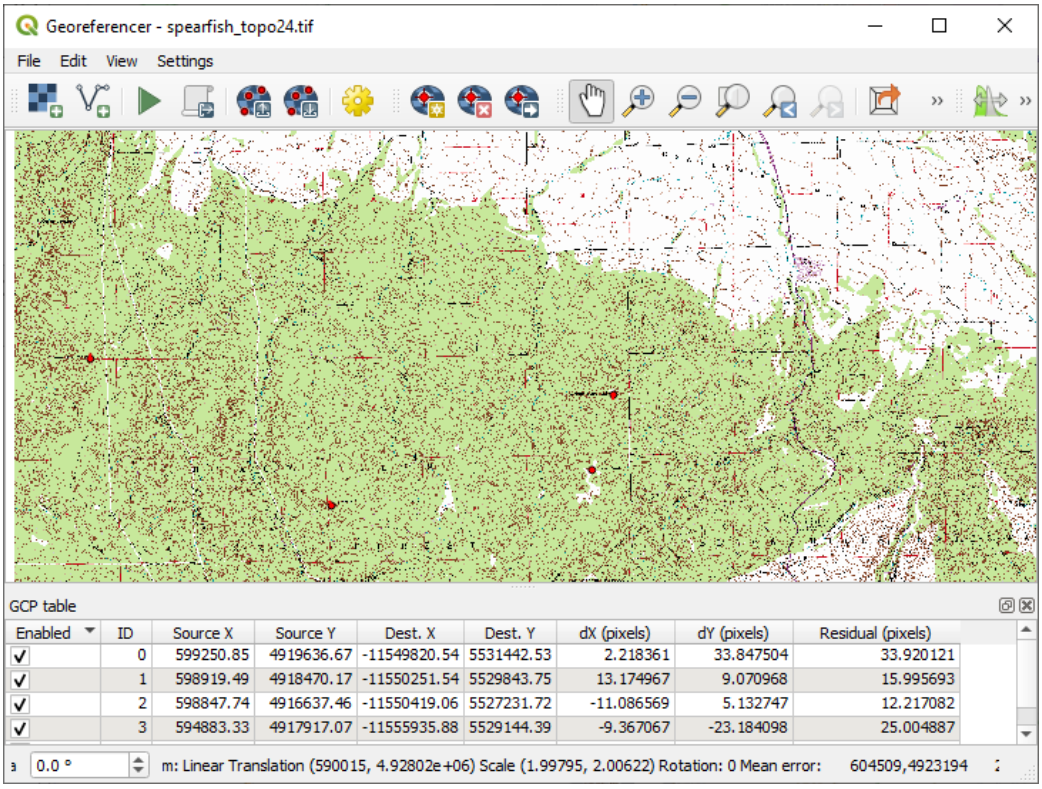


Fig. 11.38: Georeferencer Dialog

Entering ground control points (GCPs)

1. To start georeferencing an unreferenced raster, we must load it using the button. The raster will show up in the main working area of the dialog. Once the raster is loaded, we can start to enter reference points.

2. Using the Add GCP Point button, add points to the main working area and enter their coordinates (see Figure Fig. 11.39). For this procedure you have the following options:

   - Click on a point in the raster image and enter the X and Y coordinates manually, along with the CRS of the point.
   - Click on a point in the raster image and choose the From map canvas button to add the X and Y coordinates with the help of a georeferenced map already loaded in the QGIS map canvas. The CRS will be set automatically.
   - When entering GCPs from the main map canvas, you have the option to hide the georeferencer window while selecting points from the main canvas. If the *Automatically hide georeferencer window* checkbox is ticked, after clicking *From Map Canvas*, the main georeferencer window will be hidden until a point is added on the map canvas. The *Enter Map Coordinates* dialog will remain open. If the box is unchecked, both windows will remain open while selecting a point on the map canvas. This option only takes effect when the georeferencer window is not docked in the main interface.

3. Continue entering points. You should have at least four points, and the more coordinates you can provide, the better the result will be. There are additional tools for zooming and panning the working area in order to locate a relevant set of GCP points.

> **Tip**

To avoid constant switching between Pan, Add GCP point and Move GCP point buttons, you may use the keyboard arrow keys for moving and the mouse wheel for scaling the georeferenced map conveniently.

4. After you provide a few points, you can use the Link QGIS to Georeferencer and/or Link Georeferencer to QGIS buttons that will adjust, respectively, the map extent of the main QGIS window to the present view in Georeferencer and/or vice versa.
5. With the tool, you can move the GCPs in both the canvas and the georeferencing window, if you need to correct them.

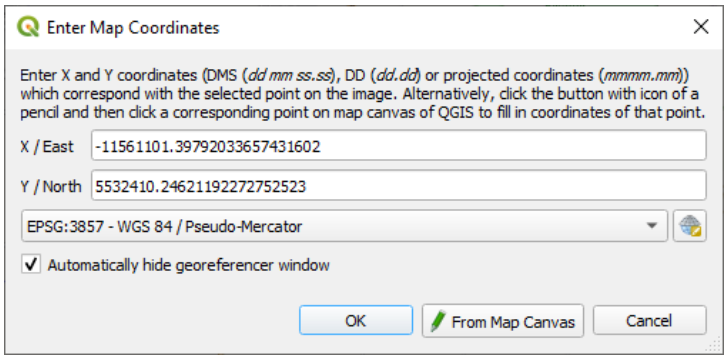


Fig. 11.39: Add points to the raster image

The points that are added to the map will be stored in a separate text file (`[filename].points`) usually together with the raster image. This allows us to reopen the Georeferencer at a later date and add new points or delete existing ones to optimize the result. The points file contains values of the form: `mapX, mapY, pixelX, pixelY`. You can use the Load GCP points and Save GCP points as buttons to manage the files.

Defining the transformation settings

After you have added your GCPs to the raster image, you need to define the transformation settings for the georeferencing process.

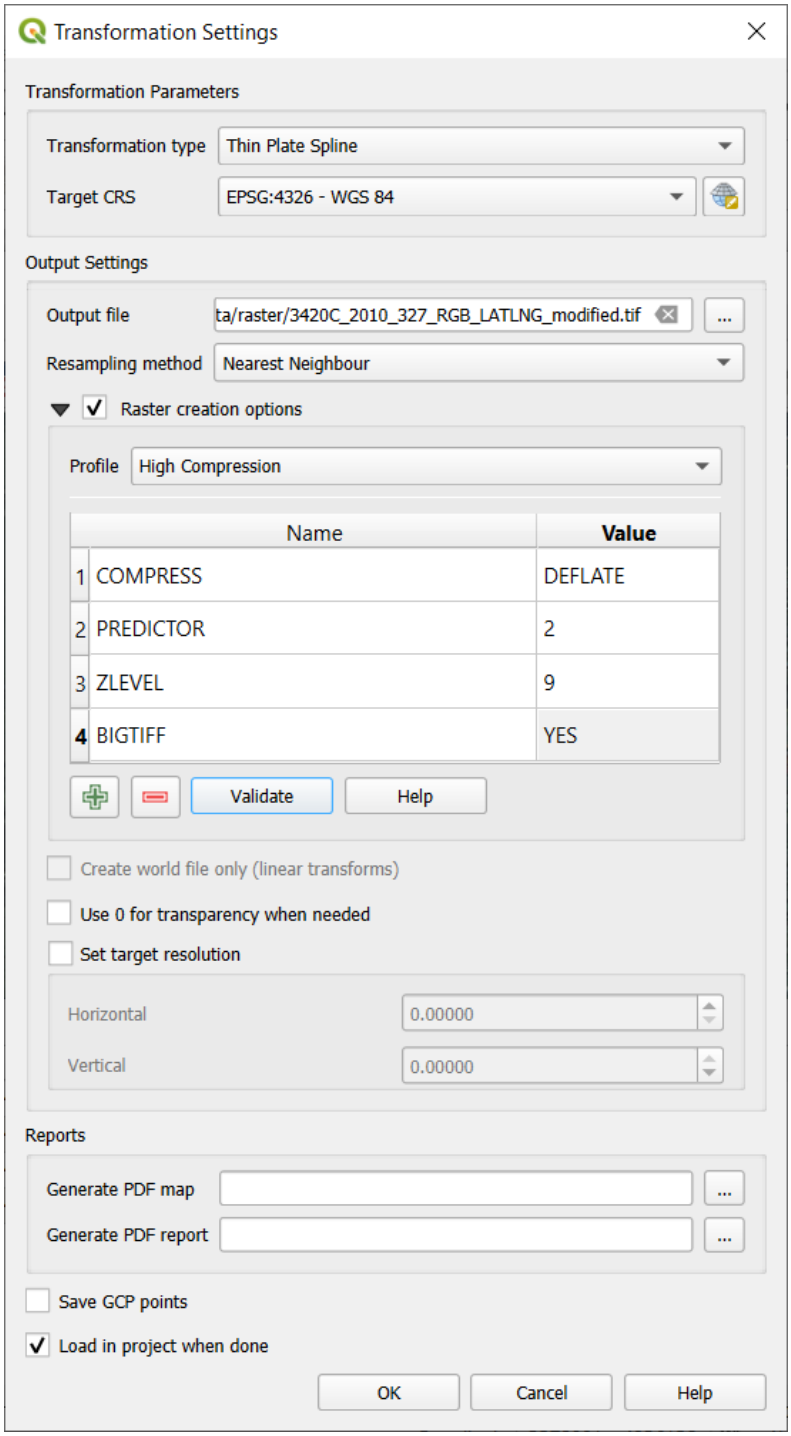


Fig. 11.40: Defining the georeferencer transformation settings

Available Transformation algorithms

A number of transformation algorithms are available, dependent on the type and quality of input data, the nature and amount of geometric distortion that you are willing to introduce to the final result, and the number of ground control points (GCPs).

Currently, the following *Transformation types* are available:

- The **Linear** algorithm is used to create a world file and is different from the other algorithms, as it does not actually transform the raster pixels. It allows positioning (translating) the image and uniform scaling, but no rotation or other transformations. It is the most suitable if your image is a good quality raster map, in a known CRS, but is just missing georeferencing information. At least 2 GCPs are needed.
- The **Helmert** transformation also allows rotation. It is particularly useful if your raster is a good quality local map or orthorectified aerial image, but not aligned with the grid bearing in your CRS. At least 2 GCPs are needed.
- The **Polynomial 1** algorithm allows a more general affine transformation, in particular also a uniform shear. Straight lines remain straight (i.e., collinear points stay collinear) and parallel lines remain parallel. This is particularly useful for georeferencing data cartograms, which may have been plotted (or data collected) with different ground pixel sizes in different directions. At least 3 GCP's are required.
- The **Polynomial** algorithms 2-3 use more general 2nd or 3rd degree polynomials instead of just affine transformation. This allows them to account for curvature or other systematic warping of the image, for instance photographed maps with curving edges. At least 6 (respectively 10) GCP's are required. Angles and local scale are not preserved or treated uniformly across the image. In particular, straight lines may become curved, and there may be significant distortion introduced at the edges or far from any GCPs arising from extrapolating the data-fitted polynomials too far.
- The **Projective** algorithm generalizes Polynomial 1 in a different way, allowing transformations representing a central projection between 2 non-parallel planes, the image and the map canvas. Straight lines stay straight, but parallelism is not preserved and scale across the image varies consistently with the change in perspective. This transformation type is most useful for georeferencing angled photographs (rather than flat scans) of good quality maps, or oblique aerial images. A minimum of 4 GCPs is required.
- Finally, the **Thin Plate Spline** (TPS) algorithm "rubber sheets" the raster using multiple local polynomials to match the GCPs specified, with overall surface curvature minimized. Areas away from GCPs will be moved around in the output to accommodate the GCP matching, but will otherwise be minimally locally deformed. TPS is most useful for georeferencing damaged, deformed, or otherwise slightly inaccurate maps, or poorly orthorectified aerials. It is also useful for approximately georeferencing and implicitly reprojecting maps with unknown projection type or parameters, but where a regular grid or dense set of ad-hoc GCPs can be matched with a reference map layer. It technically requires a minimum of 10 GCPs, but usually more to be successful.

In all of the algorithms except TPS, if more than the minimum GCPs are specified, parameters will be fitted so that the overall residual error is minimized. This is helpful to minimize the impact of registration errors, i.e. slight imprecisions in pointer clicks or typed coordinates, or other small local image deformations. Absent other GCPs to compensate, such errors or deformations could translate into significant distortions, especially near the edges of the georeferenced image. However, if more than the minimum GCPs are specified, they will match only approximately in the output. In contrast, TPS will precisely match all specified GCPs, but may introduce significant deformations between nearby GCPs with registration errors.

Define the Resampling method

The type of resampling you choose will likely depend on your input data and the ultimate objective of the exercise. If you don't want to change statistics of the raster (other than as implied by nonuniform geometric scaling if using other than the Linear, Helmert, or Polynomial 1 transformations), you might want to choose 'Nearest neighbour'. In contrast, 'cubic resampling', for instance, will usually generate a visually smoother result.

It is possible to choose between five different resampling methods:

1. Nearest neighbour
2. Bilinear (2x2 kernel)
3. Cubic (4x4 kernel)
4. Cubic B-Spline (4x4 kernel)
5. Lanczos (6x6 kernel)

Define the Raster creation options

When exporting a raster, *Raster creation options* allows you to define additional options that control how the output file is structured and compressed. See more at *Raster driver options*.

> **Tip**
>
> Select an empty entry if you want to create your own custom combination of parameters.

Define the transformation settings

There are several options that need to be defined for the georeferenced output raster.

- The *Create world file* checkbox is only available if you decide to use the linear transformation type, because this means that the raster image actually won't be transformed. In this case, the *Output raster* field is not activated, because only a new world file will be created.
- For all other transformation types, you have to define an *Output raster*. As default, a new file ([filename]_modified) will be created in the same folder together with the original raster image.
- As a next step, you have to define the *Target CRS* (Coordinate Reference System) for the georeferenced raster (see 使用投影).
- If you like, you can **generate a pdf map** and also **a pdf report**. The report includes information about the used transformation parameters, an image of the residuals and a list with all GCPs and their RMS errors.
- Furthermore, you can activate the *Set Target Resolution* checkbox and define the pixel resolution of the output raster. Default horizontal and vertical resolution is 1.
- The *Use 0 for transparency when needed* can be activated, if pixels with the value 0 shall be visualized transparent. In our example toposheet, all white areas would be transparent.
- The *Save GCP Points* will store GCP Points in a file next to the output raster.
- Finally, *Load in project when done* loads the output raster automatically into the QGIS map canvas when the transformation is done.

Running the transformation

After all GCPs have been collected and all transformation settings are defined, just press the Start georeferencing button to create the new georeferenced raster.

11.3.2 Georeferencing vector layer

Georeferencing vector layers works similarly to raster georeferencing, but instead of matching image pixels, you match vector geometries (points, lines, or polygons) to known spatial references.

The standard procedure starts the same as for raster georeferencing: open QGIS and add a layer to the map canvas to use as a reference. This can be a georeferenced raster or vector layer, or a WMS layer.

Open the Georeferencer dialog from *Layer* ► *Georeferencer*.

Start georeferencing by following these steps (in this example, we use an unreferenced `alaska.shp`):

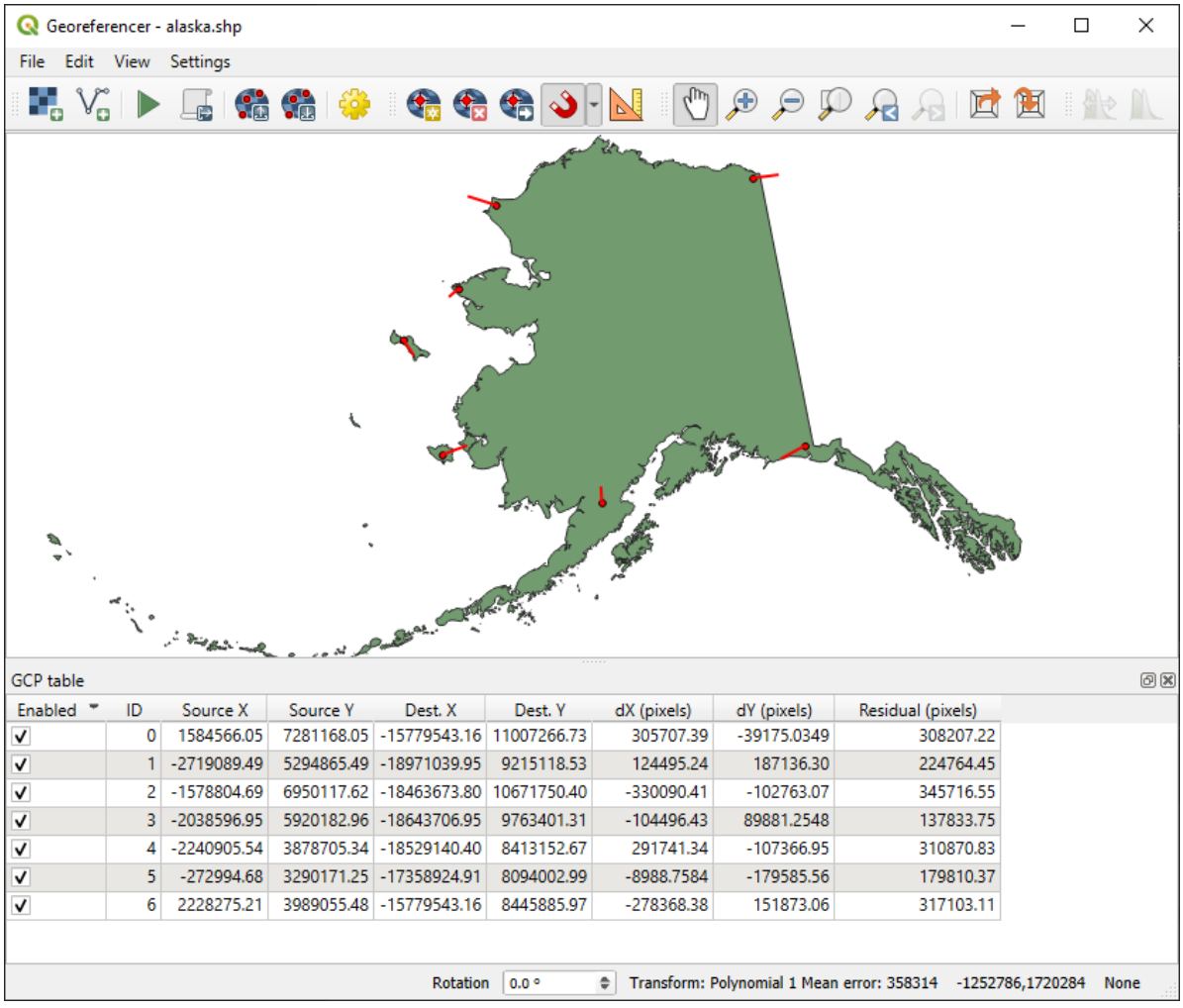


Fig. 11.41: Vector Georeferencer Dialog

1. Load the unreferenced vector layer using the button. The vector layer will appear in the main working area of the dialog.

2. Activate snapping by clicking the button and selecting the desired snapping type(s). This enables snapping to the reference layer when placing GCPs.

You can also use the Advanced Digitizing dock to ensure high-precision point selection. For more information, refer to the *Advanced Digitizing* section.

3. Use the Add GCP Point button to add a point to the working area. Enter its coordinates manually and set the CRS, or click the From map canvas button to pick the coordinates from a georeferenced layer in the main QGIS map canvas. In that case, the CRS will be set automatically.
4. Define the transformation settings:
   - Select the *Transformation type*. The transformation algorithms are the same as those for raster georeferencing. See *Defining the transformation settings* for more details.
   - Define the *Target CRS* (Coordinate Reference System) for the georeferenced vector (see 使用投影).
   - Set the output file format and path (e.g., GeoPackage, Shapefile). By default, a new file with suffix `_modified` will be created in the same folder as the original vector file.
   - Optionally, enable **Generate PDF map** and **Generate PDF report**. The report includes transformation parameters, GCP residuals, and a summary of RMS errors.
   - Enable *Save GCP Points* to store GCPs in a file alongside the output vector layer.
   - Enable *Load in project when done* to add the result directly to the map canvas.
5. Click Start georeferencing to run the transformation and generate the georeferenced vector layer.

11.3.3 Show and adapt layer properties

Clicking on the *Source properties* option in the *Settings* menu opens the *Raster Layer properties* or *Vector Layer properties* depending on the type of layer you are georeferencing.

11.3.4 Configure the georeferencer

You can customize the behavior of the georeferencer in *Settings* ► *Configure Georeferencer* (or use keyboard shortcut `Ctrl+P`).

- Under *Point Tip* you can use the checkboxes to toggle displaying GCP IDs and X/Y coordinates in both the Georeferencer window and the main map canvas.
- *Residual Units* controls whether residual units are given in pixels or map units
- *PDF Report* allows you to set margin size in mm for the report export
- *PDF Map* allows you to choose a paper size for the map export
- Finally, you can activate to *Show Georeferencer window docked*. This will dock the Georeferencer window in the main QGIS window rather than showing it as a separate window that can be minimized.

11.4 探索数据格式与字段

11.4.1 栅格数据

GIS 栅格数据是离散像元矩阵，表示地球表面上、上方或下方的要素/现象。栅格网格中的每个像元具有相同的尺寸，并且像元通常是矩形 (在 QGIS 中，它们总是矩形)。典型的栅格数据集包括遥感数据，如航空摄影，或卫星影像和建模数据，如高程或温度。

与矢量数据不同，栅格数据通常没有每个像元的关联数据库记录。它们是通过像素分辨率和栅格图层的角像素的 X/Y 坐标进行地理编码的。这允许 QGIS 在地图画布正确定位数据。

地理包格式便于在使用 QGIS 时存储栅格数据。流行且功能强大的 GeoTiff 格式是不错的选择。

QGIS 利用栅格图层 (例如 GeoTiff) 或相关 坐标信息文件中的地理配准信息来正确显示数据。

11.4.2 矢量数据

Many of the features and tools available in QGIS work the same, regardless of the vector data source. However, because of the differences in format specifications (GeoPackage, ESRI Shapefile, MapInfo and MicroStation file formats, AutoCAD DXF, PostgreSQL, SpatiaLite, Oracle Spatial, MS SQL Server, SAP HANA Spatial databases and many more), QGIS may handle some of their properties differently. Support is provided by the GDAL vector drivers. This section describes how to work with some of these specifics.

> **Note**
>
> QGIS 支持点 (集)、线 (集)、多边形 (集)、圆弧、复合曲线、曲线多边形、曲线集、表面集要素类型，所有这些都可以选择 Z 和/或 M 值。
>
> 您还应该注意，某些驱动程序不支持其中一些要素类型，如圆弧、复合曲线、曲线多边形、曲线集、表面集要素类型。QGIS 将对其进行转换。

GeoPackage

地理包 (GPKG) 格式与平台无关，作为 SQLite 数据库容器的工具，可用于储存矢量和栅格数据。该格式由开放地理空间联盟 (OGC) 定义，并于 2014 年发布。

地理包可用于在 SQLite 数据库中储存以下内容：

- **矢量**要素
- **影像瓦片矩阵集**和 **栅格**地图
- 属性 (非空间数据)
- 扩展

自 QGIS 3.8 版开始，地理包还可以储存 QGIS 工程。地理包图层可以含有 JSON 字段。

地理包是 QGIS 中矢量数据的默认格式。

ESRI 形状文件格式

ESRI 形状文件格式仍然是最常用的矢量文件格式之一，与地理包和 SpatiaLite 相比有一些限制。

ESRI 形状文件格式数据集由多个文件组成，但以下三个是必需文件：

1. `.shp` 文件包含要素几何图形
2. `.dbf` 文件包含 dBase 格式的属性
3. `.shx` 索引文件

ESRI 形状文件格式数据集还可以包括后缀名为 `.prj` 的文件，该文件包含投影信息。虽然拥有投影文件非常有用，但不是必需的。除此之外，形状文件格式的数据集可以包含其他文件。有关更多详细信息，请参阅 ESRI 技术规范 。

GDAL 支持读写压缩的 ESRI 形状文件格式 (`shz` 和 `shp.zip`) 。

改进 ESRI 形状文件格式数据集的性能

要改进 ESRI 形状文件格式数据集的绘图性能，可以创建空间索引。空间索引将提高缩放和平移的速度。QGIS 的空间索使用 `.qix` 扩展名。

使用以下步骤创建索引：

1. 加载 ESRI 形状文件格式数据集 (请参阅浏览器面板)
2. 打开 *图层属性*对话框，方法是双击图例中的图层名，或右键单击并从上下文菜单选择 *属性...*
3. 在 *源*选项卡，单击 *创建空间索引*按钮

加载.prj 文件时的问题

如果使用 `.prj` 文件加载 ESRI 形状文件格式数据集，并且 QGIS 无法从该文件读取坐标参照系，则需要在图层的 *图层属性* ► *源*选项卡通过单击 选择 CRS 按钮，手动定义正确的投影。这是因为 `.prj` 文件通常不提供在 QGIS 中使用的完整投影参数，并在 *CRS* 对话框中列出。

出于同样的原因，如果使用 QGIS 新建 ESRI 形状文件格式数据集，则会创建两个不同的投影文件：一个是具有有限投影参数的 `.prj` 文件，与 ESRI 软件兼容；另一个是提供 CRS 所有参数的 `.qpj` 文件。每当 QGIS 找到 `.qpj` 文件时，就会使用其代替 `.prj` 。

分隔文本文件

分隔文本文件非常常见并被广泛使用，是因为它们的简单性和可读性 -- 数据可以在纯文本编辑器中查看和编辑。分隔文本文件是表列数据，其列由定义的字符分隔，行由换行符分隔。首行通常包含列名。一种常见的分隔文本文件类型是 CSV (逗号分隔值)，列之间用逗号分隔。分隔文本文件还可以包含位置信息 (请参阅将几何图形信息存储在分隔文本文件中) 。

QGIS 允许将分隔文本文件加载为图层或普通表 (请参阅浏览器面板 或导入分隔文本文件) 。首先检查文件是否满足以下要求：

1. 文件必须具有字段名分隔的标题行。这必须是数据的第一行 (最好是文本文件中的第一行)。
2. 如果启用几何图形，文件必须包含定义几何图形的字段。这些字段的名称随意定义。
3. X 和 Y 坐标字段 (如果几何图形由坐标定义) 必须指定为数字。坐标系并不重要。
4. 如果您有个非字符串列的 CSV 文件，可以附带 CSVT 文件 (请参阅章节使用 *CSVT* 文件控制字段格式) 。

QGIS 示例数据集中的高程点数据文件 `elevp.csv` (请参阅章节下载示例数据) 是有效文本文件的示例：

```
X;Y;ELEV
-300120;7689960;13
-654360;7562040;52
1640;7512840;3
[...]
```

关于文本文件需要注意的一些事项：

1. 示例文本文件使用 ; (分号) 作为分隔符 (任何字符都可以用于分隔字段)。
2. 首行是标题行，包含字段 X, Y 和 ELEV 。
3. 无引号 (") 用于分隔文本字段
4. X 坐标包含在 X 字段中
5. Y 坐标包含在 Y 字段中

将几何图形信息存储在分隔文本文件中

分隔文本文件可以包含两种主要形式的几何图形信息：

- 作为单独列中的坐标 (例如 Xcol, Ycol 等), 用于点几何图形数据；
- 作为单列中的几何图形使用熟知文本 (WKT) 表示，用于几何图形类型。

支持具有曲线几何图形的要素 (圆弧、曲线多边形和复合曲线)。以下是分隔文本文件中几何图形类型的一些示例，其中几何图形编码为 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))
```

分隔文本文件还支持几何图形的 Z 和 M 坐标:

```
LINESTRINGZ(10.0 20.0 30.0, 11.0 21.0 31.0, 11.0 22.0 30.0)
```

使用 CSVT 文件控制字段格式

加载 CSV 文件时，GDAL 驱动程序假定所有字段都是字符串 (即文本)，除非另有说明。您可以创建一个 CSVT 文件来告知 GDAL (和 QGIS) 不同列的数据类型：

类型	名称	示例
整数	整数	4
布尔	整型 (布尔)	真
小数	实数	3.456
日期	日期 (YYYY-MM-DD)	2016-07-28
时间	时间 (HH:MM:SS+nn)	18:33:12+00
日期 & 时间	日期时间 (YYYY-MM-DD HH:MM:SS+nn)	2016-07-28 18:33:12+00
坐标 X	坐标 X	8.8249
坐标 Y	坐标 Y	47.2274
点 (X)	点 (X)	8.8249
点 (Y)	点 (Y)	47.2274
WKT	WKT	POINT(15 20)

CSVT 文件是 一行纯文本文件，数据类型用引号括起来，并用逗号分隔，例如:

```
"Integer","Real","String"
```

甚至可以指定每列的宽度和精度，例如:

```
"Integer(6)","Real(5.5)","String(22)"
```

此文件与 `.csv` 文件保存在同一文件夹，名称相同，但扩展名是 `.csvt` 。

You can find more information at GDAL CSV Driver.

Tip

检测字段类型

QGIS 提供了自动检测字段类型和更改假定字段类型的可能性，而不是使用 CSVT 文件告知数据类型。

PostgreSQL Database

In order to process geographical information in a PostgreSQL database, the PostGIS extension must first be installed. PostGIS extends the capabilities of the PostgreSQL relational database by adding support for storing, indexing, and querying geospatial data.

To enable PostGIS in your database you have to activate the extension in your database (open database and run `CREATE EXTENSION postgis;`). PostGIS enabled databases are also often named "PostGIS layer". Using PostGIS, vector functions such as select and identify work more accurately than they do with GDAL layers in QGIS.

Tip

PostGIS 图层

通常，PostGIS 图层由 geometry_columns 表中的条目标识。QGIS 可以加载 geometry_columns 表中没有条目的图层，这包括表和视图。有关创建视图的信息，请参阅 PostgreSQL 手册。

本节包含一些有关 QGIS 如何访问 PostgreSQL 图层的详细信息。多数情况，QGIS 应该可以简单地为您提供加载数据库表的列表，并会根据请求将其加载。但是，如果您在将 PostgreSQL 表加载到 QGIS 时遇到问题，下面的资料可能会帮助您理解 QGIS 信息，并为您提供修改 PostgreSQL 表或视图定义以允许 QGIS 将其加载。

Note

PostgreSQL 数据库也可以储存 QGIS 工程。

主键

QGIS 要求 PostgreSQL 图层包含一个可用作图层唯一键的列。对于表，通常意味着需要一个主键或一个具有唯一约束的列。在 QGIS 中，此列的类型需要为 int4 (大小为 4 字节的整数)。或者，ctid 列可以用作主键。如果表缺少这些项，将使用 oid 列代替。如果对列进行索引 (请注意，主键在 PostgreSQL 中是自动索引的)，性能将得到提升。

QGIS 提供了一个复选框 **按 ID 选择**，默认处于激活状态。此选项获取不带属性的 ID，这在大多数情况下速度会更快。

视图

如果 PostgreSQL 图层是视图，那么存在同样的要求，但视图并不总是具有主键或唯一约束的列。您必须在 QGIS 对话框中定义一个主键字段 (必须是整数)，然后才能加载视图。如果视图中不存在匹配的列，QGIS 将不会加载该图层。如果出现这种情况，解决方案是修改视图，使其包含匹配的列 (整数类型、主键或具有唯一约束，最好具有索引)。

对于表，默认会激活复选框 **按 ID 选择** (有关复选框的含义，请参阅上文)。当您使用高价值视图时，禁用此选项是明智的。

> **Note**
>
> **PostgreSQL 外部表**
>
> PostgreSQL 提供程序未明确支持 PostgreSQL 外部表，将像视图一样处理。

QGIS layer_style 表和数据库备份

If you want to make a backup of your PostgreSQL database using the `pg_dump` and `pg_restore` commands, and the default layer styles as saved by QGIS fail to restore afterwards, you need to set the XML option to `DOCUMENT` before the restore command:

1. 对 `layer_style` 表进行 PLAIN 备份
2. 在文本编辑器打开文件
3. 将行 `SET xmloption = content;` 更改为 `SET XML OPTION DOCUMENT;`
4. 保存文件
5. 使用 psql 在新数据库还原表

过滤数据库端

QGIS 允许过滤服务器端已有的要素，为此请勾选 *设置* ► *选项* ► *数据源* ► *如果允许，在服务器端执行表达式*。仅支持的表达式才会发送到数据库，使用不受支持的运算符或函数的表达式将回退到本地执行。

PostgreSQL 支持的数据类型

PostgreSQL 提供程序支持的数据类型包括：integer、float、boolean、binary object、varchar、geometry、timestamp、array、hstore 和 json。

将数据导入 PostgreSQL

Data can be imported into PostgreSQL using several tools, including the Browser, DB Manager plugin and the command line tools shp2pgsql and ogr2ogr.

数据库管理器

QGIS 自带了一个名为 数据库管理器 的核心插件。可用于加载数据，以及对架构的支持。请参阅*DB Manager Plugin* 章节了解更多信息。

shp2pgsql

PostGIS 包含一个名为 **shp2pgsql** 的实用程序，可用于将形状文件格式的数据集导入激活的 PostGIS 数据库。例如，要将名为 `lakes.shp` 的形状文件格式数据集导入到名为 `gis_data` 的 PostgreSQL 数据库，请使用以下命令:

```
shp2pgsql -s 2964 lakes.shp lakes_new | psql gis_data
```

这将在 `gis_data` 数据库创建一个名为 `lakes_new` 的新图层。新建图层具有 2964 的空间参照系标识符 (SRID)。有关空间参照系和投影的更多信息，请参阅使用投影 章节。

Tip

Exporting datasets from PostgreSQL

There is also a tool for exporting postgreSQL datasets wth geographic data to Shapefile format: **pgsql2shp**. It is shipped within your PostGIS distribution.

ogr2ogr

In addition to **shp2pgsql** and **DB Manager**, there is another tool for feeding geographical data in PostgreSQL: **ogr2ogr**. It is part of your GDAL installation.

To import a Shapefile format dataset into PostgreSQL, do the following:

```
ogr2ogr -f "PostgreSQL" PG:"dbname=postgis host=myhost.de user=postgres
password=topsecret" alaska.shp
```

This will import the Shapefile format dataset `alaska.shp` into the database *postgis* using the user *postgres* with the password *topsecret* on the host server *myhost.de*.

Note that GDAL must be built with PostgreSQL. You can verify this by typing (in):

```
ogrinfo --formats | grep -i post
```

如果您更喜欢使用 PostgreSQL 的 **COPY** 命令而不是默认的 **INSERT INTO** 方法，则可以导出以下环境变量 (至少在 和 中可用):

```
export PG_USE_COPY=YES
```

ogr2ogr 不能像 **shp2pgsl** 那样创建空间索引，作为附加操作 (如下一节提高性能 所述)，您需要使用常规的 SQL 命令 **CREATE INDEX** 手动创建。

提高性能

Retrieving features from a PostgreSQL database can be time-consuming, especially over a network. You can improve the drawing performance of PostgreSQL layers by ensuring that a spatial index exists on each layer in the database. PostGIS supports creation of a GiST (Generalized Search Tree) index to speed up spatial searching (GiST index information is taken from the PostGIS documentation available at https://postgis.net).

> **Tip**
>
> 可以使用数据库管理器为图层创建索引。您应该首先选择图层，然后单击 *表* ► *编辑表*，并转到 *索引* 选项卡，之后单击 *添加空间索引*。

创建 GiST 索引的语法为:

```
CREATE INDEX [indexname] ON [tablename]
  USING GIST ( [geometryfield] GIST_GEOMETRY_OPS );
```

请注意，对于较大的表，创建索引可能需要很长时间。创建索引后，应该执行 *优化分析* (`Vacuum`)。更多信息请参阅 PostGIS 文档 (*文献和网络参考* 中的 POSTGIS-PROJECT)。

以下示例为创建 GiST 索引:

```
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 图层

如果要使用 SpatiaLite 格式保存矢量图层，可以参照*从现有图层创建新图层* 操作指南。选择 `SpatiaLite` 作为 *格式* 类型，然后输入 *文件名* 和 *图层名*。

Also, you can select `SQLite` as format and then add `SPATIALITE=YES` in the *Custom Options* ► *Data source* field. This tells GDAL to create a SpatiaLite database. See also https://gdal.org/en/latest/drivers/vector/sqlite.html.

QGIS 还支持 SpatiaLite 中的可编辑视图。对于 SpatiaLite 数据管理，可以使用核心插件*数据库管理器* 。

如果要新建 SpatiaLite 图层，请参阅*新建 SpatiaLite 图层* 章节。

GeoJSON 特定参数

导出图层 到 GeoJSON 时，有一些特定的 图层选项可用。这些选项来自负责写入文件的 GDAL：

- *COORDINATE_PRECISION* 写入坐标的小数分隔符之后的最大位数。默认值为 15 (注意：对于 Lat Lon 坐标是 6 就足够了)。其余将截断以移除尾随零。
- *RFC7946* by default GeoJSON 2008 will be used. If set to YES, the updated RFC 7946 standard will be used. Default is NO (thus GeoJSON 2008). See https://gdal.org/en/latest/drivers/vector/geojson.html#rfc-7946-write-support for the main differences, in short: only EPSG:4326 is allowed, other crs's will be transformed, polygons will be written such as to follow the right-hand rule for orientation, values of a "bbox" array are [west, south, east, north], not [minx, miny, maxx, maxy]. Some extension member names are forbidden in FeatureCollection, Feature and Geometry objects, the default coordinate precision is 7 decimal digits
- *WRITE_BBOX* 设置为 YES，可以包括要素和要素集层级的几何图形的边界框。

Besides GeoJSON there is also an option to export to "GeoJSON - Newline Delimited" (see https://gdal.org/en/latest/drivers/vector/geojsonseq.html). Instead of a FeatureCollection with Features, you can stream one type (probably only Features) sequentially separated with newlines.

GeoJSON - 换行符分隔也有一些特定的图层选项可用：

- *COORDINATE_PRECISION* 请参阅上文 (与 GeoJSON 相同)
- *RS* 是否以 RS=0x1E 字符开始记录。区别在于要素的分隔方式：只通过换行 (LF) 字符 (换行分隔的 JSON, geojsons)，或通过准备记录分隔 (RS) 字符 (给定 GeoJSON 文本序列, geojsons)。默认是 NO。如果未提供扩展名，则为文件提供 `.json` 扩展名。

SAP HANA 空间图层

本节包含一些有关 QGIS 如何访问 SAP HANA 图层的详细信息。多数情况，QGIS 应该可以简单地为您提供加载数据库表和视图的列表，并会根据请求将其加载。但是，如果您在将 SAP HANA 表或视图加载到 QGIS 中时遇到问题，以下信息有助于您了解根本原因并帮助解决。

要素识别

如果您想使用所有 QGIS 的要素编辑功能，QGIS 必须能够明确识别图层中的每个要素。在内部，QGIS 使用 64 位符号整数识别要素，而负范围则保留用于特殊用途。

因此，SAP HANA 提供程序需要可以映射到正 64 位整数的唯一密钥，以完全支持 QGIS 的要素编辑功能。如果无法创建此类映射，您仍然可以查看要素，但可能无法进行编辑操作。

添加表

将表添加为图层时，SAP HANA 提供程序使用表的主键将其映射到唯一要素的 ID。因此，要获得完整要素编辑支持，您需要具有表定义的主键。

SAP HANA 提供程序支持多列主键，但如果想获得最佳性能，主键应为 整数类型的单列。

添加视图

将视图添加为图层时，SAP HANA 提供程序无法自动识别明确识别要素的列。此外，有些视图是只读的，无法编辑。

要获得完整要素编辑支持，视图必须是可更新的 (请检查问题视图 `SYS.VIEWS` 在系统视图中的列 `IS_READ_ONLY`)，并且必须手动为 QGIS 提供一个或多个用于识别要素的列。可以通过使用 *图层* ► *添加图层* ► *添加 SAP HANA 空间图层*给定列，然后选择 *要素 ID* 列中的列。为了获得最佳性能，*要素 ID* 值应该是单 整数列。

11.4.3 图层横渡 180° 经度

许多 GIS 软件包不使用横渡 180 度经度线的地理参照系 (lat/lon) 围绕图层。因此，如果在 QGIS 打开这样的图层，可以看到彼此位置相距甚远，它们应该相互靠近。在 Fig. 11.42 中，地图画布最左侧的小点 (查塔姆群岛) 应该在新西兰主要岛屿右侧的网格内。

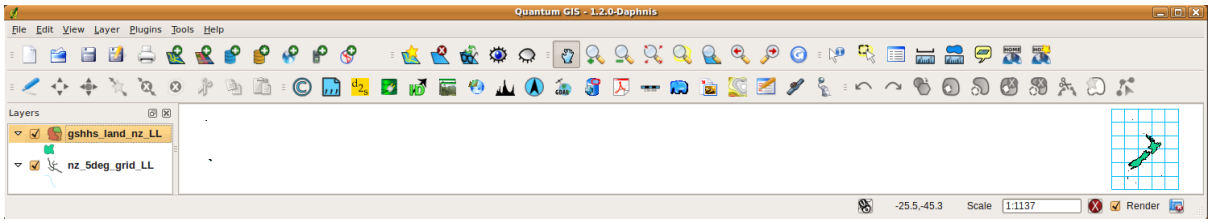


Fig. 11.42: 经纬度地图横渡 180° 经度线

Solving in PostgreSQL

A work-around is to transform the longitude values using PostgreSQL and the ST_ShiftLongitude function. This function reads every point/vertex in every component of every feature in a geometry, and shifts its longitude coordinate from -180..0° to 180..360° and vice versa if between these ranges. This function is symmetrical so the result is a 0..360° representation of a -180..180° data and a -180..180° representation of a 0..360° data.

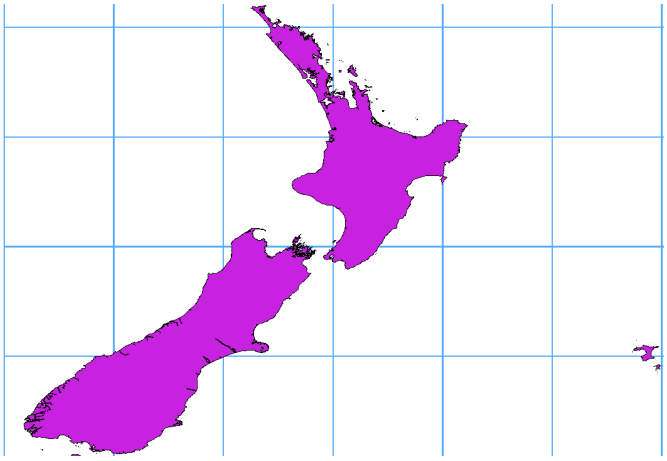


Fig. 11.43: 应用 **ST_ShiftLongitude** 函数横渡 180° 经度

1. Import data into PostgreSQL (*将数据导入 PostgreSQL*) using, for example, the DB Manager plugin.
2. Use the PostgreSQL command line interface to issue the following command:

```
-- In this example, "TABLE" is the actual name of your PostgreSQL table
update TABLE set geom=ST_ShiftLongitude(geom);
```

3. 如果一切顺利，应该会收到更新要素数量的确认窗口。然后，您将可以加载地图并查看差异 (*Figure_vector_crossing_map*) 。

CHAPTER 12

使用矢量数据

12.1 矢量属性对话框

矢量图层的 图层属性对话框提供常规设置，用于管理地图中图层的要素外观 (符号、标注、图表)、与鼠标交互 (动作、地图提示、表单设计)。它还提供有关该图层的信息。

要访问 图层属性对话框，请执行以下操作：

- 在 图层面板，双击该图层或右键单击并从弹出式菜单选择 属性... ；
- 选择图层后，转到 图层 ► 图层属性... 菜单。

矢量 图层属性对话框提供以下选项卡部分：

信息	源	*Symbology* [1]
Labels [1]	*Mask* [1]	*3D View* [1]
图表	字段	属性表单
Joins	附加存储	动作
显示	渲染	时态
变量	*Elevation*	元数据
相依性	图例	*QGIS* 服务器
数字化	*External plugins* [2] tabs	

Tip

分享图层样式的全部或部分属性

[1] Also available in the *Layer styling panel*
[2] *External plugins* you install can optionally add tabs to this dialog. Those are not presented in this document. Refer to their documentation.

对话框底部的 样式菜单允许您从 / 向多个目标 (文件、剪贴板、数据库) 导入或导出全部或部分属性。请参阅管理自定义样式 。

Note

因为嵌入图层 (请参阅从外部工程嵌入图层) 的属性 (符号、标注、动作、默认值、表单等) 是从原工程文件中提取的，为了避免可能破坏此行为的更改，图层属性对话框对这些图层不可用。

12.1.1 信息属性

The *Information* tab is read-only and represents an interesting place to quickly grab summarized information and metadata on the current layer. Provided information are:

- 常规：例如工程中的名称、源路径、辅助文件列表、上次保存的时间和大小、所使用的提供程序
- custom properties, used to store in the active project additional information about the layer. Default custom properties may include *layer notes*, *legend widgets*, *layer variables*, *form properties*... More custom properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- 基于图层的提供信息：存储格式、几何图形类型、数据源编码、范围、要素总数等
- 坐标参照系：名称、单位、方法、精度、参照 (即静态还是动态)
- 从填充的元数据 中提取：访问、范围、链接、联系人、历史等
- 并与其几何图形 (空间范围、CRS 等) 或其属性 (字段数量、每个特性等) 有关。

12.1.2 源属性

使用此选项卡可以定义矢量图层的常规设置。

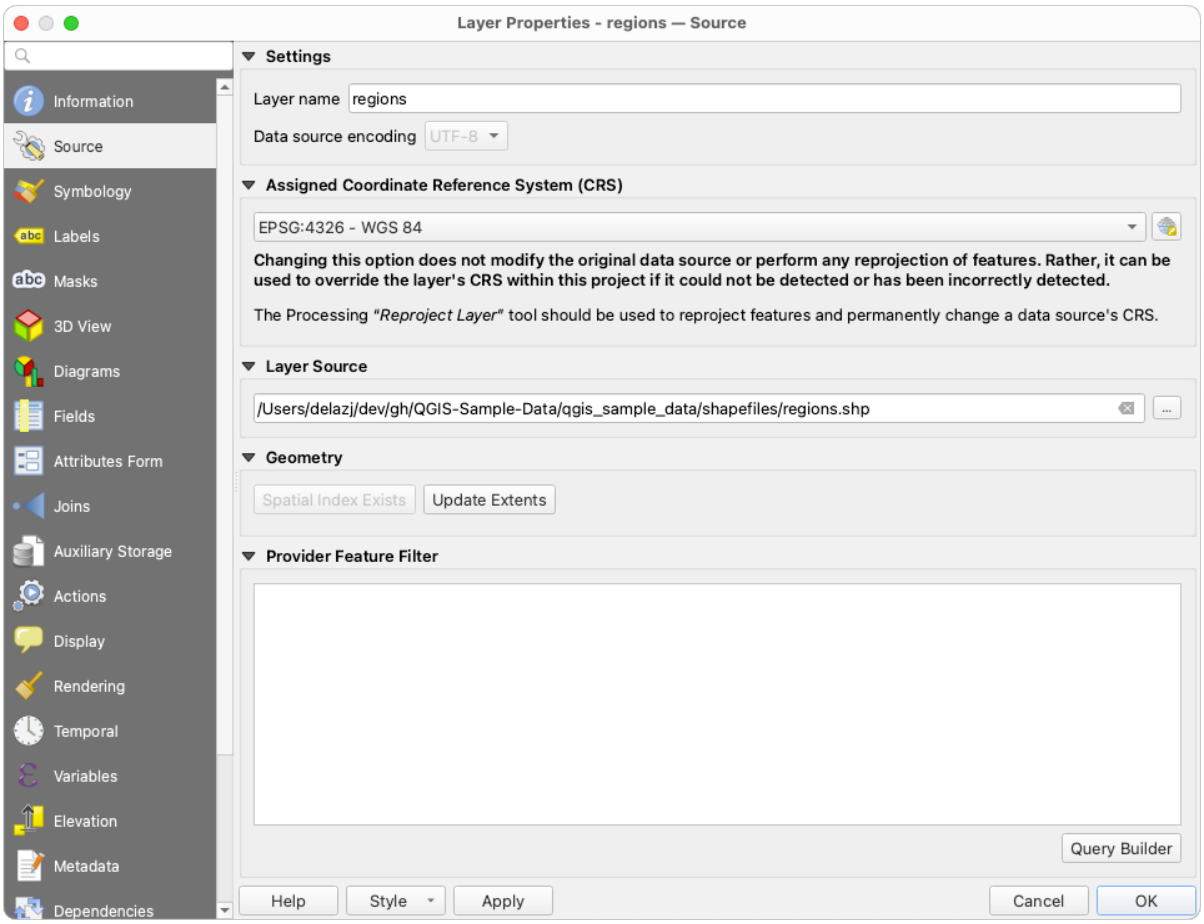


Fig. 12.1: 矢量图层属性对话框中的源选项卡

设置

- 设置 *图层名* 与用于识别工程中的图层文件名不同 (在 *图层面板* 中，使用表达式，在打印布局图例中，等等)
- 具体取决于数据格式，如果 QGIS 未正确检测到，请选择 *数据源编码*。

Coordinate Reference System

- 显示图层的*指定坐标参照系 (CRS)* 。您可以更改图层的 CRS，在下拉列表中选择最近使用的 CRS，或单击 选择 CRS 按钮 (请参阅*坐标参照系选择器*)。仅当应用于图层的 CRS 错误或未应用任何 CRS 时，才使用此方法。如果您希望将数据重投影到另一个 CRS，请使用图层重投影算法进行处理或*将其另存为其他图层* 。

Layer source and Geometry

- Depending on the data provider, a *Layer source* group indicates the path to the source of the dataset and allows for replacing the loaded layer:
  - When the layer is stored as file on disk, edit the path shown in the text box or press ... Browse to select another file on the disk. Both layers do not need to share attribute fields, geometry type or file formats.
  - When the layer is provided by an ArcGIS Feature service, it is possible to modify its *authentication settings*, while keeping unchanged the details for connecting to the service.
- *Create spatial index* (only for OGR-supported formats): helps speeding layer rendering and features' geometry retrieval.
- 为图层 更新范围信息。

查询构建器

The Query Builder dialog is accessible through the *Query Builder* button at the bottom of the *Source* tab in the Layer Properties dialog, under the *Provider feature filter* group.

查询构建器提供了一个界面，允许您使用类似 SQL 的 WHERE 子句定义图层中要素的子集，并在主窗口中显示结果。只要查询处于激活状态，工程中就只有与其结果相对应的要素可用。

可以使用一个或多个图层属性在 查询构建器中定义过滤条件。多个属性的使用如 Fig. 12.2 所示。在该示例中，过滤器组合了多个属性。

- `toa` (`DateTime` 字段: `cast("toa" as character) > '2017-05-17'` 和 `cast("toa" as character) < '2019-12-24T18:00:00'`),
- `name` (`String` field: `"name" > 'S'`) 和
- `FID` (`Integer` field: `FID > 10`)

使用 AND、OR 和 NOT 运算符和括号。此语法 (包括 `toa` 字段的日期时间格式) 适用于 GeoPackage 数据集。

The filter is made at the data provider (OGR, PostgreSQL, MS SQL Server...) level. So the syntax depends on the data provider (DateTime is for instance not supported for the ESRI Shapefile format). The complete expression:

```
cast("toa" as character) > '2017-05-17' AND
cast("toa" as character) < '2019-12-24T18:00:00' AND
NOT ("name" > 'S' OR FID > 10)
```

You can also open the *Query Builder* dialog using the *Filter...* option from the *Layer* menu or the layer contextual menu. The *Fields*, *Values* and *Operators* sections in the dialog help you to construct the SQL-like query exposed in the *Provider specific filter expression* box.

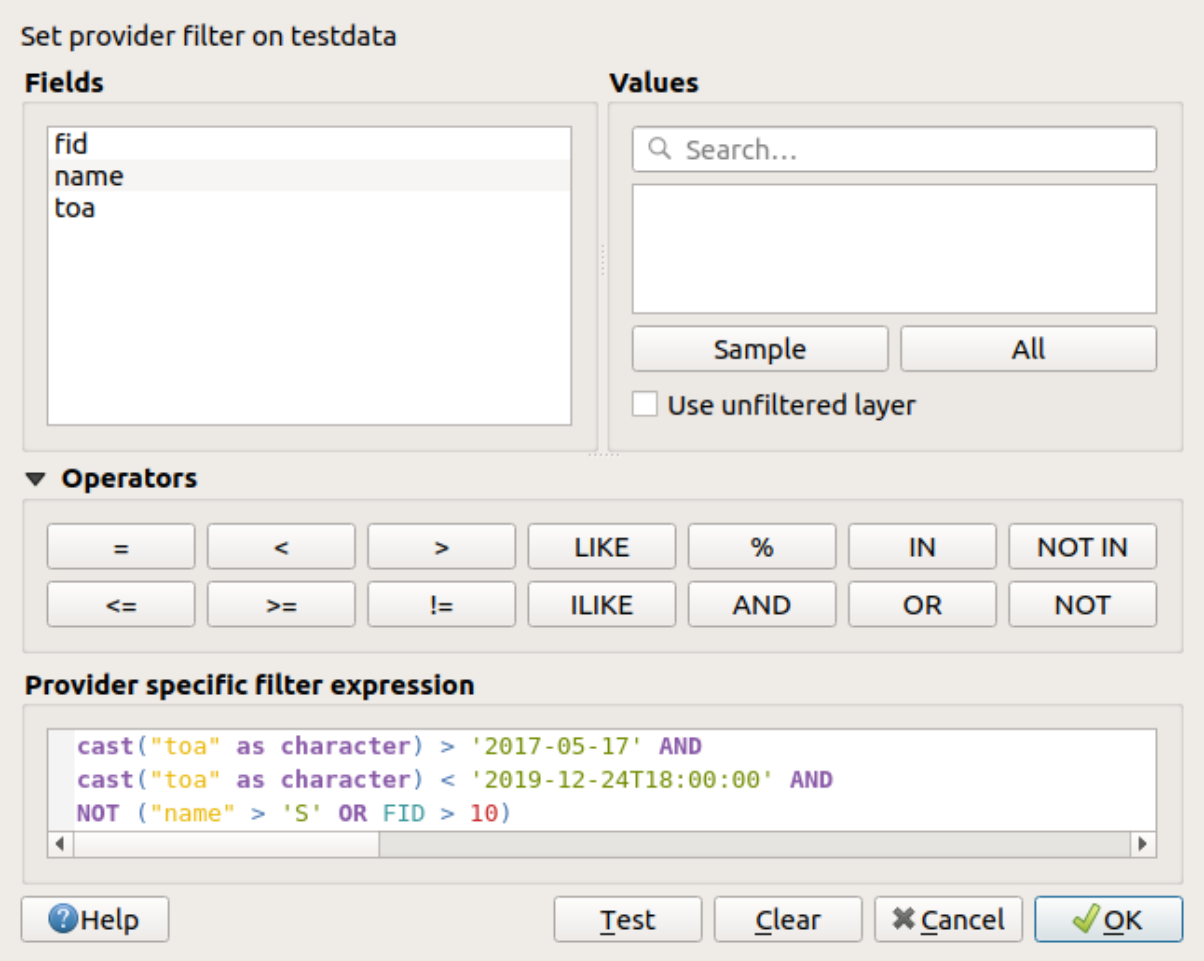


Fig. 12.2: 查询构建器

The **Fields** list contains all the fields of the layer. To add an attribute column to the SQL WHERE clause field, double-click its name or just type it into the SQL box.

The **Values** frame lists the values of the currently selected field. To list all unique values of a field, click the *All* button. To instead list the first 25 unique values of the column, click the *Sample* button. To add a value to the SQL WHERE clause field, double click its name in the Values list. You can use the search box at the top of the Values frame to easily browse and find attribute values in the list.

The **Operators** section contains all usable operators. To add an operator to the SQL WHERE clause field, click the appropriate button. Relational operators (= , > , ...), string comparison operator (`LIKE`), and logical operators (`AND`, `OR`, ...) are available.

The *Test* button helps you check your query and displays a message box with the number of features satisfying the current query. Use the *Clear* button to wipe the SQL query and revert the layer to its original state (ie, fully load all the features). It is possible to *Save...* the query as a `.QQF` file, or *Load...* the query from a file into the dialog.

When a filter is applied, QGIS treats the resulting subset acts as if it were the entire layer. For example if you applied the filter above for 'Borough' (`"TYPE_2" = 'Borough'`), you can not display, query, save or edit `Anchorage`, because that is a 'Municipality' and therefore not part of the subset.

Tip

Filtered layers are indicated in the Layers Panel

In the *Layers* panel, filtered layer is listed with a Filter icon next to it indicating the query used when the mouse

hovers over the button. Double-click the icon opens the *Query Builder* dialog for edit. This can also be achieved through the *Layer* ► *Filter...* menu.

12.1.3 Symbology Properties

The *Symbology* tab provides you with a comprehensive tool for rendering and symbolizing your vector data. You can use tools that are common to all vector data, as well as special symbolizing tools that were designed for the different kinds of vector data. However all types share the following dialog structure: in the upper part, you have a widget that helps you prepare the classification and the symbol to use for features and at the bottom the *Layer rendering* widget.

Tip

Switch quickly between different layer representations

Using the *Styles* ► *Add* menu at the bottom of the *Layer Properties* dialog, you can save as many styles as needed. A style is the combination of all properties of a layer (such as symbology, labeling, diagram, fields form, actions...) as you want. Then, simply switch between styles from the context menu of the layer in *Layers Panel* to automatically get different representations of your data.

Tip

Export vector symbology

You have the option to export vector symbology from QGIS into Google *.kml, *.dxf and MapInfo *.tab files. Just open the right mouse menu of the layer and click on *Save As...* to specify the name of the output file and its format. In the dialog, use the *Symbology export* menu to save the symbology either as *Feature symbology* ► or as *Symbol layer symbology* ►. If you have used symbol layers, it is recommended to use the second setting.

Features rendering

The renderer is responsible for drawing a feature together with the correct symbol. Regardless layer geometry type, there are four common types of renderers: single symbol, categorized, graduated and rule-based. For point layers, there are point displacement, point cluster and heatmap renderers available while polygon layers can also be rendered with the merged features, inverted polygons and 2.5 D renderers.

There is no continuous color renderer, because it is in fact only a special case of the graduated renderer. The categorized and graduated renderers can be created by specifying a symbol and a color ramp - they will set the colors for symbols appropriately. For each data type (points, lines and polygons), vector symbol layer types are available. Depending on the chosen renderer, the dialog provides different additional sections.

Note

If you change the renderer type when setting the style of a vector layer the settings you made for the symbol will be maintained. Be aware that this procedure only works for one change. If you repeat changing the renderer type the settings for the symbol will get lost.

Single Symbol Renderer

The *Single Symbol* renderer is used to render all features of the layer using a single user-defined symbol. See 符号选择器 for further information about symbol representation.

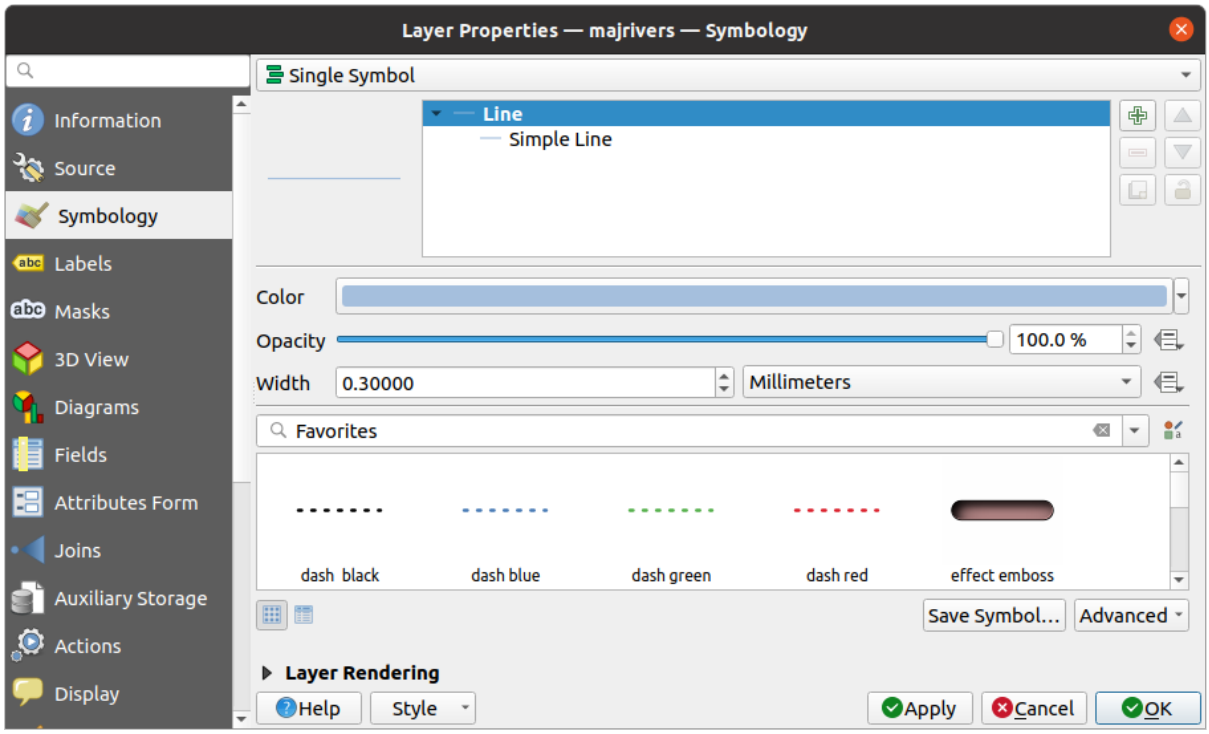


Fig. 12.3: Single symbol line properties

No Symbols Renderer

The *No Symbols* renderer is a special use case of the Single Symbol renderer as it applies the same rendering to all features. Using this renderer, no symbol will be drawn for features, but labeling, diagrams and other non-symbol parts will still be shown.

Selections can still be made on the layer in the canvas and selected features will be rendered with a default symbol. Features being edited will also be shown.

This is intended as a handy shortcut for layers which you only want to show labels or diagrams for, and avoids the need to render symbols with totally transparent fill/border to achieve this.

Categorized Renderer

The *Categorized* renderer is used to render the features of a layer, using a user-defined symbol whose aspect reflects the discrete values of a field or an expression.

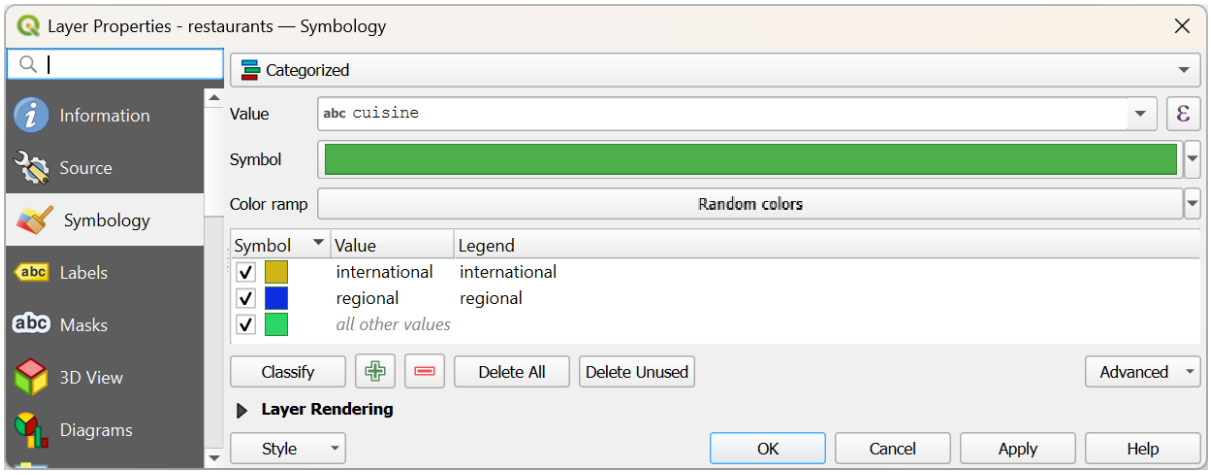


Fig. 12.4: Categorized Symbolizing options

To use categorized symbology for a layer:

1. Select the *Value* of classification: it can be an existing field or an *expression* you can type in the box or build using the associated ε button. Using expressions for categorizing avoids the need to create a field for symbology purposes only (eg, if your classification criteria are derived from one or more attributes).

   The expression used to classify features can be of any type; eg, it can:

   - be a comparison. In this case, QGIS returns values `1` (**True**) and `0` (**False**). Some examples:

     ```
     myfield >= 100
     $id = @atlas_featureid
     myfield % 2 = 0
     within( $geometry, @atlas_geometry )
     ```

   - combine different fields:

     ```
     concat( field_1, ' ', field_2 )
     ```

   - be a calculation on fields:

     ```
     myfield % 2
     year( myfield )
     field_1 + field_2
     substr( field_1, -3 )
     ```

   - be used to transform linear values to discrete classes, e.g.:

     ```
     CASE WHEN x > 1000 THEN 'Big' ELSE 'Small' END
     ```

   - combine several discrete values into a single category, e.g.:

     ```
     CASE
     WHEN building IN ('residence', 'mobile home') THEN 'residential'
     WHEN building IN ('commercial', 'industrial') THEN 'Commercial and␣
     ↪Industrial'
     END
     ```

> **Tip**
>
> While you can use any kind of expression to categorize features, for some complex expressions it might be simpler to use *rule-based rendering*.

2. Configure the *Symbol*, which will be used as base symbol for all the classes;
3. Indicate the *Color ramp*, i.e. the range of colors from which the color applied to each symbol is selected.

   Besides the common options of the *color ramp widget*, you can apply a *Random Color Ramp* to the categories. You can click the *Shuffle Random Colors* entry to regenerate a new set of random colors if you are not satisfied.
4. Then click on the *Classify* button to create classes from the distinct values of the provided field or expression.
5. For each class, you can edit the *Legend* column to a more meaningful label (used in the *Layers* panel and the print layout).

   Note

   Keep in mind that some values may use widgets that do not display the actual value stored in the field. For example, a checkbox widget may store `1` and `0` for checked and unchecked states, while displaying `True` and `False` labels. In this case, to categorize features based on the checkbox state, you need to use the stored values (`1` and `0`) in the expression. QGIS will automatically use the display value for the legend column.
6. *Apply* the changes if the *live update* is not in use and each feature on the map canvas will be rendered with the symbol of its class.

   By default, QGIS appends an *all other values* class to the list. While empty at the beginning, this class is used as a default class for any feature not falling into the other classes (eg, when you create features with new values for the classification field / expression).

Further tweaks can be done to the default classification:

- You can Add new categories, Remove selected categories, *Delete All* of them or *Delete Unused* categories.
- A class can be disabled by unchecking the checkbox to the left of the class name; the corresponding features are hidden on the map.
- Drag-and-drop the rows to reorder the classes
- To change the symbol, the value or the legend of a class, double click the item.

Right-clicking over selected item(s) shows a contextual menu to:

- *Copy Symbol* and *Paste Symbol*, a convenient way to apply the item's representation to others
- *Change Color...* of the selected symbol(s)
- *Change Opacity...* of the selected symbol(s)
- *Change Output Unit...* of the selected symbol(s)
- *Change Width...* of the selected line symbol(s)
- *Change Size...* of the selected point symbol(s)
- *Change Angle...* of the selected point symbol(s)
- *Merge Categories*: Groups multiple selected categories into a single one. This allows simpler styling of layers with a large number of categories, where it may be possible to group numerous distinct categories into a smaller and more manageable set of categories which apply to multiple values.

  Tip

  Since the symbol kept for the merged categories is the one of the topmost selected category in the list, you may want to move the category whose symbol you wish to reuse to the top before merging.
- *Unmerge Categories* that were previously merged

The created classes also appear in a tree hierarchy in the *Layers* panel. Double-click an entry in the map legend to edit the assigned symbol. Right-click and you will get some *more options*.

The *Advanced* menu gives access to options to speed classification or fine-tune the symbols rendering:

- *Match to saved symbols*: Using the *symbols library*, assigns to each category a symbol whose name represents the classification value of the category
- *Match to symbols from file...*: Provided a file with symbols, assigns to each category a symbol whose name represents the classification value of the category
- *Symbol levels...* to define the order of symbols rendering.

Graduated Renderer

The *Graduated* renderer is used to render all the features from a layer, using an user-defined symbol whose color or size reflects the assignment of a selected feature's attribute to a class.

Like the Categorized Renderer, the Graduated Renderer allows you to define rotation and size scale from specified columns.

Also, analogous to the Categorized Renderer, it allows you to select:

- The *Value* of classification: it can be an existing field or an *expression* you can type in the box or build using the associated button. Using expressions for graduating avoids the need to create a field for symbology purposes only (eg, if your classification criteria are derived from one or more attributes).
- The symbol (using the Symbol selector dialog)
- The legend format and the precision
- The method to use to change the symbol: color or size
- The colors (using the color Ramp list) if the color method is selected
- The size (using the size domain and its unit)

Then you can use the Histogram tab which shows an interactive histogram of the values from the assigned field or expression. Class breaks can be moved or added using the histogram widget.

> **Note**
>
> You can use Statistical Summary panel to get more information on your vector layer. See 统计摘要面板.

Back to the Classes tab, you can specify the number of classes and also the mode for classifying features within the classes (using the Mode list). The available modes are:

- Equal Count (Quantile): each class will have the same number of elements (the idea of a boxplot).
- Equal Interval: each class will have the same size (e.g. with the values from 1 to 16 and four classes, each class will have a size of four).
- Fixed Interval: each class will have a fixed range of values (e.g. with the values from 1 to 16 and an interval size of 4, the classes will be 1-4, 5-8, 9-12 and 13-16).
- Logarithmic scale: suitable for data with a wide range of values. Narrow classes for low values and wide classes for large values (e.g. for decimal numbers with range [0..100] and two classes, the first class will be from 0 to 10 and the second class from 10 to 100).
- Natural Breaks (Jenks): the variance within each class is minimized while the variance between classes is maximized.

- Pretty Breaks: computes a sequence of about n+1 equally spaced nice values which cover the range of the values in x. The values are chosen so that they are 1, 2 or 5 times a power of 10. (based on pretty from the R statistical environment https://www.rdocumentation.org/packages/base/topics/pretty).
- Standard Deviation: classes are built depending on the standard deviation of the values.

The listbox in the center part of the *Symbology* tab lists the classes together with their ranges, labels and symbols that will be rendered.

Click on **Classify** button to create classes using the chosen mode. Each classes can be disabled unchecking the checkbox at the left of the class name.

To change symbol, value and/or label of the class, just double click on the item you want to change.

Right-clicking over selected item(s) shows a contextual menu to:

- *Copy Symbol* and *Paste Symbol*, a convenient way to apply the item's representation to others
- *Change Color...* of the selected symbol(s)
- *Change Opacity...* of the selected symbol(s)
- *Change Output Unit...* of the selected symbol(s)
- *Change Width...* of the selected line symbol(s)
- *Change Size...* of the selected point symbol(s)
- *Change Angle...* of the selected point symbol(s)

The example in Fig. 12.5 shows the graduated rendering dialog for the major_rivers layer of the QGIS sample dataset.

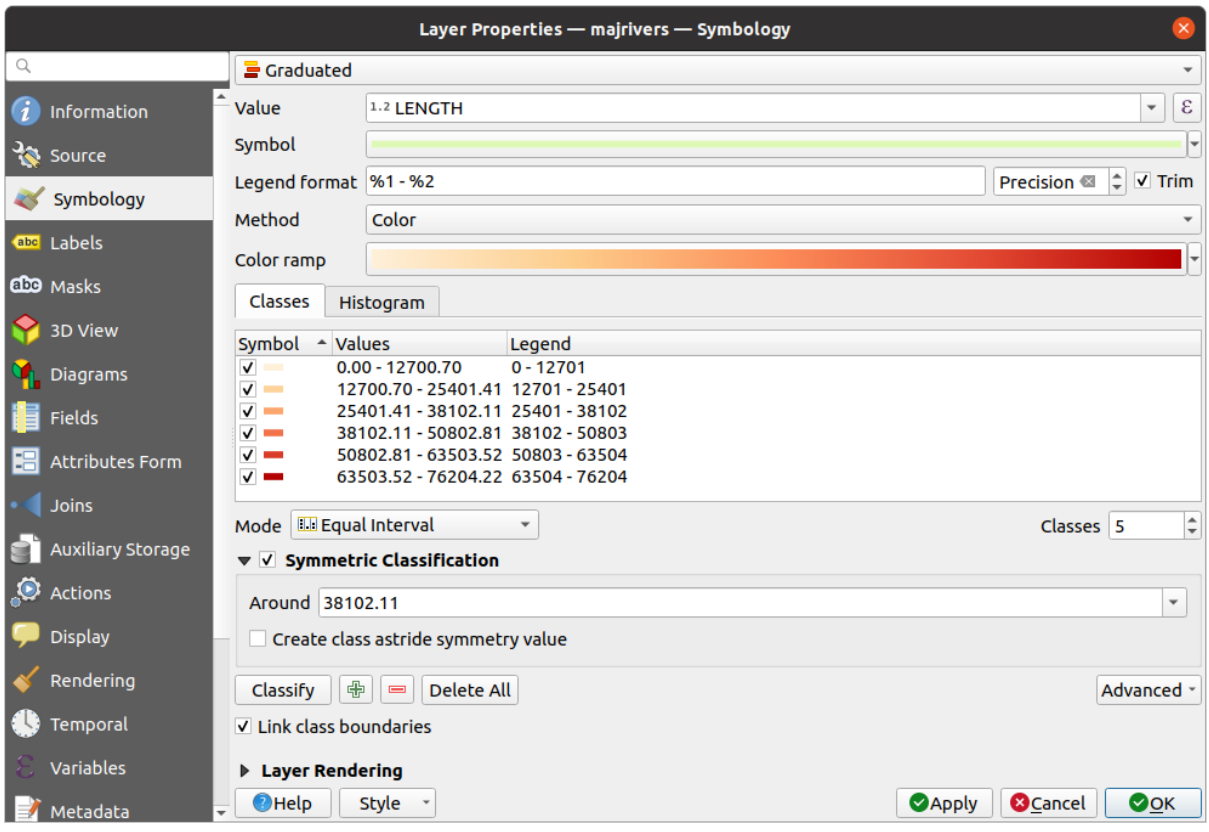


Fig. 12.5: Graduated Symbolizing options

The created classes also appear in a tree hierarchy in the *Layers* panel. Double-click an entry in the map legend to edit the assigned symbol. Right-click and you will get some *more options*.

Proportional Symbol and Multivariate Analysis

Proportional Symbol and Multivariate Analysis are not rendering types available from the Symbology rendering drop-down list. However with the *data-defined override* options applied over any of the previous rendering options, QGIS allows you to display your point and line data with such representation.

Creating proportional symbol

To apply a proportional rendering:

1. First apply to the layer the *single symbol renderer*.
2. Then set the symbol to apply to the features.
3. Select the item at the upper level of the symbol tree, and use the Data-defined override *button* next to the *Size* (for point layer) or *Width* (for line layer) option.
4. Select a field or enter an expression, and for each feature, QGIS will apply the output value to the property and proportionally resize the symbol in the map canvas.

   If need be, use the *Size assistant...* option of the menu to apply some transformation (exponential, flannery...) to the symbol size rescaling (see *Using the data-defined assistant interface* for more details).

You can choose to display the proportional symbols in the *Layers panel* and the *print layout legend item*: unfold the *Advanced* drop-down list at the bottom of the main dialog of the *Symbology* tab and select **Data-defined size legend...** to configure the legend items (see *Data-defined size legend* for details).

Creating multivariate analysis

A multivariate analysis rendering helps you evaluate the relationship between two or more variables e.g., one can be represented by a color ramp while the other is represented by a size.

The simplest way to create multivariate analysis in QGIS is to:

1. First apply a categorized or graduated rendering on a layer, using the same type of symbol for all the classes.
2. Then, apply a proportional symbology on the classes:
   1. Click on the *Change* button above the classification frame: you get the 符号选择器 dialog.
   2. Rescale the size or width of the symbol layer using the *data defined override* widget as seen above.

Like the proportional symbol, the scaled symbology can be added to the layer tree, on top of the categorized or graduated classes symbols using the *data defined size legend* feature. And both representation are also available in the print layout legend item.

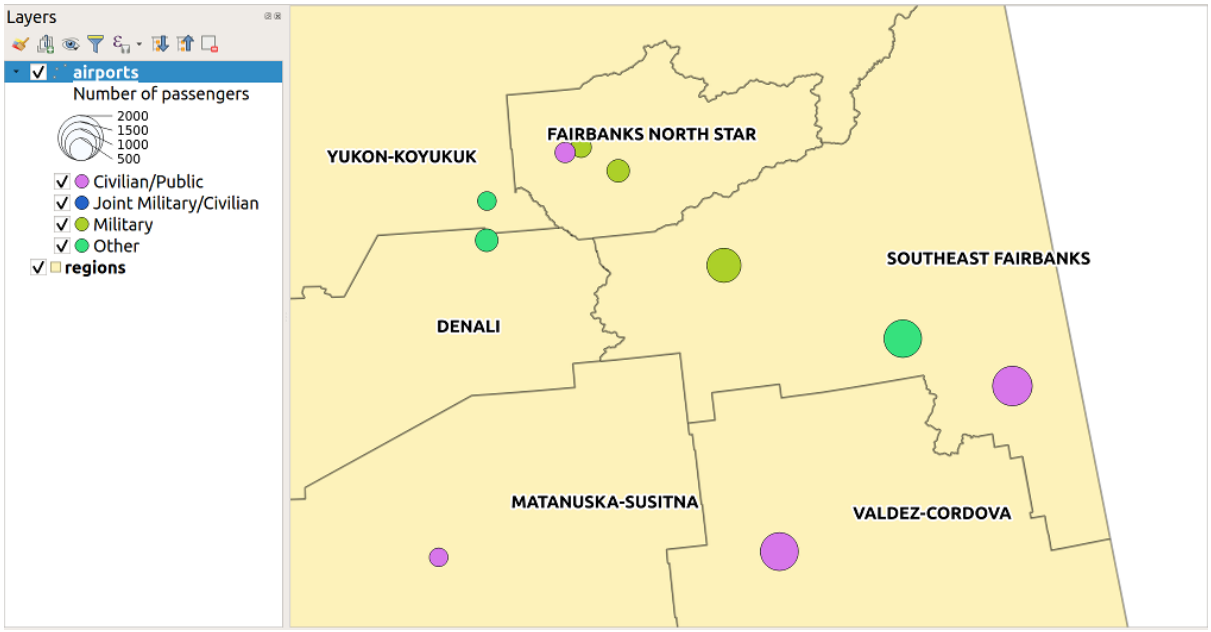


Fig. 12.6: Multivariate example with scaled size legend

Rule-based Renderer

Rules are QGIS *expressions* used to discriminate features according to their attributes or properties in order to apply specific rendering settings to them. Rules can be nested, and features belong to a class if they belong to all the upper nesting level(s).

The *Rule-based* renderer is thus designed to render all the features from a layer, using symbols whose aspect reflects the assignment of a selected feature to a fine-grained class.

To create a rule:

1. Activate an existing row by double-clicking it (by default, QGIS adds a symbol without a rule when the rendering mode is enabled) or click the Edit rule or Add rule button.
2. In the *Edit Rule* dialog that opens, you can define a label to help you identify each rule. This is the label that will be displayed in the *Layers Panel* and also in the print composer legend.
3. Manually enter an expression in the text box next to the *Filter* option or press the button next to it to open the expression string builder dialog.
4. Use the provided functions and the layer attributes to build an *expression* to filter the features you'd like to retrieve. Press the *Test* button to check the result of the query.
5. You can enter a longer label to complete the rule description.
6. You can use the *Scale Range* option to set scales at which the rule should be applied.
7. Finally, configure the *symbol to use* for these features.
8. And press *OK*.

A new row summarizing the rule is added to the Layer Properties dialog. You can create as many rules as necessary following the steps above or copy pasting an existing rule. Drag-and-drop the rules on top of each other to nest them and refine the upper rule features in subclasses.

The rule-based renderer can be combined with categorized or graduated renderers. Selecting a rule, you can organize its features in subclasses using the *Refine selected rules* drop-down menu. Refined classes appear like sub-items of the

rule, in a tree hierarchy and like their parent, you can set the symbology and the rule of each class. Automated rule refinement can be based on:

- **scales**: given a list of scales, this option creates a set of classes to which the different user-defined scale ranges apply. Each new scale-based class can have its own symbology and expression of definition. This can e.g. be a convenient way to display the same features with various symbols at different scales, or display only a set of features depending on the scale (e.g. local airports at large scale vs international airports at small scale).
- **categories**: applies a *categorized renderer* to the features falling in the selected rule.
- or **ranges**: applies a *graduated renderer* to the features falling in the selected rule.

Refined classes appear like sub-items of the rule, in a tree hierarchy and like above, you can set symbology of each class. Symbols of the nested rules are stacked on top of each other so be careful in choosing them. It is also possible to uncheck *Symbols* in the *Edit rule* dialog to avoid rendering a particular symbol in the stack.

In the *Edit rule* dialog, you can avoid writing all the rules and make use of the *Else* option to catch all the features that do not match any of the other rules, at the same level. This can also be achieved by writing `Else` in the *Rule* column of the *Layer Properties* ► *Symbology* ► *Rule-based* dialog.

Right-clicking over selected item(s) shows a contextual menu to:

- *Copy* and *Paste*, a convenient way to create new item(s) based on existing item(s)
- *Copy Symbol* and *Paste Symbol*, a convenient way to apply the item's representation to others
- *Change Color...* of the selected symbol(s)
- *Change Opacity...* of the selected symbol(s)
- *Change Output Unit...* of the selected symbol(s)
- *Change Width...* of the selected line symbol(s)
- *Change Size...* of the selected point symbol(s)
- *Change Angle...* of the selected point symbol(s)
- *Refine Current Rule*: open a submenu that allows to refine the current rule with **scales**, **categories** or **Ranges**. Same as selecting the *corresponding menu* at the bottom of the dialog.

Unchecking a row in the rule-based renderer dialog hides in the map canvas the features of the specific rule and the nested ones.

The created rules also appear in a tree hierarchy in the map legend. Double-click an entry in the map legend to edit the assigned symbol. Right-click and you will get some *more options*.

The example in Fig. 12.7 shows the rule-based rendering dialog for the rivers layer of the QGIS sample dataset.

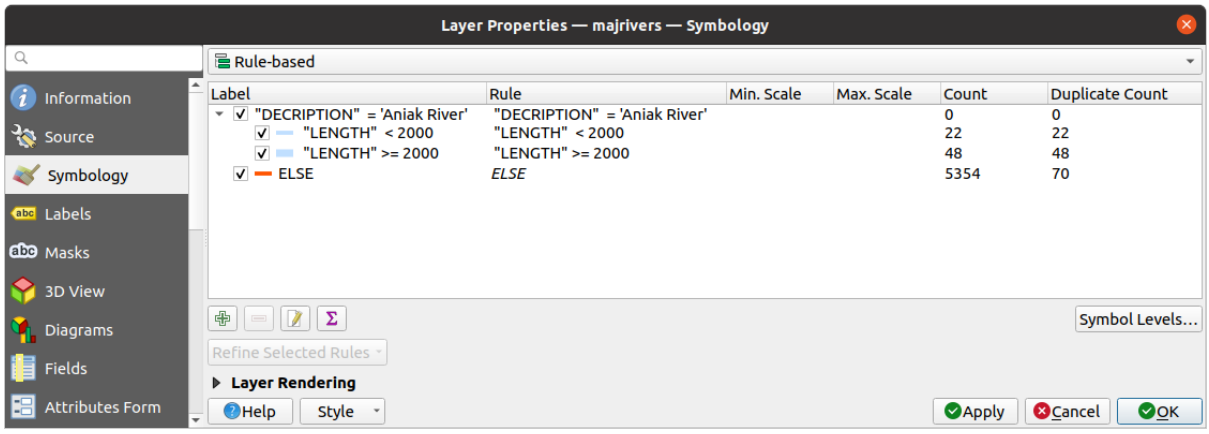


Fig. 12.7: Rule-based Symbolizing options

Point displacement Renderer

The *Point Displacement* renderer takes the point features falling in a given distance tolerance from each other and places their symbols around their barycenter, following different placement methods. This can be a convenient way to visualize all the features of a point layer, even if they have the same location (e.g. amenities in a building).

To configure a point displacement renderer, you have to:

1. Set the *Center symbol*: how the virtual point at the center will look like
2. Select the *Renderer* type: how you want to classify features in the layer (single, categorized, rule-based...)
3. Press the *Renderer Settings...* button to configure features' symbology according to the selected renderer
4. Indicate the *Distance* tolerance in which close features are considered overlapping and then displaced over the same virtual point. Supports common symbol units.
5. Configure the *Placement methods*:
   - **Ring**: places all the features on a circle whose radius depends on the number of features to display.
   - **Concentric rings**: uses a set of concentric circles to show the features.
   - **Grid**: generates a regular grid with a point symbol at each intersection.
6. Displaced symbols are placed on the *Displacement lines*. While the minimal spacing of the displacement lines depends on the point symbols renderer, you can still customize some of their settings such as the *Stroke width*, *Stroke color* and *Size adjustment* (e.g., to add more spacing between the rendered points).
7. Use the *Labels* group options to perform points labeling: the labels are placed near the displaced symbol, and not at the feature real position.
   1. Select the *Label attribute*: a field of the layer to use for labeling
   2. Indicate the *Label font* properties and size
   3. Pick a *Label color*
   4. Set a *Label distance factor*: for each point feature, offsets the label from the symbol center proportionally to the symbol's diagonal size.
   5. Turn on *Use scale dependent labeling* if you want to display labels only on scales larger than a given *Minimum map scale*.

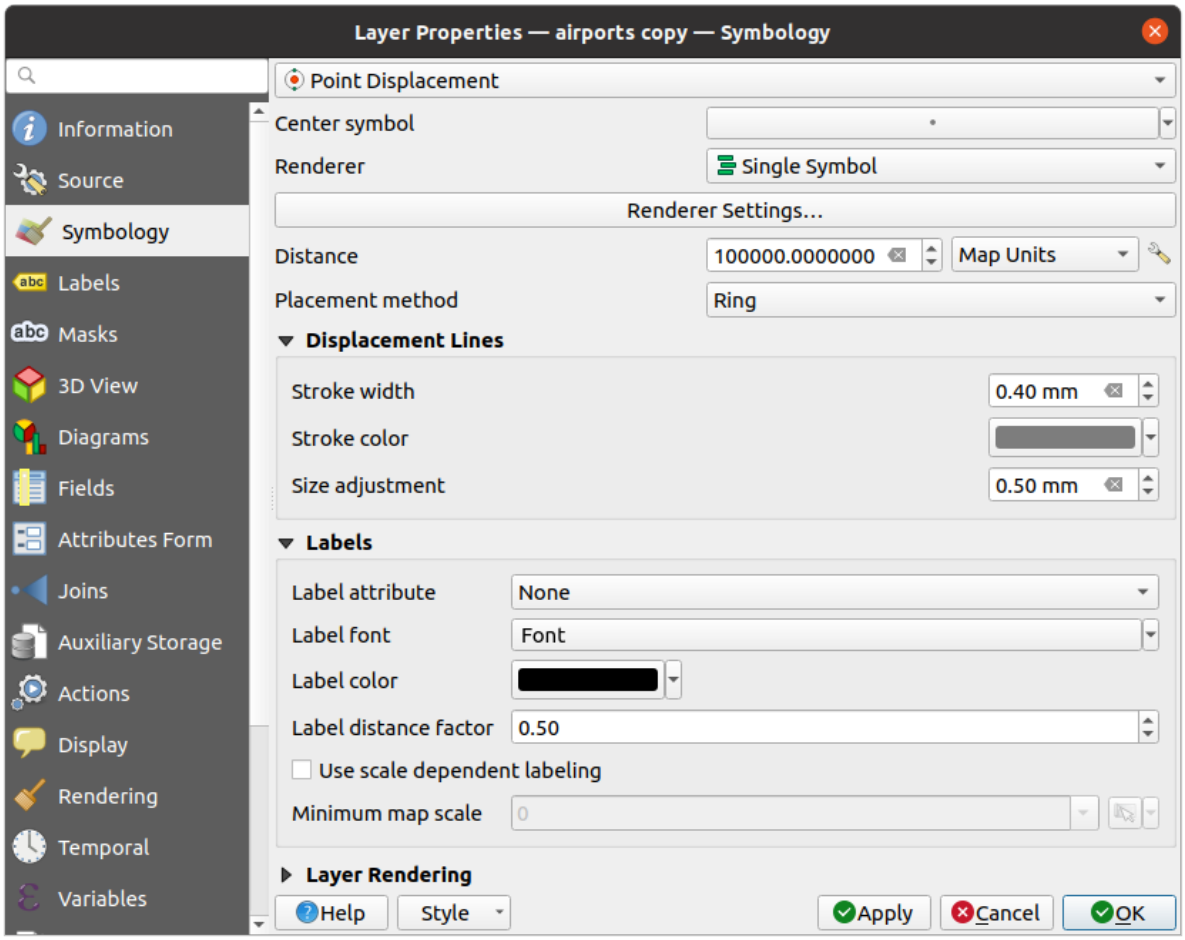


Fig. 12.8: Point displacement dialog

Note

Point Displacement renderer does not alter feature geometry, meaning that points are not moved from their position. They are still located at their initial place. Changes are only visual, for rendering purpose. Use instead the Processing *Points displacement* algorithm if you want to create displaced features.

Point Cluster Renderer

Unlike the *Point Displacement* renderer which blows up nearest or overlaid point features placement, the *Point Cluster* renderer groups nearby points into a single rendered marker symbol. Points that fall within a specified distance from each others are merged into a single symbol. Points aggregation is made based on the closest group being formed, rather than just assigning them the first group within the search distance.

From the main dialog, you can:

1. Set the symbol to represent the point cluster in the *Cluster symbol*; the default rendering displays the number of aggregated features thanks to the `@cluster_size` *variable* on Font marker symbol layer.
2. Select the *Renderer* type, i.e. how you want to classify features in the layer (single, categorized, rule-based...)
3. Press the *Renderer Settings...* button to configure features' symbology as usual. Note that this symbology is only visible on features that are not clustered, the *Cluster symbol* being applied otherwise. Also, when all the point

features in a cluster belong to the same rendering class, and thus would be applied the same color, that color represents the `@cluster_color` variable of the cluster.

4. Indicate the maximal *Distance* to consider for clustering features. Supports common symbol units.

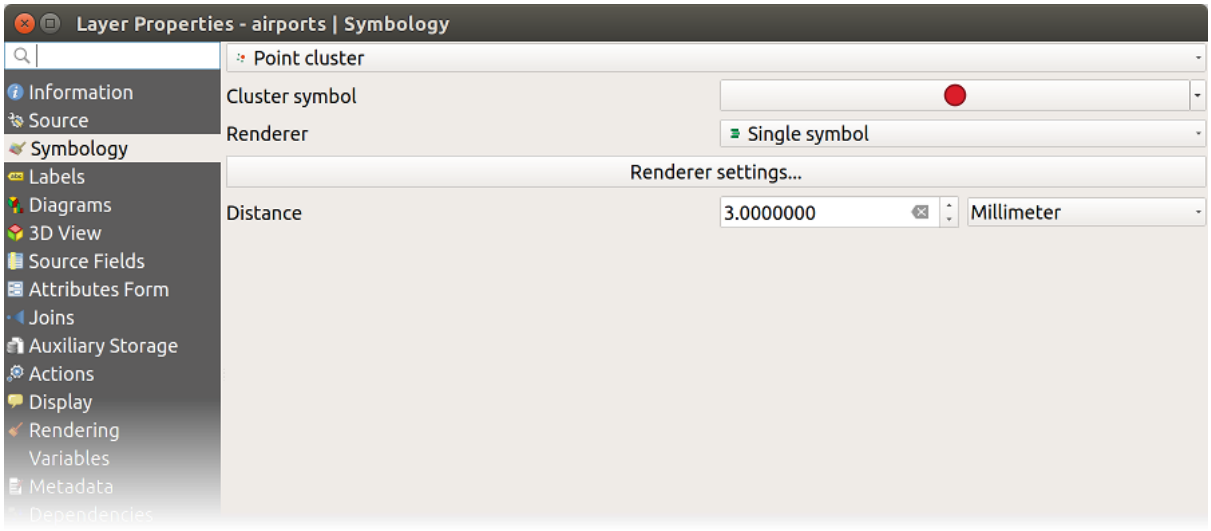


Fig. 12.9: Point Cluster dialog

Note

Point Cluster renderer does not alter feature geometry, meaning that points are not moved from their position. They are still located at their initial place. Changes are only visual, for rendering purpose. Use instead the Processing *K-means clustering* or *DBSCAN clustering* algorithm if you want to create cluster-based features.

Merged Features Renderer

The *Merged Features* renderer allows area and line features to be "dissolved" into a single object prior to rendering to ensure that complex symbols or overlapping features are represented by a uniform and contiguous cartographic symbol.

Inverted Polygon Renderer

The *Inverted Polygon* renderer allows user to define a symbol to fill in outside of the layer's polygons. As above you can select subrenderers, namely Single symbol, Graduated, Categorized, Rule-Based or 2.5D renderer.

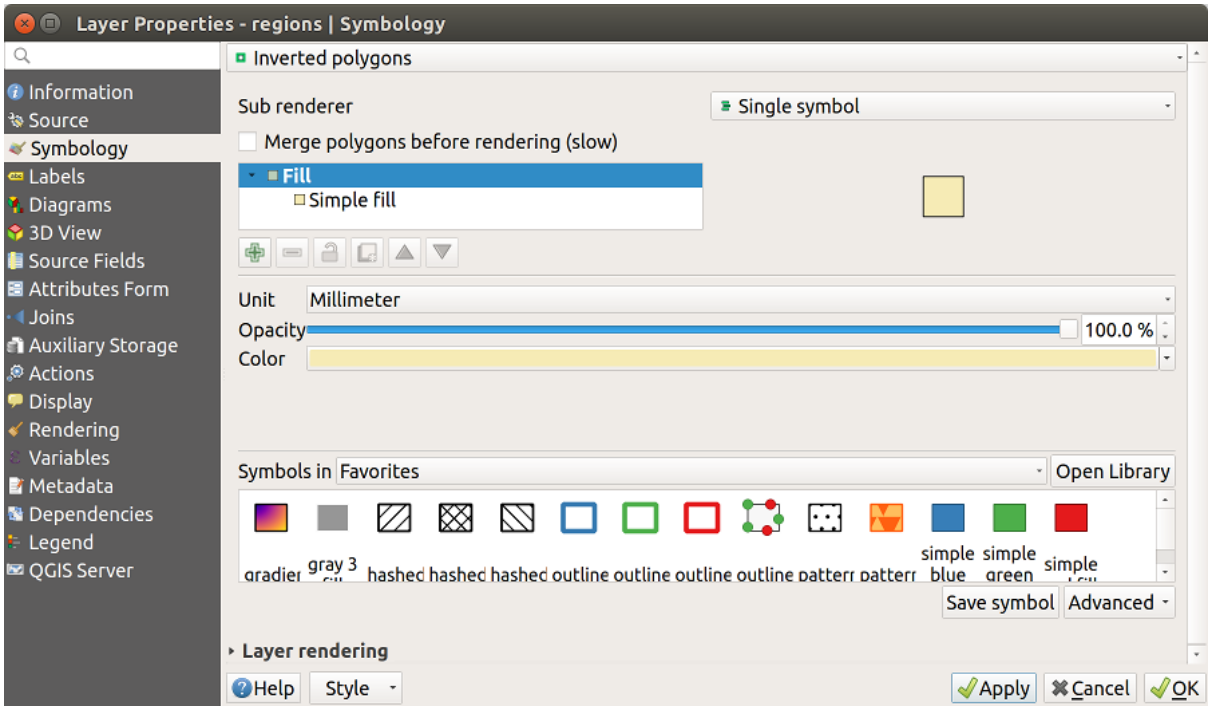


Fig. 12.10: Inverted Polygon dialog

Heatmap Renderer

With the *Heatmap* renderer you can create live dynamic heatmaps for (multi)point layers. You can specify the heatmap *Radius* in millimeters, points, pixels, map units or inches, choose and edit a *Color ramp* for the heatmap style and use a slider for selecting a trade-off between render speed and quality. You can also define a *Maximum value* limit and *Weight points by* using a field or an expression.

Use Data defined override to dynamically control *Radius* and *Maximum value* based on the attributes of your data. For example, the radius of a heatmap point could be determined by its population attribute, or the maximum value could be based on a temporal range.

When adding or removing a feature the heatmap renderer updates the heatmap style automatically. The *Color ramp* will be shown as a legend bar and in the *Legend settings* you can set the *Labels* for the *Maximum* and *Minimum* values. You can also change the orientation and direction of the legend in the *Layout*.

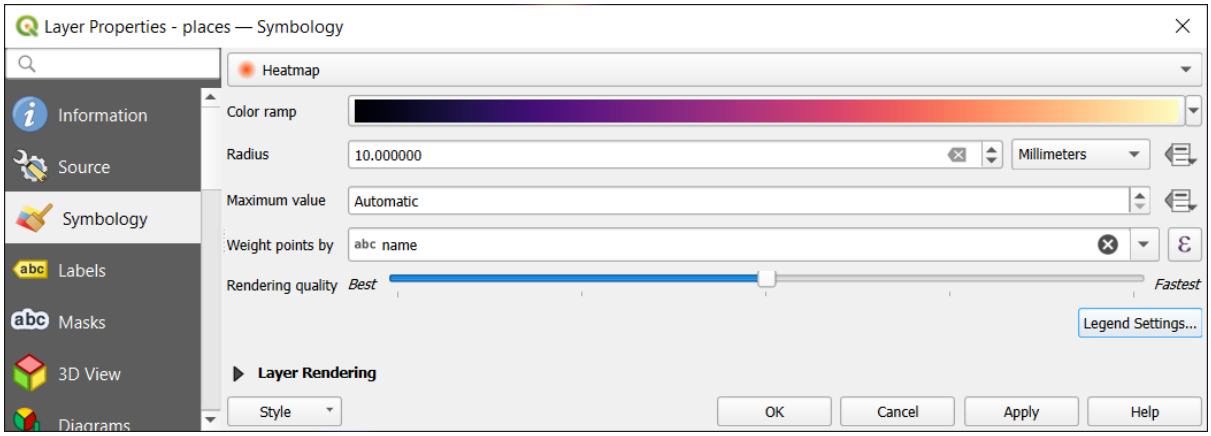


Fig. 12.11: Heatmap dialog

2.5D Renderer

Using the *2.5D* renderer it's possible to create a 2.5D effect on your layer's features. You start by choosing a *Height* value (in map units). For that you can use a fixed value, one of your layer's fields, or an expression. You also need to choose an *Angle* (in degrees) to recreate the viewer position (0° means west, growing in counter clock wise). Use advanced configuration options to set the *Roof Color* and *Wall Color*. If you would like to simulate solar radiation on the features walls, make sure to check the *Shade walls based on aspect* option. You can also simulate a shadow by setting a *Color* and *Size* (in map units).

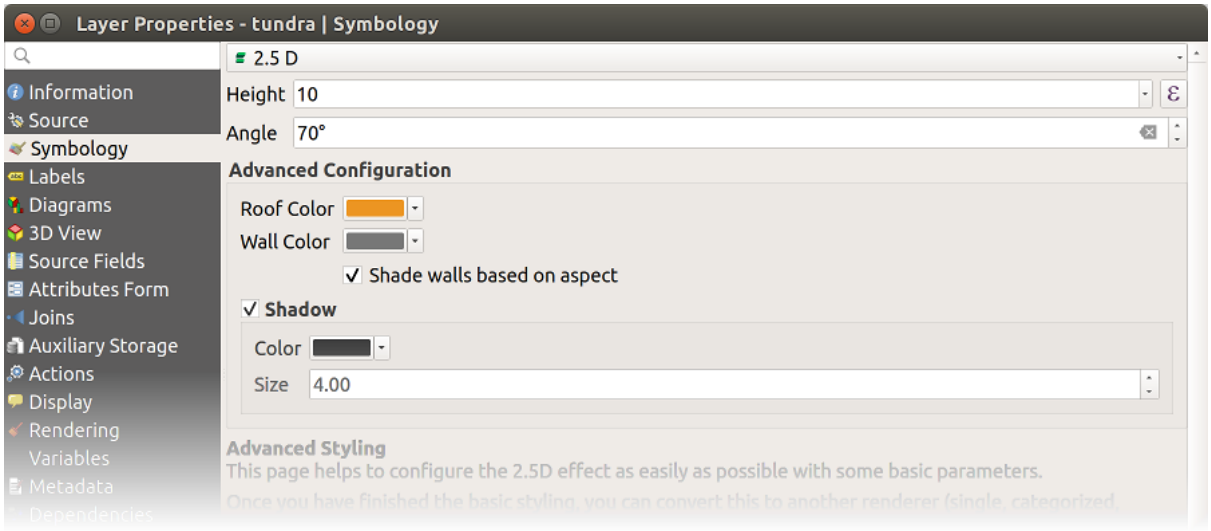


Fig. 12.12: 2.5D dialog

> **Tip**
>
> **Using 2.5D effect with other renderers**
>
> Once you have finished setting the basic style on the 2.5D renderer, you can convert this to another renderer (single, categorized, graduated). The 2.5D effects will be kept and all other renderer specific options will be available for you to fine tune them (this way you can have for example categorized symbols with a nice 2.5D representation or add some extra styling to your 2.5D symbols). To make sure that the shadow and the "building"

itself do not interfere with other nearby features, you may need to enable Symbols Levels (*Advanced* ► *Symbol levels...*). The 2.5D height and angle values are saved in the layer's variables, so you can edit it afterwards in the variables tab of the layer's properties dialog.

Embedded Renderer

The *Embedded Symbols* renderer allows to display the 'native' symbology of a provided datasource. This is mostly the case with `KML` and `TAB` datasets that have predefined symbology.

Layer rendering

From the Symbology tab, you can also set some options that invariably act on all features of the layer:

- *Opacity* : You can make the underlying layer in the map canvas visible with this tool. Use the slider to adapt the visibility of your vector layer to your needs. You can also make a precise definition of the percentage of visibility in the menu beside the slider.
- *Blending mode* at the *Layer* and *Feature* levels: You can achieve special rendering effects with these tools that you may previously only know from graphics programs. The pixels of your overlaying and underlaying layers are mixed through the settings described in *Blending Modes*.
- Apply *paint effects* on all the layer features with the *Draw Effects* button.
- *Control feature rendering order* allows you, using features attributes, to define the z-order in which they shall be rendered. Activate the checkbox and click on the button beside. You then get the *Define Order* dialog in which you:

  1. Choose a field or build an expression to apply to the layer features.
  2. Set in which order the fetched features should be sorted, i.e. if you choose **Ascending** order, the features with lower value are rendered under those with higher value.
  3. Define when features returning NULL value should be rendered: **first** (bottom) or **last** (top).
  4. Repeat the above steps as many times as rules you wish to use.

  The first rule is applied to all the features in the layer, z-ordering them according to their returned value. Then, within each group of features with the same value (including those with NULL value) and thus the same z-level, the next rule is applied to sort them. And so on...

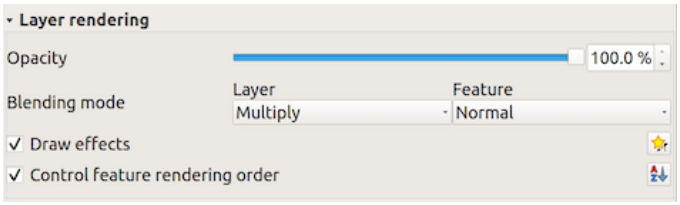


Fig. 12.13: Layer rendering options

Other Settings

Symbol levels

For renderers that allow stacked symbol layers (only heatmap doesn't) there is an option to control the rendering order of each symbol's levels.

For most of the renderers, you can access the Symbols levels option by clicking the *Advanced* button below the saved symbols list and choosing *Symbol levels*. For the *Rule-based Renderer* the option is directly available through *Symbols Levels...* button, while for *Point displacement Renderer* renderer the same button is inside the *Rendering settings* dialog.

To activate symbols levels, select the *Enable symbol levels*. Each row will show up a small sample of the combined symbol, its label and the individual symbols layer divided into columns with a number next to it. The numbers represent the rendering order level in which the symbol layer will be drawn. Lower values levels are drawn first, staying at the bottom, while higher values are drawn last, on top of the others.

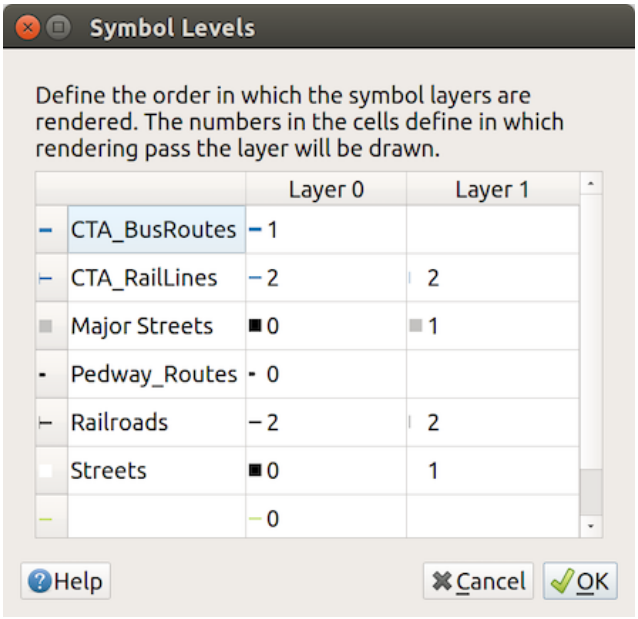


Fig. 12.14: Symbol levels dialog

Note

If symbols levels are deactivated, the complete symbols will be drawn according to their respective features order. Overlapping symbols will simply obfuscate to other below. Besides, similar symbols won't "merge" with each other.

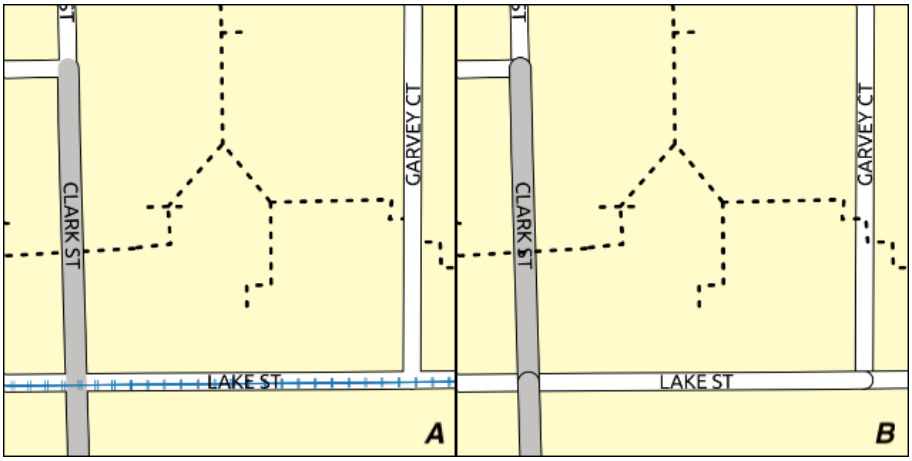


Fig. 12.15: Symbol levels activated (A) and deactivated (B) difference

Data-defined size legend

When a layer is rendered with the *proportional symbol or the multivariate rendering* or when a *scaled size diagram* is applied to the layer, you can allow the display of the scaled symbols in both the *Layers panel* and the *print layout legend*.

To enable the *Data-defined Size Legend* dialog to render symbology, select the eponym option in the *Advanced* button below the saved symbols list. For diagrams, the option is available under the *Legend* tab. The dialog provides the following options to:

- select the type of legend: *Legend not enabled*, *Separated legend items* and *Collapsed legend*. For the latter option, you can select whether the legend items are aligned at the **Bottom** or at the **Center**;
- preview the *symbol to use* for legend representation;
- insert the title in the legend;
- resize the classes to use: by default, QGIS provides you with a legend of five classes (based on natural pretty breaks) but you can apply your own classification using the *Manual size classes* option. Use the and buttons to set your custom classes values and labels.
- For collapsed legend, it's possible to:
  - *Align symbols* in the center or the bottom
  - configure the horizontal leader *Line symbol* from the symbol to the corresponding legend text.

A preview of the legend is displayed in the right panel of the dialog and updated as you set the parameters.

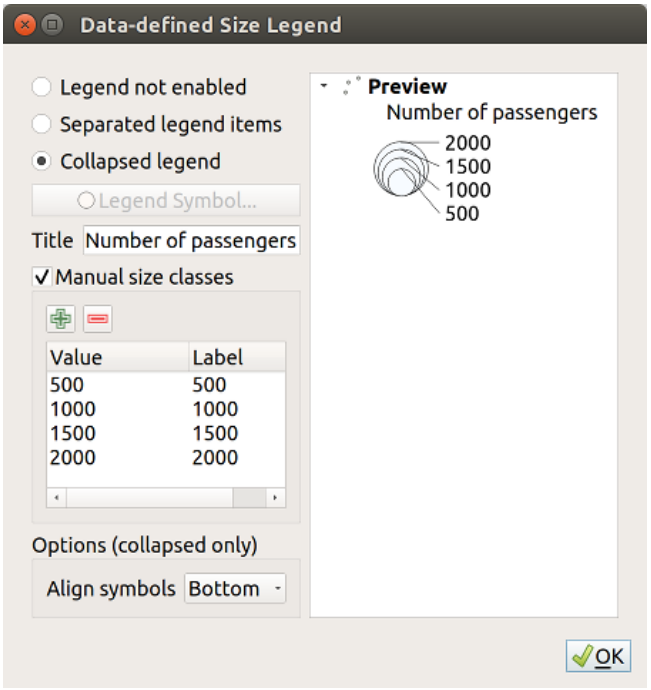


Fig. 12.16: Setting size scaled legend

Note

Currently, data-defined size legend for layer symbology can only be applied to point layer using single, categorized or graduated symbology.

Animation settings

To allow any symbol to become an *animated symbol*, you can utilize *Animation settings* panel. In this panel, you can enable animation for the symbol and set a specific frame rate for the symbol's redrawing.

1. Start by going to the top symbol level and select *Advanced* menu in the bottom right of the dialog
2. Find *Animation settings* option
3. Check *Is Animated* to enable animation for the symbol
4. Configure the *Frame rate*, i.e. how fast the animation would be played
5. You can now use `@symbol_frame` variable in any sub-symbol data defined property in order to animate that property.

For example, setting the symbol's rotation to data defined expression `@symbol_frame % 360` will cause the symbol to rotate over time, with rotation speed dictated by the symbol's frame rate:

Extent buffer

You may set an extent buffer for a *symbol*. This means that a buffer is applied to the current map extent so that if a feature is outside of the actual map extent but inside the buffered extent it will still be rendered. This is useful for example with symbols which use the geometry generator where you would like to still see the generated geometries even if the actual feature is outside of the map extent.

To edit the extent buffer you can utilize the *Extent buffer* panel.

1. Start by going to the top symbol level and select *Advanced* menu in the bottom right of the dialog
2. Find *Extent buffer* option
3. In the new panel you can set the buffer distance

The buffer distance units can be changed. You can also control the distance value by using the data defined override widget. For example you can change the value based on the current map scale `if(@map_scale > 50000, 5000, 0)`:

Draw effects

In order to improve layer rendering and avoid (or at least reduce) the resort to other software for final rendering of maps, QGIS provides another powerful functionality: the *Draw Effects* options, which adds paint effects for customizing the visualization of vector layers.

The option is available in the *Layer Properties* ► *Symbology* dialog, under the *Layer rendering* group (applying to the whole layer) or in *symbol layer properties* (applying to corresponding features). You can combine both usage.

Paint effects can be activated by checking the *Draw effects* option and clicking the Customize effects button. That will open the *Effect Properties* Dialog (see Fig. 12.17). The following effect types, with custom options are available:

- **Source**: Draws the feature's original style according to the configuration of the layer's properties. The *Opacity* of its style can be adjusted as well as the *Blend mode* and *Draw mode*. These are common properties for all types of effects.

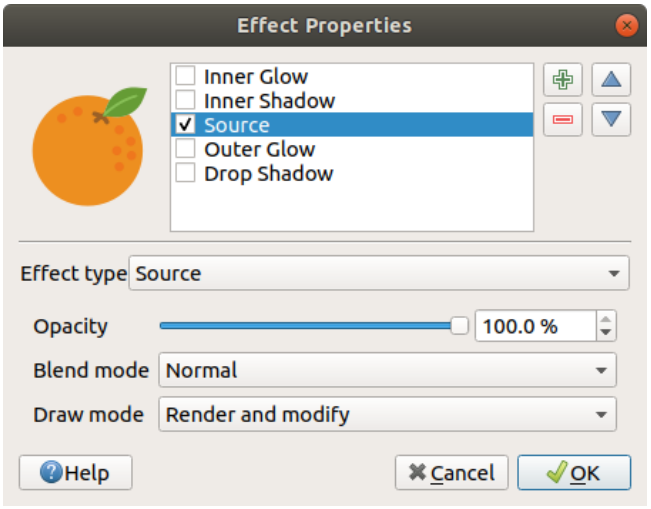


Fig. 12.17: Draw Effects: Source dialog

- **Blur**: Adds a blur effect on the vector layer. The custom options that you can change are the *Blur type* (*Stack blur (fast)* or *Gaussian blur (quality)*) and the *Blur strength*.

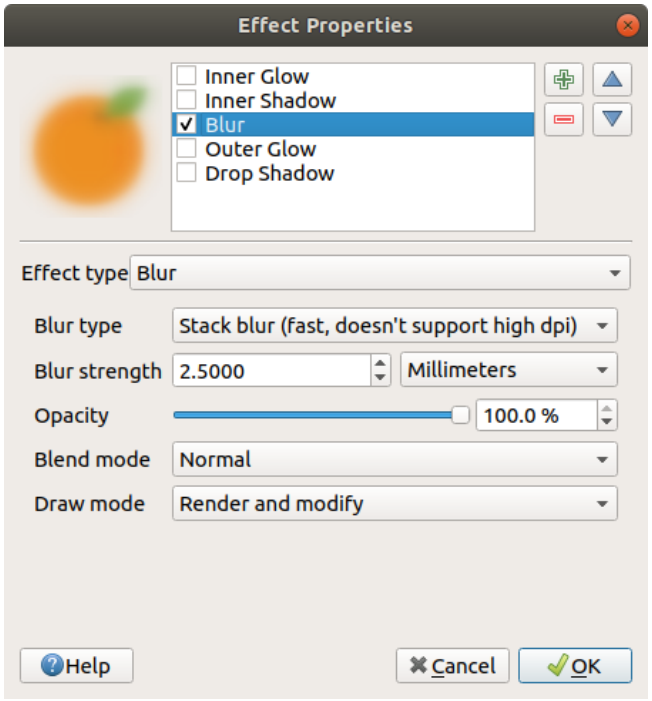


Fig. 12.18: Draw Effects: Blur dialog

- **Colorise**: This effect can be used to make a version of the style using one single hue. The base will always be a grayscale version of the symbol and you can:
  - Use the *Grayscale* to select how to create it: options are 'By lightness', 'By luminosity', 'By average' and 'Off'.
  - If *Colorise* is selected, it will be possible to mix another color and choose how strong it should be.
  - Control the *Brightness*, *Contrast* and *Saturation* levels of the resulting symbol.

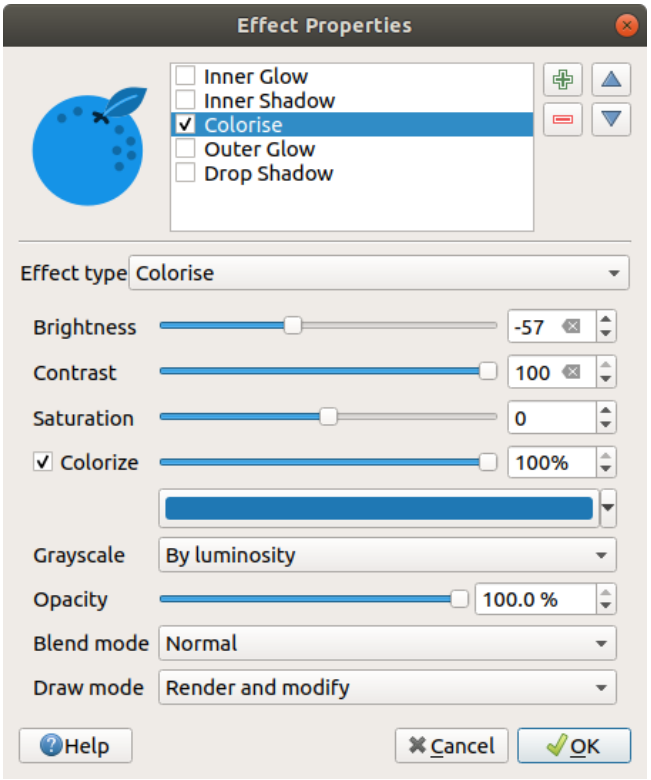


Fig. 12.19: Draw Effects: Colorize dialog

- **Drop Shadow**: Using this effect adds a shadow on the feature, which looks like adding an extra dimension. This effect can be customized by changing the *Offset* angle and distance, determining where the shadow shifts towards to and the proximity to the source object. *Drop Shadow* also has the option to change the *Blur radius* and the *Color* of the effect.

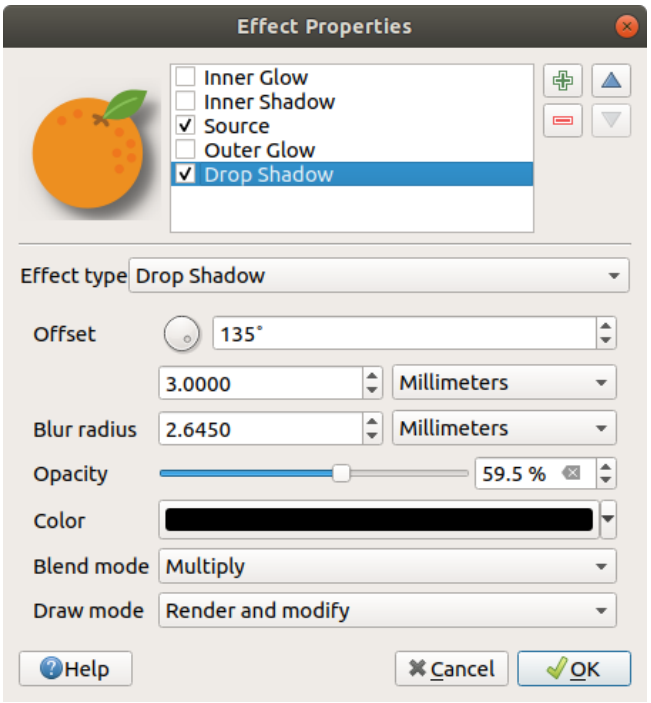


Fig. 12.20: Draw Effects: Drop Shadow dialog

- **Inner Shadow**: This effect is similar to the *Drop Shadow* effect, but it adds the shadow effect on the inside of the edges of the feature. The available options for customization are the same as the *Drop Shadow* effect.

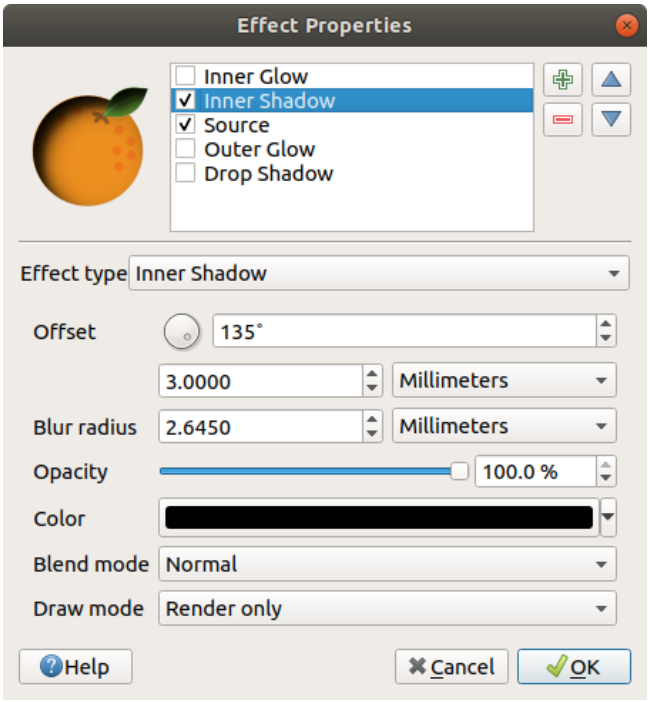


Fig. 12.21: Draw Effects: Inner Shadow dialog

- **Inner Glow**: Adds a glow effect inside the feature. This effect can be customized by adjusting the *Spread* (width) of the glow, or the *Blur radius*. The latter specifies the proximity from the edge of the feature where you want any blurring to happen. Additionally, there are options to customize the color of the glow using a *Single color* or a *Color ramp*.

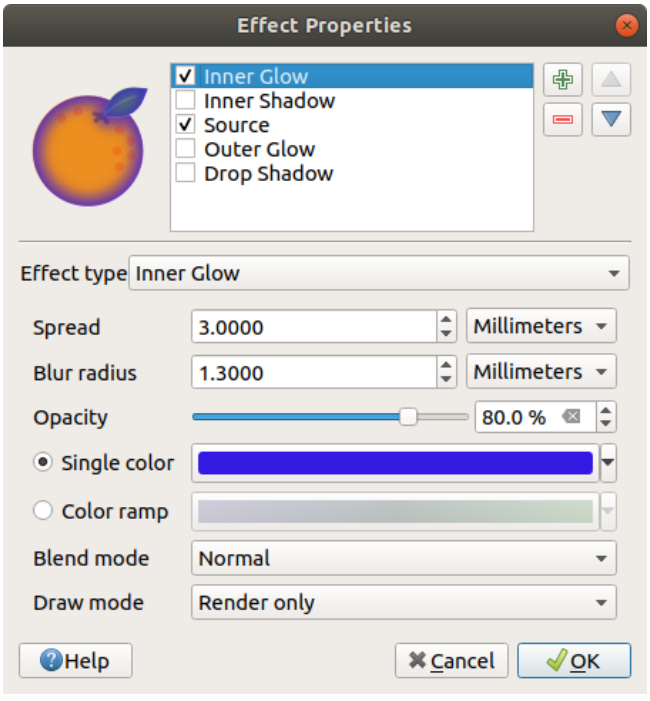


Fig. 12.22: Draw Effects: Inner Glow dialog

- **Outer Glow**: This effect is similar to the *Inner Glow* effect, but it adds the glow effect on the outside of the edges of the feature. The available options for customization are the same as the *Inner Glow* effect.

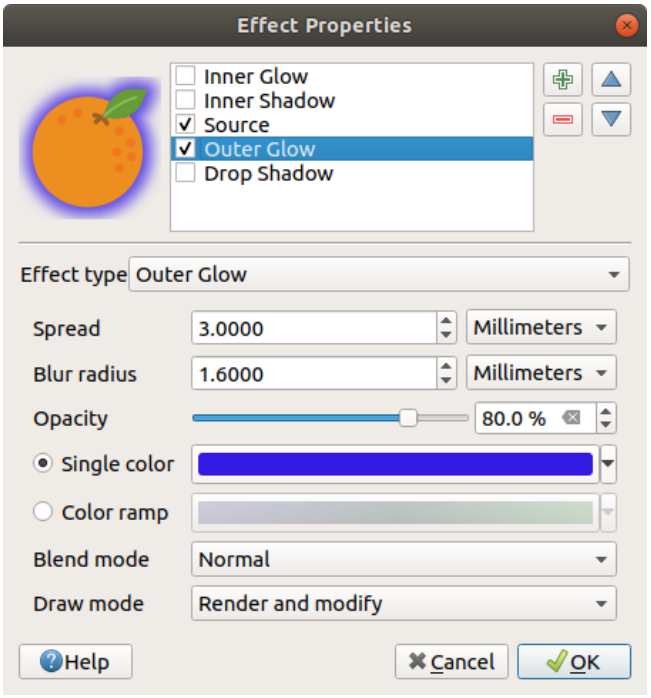


Fig. 12.23: Draw Effects: Outer Glow dialog

- **Transform**: Adds the possibility of transforming the shape of the symbol. The first options available for customization are the *Reflect horizontal* and *Reflect vertical*, which actually create a reflection on the horizontal and/or vertical axes. The other options are:
  - *Shear X,Y*: Slants the feature along the X and/or Y axis.
  - *Scale X,Y*: Enlarges or minimizes the feature along the X and/or Y axis by the given percentage.
  - *Rotation*: Turns the feature around its center point.
  - and *Translate X,Y* changes the position of the item based on a distance given on the X and/or Y axis.

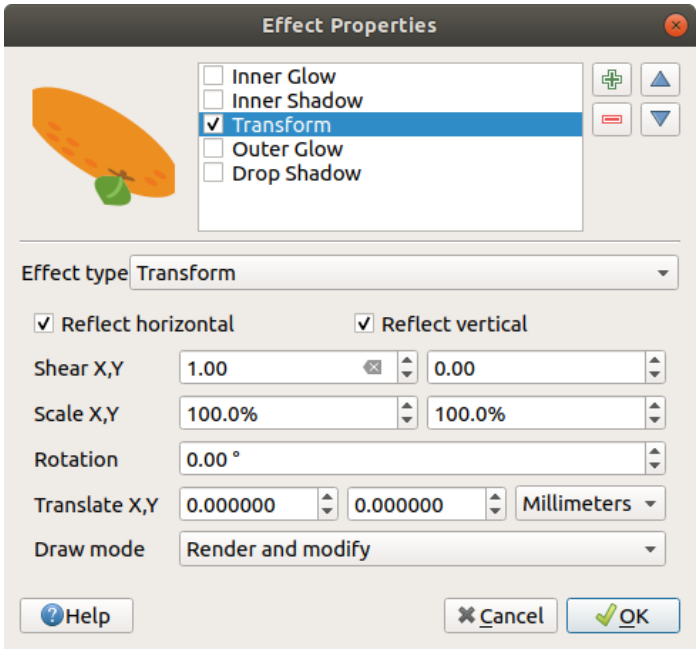


Fig. 12.24: Draw Effects: Transform dialog

One or more effect types can be used at the same time. You (de)activate an effect using its checkbox in the effects list. You can change the selected effect type by using the *Effect type* option. You can reorder the effects using Move up and Move down buttons, and also add/remove effects using the Add new effect and Remove effect buttons.

There are some common options available for all draw effect types. *Opacity* and *Blend mode* options work similar to the ones described in *Layer rendering* and can be used in all draw effects except for the transform one.

There is also a *Draw mode* option available for every effect, and you can choose whether to render and/or modify the symbol, following some rules:

- Effects render from top to bottom.
- *Render only* mode means that the effect will be visible.
- *Modifier only* mode means that the effect will not be visible but the changes that it applies will be passed to the next effect (the one immediately below).
- The *Render and Modify* mode will make the effect visible and pass any changes to the next effect. If the effect is at the top of the effects list or if the immediately above effect is not in modify mode, then it will use the original source symbol from the layers properties (similar to source).

12.1.4 Labels Properties

The *Labels* properties provides you with all the needed and appropriate capabilities to configure smart labeling on vector layers. This dialog can also be accessed from the *Layer Styling* panel, or using the Layer Labeling Options button of the **Labels toolbar**.

At the top of the dialog, you have:

- a combobox for selecting the appropriate labeling method for the active layer
- the Configure project labeling rules button: helps you control interactions between labels and features across the layers in the project. More details at *Configuring project labeling rules*.
- the Automated placement settings (applies to all layers) button: configure general properties on label placement and conflicts resolution. More details at *Setting the automated placement engine*.

The first step is to choose the labeling method from the drop-down list. Available methods are:

- *No labels*: the default value, showing no labels from the layer
- *Single labels*: Show labels on the map using a single attribute or an expression
- *Rule-based labeling*
- and *Blocking*: allows to set a layer as just an obstacle for other layer's labels without rendering any labels of its own.

The next steps assume you select the *Single labels* option, opening the following dialog.

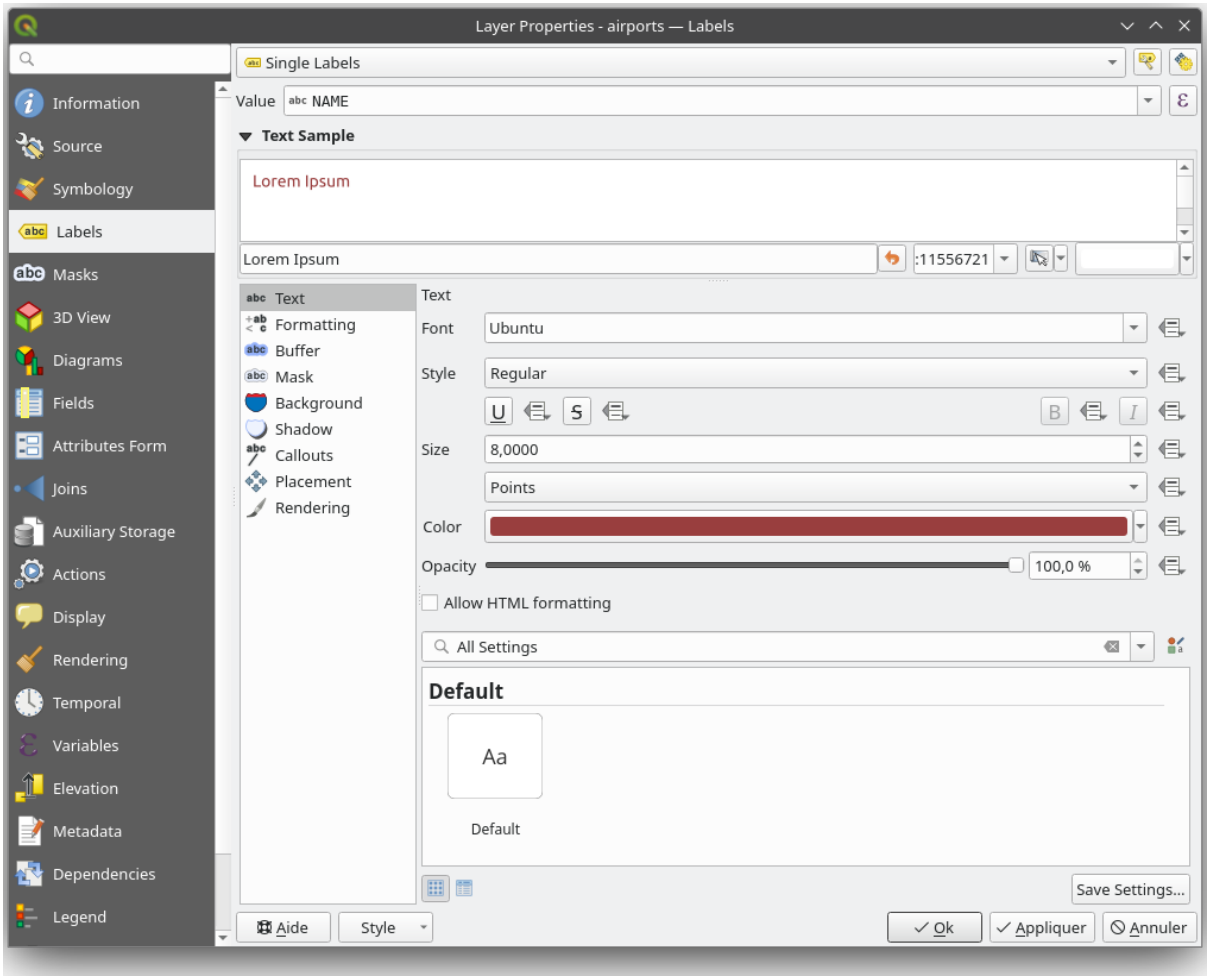


Fig. 12.25: Layer labeling settings - Single labels

At the top of the dialog, a *Value* drop-down list is enabled. You can select an attribute column to use for labeling. By default, the *display field* is used. Click ε if you want to define labels based on expressions - See *Define labels based on expressions*.

> **Note**
>
> Labels with their formatting can be displayed as entries in the legends, if enabled in the *Legend* tab.

Below are displayed options to customize the labels, under various tabs:

- abc *Text*

- *Formatting*
- *Buffer*
- *Mask*
- *Background*
- *Shadow*
- *Callouts*
- *Placement*
- *Rendering*

Description of how to set each property is exposed at 设置标注.

Configuring project labeling rules

Pressing the Configure project labeling rules button next to the labeling method drop-down selector, you can create rules that controls how labels from a layer can interact with labels or features from another layer.

1. Press the Add rule button and in the drop-down menu, select one of the rule types:
   - *Prevent labels overlapping features*: prevents labels being placed overlapping features from a different layer.
   - *Pull labels towards features*: prevents labels being placed too far from features from a different layer. The maximum distance can be set in the unit of your choice.
   - *Push labels away from features*: prevents labels being placed too close to features from a different layer. The minimum distance can be set in the unit of your choice, as well as the *rule's priority* (The highest-priority rules are more important to respect in the event of a label placement conflict).
   - *Push labels away from other labels*: prevents labels being placed too close to labels from a different layer.

   Attention

   The last three options are only available on QGIS installed with GEOS >= 3.10 (see *Help* ► *About* menu).

2. Fill the properties at your will; you can provide a more meaningful name to the rule.
3. 按 确定。
4. Add as many rules as necessary.
5. If necessary, press Edit rule to modify the selected rule or Remove rule to delete it from the project.

The set rules are available from any layer *Labels* properties tab, pressing the Configure project labeling rules button. You can temporary enable or disable any of them, using the checkbox next to the name. Hover over a rule to preview its details.

Setting the automated placement engine

You can use the automated placement settings to configure a project-level automated behavior of the labels. In the top right corner of the *Labels* tab, click the Automated placement settings (applies to all layers) button, opening a dialog with the following options:

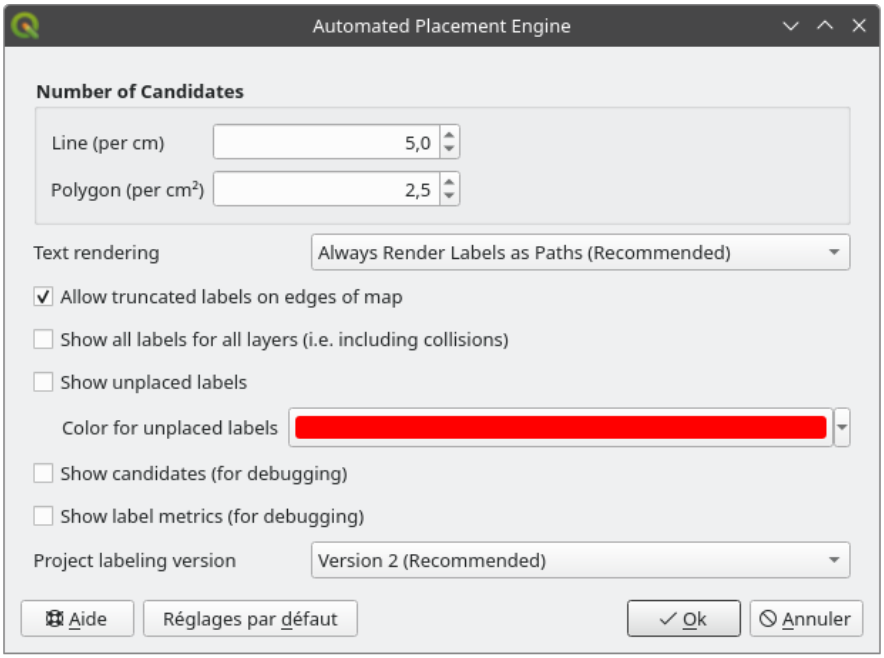


Fig. 12.26: The labels automated placement engine

- *Number of candidates*: calculates and assigns to line and polygon features the number of possible labels placement based on their size. The longer or wider a feature is, the more candidates it has, and its labels can be better placed with less risk of collision.
- *Text rendering*: sets the default value for label rendering widgets when *exporting a map canvas* or *a layout* to PDF or SVG. If *Always render labels as text* is selected then labels can be edited in external applications (e.g. Inkscape) as normal text. BUT the side effect is that the rendering quality is decreased, and there are issues with rendering when certain text settings like buffers are in place. That's why *Always render labels as paths (recommended)* which exports labels as outlines but guarantees complete compatibility with the full range of formatting options available, is recommended. With *Prefer rendering labels as text* option, labels are rendered as text objects, unless doing so results in rendering artifacts or poor quality rendering (depending on text format settings).

> **Note**
>
> When rendering labels as text to a vector based device (e.g. PDF or SVG), care must be taken to ensure that the required fonts are available to users when opening the created files, or default fallback fonts will be used to display the output instead. (Although PDF exports MAY automatically embed some fonts when possible, depending on the user's platform).

- *Allow truncated labels on edges of map*: controls whether labels which fall partially outside of the map extent should be rendered. If checked, these labels will be shown (when there's no way to place them fully within the visible area). If unchecked then partially visible labels will be skipped. Note that this setting has no effects on labels' display in the *layout map item*.
- *Show all labels for all layers (i.e. including colliding objects)*. Note that this option can be also set per layer (see 渲染选项卡)

- *Show unplaced labels*: allows to determine whether any important labels are missing from the maps (e.g. due to overlaps or other constraints). They are displayed using a customizable color.
- *Show candidates (for debugging)*: controls whether boxes should be drawn on the map showing all the candidates generated for label placement. Like the label says, it's useful only for debugging and testing the effect different labeling settings have. This could be handy for a better manual placement with tools from the *label toolbar*.
- *Show label metrics (for debugging)*: displays the text bounds of the label in red and baselines in blue
- *Project labeling version*: QGIS supports two different versions of label automatic placement:
  - *Version 1*: the old system (used by QGIS versions 3.10 and earlier, and when opening projects created in these versions in QGIS 3.12 or later). Version 1 treats label and obstacle priorities as "rough guides" only, and it's possible that a low-priority label will be placed over a high-priority obstacle in this version. Accordingly, it can be difficult to obtain the desired labeling results when using this version and it is thus recommended only for compatibility with older projects.
  - *Version 2 (recommended)*: this is the default system in new projects created in QGIS 3.12 or later. In version 2, the logic dictating when labels are allowed to overlap *obstacles* has been reworked. The newer logic forbids any labels from overlapping any obstacles with a greater obstacle weight compared to the label's priority. As a result, this version results in much more predictable and easier to understand labeling results.

Rule-based labeling

With rule-based labeling multiple label configurations can be defined and applied selectively on the base of expression filters and scale range, as in *Rule-based rendering*.

To create a rule:

1. Select the **Rule-based labeling** option in the main drop-down list from the *Labels* tab
2. Click the Add rule button at the bottom of the dialog.
3. Fill the new dialog with:
   - *Description*: a text used to identify the rule in the *Labels* tab and as a *label legend entry* in the print layout legend
   - *Filter*: an expression to select the features to apply the label settings to
   - If there are rules already set, the *Else* option can be used to select all the features not matching any filter of the rules in the same group.
4. You can set a *scale range* in which the label rule should be applied.
5. The options available under the *Labels* group box are the usual *label settings*. Configure them and press *OK*.

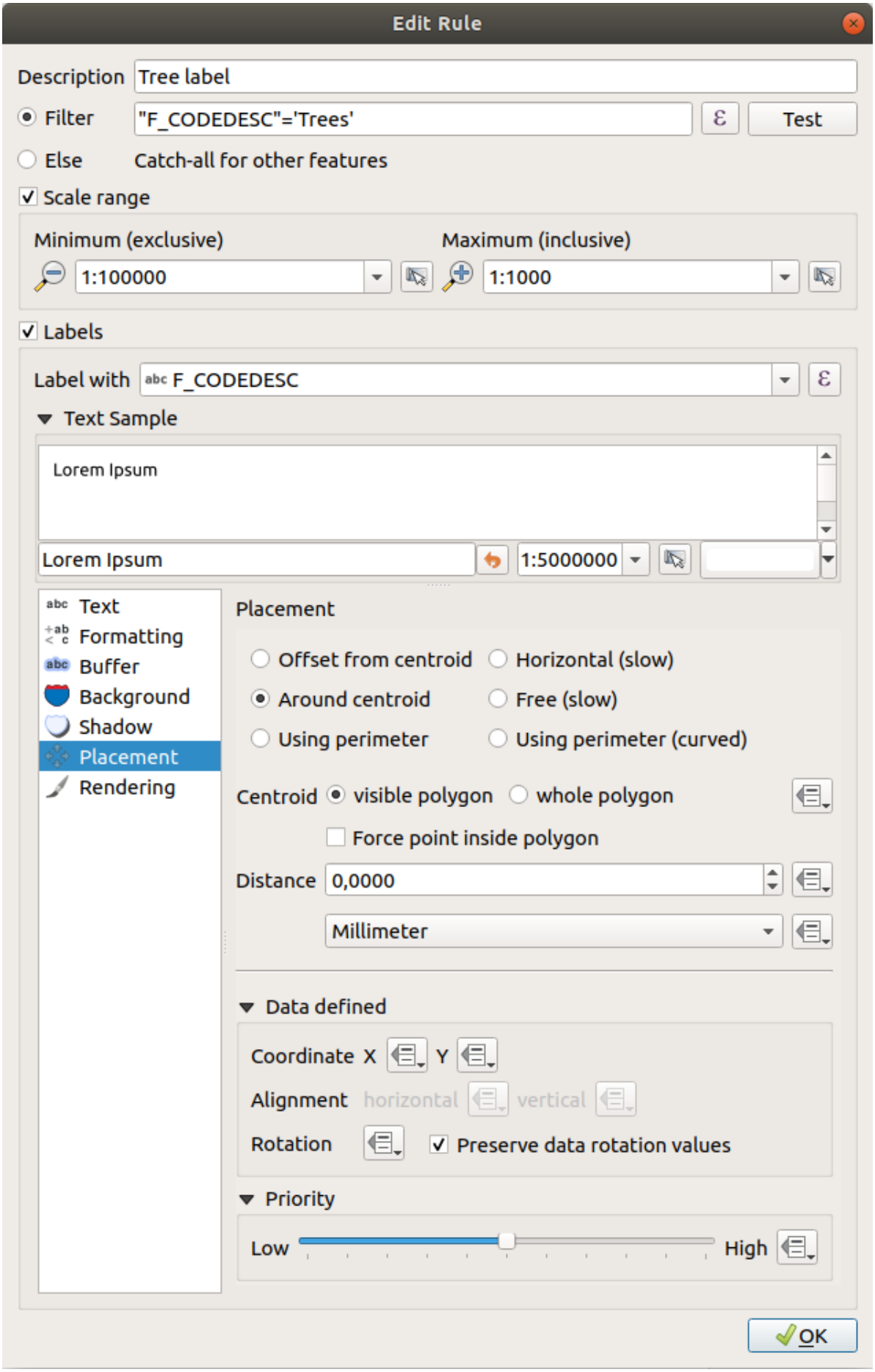


Fig. 12.27: Rule settings

A summary of existing rules is shown in the main dialog (see Fig. 12.28). You can add multiple rules, reorder or imbricate them with a drag-and-drop. You can as well remove them with the button or edit them with button or a double-click.

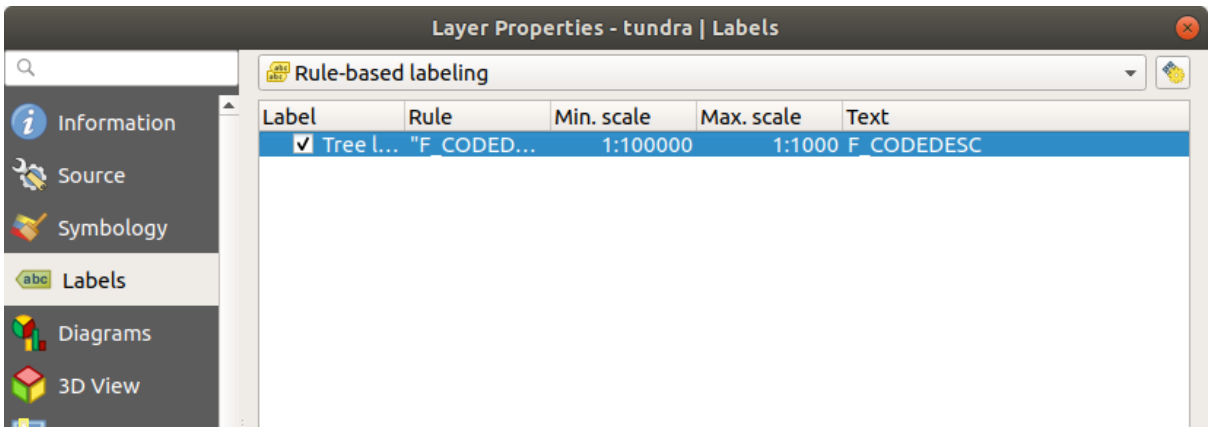


Fig. 12.28: Rule based labeling panel

Define labels based on expressions

Whether you choose single or rule-based labeling type, QGIS allows using expressions to label features.

Assuming you are using the *Single labels* method, click the button near the *Value* drop-down list in the *Labels* tab of the properties dialog.

In Fig. 12.29, you see a sample expression to label the alaska trees layer with tree type and area, based on the field 'VEGDESC', some descriptive text, and the function `$area` in combination with `format_number()` to make it look nicer.

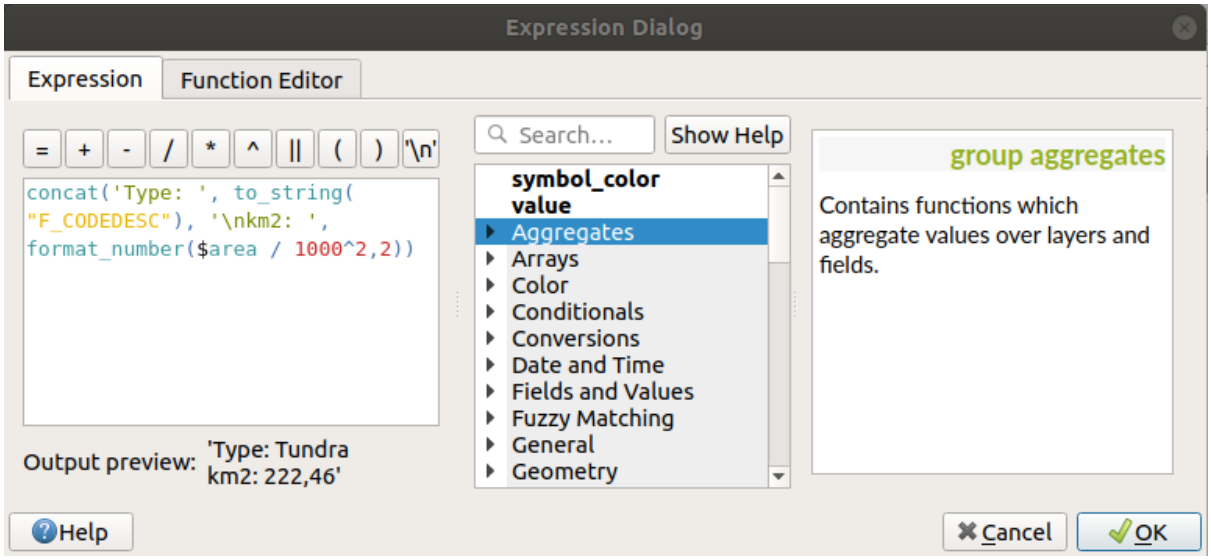


Fig. 12.29: Using expressions for labeling

Expression based labeling is easy to work with. All you have to take care of is that:

- You may need to combine all elements (strings, fields, and functions) with a string concatenation function such as `concat`, `+` or `||`. Be aware that in some situations (when null or numeric value are involved) not all of these tools will fit your need.

- Strings are written in 'single quotes'.
- Fields are written in "double quotes" or without any quote.

Let's have a look at some examples:

1. Label based on two fields 'name' and 'place' with a comma as separator:

```
"name" || ', ' || "place"
```

   Returns:

```
John Smith, Paris
```

2. Label based on two fields 'name' and 'place' with other texts:

```
'My name is ' + "name" + 'and I live in ' + "place"
'My name is ' || "name" || 'and I live in ' || "place"
concat('My name is ', name, ' and I live in ', "place")
```

   Returns:

```
My name is John Smith and I live in Paris
```

3. Label based on two fields 'name' and 'place' with other texts combining different concatenation functions:

```
concat('My name is ', name, ' and I live in ' || place)
```

   Returns:

```
My name is John Smith and I live in Paris
```

   Or, if the field 'place' is NULL, returns:

```
My name is John Smith
```

4. Multi-line label based on two fields 'name' and 'place' with a descriptive text:

```
concat('My name is ', "name", '\n' , 'I live in ' , "place")
```

   Returns:

```
My name is John Smith
I live in Paris
```

5. Label based on a field and the $area function to show the place's name and its rounded area size in a converted unit:

```
'The area of ' || "place" || ' has a size of '
|| round($area/10000) || ' ha'
```

   Returns:

```
The area of Paris has a size of 10500 ha
```

6. Create a CASE ELSE condition. If the population value in field *population* is <= 50000 it is a town, otherwise it is a city:

```
concat('This place is a ',
CASE WHEN "population" <= 50000 THEN 'town' ELSE 'city' END)
```

   Returns:

```
This place is a town
```

7. Display name for the cities and no label for the other features (for the "city" context, see example above):

```
CASE WHEN "population" > 50000 THEN "NAME" END
```

Returns:

```
Paris
```

As you can see in the expression builder, you have hundreds of functions available to create simple and very complex expressions to label your data in QGIS. See 表达式 chapter for more information and examples on expressions.

Using data-defined override for labeling

With the Data defined override function, the settings for the labeling are overridden by entries in the attribute table or expressions based on them. This feature can be used to set values for most of the labeling options described above.

For example, using the Alaska QGIS sample dataset, let's label the `airports` layer with their name, based on their militarian `USE`, i.e. whether the airport is accessible to :

- military people, then display it in gray color, size 8;
- others, then show in blue color, size 10.

To do this, after you enabled the labeling on the `NAME` field of the layer (see 设置标注):

1. Activate the *Text* tab.
2. Click on the icon next to the *Size* property.
3. Select *Edit...* and type:

```
CASE
  WHEN "USE" like '%Military%' THEN 8 -- because compatible values are
↪'Military'
                                      -- and 'Joint Military/Civilian'
  ELSE 10
END
```

4. Press *OK* to validate. The dialog closes and the button becomes meaning that an rule is being run.
5. Then click the button next to the color property, type the expression below and validate:

```
CASE
  WHEN "USE" like '%Military%' THEN '150, 150, 150'
  ELSE '0, 0, 255'
END
```

Likewise, you can customize any other property of the label, the way you want. See more details on the Data-define override widget's description and manipulation in *Data defined override setup* section.

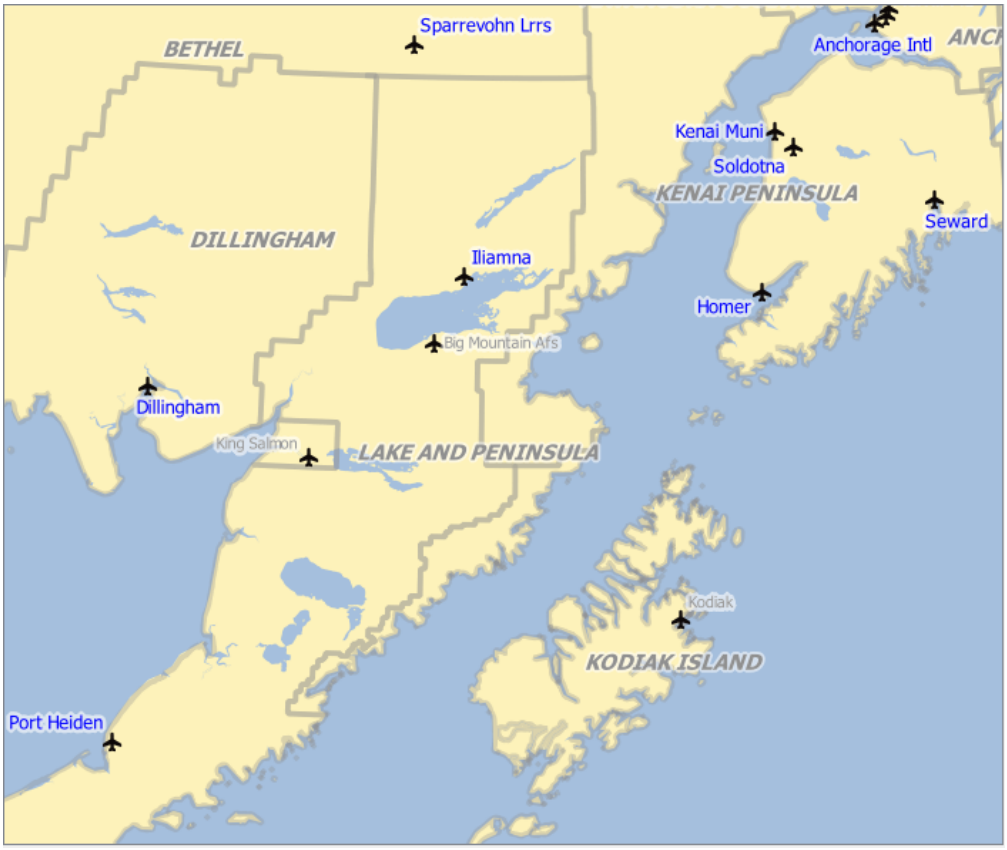


Fig. 12.30: Airports labels are formatted based on their attributes

Tip

Use the data-defined override to label every part of multi-part features

There is an option to set the labeling for multi-part features independently from your label properties. Choose the *Rendering*, `Feature options`, go to the Data-define override button next to the checkbox *Label every part of multipart-features* and define the labels as described in *Data defined override setup*.

The Label Toolbar

The *Label Toolbar* provides some tools to manipulate *label* (including their *callout*) or *diagram* properties:

Fig. 12.31: The Label toolbar

- Highlight Pinned Labels, Diagrams and Callouts. If the vector layer of the item is editable, then the highlighting is green, otherwise it's blue.
- Toggle Display of Unplaced Labels: Allows to determine whether any important labels are missing from the maps (e.g. due to overlaps or other constraints). They are displayed with a customizable color (see *Setting the automated placement engine*).

- Pin/Unpin Labels and Diagrams. By clicking or draging an area, you pin overlaid items. If you click or drag an area holding `Shift`, the items are unpinned. Finally, you can also click or drag an area holding `Ctrl` to toggle their pin status.
- Show/Hide Labels and Diagrams. If you click on the items, or click and drag an area holding `Shift`, they are hidden. When an item is hidden, you just have to click on the feature to restore its visibility. If you drag an area, all the items in the area will be restored.
- Move a Label, Diagram or Callout: click to select the item and click to move it to the desired place. The new coordinates are stored in *auxiliary fields*. Selecting the item with this tool and hitting the `Delete` key will delete the stored position value.
- Rotate a Label. Click to select the label and click again to apply the desired rotation. Likewise, the new angle is stored in an auxiliary field. Selecting a label with this tool and hitting the `Delete` key will delete the rotation value of this label.
- Change Label Properties. It opens a dialog to change the clicked label properties; it can be the label itself, its coordinates, angle, font, size, multiline alignment ... as long as this property has been mapped to a field. Here you can set the option to *Label every part of a feature*.

Warning

Label tools overwrite current field values

Using the *Label toolbar* to customize the labeling actually writes the new value of the property in the mapped field. Hence, be careful to not inadvertently replace data you may need later!

Note

The *Auxiliary Storage Properties* mechanism may be used to customize labeling (position, and so on) without modifying the underlying data source.

Customize the labels from the map canvas

Combined with the *Label Toolbar*, the data defined override setting helps you manipulate labels in the map canvas (move, edit, rotate). We now describe an example using the data-defined override function for the Move Label, Diagram or Callout function (see Fig. 12.32).

1. Import `lakes.shp` from the QGIS sample dataset.
2. Double-click the layer to open the Layer Properties. Click on *Labels* and *Placement*. Select *Offset from centroid*.
3. Look for the *Data defined* entries. Click the icon to define the field type for the *Coordinate*. Choose `xlabel` for X and `ylabel` for Y. The icons are now highlighted in yellow.

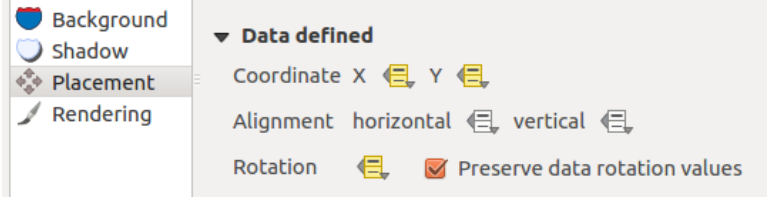


Fig. 12.32: Labeling of vector polygon layers with data-defined override

4. Zoom into a lake.
5. Set editable the layer using the Toggle Editing button.
6. Go to the Label toolbar and click the icon. Now you can shift the label manually to another position (see Fig. 12.33). The new position of the label is saved in the `xlabel` and `ylabel` columns of the attribute table.
7. It's also possible to add a line connecting each lake to its moved label using:
   - the label's *callout property*
   - or the *geometry generator symbol layer* with the expression below:

```
make_line( centroid( $geometry ), make_point( "xlabel", "ylabel" ) )
```

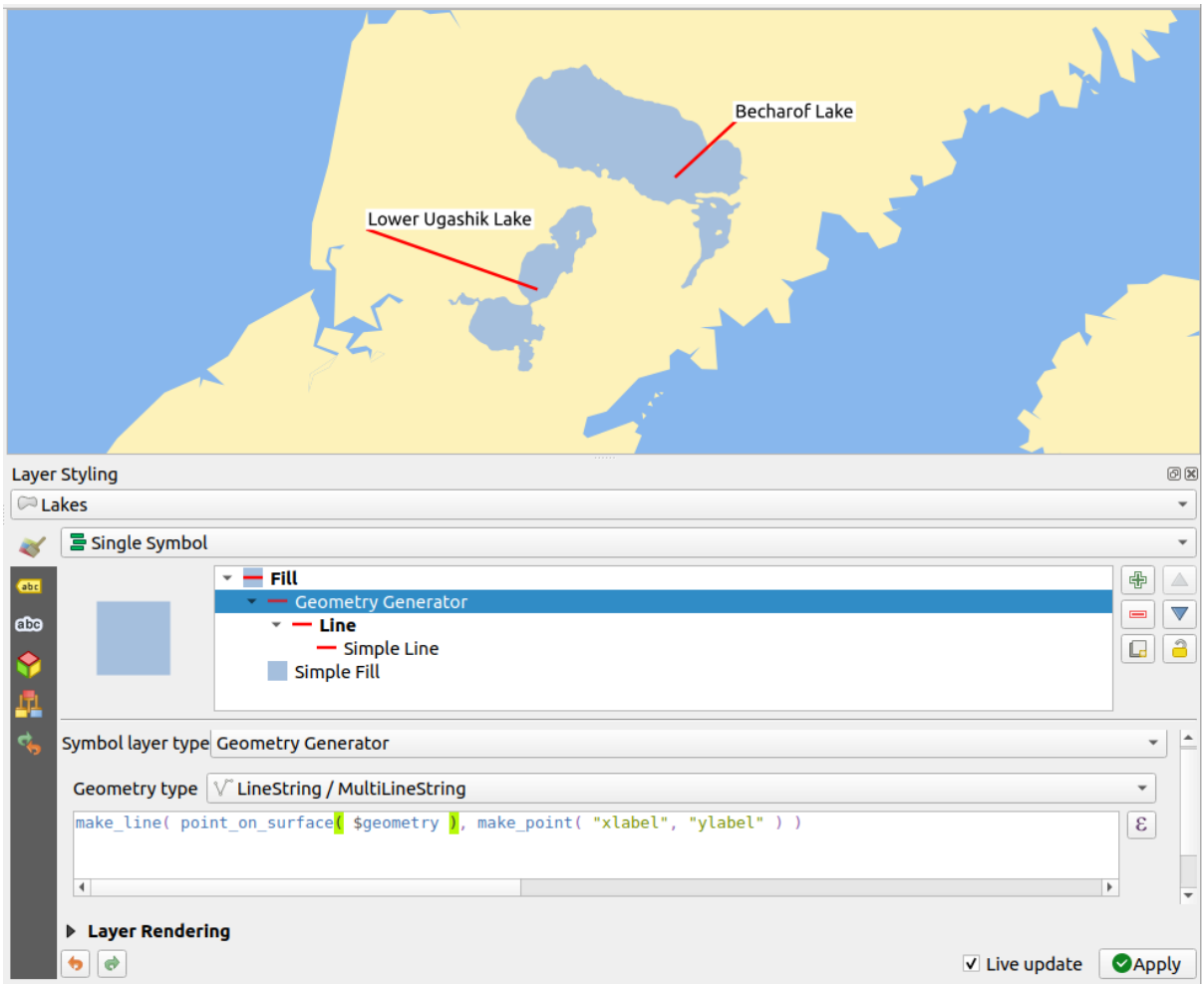


Fig. 12.33: Moved labels

Note

The *Auxiliary Storage Properties* mechanism may be used with data-defined properties without having an editable data source.

12.1.5 Diagrams Properties

The *Diagrams* tab allows you to add a graphic overlay to a vector layer (see Fig. 12.34). This dialog can also be accessed from the *Layer Styling* panel, or using the Layer Diagram Options button of the **Labels toolbar**.

The current core implementation of diagrams provides support for:

- *No diagrams*: the default value with no diagram displayed over the features;
- *Pie chart*, a circular statistical graphic divided into slices to illustrate numerical proportion. The arc length of each slice is proportional to the quantity it represents;
- *Text diagram*, a horizontally divided circle showing statistic values inside;
- *Histogram*, bars of varying colors for each attribute aligned next to each other;
- *Stacked bars*, stacks bars of varying colors for each attribute on top of each other vertically or horizontally;
- *Stacked diagram*, stacks diagrams of equal or varying types, next to each other, vertically or horizontally. More details at *Stacked Diagrams*.

In the top right corner of the *Diagrams* tab, the Automated placement settings (applies to all layers) button provides means to control diagram *labels placement* on the map canvas.

> **Tip**
>
> **Switch quickly between types of diagrams**
>
> Given that the settings are almost common to the different types of diagram, when designing your diagram, you can easily change the diagram type and check which one is more appropriate to your data without any loss.

For each type of diagram, the properties are divided into several tabs:

- *Attributes*
- *Rendering*
- *Size*
- *Placement*
- *Options*
- *Legend*

Attributes

Attributes defines which variables to display in the diagram. Use add item button to select the desired fields into the 'Assigned Attributes' panel. Generated attributes with 表达式 can also be used.

You can move up and down any row with click and drag, sorting how attributes are displayed. You can also change the label in the 'Legend' column or the attribute color by double-clicking the item.

This label is the default text displayed in the legend of the print layout or of the layer tree.

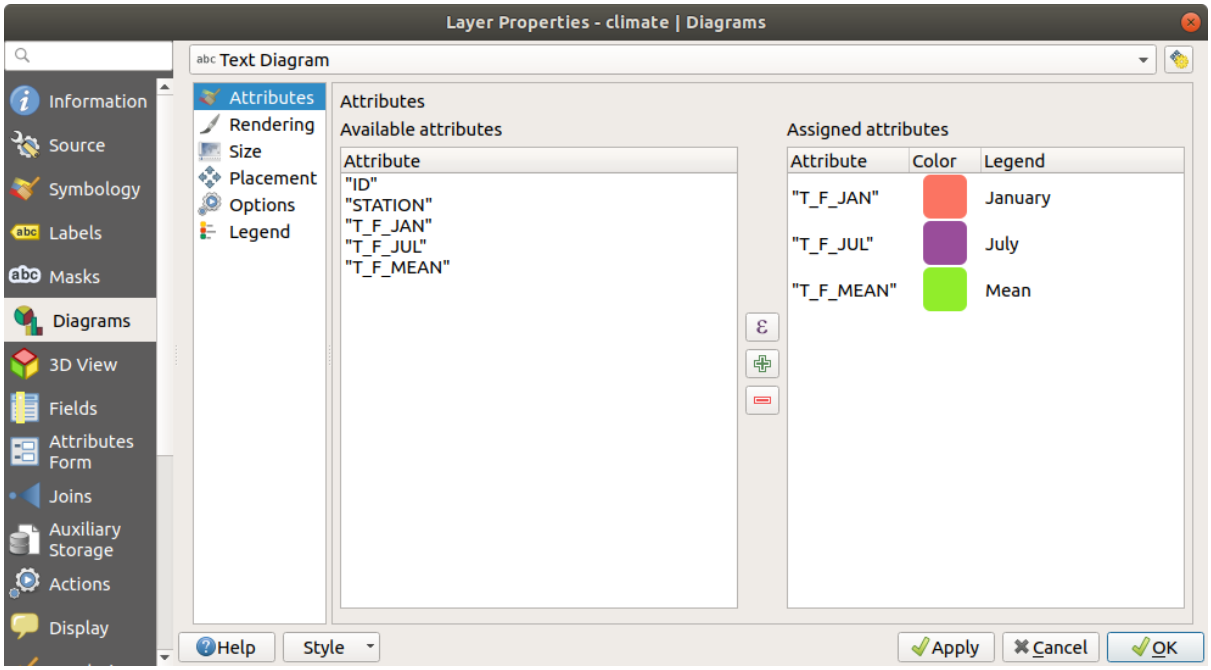


Fig. 12.34: Diagram properties - Attributes tab

Rendering

Rendering defines how the diagram looks like. It provides general settings that do not interfere with the statistic values such as:

- the graphic's opacity, its outline width and color;
- depending on the type of diagram:
  - for histogram and stacked bars, the width of the bar and the spacing between the bars. You may want to set the spacing to `0` for stacked bars. Moreover, the *Axis line symbol* can be made visible on the map canvas and customized using *line symbol properties*.
  - for text diagram, the circle background color and the *font* used for texts;
  - for pie charts, the *Start angle* of the first slice and their *Direction* (clockwise or not).
- the use of *paint effects* on the graphics.

In this tab, you can also manage and fine tune the diagram visibility with different options:

- *Diagram z-index*: controls how diagrams are drawn on top of each other and on top of labels. A diagram with a high index is drawn over other diagrams and labels;
- *Show all diagrams*: shows all the diagrams even if they overlap each other;
- *Show diagram*: allows only specific diagrams to be rendered;
- *Always Show*: selects specific diagrams to always render, even when they overlap other diagrams or map labels;
- setting the *Scale dependent visibility*;

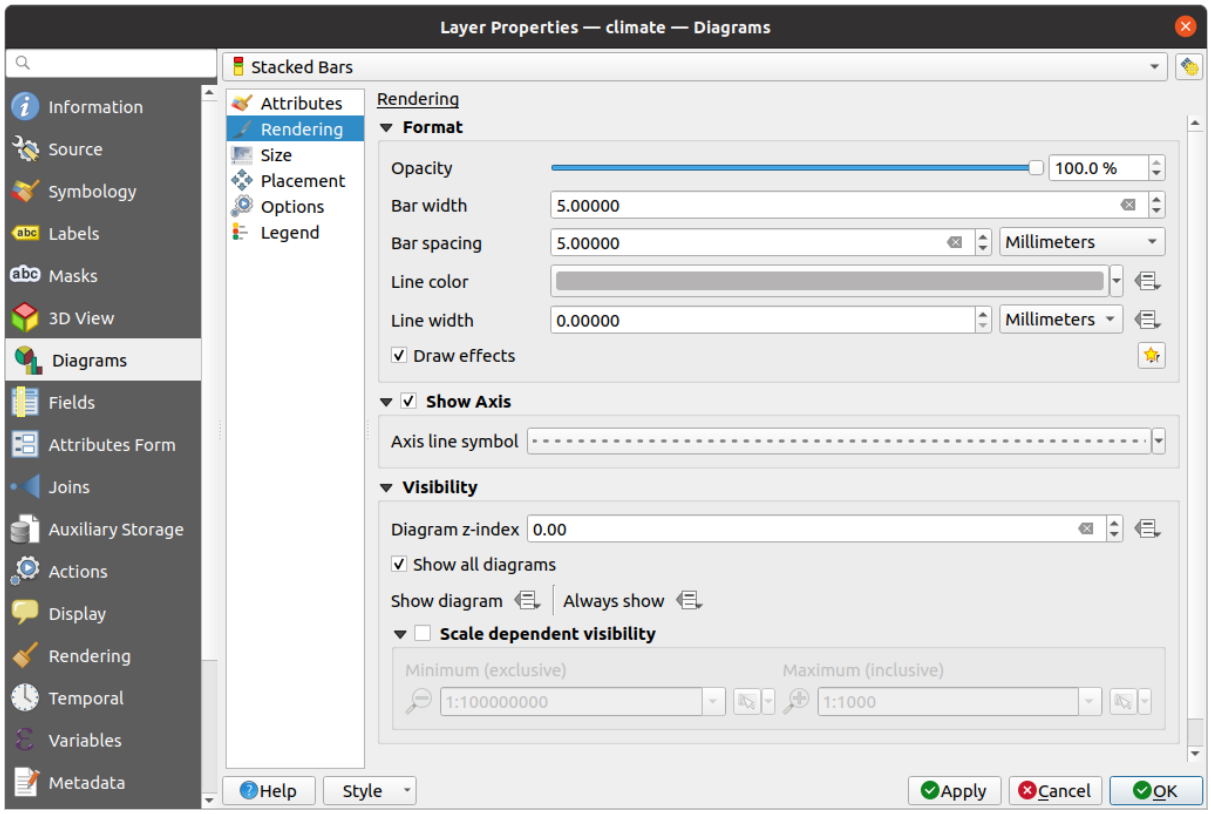


Fig. 12.35: Diagram properties - Rendering tab

Size

Size is the main tab to set how the selected statistics are represented. The diagram size *units* can be 'Millimeters', 'Points', 'Pixels', 'Map Units' or 'Inches'. You can use:

- *Fixed size*, a unique size to represent the graphic of all the features (not available for histograms)
- or *Scaled size*, based on an expression using layer attributes:

  1. In *Attribute*, select a field or build an expression
  2. Press *Find* to return the *Maximum value* of the attribute or enter a custom value in the widget.
  3. For histogram and stacked bars, enter a *Bar length* value, used to represent the *Maximum value* of the attributes. For each feature, the bar length will then be scaled linearly to keep this matching.
  4. For pie chart and text diagram, enter a *Size* value, used to represent the *Maximum value* of the attributes. For each feature, the circle area or diameter will then be scaled linearly to keep this matching (from `0`). A *Minimum size* can however be set for small diagrams.

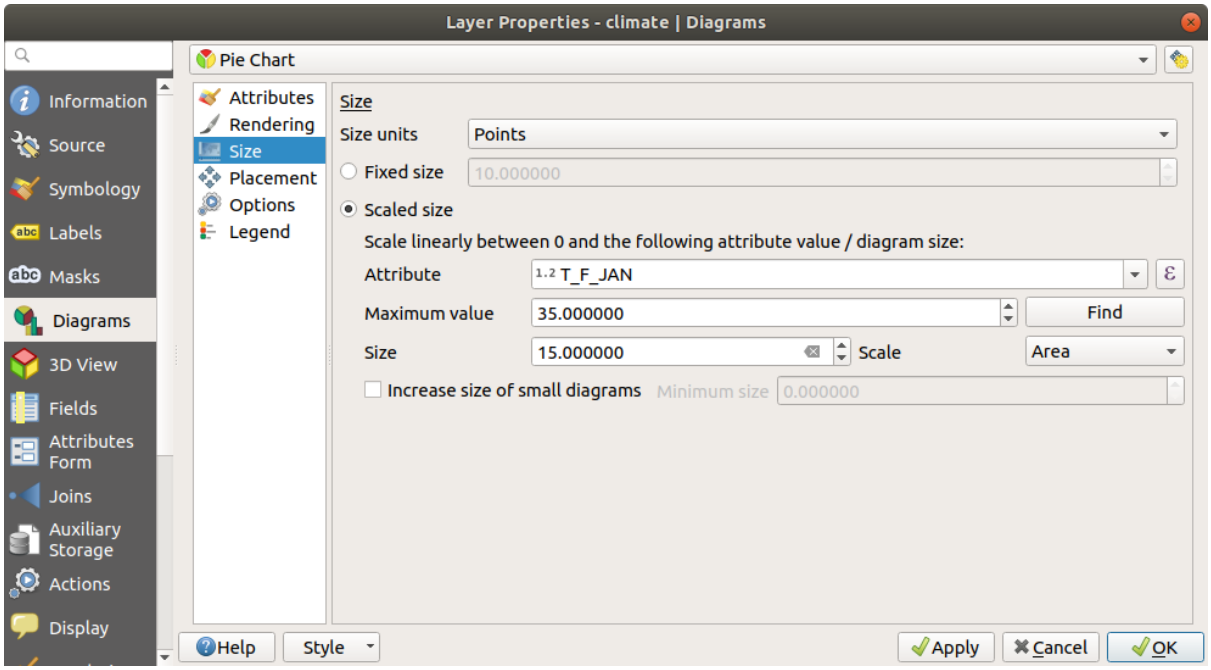


Fig. 12.36: Diagram properties - Size tab

Placement

Placement defines the diagram position. Depending on the layer geometry type, it offers different options for the placement (more details at *Placement*):

- *Around point* or *Over point* for point geometry. The former variable requires a radius to follow.
- *Around line* or *Over line* for line geometry. Like point feature, the first variable requires a distance to respect and you can specify the diagram placement relative to the feature ('above', 'on' and/or 'below' the line) It's possible to select several options at once. In that case, QGIS will look for the optimal position of the diagram. Remember that you can also use the line orientation for the position of the diagram.
- *Around centroid* (at a set *Distance*), *Over centroid*, *Using perimeter* and *Inside polygon* are the options for polygon features.

The *Coordinate* group provides direct control on diagram placement, on a feature-by-feature basis, using their attributes or an expression to set the *X* and *Y* coordinate. The information can also be filled using the *Move labels and diagrams* tool.

In the *Priority* section, you can define the placement priority rank of each diagram, i.e. if there are different diagrams or labels candidates for the same location, the item with the higher priority will be displayed and the others could be left out.

Discourage diagrams and labels from covering features defines features to use as *obstacles*, i.e. QGIS will try to not place diagrams nor labels over these features. The priority rank is then used to evaluate whether a diagram could be omitted due to a greater weighted obstacle feature.

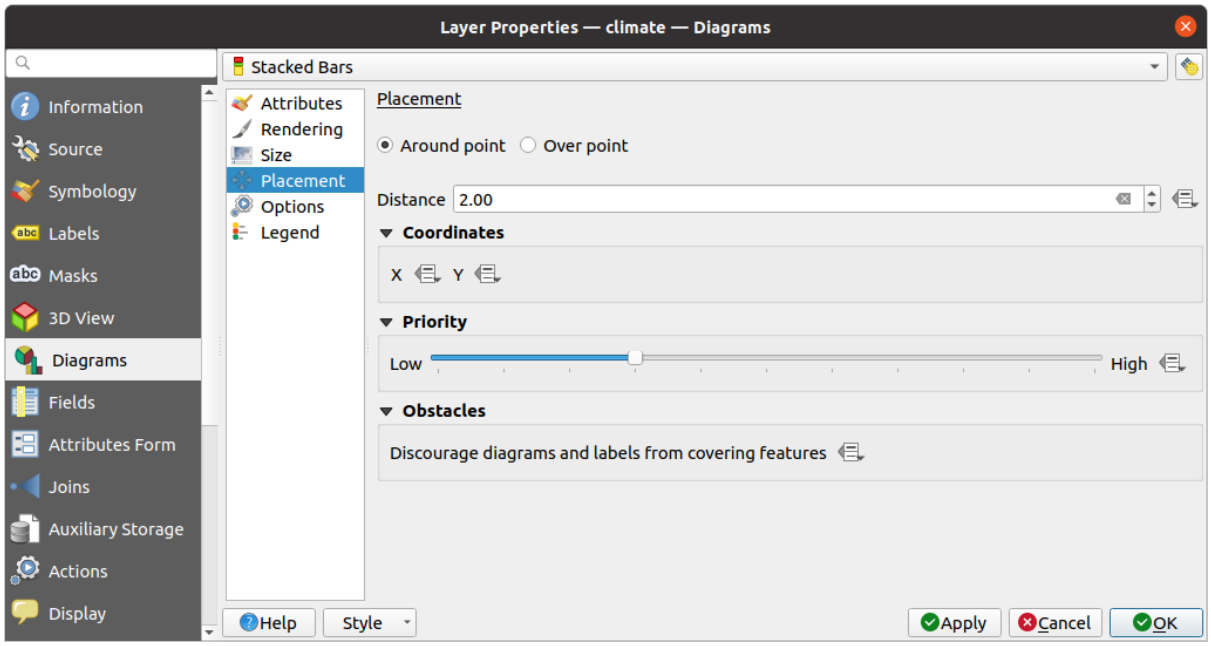


Fig. 12.37: Vector properties dialog with diagram properties, Placement tab

选项

The *Options* tab has settings for histograms and stacked bars. You can choose whether the *Bar orientation* should be *Up*, *Down*, *Right* or *Left*, for horizontal and vertical diagrams.

Legend

From the *Legend* tab, you can choose to display items of the diagram in the *Layers panel*, and in the *print layout legend*, next to the layer symbology:

- check *Show legend entries for diagram attributes* to display in the legends the `Color` and `Legend` properties, as previously assigned in the *Attributes* tab;
- and, when a *scaled size* is being used for the diagrams, push the *Legend Entries for Diagram Size...* button to configure the diagram symbol aspect in the legends. This opens the *Data-defined Size Legend* dialog whose options are described in *Data-defined size legend*.

When set, the diagram legend items (attributes with color and diagram size) are also displayed in the print layout legend, next to the layer symbology.

Stacked Diagrams

Stacked diagrams allow users to create complex diagrams like population pyramids, where two subdiagrams, namely histograms, are located side by side and displayed horizontally.

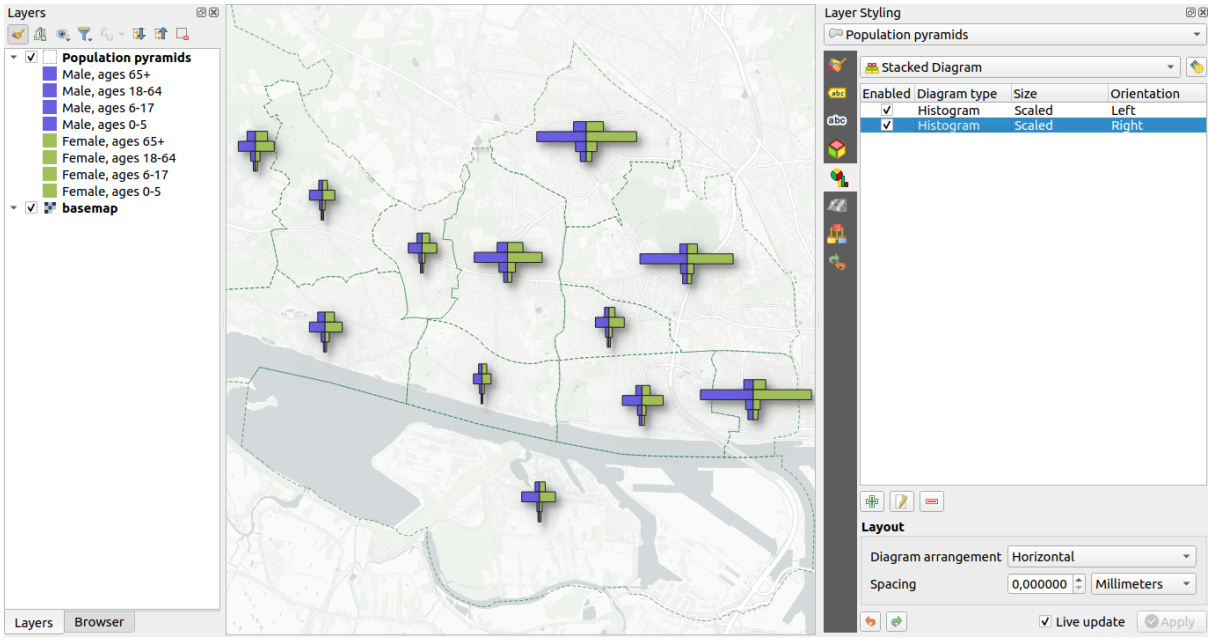


Fig. 12.38: Population pyramids built for each layer feature

Multi-temporal diagrams can also be constructed as stacked diagrams. The number of subdiagrams, as well as the spacing between them can be configured.

Moreover, subdiagrams can have different types (e.g., a pie chart alongside a histogram) and have their own independent settings like *Attributes*, *Rendering*, *Size*, *Options* and *Legend*.

Placement settings in a stacked diagram, as well as some visibility settings (located in the *Rendering* tab), are determined by the placement and visibility settings of the first subdiagram in the stack.

Finally, subdiagram ordering is given by the item ordering in the Stacked Diagram's list. The first subdiagram appears to the left in a horizontal stacked diagram, or in the upper part of a vertical one.

12.1.6 Masks Properties

The *Masks* tab helps you configure the current layer symbols overlay with other symbol layers or labels, from any layer. This is meant to improve the readability of symbols and labels whose colors are close and can be hard to decipher when overlapping; it adds a custom and transparent mask around the items to "hide" parts of the symbol layers of the current layer.

To apply masks on the active layer, you first need to enable in the project either *mask symbol layers* or *mask labels*. Then, from the *Masks* tab, check:

- the *Masked symbol layers*: lists in a tree structure all the symbol layers of the current layer. There you can select the symbol layer item you would like to transparently "cut out" when they overlap the selected mask sources
- the *Mask sources* tab: list all the mask labels and mask symbol layers defined in the project. Select the items that would generate the mask over the selected masked symbol layers

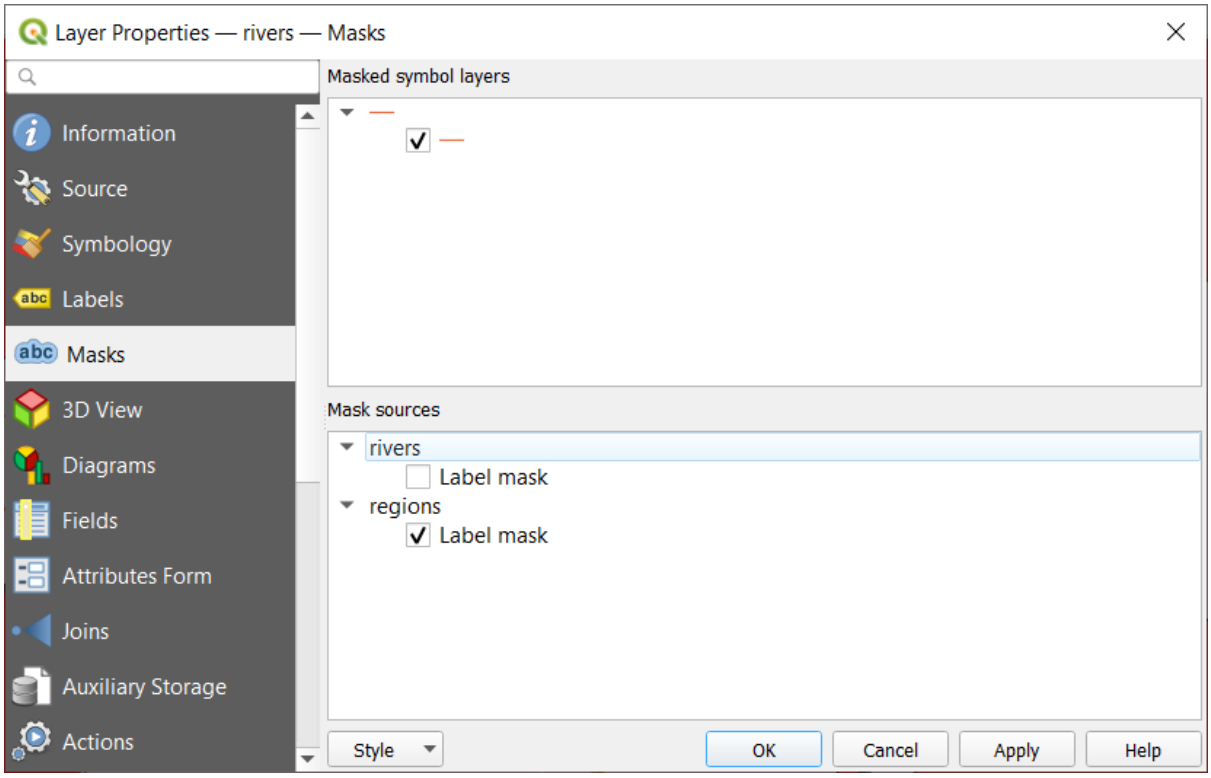


Fig. 12.39: Layer properties - Masks tab

12.1.7 3D View Properties

The *3D View* tab provides settings for vector layers that should be depicted in the *3D Map view* tool.

To display a layer in 3D, select from the combobox at the top of the tab, either:

- *Single symbol*: features are rendered using a common 3D symbol whose properties can be *data-defined* or not. Read details on *setting a 3D symbol* for each layer geometry type.
- *Rule-based*: multiple symbol configurations can be defined and applied selectively based on expression filters and scale range. More details on how-to at *Rule-based rendering*.

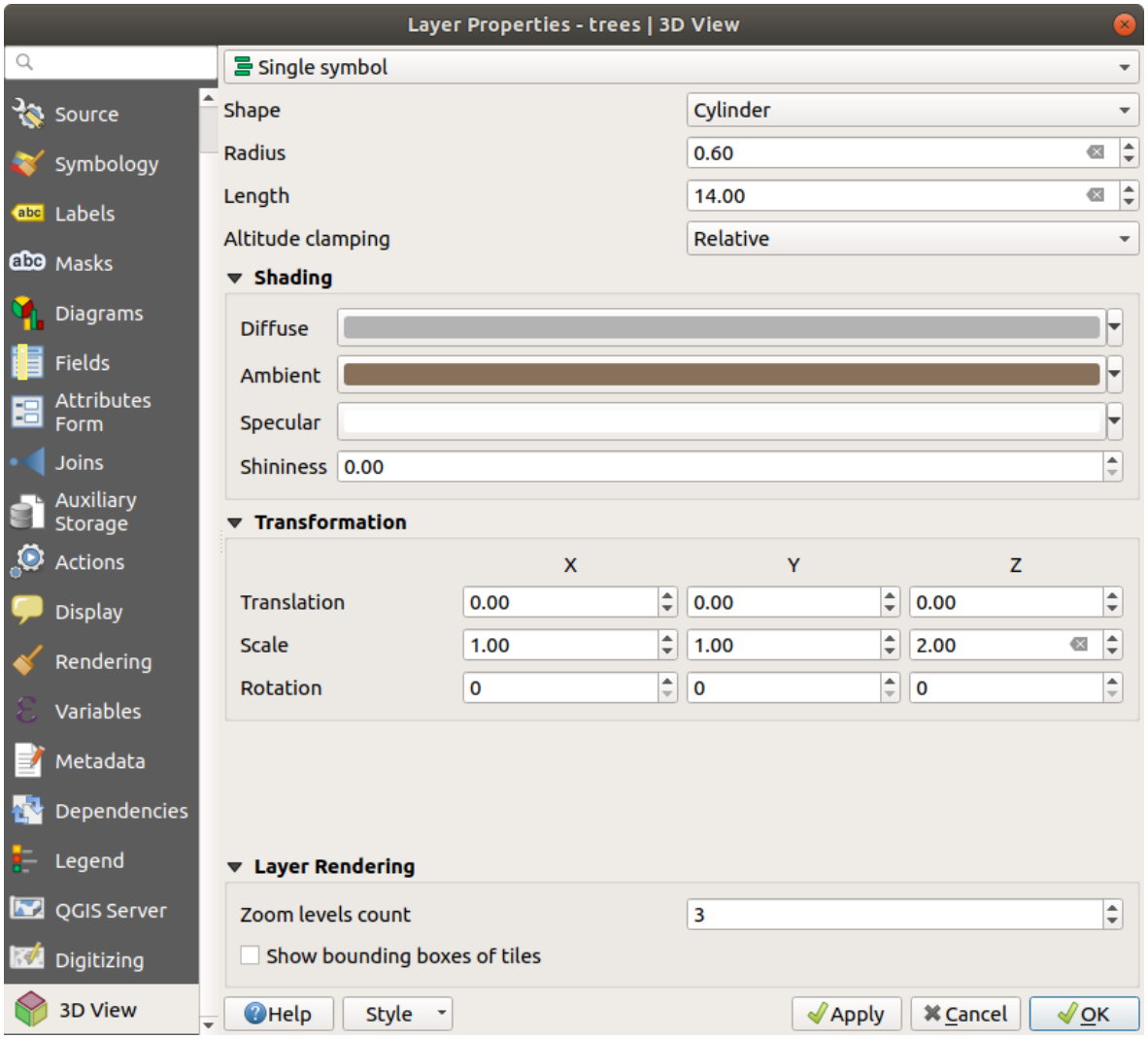


Fig. 12.40: 3D properties of a point layer

Attention

Prefer the *Elevation* tab for symbol elevation and terrain settings

Features' elevation and altitude related properties (*Altitude clamping*, *Altitude binding*, *Extrusion* or *Height*) in the *3D View* tab inherit their default values from the layer's *Elevation properties* and should preferably be set from within the *Elevation* tab.

For better performance, data from vector layers are loaded in the background, using multithreading, and rendered in tiles whose size can be controlled from the *Layer rendering* section of the tab:

- *Zoom levels count*: determines how deep the quadtree will be. For example, one zoom level means there will be a single tile for the whole layer. Three zoom levels means there will be 16 tiles at the leaf level (every extra zoom level multiplies that by 4). The default is `3` and the maximum is `8`.
- *Show bounding boxes of tiles*: especially useful if there are issues with tiles not showing up when they should.

12.1.8 Fields Properties

The *Fields* tab provides information on fields related to the layer and helps you organize them.

The layer can be made *editable* using the Toggle editing mode. At this moment, you can modify its structure using the New field and Delete field buttons.

When creating New field, the *Comment* option is available only for data sources that allow editing comments (See 数据库条目 for more details). You can also set aliases within *Add Field* dialog, for supported OGR formats (GeoPackage and ESRI File Geodatabase).

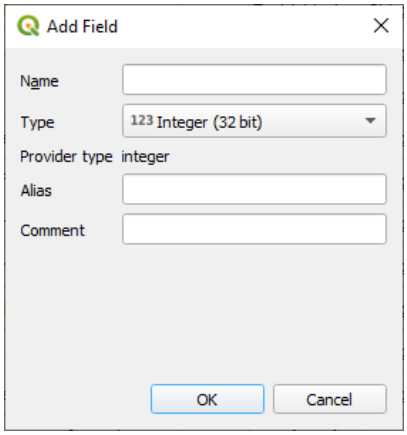


Fig. 12.41: Add Field Dialog

You can also rename fields by double-clicking its name. This is only supported for data providers like PostgreSQL, Oracle, Memory layer and some GDAL layers depending on the GDAL version.

If set in the underlying data source or in the *forms properties*, the field's alias is also displayed. An alias is a human readable field name you can use in the feature form or the attribute table. Aliases are saved in the project file.

Other than the fields contained in the dataset, *virtual fields* and *Auxiliary Storage* included, the *Fields* tab also lists fields from any *joined layers*. Depending on the origin of the field, a different background color is applied to it.

For each listed field, the dialog also lists read-only characteristics such as its *Type*, *Type name*, *Length* and *Precision*'.

Depending on the data provider, you can associate a comment with a field, for example at its creation. This information is retrieved and shown in the *Comment* column and is later displayed when hovering over the field label in a feature form.

Under the *Configuration* column, you can set how the field should behave in certain circumstances:

- `Not searchable`: check this option if you do not want this field to be queried by the *search locator bar*
- `Do not expose via WMS`: check this option if you do not want to display this field if the layer is served as WMS from QGIS server
- `Do not expose via WFS`: check this option if you do not want to display this field if the layer is served as WFS from QGIS server

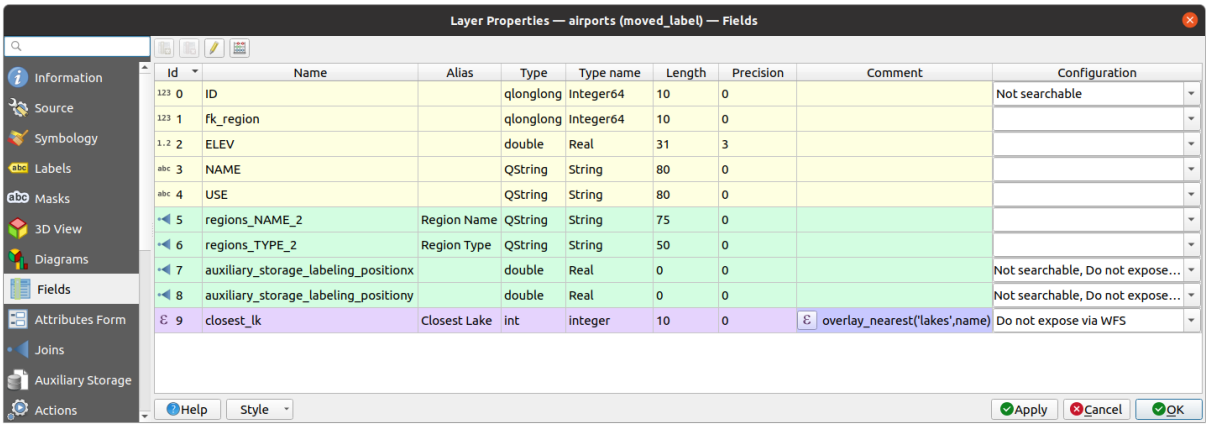


Fig. 12.42: Fields properties tab

12.1.9 Attributes Form Properties

The *Attributes Form* tab helps you set up the form to display when creating new features, editing or querying existing one. This affects them in both *table and form views*. You can define:

- the look and the behavior (label, visibility, widget, constraints...) of each field in the current layer, the *joined layers* and the *relational ones*;
- the placement of enabled *actions*;
- the form's structure (custom or autogenerated);
- extra HTML, QML, text widgets to enhance the form;
- extra logic in Python to handle interaction with the form or field widgets.

At the top right of the dialog, you can set whether the form is opened by default when creating new features. This can be configured per layer or globally with the *Suppress attribute form pop-up after feature creation* option in the *Settings* ► *Options* ► *Digitizing* menu.

Customizing a form for your data

By default, when you click on a feature with the Identify Features tool or switch the attribute table to the *form view* mode, QGIS displays a basic form with predefined widgets (generally spinboxes and textboxes --- each field is represented on a dedicated row by its label next to the widget). If *relations* are set on the layer, fields from the referencing layers are shown in an embedded frame at the bottom of the form, following the same basic structure.

This rendering is the result of the default `Autogenerate` value of the *Attribute editor layout* setting in the *Layer properties* ► *Attributes Form* tab. This property holds three different values:

- `Autogenerate`: keeps the basic structure of "one row - one field" for the form but allows to customize each corresponding widget.
- `Drag-and-drop designer`: other than widget customization, the form structure can be made more complex eg, with widgets embedded in groups and tabs.
- `Provide ui file`: allows to use a Qt designer file, hence a potentially more complex and fully featured template, as feature form.

Just below the *Attribute editor layout* setting, a search box allows you to filter fields, relations, containers and other available widgets.

For items that support aliases, like fields or relations, the filter targets both names and aliases.

The Show field aliases instead of names button allows you to toggle field aliases and field names.

When showing aliases, fields without an alias set will have their names displayed in light gray to help identify them easily.

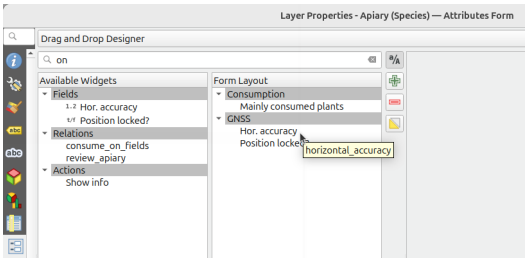


Fig. 12.43: Search fields using aliases and names in Attributes Form page

The autogenerated form

When the `Autogenerate` option is on, the *Available widgets* panel displays lists of fields (from the current layer, the join and relation layers) and the enabled actions that would be shown in the form. Select a field and you can configure its appearance and behavior in the right panel:

- adding *custom label and automated checks* to the field;
- setting a *particular widget* to use.

In this editor mode, only the properties can be viewed for the actions. Moreover, they will be displayed through an *Actions* drop-down menu in the feature form, and following the *Show in attribute table* property in other contexts.

The drag and drop designer

The drag and drop designer allows you to create a form with several containers (tabs or groups) to present the attribute fields or other widgets that are not directly linked to a particular field (like the HTML/QML widgets or the *actions* defined for the layer), as shown for example in Fig. 12.44.

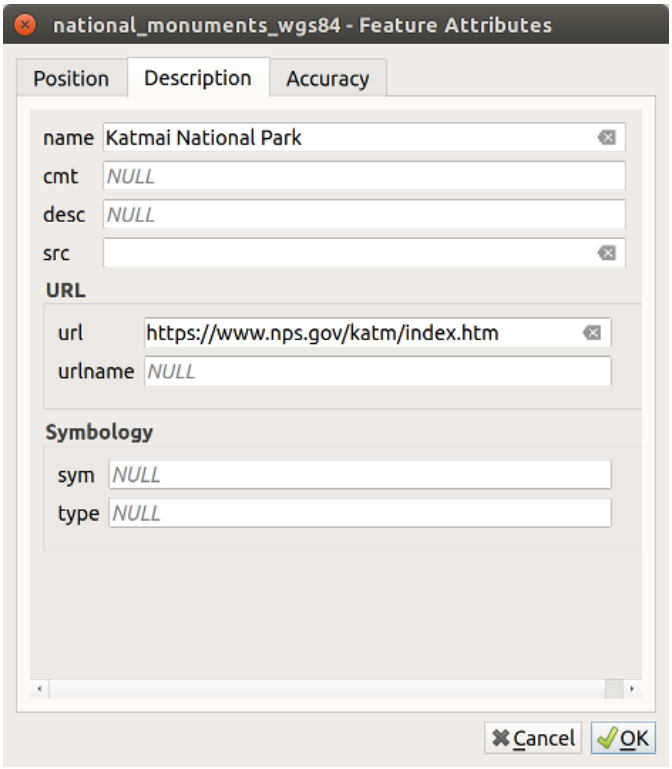


Fig. 12.44: Resulting built-in form with tabs and named groups

1. Choose `Drag and drop designer` from the *Select attribute layout editor* combobox. This enables the *Form Layout* panel next to the *Available widgets* panel, filled with existing fields. The selected field displays its *properties* in a third panel.

2. Select fields you do not want to use in your *Form Layout* panel and hit the button to remove them. You can also toggle the selection with the Invert selection button.

3. Drag and drop fields from the first panel to the *Form Layout* one to re-add them. The same field can be added multiple times.

4. Drag and drop fields within the *Form Layout* panel to reorder their position.

5. Add containers to associate fields that belong to the same category and better structure the form.

    1. The first step is to use the Add a new tab or group to the form layout icon. Fields and other groups will be displayed in it.

    2. Then set the properties of the container, i.e.:

        - the *Label*: the title that will be used for the container
        - the *Container Type*: it can be a *Tab*, *Group box in container* (a collapsible group box inside a tab or another group) or a *Row* (a container type that allows you to arrange your widgets in a horizontal row, automatically determining the number of columns based on the number of widgets),
        - the *Within*: this optional feature allows you to select an existing container in which the new container (*Group box in container* or *Row*) will be embedded.
        - and the *Number of columns* the embedded fields should be distributed over

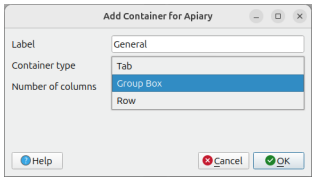


Fig. 12.45: Dialog to create containers with the **Attribute editor layout**

These, and other properties can later be updated by selecting the item and, from the third panel:

- hide or show the container's label
- rename the container
- set the number of columns
- enter an expression to control the container's visibility. The expression will be re-evaluated every time values in the form change, and the tab or group box shown/hidden accordingly
- *Show as Group Box*: converts a tab into a top-level group box and vice versa
- in case of a group box, set whether at the form opening, it should show as *Collapsed* for every features, or only for those matching an expression (*Control collapsed by expression*).
- configure the *Style* look of the container, with custom *Background color*, label color and font properties

3. You can create and embed as many containers as you want by pressing again the Add a new tab or group to the form layout icon.

6. The next step is to assign the relevant fields to each container, by simple drag and drop. Group boxes and tabs can also be moved the same way.
7. *Customize the widget* of the fields in use
8. In case the layer is involved in a *one or many to many relation*, drag-and-drop the relation name from the *Available Widgets* panel to the *Form Layout* panel. The associated layer attribute form will be embedded at the chosen place in the current layer's form. As for the other items, select the relation label to configure some properties:
   - hide or show the relation label
   - show the link button
   - show the unlink button
9. In case the layer has one or more *actions* enabled for `Layer` or `Feature` scope, the actions will be listed under *Actions* and you can drag and drop them as with the other fields. The associated action will be embedded at the chosen place in the current layer's form.
10. Further customize the form by adding one or more widgets from *Other Widgets* (see *Other Widgets*)
11. Apply the layer's properties dialog
12. Open a feature attribute form (eg, using the Identify features tool) and it should display the new form.

Other Widgets

The drag and drop designer offers a number of widgets that are not connected to a particular field. They can be used to enhance the appearance of the form or to display dynamically calculated values.

- *HTML Widget*: embeds an HTML page, the HTML source may contain the result of dynamically calculated expressions.

  HTML widgets can be used for example to display images stored as BLOB in a field (let's call it `photo`):

  1. In the `Drag-and-drop designer` mode, add a *HTML Widget* to your *Form Layout*.
  2. Double-click on the *HTML Widget* to configure it.
  3. Change the default *Title* or hide it.
  4. Press the button and enter the following QGIS expression:

     ```
     '<img src= "' || to_base64("photo") || '">'
     ```

     Ensure that you replace *photo* with your own BLOB field name. The above expression creates a string with HTML image tag in which the BLOB file is encoded.
  5. Apply the dialog and then press the button.
  6. QGIS automatically applies HTML formatting and functions to evaluate your expression, resulting in following code:

     ```
     <script>document.write(expression.evaluate("'<img src=\"data:image/png;
     ↪base64,' || to_base64(\"photo\") || '\">'"));</script>
     ```

     A preview of your image is displayed on the right.

- *QML Widget*: embeds a Qt QML document, displaying graphical elements in the attribute form. Beside the custom *Title* of the added widget and whether it should be shown or not, you can select from predefined *QML code* elements:
  - *Free text...*: allows you to write from scratch or paste an existing code
  - *Rectangle*: provides a minimal code for displaying a rectangle
  - *Bar chart*: provides a minimal code for displaying a bar chart
  - *Pie chart*: provides a minimal code for displaying a pie chart

  You can extend the code with QML syntax, use layer fields or QGIS expressions that are dynamically calculated.

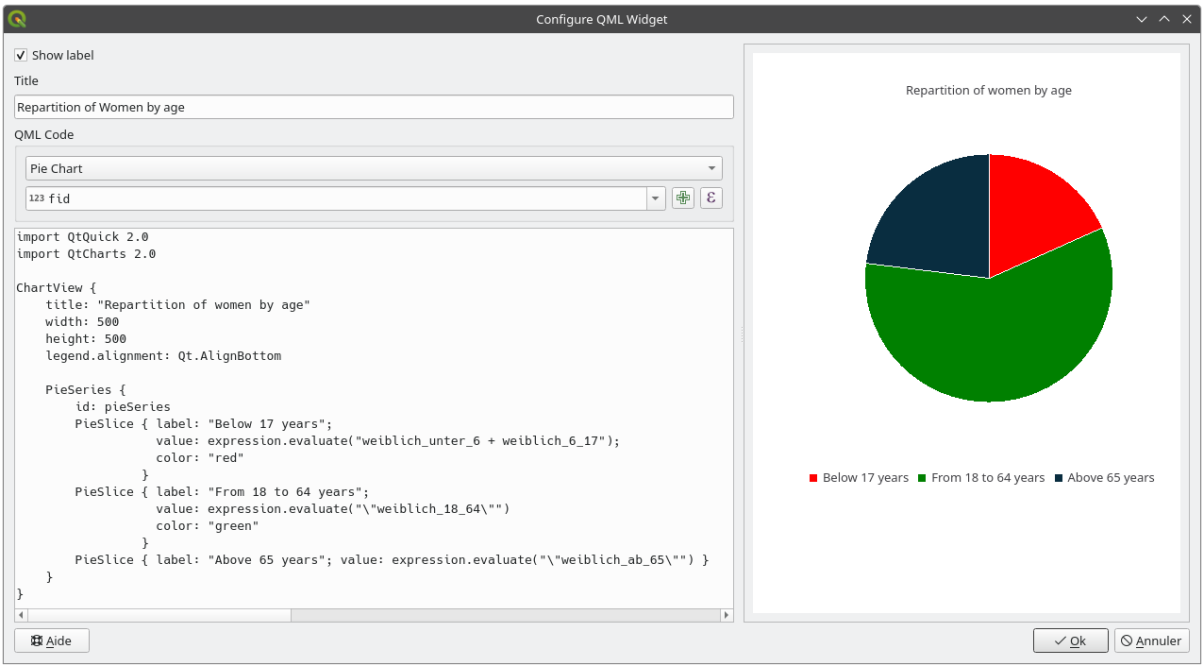


Fig. 12.46: Setting a QML graph to display in attribute form

- *Text Widget*: displays a text widget which supports basic HTML markup and may contain the result of dynamically calculated expressions.
- *Spacer Widget*: inserts an empty transparent rectangle, increasing the vertical distance between two widgets.

Tip

Display Dynamic Content

The widgets mentioned above (except the *Spacer Widget*) support expressions that can be used to display content that dynamically changes whenever another field in the form changes. This can be achieved using the `current_value('field_name')` function in the expression to examine the value of another field.

Using custom ui-file

The `Provide ui-file` option allows you to use complex dialogs made with Qt-Designer. Using a UI-file allows a great deal of freedom in creating a dialog. Note that, in order to link the graphical objects (textbox, combobox...) to the layer's fields, you need to give them the same name.

Use the *Edit UI* to define the path to the file to use.

UI-files can also be hosted on a remote server. In this case, you provide the URL of the form instead of the file path in *Edit UI*.

You'll find some example in the Creating a new form lesson of the QGIS-training-manual-index-reference. For more advanced information, see https://woostuff.wordpress.com/2011/09/05/qgis-tips-custom-feature-forms-with-python-logic/.

Enhance your form with custom functions

QGIS forms can have a Python function that is called when the dialog is opened. Use this function to add extra logic to your dialogs. The form code can be specified in three different ways:

- `load from the environment`: use a function, for example in `startup.py` or from an installed plugin
- `load from an external file`: a file chooser will let you select a Python file from your filesystem or enter a URL for a remote file.
- `provide code in this dialog`: a Python editor will appear where you can directly type the function to use.

In all cases you must enter the name of the function that will be called (`open` in the example below).

An example is (in module MyForms.py):

```
def open(dialog,layer,feature):
    geom = feature.geometry()
    control = dialog.findChild(QWidget,"My line edit")
```

Reference in Python Init Function like so: `open`

Configure the field behavior

The main part of the *Attributes Form* tab helps you set the type of widget used to fill or display values of the field, in the attribute table or the feature form: you can define how user interacts with each field and the values or range of values that are allowed to be added to each.

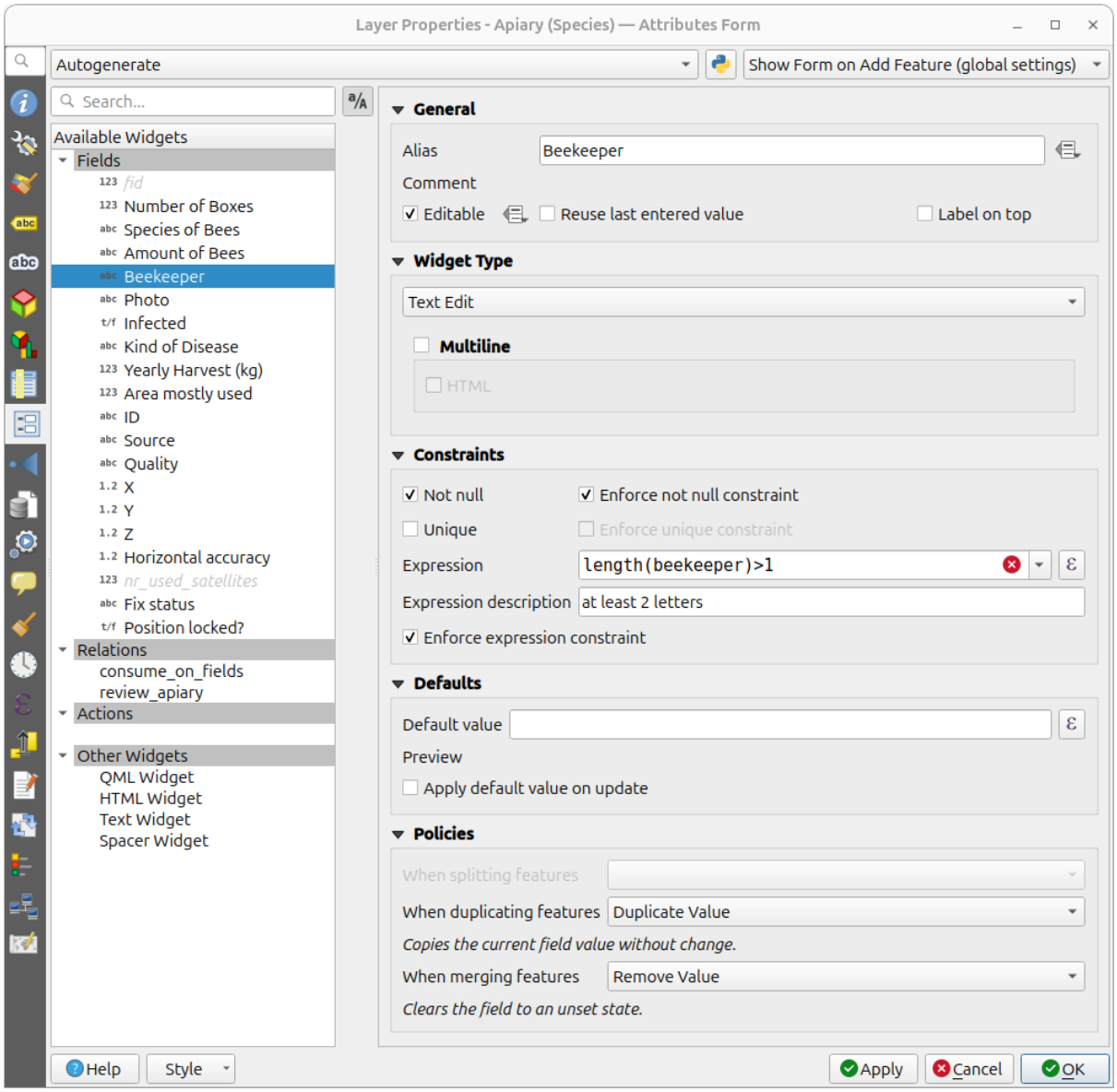


Fig. 12.47: Dialog to select an edit widget for an attribute column

Tip

Copy and paste widget configuration

The field configuration can be copied to the clipboard and then pasted to another field via right click on a field in the *Available widgets* panel.

This comes handy to save time and to make sure that common settings are correctly applied to several related fields, e.g., when configuring latitude and longitude field widgets.

Copy and paste of widget configuration can be done between fields of the same layer, between fields from different layers in the QGIS project, or between fields from layers in different QGIS instances.

Note that neither aliases nor comments are copied, since those are specific to each particular field.

Common settings

Regardless the type of widget applied to the field, there are some common properties you can set to control whether and how a field can be edited.

Widget display

Only available for the *Drag and drop* designer mode, this group helps you configure the look of the widget assigned to the field:

- *Show label*: indicates whether the field name should be displayed in the form
- *Override label color*: applies specific color to the field's label
- *Override label font*: applies specific font properties (bold, italic, underline, strikethrough, or font family) to the field's label
- *Size*: allows to control how widgets will relatively resize when resizing an attribute form.
  - *Horizontal stretch*: sets a higher horizontal value for widgets that need more horizontal space.
  - *Vertical stretch*: determines how widgets resize vertically when the form is resized.

General options

- *Alias*: a human readable name to use for fields. The alias will be displayed in the feature form, the attribute table, or in the *Identify results* panel. It can also be used as field name replacement in the *expression builder*, easing expressions understanding and reviews. Aliases are saved in project file.
- *Comment*: displays the field's comment as shown in the *Fields* tab, in a read-only state. This information is shown as tooltip when hovering over the field label in a feature form.
- *Editable*: uncheck this option to set the field read-only (not manually modifiable) even when the layer is in edit mode. Note that checking this setting doesn't override any edit limitation from the provider. This option can be controlled by a data-defined property thanks to the Data-defined override button.
- *Reuse last entered value*: remembers the last value entered in this field and uses it as default for the next feature being edited in the layer.
- *Label on top*: places the field name above or beside the widget in the feature form.

Constraints

You can constrain the value to insert in the field. This constraint can be:

- *Not null*: requires the user to provide a value;
- *Unique*: guarantee the inserted value to be unique throughout the field;
- based on a custom *expression*: e.g. `not regexp_match(col0,'[^A-Za-z]')` will ensure that the value of the field *col0* has only alphabet letters. A short description can be added to help you remember the constraint.

Whenever a value is added or edited in a field, it's submitted to the existing constraints and:

- if it meets all the requirements, a green check is shown beside the field in the form;
- if it does not meet all the requirements, then the field is colored in yellow or orange and a corresponding cross is displayed next to the widget. You can hover over the cross to remind which constraints are applied to the field and fix the value:

  - A yellow cross appears when the unmet constraint is an unenforced one (`soft constraint`) and it does not prevent you to save the changes with the "wrong" values;
  - An orange cross can not be ignored and does not allow you to save your modifications until they meet the constraints. It appears when the *Enforce constraint* option is checked (`hard constraint`).

Default values

- *Default value*: for new features, automatically populates by default the field with a predefined value or an *expression-based one*. For example, you can:
  - use `$x`, `$length`, `$area` to automatically populate a field with the feature's X coordinate, length, area or any geometric information at its creation;
  - increment a field by 1 for each new feature using `maximum("field")+1`;
  - save the feature creation datetime using `now()`;
  - use *variables* in expressions, making it easier to e.g. insert the operator name (`@user_full_name`), the project file path (`@project_path`), ...

  A preview of the resulting default value is displayed at the bottom of the widget.

  > **Note**
  >
  > The `Default value` option is not aware of the values in any other field of the feature being created so it won't be possible to use an expression combining any of those values i.e using an expression like `concat(field1, field2)` may not work.

- *Apply default value on update*: whenever the feature attribute or geometry is changed, the default value is recalculated. This could be handy to save values like last user that modifies data, last time it was changed...

Policies

Field policies determine how values are assigned to fields during various editing operations:

Split and Duplicate Policies

These policies apply when *Splitting features* or *Duplicating features*:

- *Duplicate Values*: Keeps the existing value of the field for the new features.
- *Use Default Value*: Resets the field by recalculating its *default value*. If no default value clause exists, the existing value is kept for the new features.
- *Remove Value*: Clears the field to an unset state.
- *Use Ratio Geometries*: Recalculates the field value for all split portions by multiplying the existing value by ratio of the split parts lengths or areas.

Merge Policies

These policies determine initial values when *Merging features*:

- *Remove Value*: Clears the field to an unset state (data provider may populate default value).
- *Use Default Value*: Uses the default field value set in QGIS. (expressions use a dummy feature with the merged geometry; references to other fields won't work).
- *Use Sum* (numeric fields only): Sums values from all merged features.
- *Use Minimum Value* (numeric fields only): Uses smallest value from the selected features.
- *Use Maximum Value* (numeric fields only): Uses largest value from the selected features.

- *Average Weighted by Geometry* (numeric fields only): New values are computed as the weighted average of the source values.
- *Use Largest Feature*: Uses value from feature with the largest geometry.
- *Set to NULL*: Sets attribute to NULL.

Edit widgets

Based on the field type, QGIS automatically determines and assigns a default widget type to it. You can then replace the widget with any other compatible with the field type. The available widgets are:

- **Binary (BLOB)**: Available only for binary fields, it displays by default a label with the size of the embedded data, if not empty. A drop-down button next to the label allows to:
  - *Embed file*, replacing or filling the field
  - *Clear contents*, removing any data in the field
  - *Save contents to file*, exporting the data as a file on disk

  It is also possible to preview the embedded binary file in the field, if combined in a drag-and-drop form with e.g. a *QML or HTML widget*.
- **Checkbox**: Displays a checkbox whose state defines the value to insert. Internally, the widget stores a value (e.g. 1 for checked, 0 for unchecked) while representing it visually as a toggled box. You can configure the stored values for the checked and unchecked states in the representation. It is also possible to display the checkbox state as a plain text using "True"/"False" labels. These settings influence how the values are shown in classification renderers.
- **Classification**: Only available when a *categorized symbology* is applied to the layer, displays a combo box with the values of the classes.
- **Color**: Displays a *color widget* allowing to select a color; the color value is stored as a html notation in the attribute table.
- **Date/Time**: Displays a line field which can open a calendar widget to enter a date, a time or both. Column type must be text. You can select a custom format, pop-up a calendar, etc.
- **Enumeration**: Opens a combo box with predefined values fetched from the database. This is currently only supported by the PostgreSQL provider, for fields of `enum` type.
- **Attachment**: Uses a "Open file" dialog to store file path in a relative or absolute mode. It can be used to display the document path as a hyperlink or render the document within a dedicated widget in the form. Supported document types are image, web page, audio and video, and the supported file formats depend on the Operating System. You can also configure an *external storage system* to fetch/store resources.

  > **Tip**
  >
  > **Relative Path in Attachment widget**
  >
  > If the path which is selected with the file browser is located in the same directory as the `.qgs` project file or below, paths are converted to relative paths. This increases portability of a `.qgs` project with multimedia information attached.

- **Geometry**: Compatible with the *Geometry* field type. For example, PostgreSQL and Memory layers can contain multiple geometry fields. QGIS uses the first geometry column as the layer geometry. Applying the Geometry widget to additional geometry fields allows lossless use of the data in the attribute form:
  - Copy in the formats WKT (Well-known text representation, e.g. `POINT (30 10)`) or GeoJSON
  - Paste
  - Clear

- **Hidden**: A hidden attribute column is invisible. The user is not able to see its contents.
- **Key/Value**: Displays a two-columns table to store sets of key/value pairs within a single field. This is currently supported by the PostgreSQL provider, for fields of `hstore` type.
- **JSON View**: Displays JSON data in a syntax highlighted text edit or in tree view. This widget is currently read only. Several options are available to change how the data is displayed. 'Default view' specify if the widget should appear in Text or Tree mode. 'Format JSON' has three options which are related to the tree view only:
  - Indented: Display data in a human readable form with newlines and four space characters for indentation.
  - Compact: Display data in a one-line size optimized string without newlines or spaces.
  - Disabled: Display data as it comes from the provider.
- **List**: Displays a single column table to add different values within a single field. This is currently supported by the PostgreSQL provider, for fields of `array` type.
- **Range**: Allows you to set numeric values from a specific range. The edit widget can be either a slider or a spin box.

  > **Note**
  >
  > Some layers, such as GeoPackage or ESRI File Geodatabase, with predefined **range Field Domains** will be automatically recognized by QGIS and assigned a **Range** widget for the relevant fields. The widget will be prefilled with the minimum and maximum values specified in the domain.

- **Relation Reference**: This is the default widget assigned to the referencing field (i.e., the foreign key in the child layer) when a *relation* is set. It provides direct access to the parent feature's form which in turn embeds the list and form of its children. The number of entries in the widget can be limited for efficiency, and if limit is not set, all entries will be loaded. The field defined in the display expression is used as representable value in classification renderers.
- **Text Edit** (default): This opens a text edit field that allows simple text or multiple lines to be used. If you choose multiple lines you can also choose html content.
- **Unique Values**: You can select one of the values already used in the attribute table. If 'Editable' is activated, a line edit is shown with autocompletion support, otherwise a combo box is used.
- **Uuid Generator**: Generates a read-only UUID (Universally Unique Identifiers) field, if empty.
- **Value Map**: A combo box with predefined items. The value is stored in the attribute, the description is shown in the combo box. You can define values manually or load them from a layer or a CSV file. The description is used as the representable value, meaning it will be shown in classification renderers.

  > **Note**
  >
  > Some layers, such as GeoPackage or ESRI File Geodatabase, with predefined **coded Field Domains** will be automatically recognized by QGIS and assigned a **Value Map** widget for the relevant fields.

- **Value Relation**: Offers values from a related table in a combobox. You can select layer, key column and value column. The key column is the stored value in the attribute table, while the value column is the label shown in the combobox. Value column is also the representable value that will be shown in the classification renderers. You may also choose to order the list of displayed values by the *Key* column (the default sort order), the *Value* column, or a specific *Field* of the layer. Enable the *Descending order* if you wish the values to be sorted in reverse order.

  Several options are available to change the standard behavior: allow NULL value, allow multiple selections and use of auto-completer. The forms will display either a drop-down list or a line edit field when completer checkbox is enabled.

If a layer that is stored in PostgreSQL, GeoPackage or SpatiaLite is configured to use a value relation widget, but the required layer is not already loaded into the project, QGIS will automatically search for the layer in the same database/connection.

12.1.10 Joins Properties

The *Joins* tab allows you to associate features of the current layer to features from another loaded vector layer (or table). The join is based on an attribute that is shared by the layers, in a one-to-one relationship. For more details on joins, please read *Joining features between two layers*.

12.1.11 Auxiliary Storage Properties

The regular way to customize styling and labeling is to use data-defined properties as described in *Data defined override setup*. However, it may not be possible if the underlying data is read only. Moreover, configuring these data-defined properties may be very time consuming or not desirable! For example, if you want to fully use map tools coming with *The Label Toolbar*, then you need to add and configure more than 20 fields in your original data source (X and Y positions, rotation angle, font style, color and so on).

The Auxiliary Storage mechanism provides the solution to these limitations and awkward configurations. Auxiliary fields are a roundabout way to automatically manage and store these data-defined properties (labels, diagram, symbology...) in a SQLite database thanks to editable joins. This allows you to store properties for layers that aren't editable.

A tab is available in vector layer properties dialog to manage auxiliary storage:

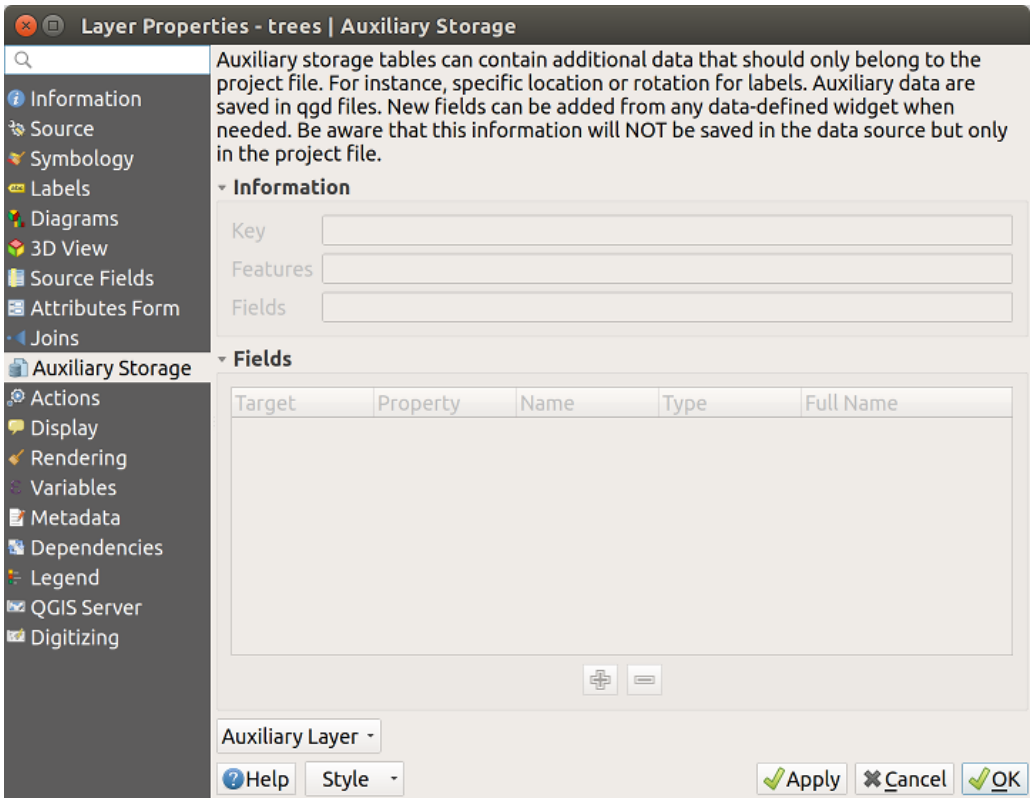


Fig. 12.48: Auxiliary Storage tab

Labeling

Considering that the data source may be customized thanks to data-defined properties without being editable, labeling map tools described in *The Label Toolbar* are always available as soon as labeling is activated.

Actually, the auxiliary storage system needs an auxiliary layer to store these properties in a SQLite database (see *Auxiliary storage database*). Its creation process is run the first time you click on the map while a labeling map tool is currently activated. Then, a window is displayed, allowing you to select the primary key to use for joining (to ensure that features are uniquely identified):

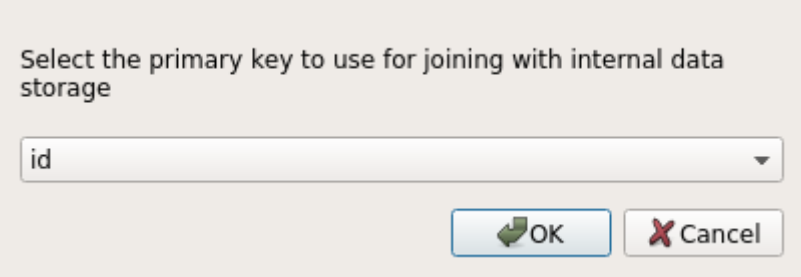


Fig. 12.49: Auxiliary Layer creation dialog

As soon as an auxiliary layer is configured for the current data source, you can retrieve its information in the tab:

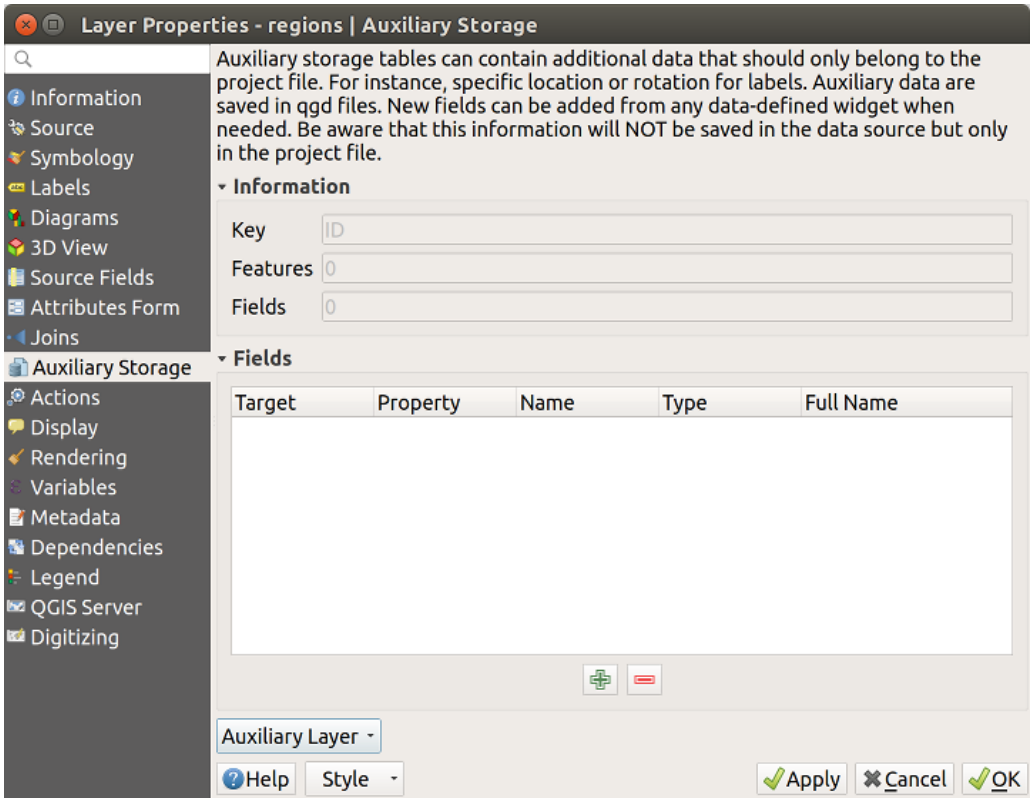


Fig. 12.50: Auxiliary Layer key

The auxiliary layer now has these characteristics:

- the primary key is `ID`,
- there are `0` features using an auxiliary field,
- there are `0` auxiliary fields.

Now that the auxiliary layer is created, you can edit the layer labels. Click on a label while the Change Label map tool is activated, then you can update styling properties like sizes, colors, and so on. The corresponding data-defined properties are created and can be retrieved:

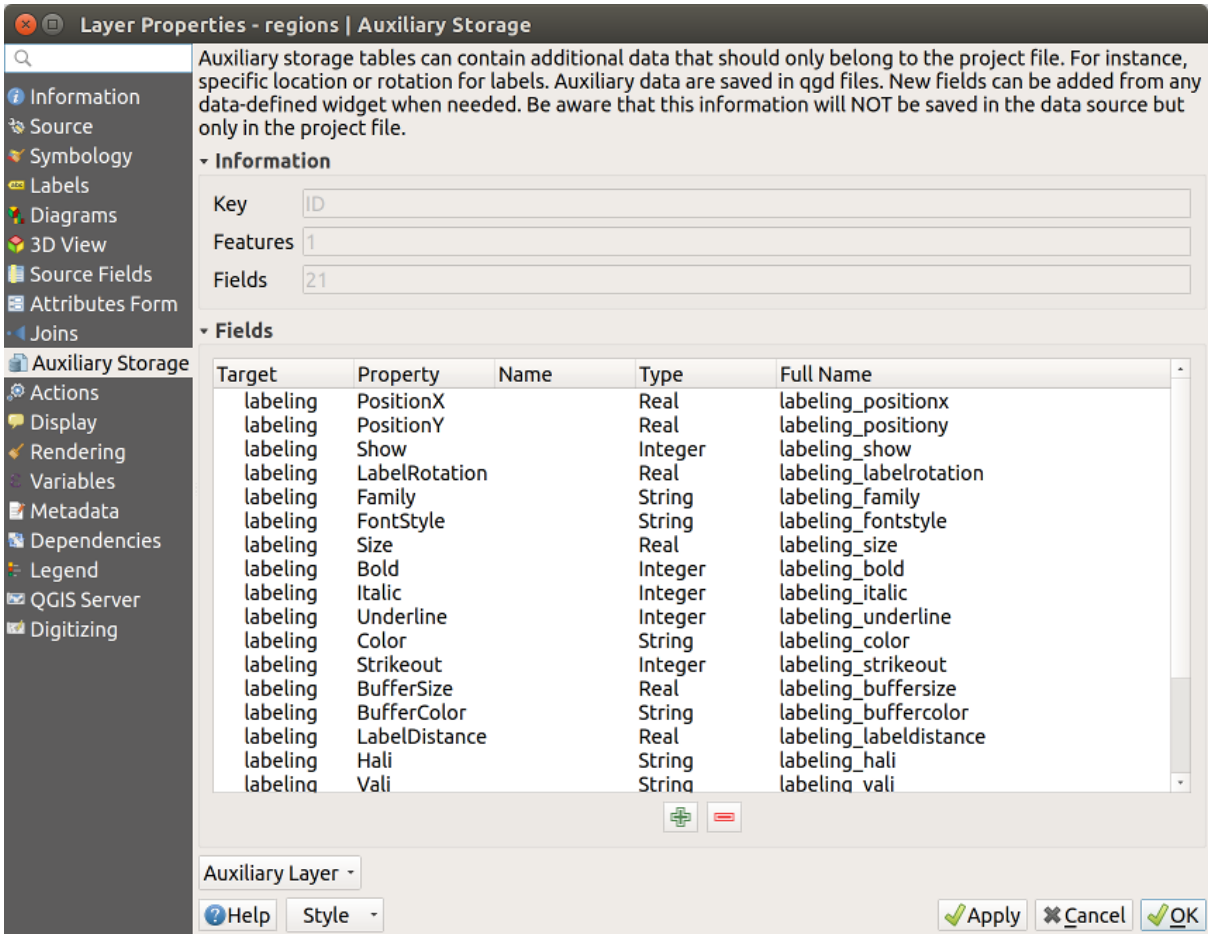


Fig. 12.51: Auxiliary Fields

As you can see in the figure above, `21` fields are automatically created and configured for labeling. For example, the `FontStyle` auxiliary field type is a `String` and is named `labeling_fontstyle` in the underlying SQLite database. There is also `1` feature which is currently using these auxiliary fields.

Notice that the icon is displayed in the *Labels* properties tab indicating that the data-defined override options are set correctly:

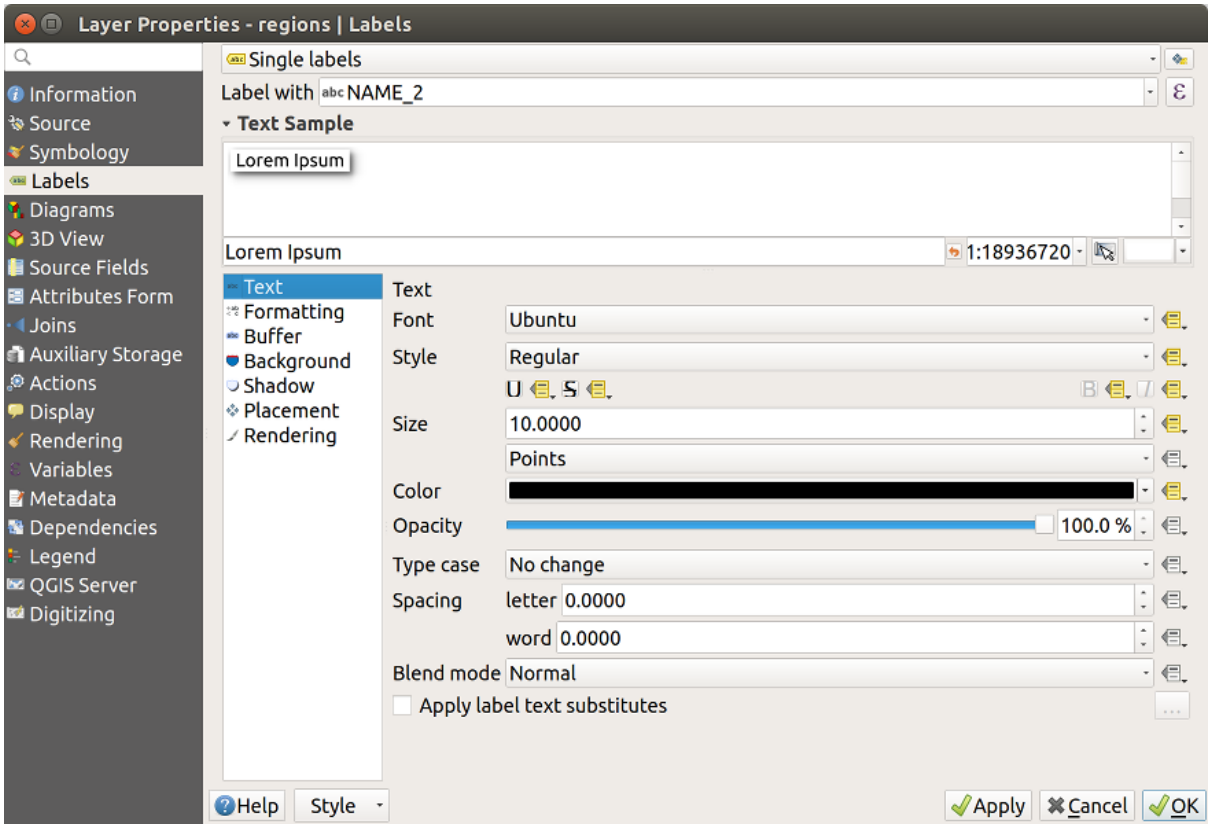


Fig. 12.52: Data-defined properties automatically created

Otherwise, there's another way to create an auxiliary field for a specific property thanks to the Data-defined override button. By clicking on *Store data in the project*, an auxiliary field is automatically created for the *Opacity* field. If you click on this button and the auxiliary layer is not created yet, a window (Fig. 12.49) is first displayed to select the primary key to use for joining.

Symbology

Like the method described above for customizing labels, auxiliary fields can also be used to stylize symbols and diagrams. To do this, click on Data-defined override and select *Store data in the project* for a specific property. For example, the *Fill color* field:

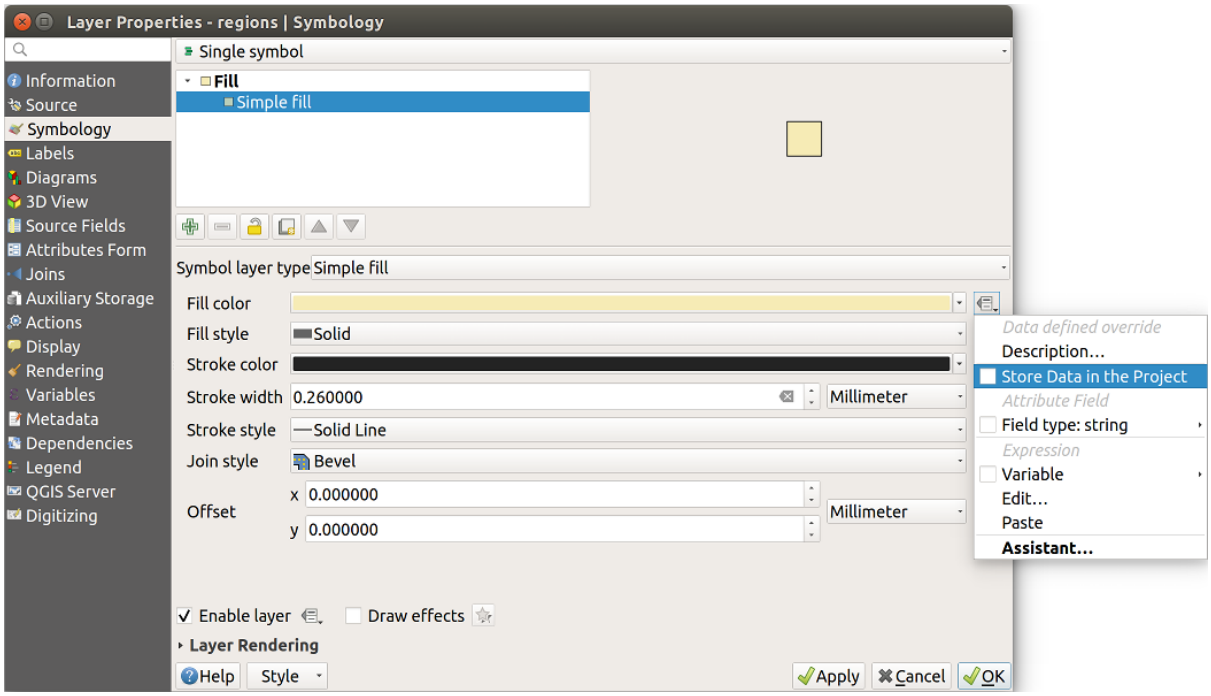


Fig. 12.53: Data-defined property menu for symbol

There are different attributes for each symbol (e.g. fill style, fill color, stroke color, etc...), so each auxiliary field representing an attribute requires a unique name to avoid conflicts. After selecting *Store data in the project*, a window opens and displays the *Type* of the field and prompts you to enter a unique name for the auxiliary field. For example, when creating a *Fill color* auxiliary field the following window opens:

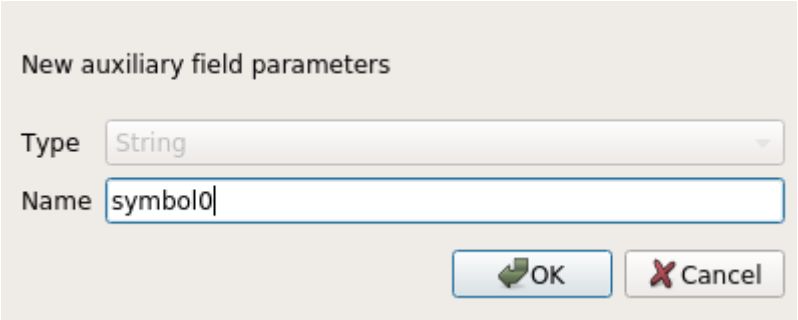


Fig. 12.54: Name of the auxiliary field for a symbol

Once created, the auxiliary field can be retrieved in the auxiliary storage tab:

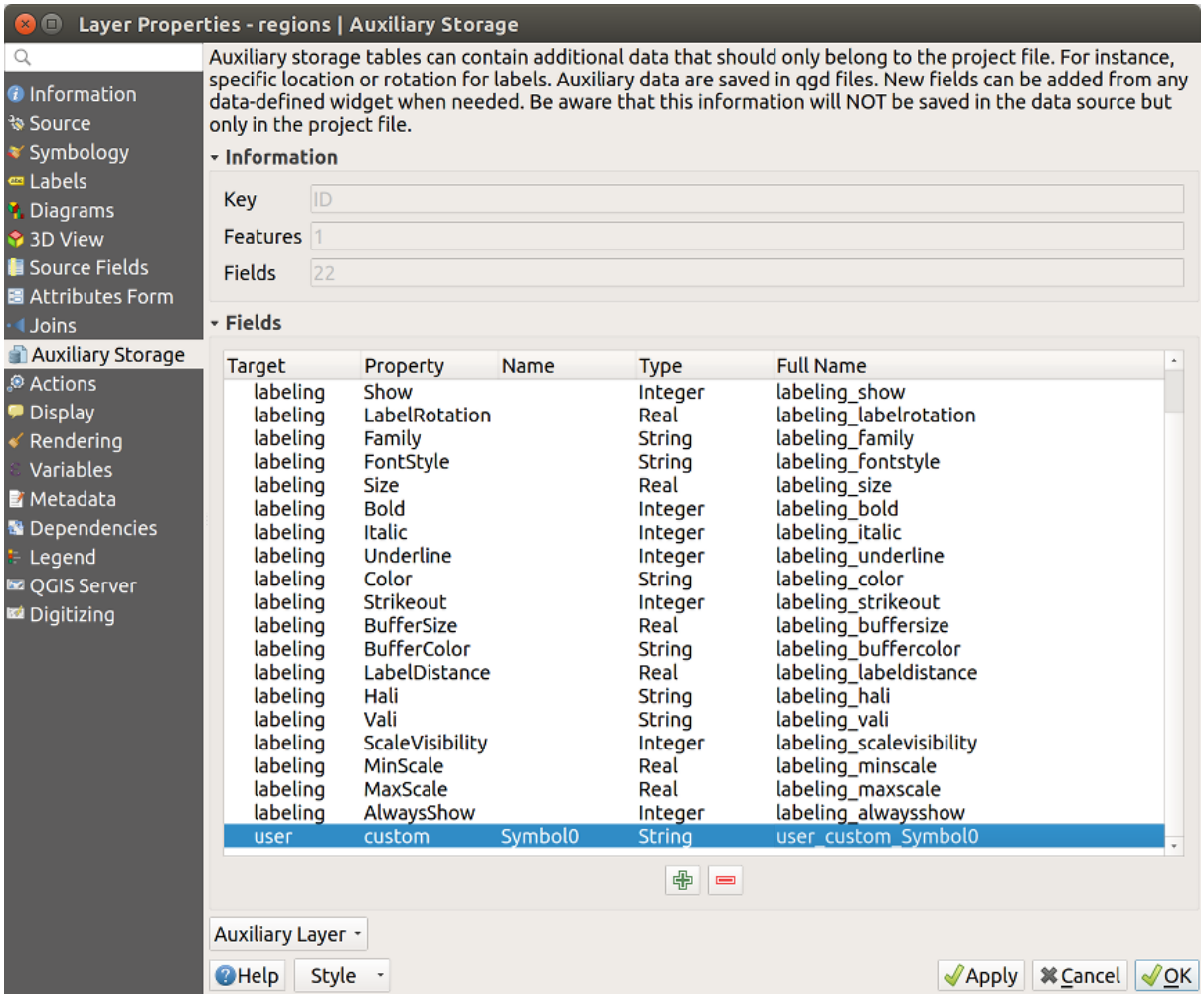


Fig. 12.55: Auxiliary field symbol

Attribute table and widgets

Auxiliary fields can be edited using the *attribute table*. However, not all auxiliary fields are initially visible in the attribute table.

Auxiliary fields representing attributes of a layer's symbology, labeling, appearance, or diagrams will appear automatically in the attribute table. The exception are attributes that can be modified using the *Label Toolbar* which are hidden by default. Auxiliary fields representing a `Color` have a widget **Color** set by default, otherwise auxiliary fields default to the **Text Edit** widget.

Auxiliary fields that represent attributes that can be modified using the *Label toolbar* are **Hidden** in the attribute table by default. To make a field visible, open the *Attribute Form properties tab* and change the value of an auxiliary field *Widget Type* from **Hidden** to another relevant value. For example, change the **auxiliary_storage_labeling_size** to **Text Edit** or change **auxiliary_storage_labeling_color** to the **Color** widget. Those fields will now be visible in the attribute table.

Auxiliary fields in the attribute table will appear like the following image:

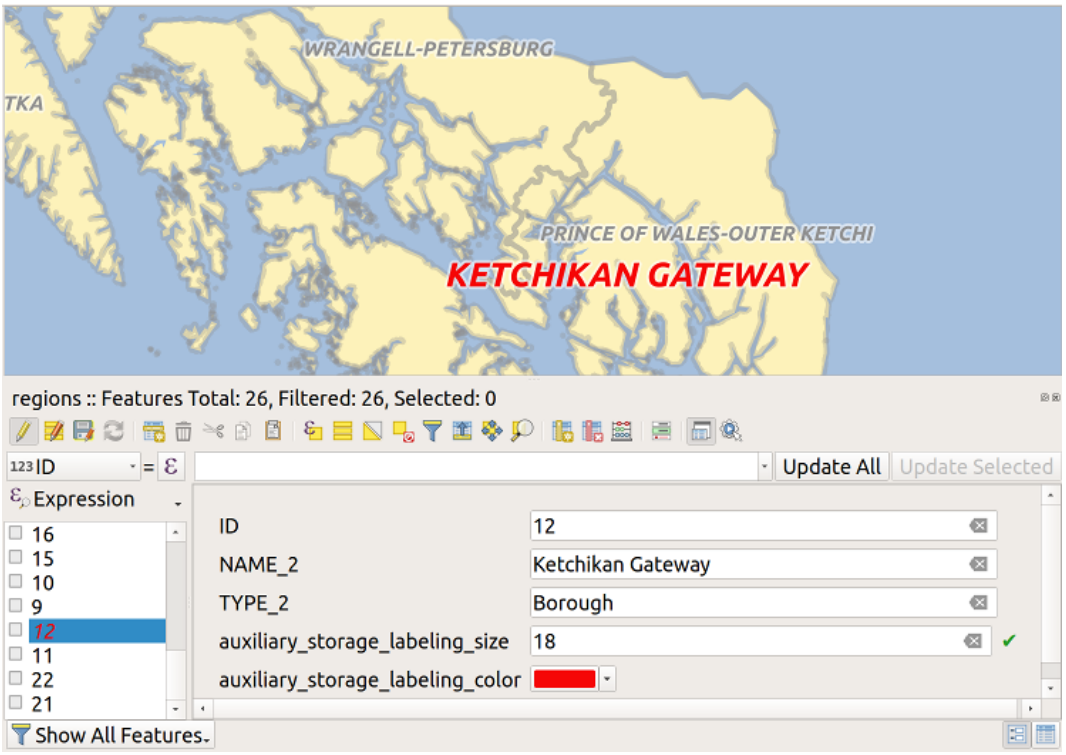


Fig. 12.56: Form with auxiliary fields

Management

The *Auxiliary Layer* menu allows you to manage the auxiliary fields:

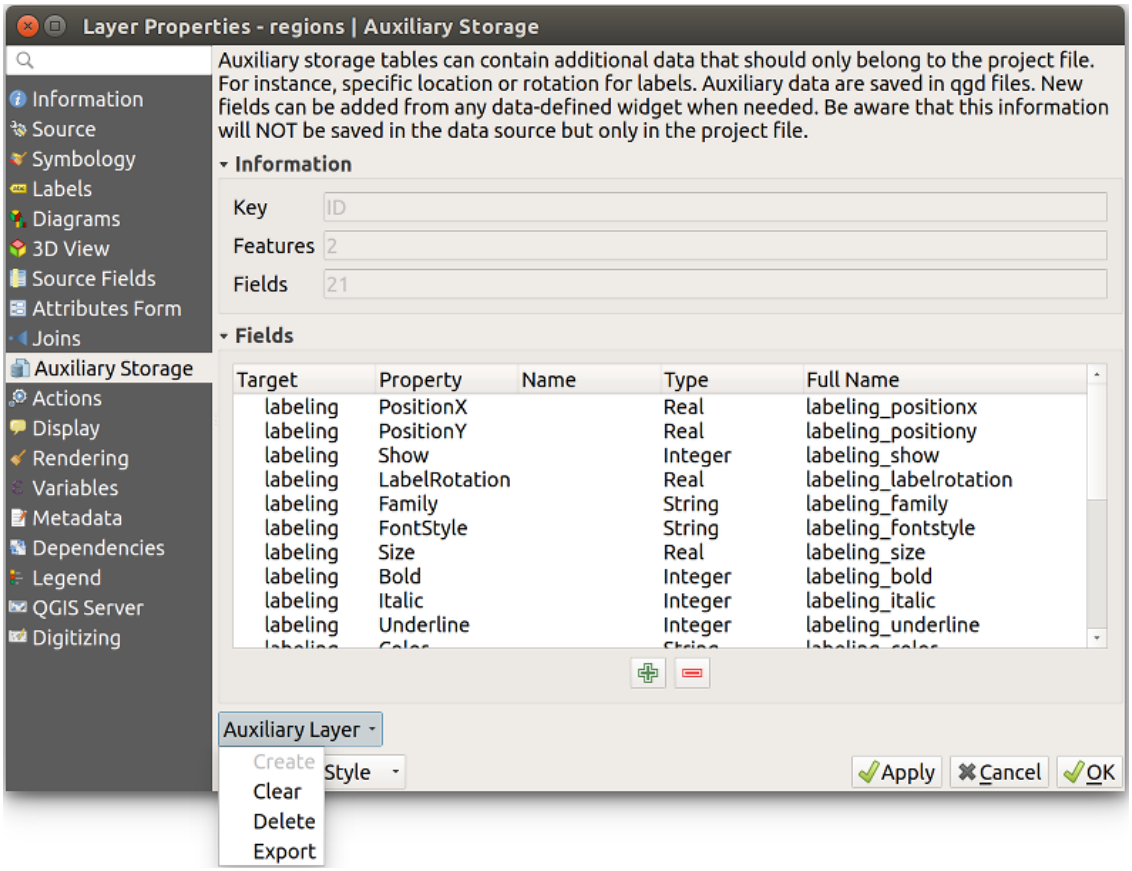


Fig. 12.57: Auxiliary layer management

The first item *Create* is disabled in this case because the auxiliary layer is already created. But in case of a fresh work, you can use this action to create an auxiliary layer. As explained in *Labeling*, a primary key will be needed then.

The *Clear* action allows to keep all auxiliary fields, but remove their contents. This way, the number of features using these fields will fall to `0`.

The *Delete* action completely removes the auxiliary layer. In other words, the corresponding table is deleted from the underlying SQLite database and properties customization are lost.

Finally, the *Export* action allows to save the auxiliary layer as a *new vector layer*. Note that geometries are not stored in auxiliary storage. However, in this case, geometries are exported from the original data source too.

Auxiliary storage database

When you save your project with the `.qgs` format, the SQLite database used for auxiliary storage is saved at the same place but with the extension `.qgd`.

For convenience, an archive may be used instead thanks to the `.qgz` format. In this case, `.qgd` and `.qgs` files are both embedded in the archive.

12.1.12 Actions Properties

The *Actions* tab provides the ability to perform an action based on the attributes of a feature. This can be used to perform any number of actions, for example, running a program with arguments built from the attributes of a feature or passing parameters to a web reporting tool.

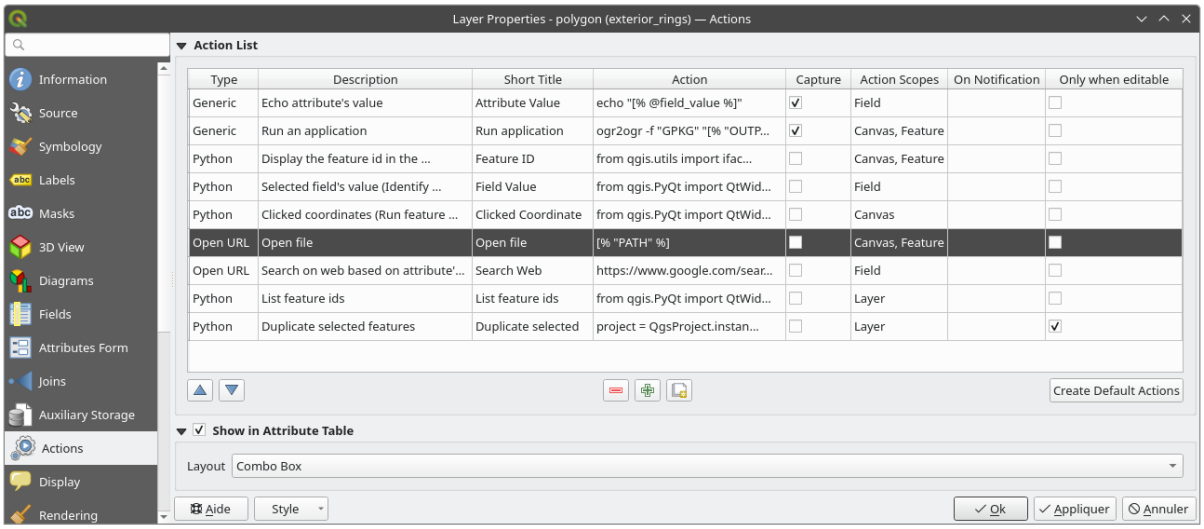


Fig. 12.58: Overview action dialog with some sample actions

Actions are useful when you frequently want to run an external application or view a web page based on one or more values in your vector layer. They are of different types and can be used like this:

- *Generic*, *macOS*, *Windows* and *Unix* actions start an external process.
- *Python* actions execute a Python expression.
- *Generic* and *Python* actions are visible everywhere.
- *macOS*, *Windows* and *Unix* actions are visible only on the respective platform (i.e., you can define three ”Edit” actions to open an editor and the users can only see and execute the one ”Edit” action for their platform to run the editor).
- *Open URL*: Uses a HTTP GET request to open a provided URL.
- *Submit URL (urlencoded or JSON)*: Same as the *Open URL* action but using a HTTP POST request. Data are posted to a URL, using ”application/x-www-form-urlencoded” or ”application/json” if the body is a valid JSON.

  An example of action call could be:

  ```
  http://localhost:8000?/[% url_encode(map('file', 'index.php')) %]
  ```

- *Submit URL (multipart)*: Same as the *Open URL* action but using a HTTP POST request. Data are posted to a URL, using ”multipart/form-data”.

There are several examples included in the dialog. You can load them by clicking on *Create Default Actions*. To edit any of the examples, double-click its row. One example is performing a search based on an attribute value. This concept is used in the following discussion.

The *Show in Attribute Table* allows you to display in the attribute table dialog the checked feature-scoped actions, either as *Combo Box* or as *Separate Buttons* (see *Configuring the columns*).

Defining Actions

To define an attribute action, open the vector *Layer Properties* dialog and click on the *Actions* tab. In the *Actions* tab, click the Add a new action to open the *Edit Action* dialog.

Select the action *Type* and provide a descriptive name for the action. The action itself must contain the name of the application that will be executed when the action is invoked. You can add one or more attribute field values as arguments to the application. When the action is invoked, any set of characters that start with a `%` followed by the name of a field will be replaced by the value of that field. The special characters `%%` will be replaced by the value of the field that was selected from the identify results or attribute table (see *Using Actions*). Double quote marks can be used to group text into a single argument to the program, script or command. Double quotes will be ignored if preceded by a backslash.

Actions can invoke a single process, with arguments, so Boolean operators (such as `&`, `&&`, `;`, `|`) will not work. In UNIX-like operating systems multiple commands can be executed via `bash -c`.

The *Action Scopes* allows you to define where the action should be available. You have following choices:

1. *Field*: action is available when right click in the cell within the attribute table, in the feature form and in the default action button of the main toolbar.
2. *Feature*: action is available when right click in the cell within the attribute table.
3. *Canvas*: action is available in the main action button in the toolbar.
4. *Form*: action is available only in a feature form designed using the *drag-and-drop* mode.
5. *Layer*: action is available in the action button in the attribute table toolbar. Be aware that this type of action involves the entire layer and not the single features.

If you have field names that are substrings of other field names (e.g., `col1` and `col10`), you should indicate that by surrounding the field name (and the % character) with square brackets (e.g., `[%col10]`). This will prevent the `%col10` field name from being mistaken for the `%col1` field name with a `0` on the end. The brackets will be removed by QGIS when it substitutes in the value of the field. If you want the substituted field to be surrounded by square brackets, use a second set like this: `[[%col10]]`.

Using the *Identify Features* tool, you can open the *Identify Results* dialog. It includes a *(Derived)* item that contains information relevant to the layer type. The values in this item can be accessed in a similar way to the other fields by proceeding the derived field name with `(Derived).`. For example, a point layer has an `X` and `Y` field, and the values of these fields can be used in the action with `%(Derived).X` and `%(Derived).Y`. The derived attributes are only available from the *Identify Results* dialog box, not the *Attribute Table* dialog box.

Two example actions are shown below:

- `konqueror https://www.google.com/search?q=%nam`
- `konqueror https://www.google.com/search?q=%%`

In the first example, the web browser konqueror is invoked and passed a URL to open. The URL performs a Google search on the value of the `nam` field from our vector layer. Note that the application or script called by the action must be in the path, or you must provide the full path. To be certain, we could rewrite the first example as: `/opt/kde3/bin/konqueror https://www.google.com/search?q=%nam`. This will ensure that the konqueror application will be executed when the action is invoked.

The second example uses the %% notation, which does not rely on a particular field for its value. When the action is invoked, the %% will be replaced by the value of the selected field in the identify results or attribute table.

Duplicating Actions

QGIS allows you to duplicate existing actions. To duplicate an attribute action, open the vector *Layer Properties* dialog and click on the *Actions* tab. In the *Actions* tab, click the Duplicate an action to open the *Duplicate Action* dialog. You must have selected at least one existing action in order to create a duplicate.

In the dialogue that appears, make any changes that are necessary. See *Defining Actions* for further information. Once finished, press *OK* to create a duplicate of the action with any changes that you made. If you did not edit the description, or if you changed it to be identical to the description of any other existing action, "_1" will be added to the end of it.

Using Actions

QGIS offers many ways to execute actions you enabled on a layer. Depending on their settings, they can be available:

- in the drop-down menu of Run Feature Action button from the *Attributes toolbar* or *Attribute table* dialog;
- when right-clicking a feature with the Identify Features tool (see 识别要素 for more information);
- from the *Identify Results* panel, under the *Actions* section;
- as items of an *Actions* column in the *Attribute Table* dialog.

If you are invoking an action that uses the `%%` notation, right-click on the field value in the *Identify Results* dialog or the *Attribute Table* dialog that you wish to pass to the application or script.

Here is another example that pulls data out of a vector layer and inserts it into a file using bash and the `echo` command (so it will only work on or perhaps). The layer in question has fields for a species name `taxon_name`, latitude `lat` and longitude `long`. We would like to be able to make a spatial selection of localities and export these field values to a text file for the selected record (shown in yellow in the QGIS map area). Here is the action to achieve this:

```
bash -c "echo \"%taxon_name %lat %long\" >> /tmp/species_localities.txt"
```

After selecting a few localities and running the action on each one, opening the output file will show something like this:

```
Acacia mearnsii -34.0800000000 150.0800000000
Acacia mearnsii -34.9000000000 150.1200000000
Acacia mearnsii -35.2200000000 149.9300000000
Acacia mearnsii -32.2700000000 150.4100000000
```

As an exercise, we can create an action that does a Google search on the `lakes` layer. First, we need to determine the URL required to perform a search on a keyword. This is easily done by just going to Google and doing a simple search, then grabbing the URL from the address bar in your browser. From this little effort, we see that the format is https://www.google.com/search?q=QGIS, where `QGIS` is the search term. Armed with this information, we can proceed:

1. Make sure the `lakes` layer is loaded.
2. Open the *Layer Properties* dialog by double-clicking on the layer in the legend, or right-click and choose *Properties* from the pop-up menu.
3. Click on the *Actions* tab.
4. Click Add a new action.
5. Choose the *Open URL* action type,
6. Enter a name for the action, for example `Google Search`.
7. Additionally you can add a *Short Name* or even an *Icon*.

8. Choose the *Action Scopes*. See *Defining Actions* for further information. Leave the default settings for this example.
9. For the action, add the URL used for doing a Google search, up to but not including the search term: `https://www.google.com/search?q=`
10. The text in the *Action* field should now look like this:

```
https://www.google.com/search?q=
```

11. Click on the drop-down box containing the field names for the `lakes` layer. It's located just to the left of the *Insert* button.
12. From the drop-down box, select *NAMES* and click *Insert*.
13. Your action text now looks like this:

```
https://www.google.com/search?q=[%NAMES%]
```

14. To finalize and add the action, click the *OK* button.

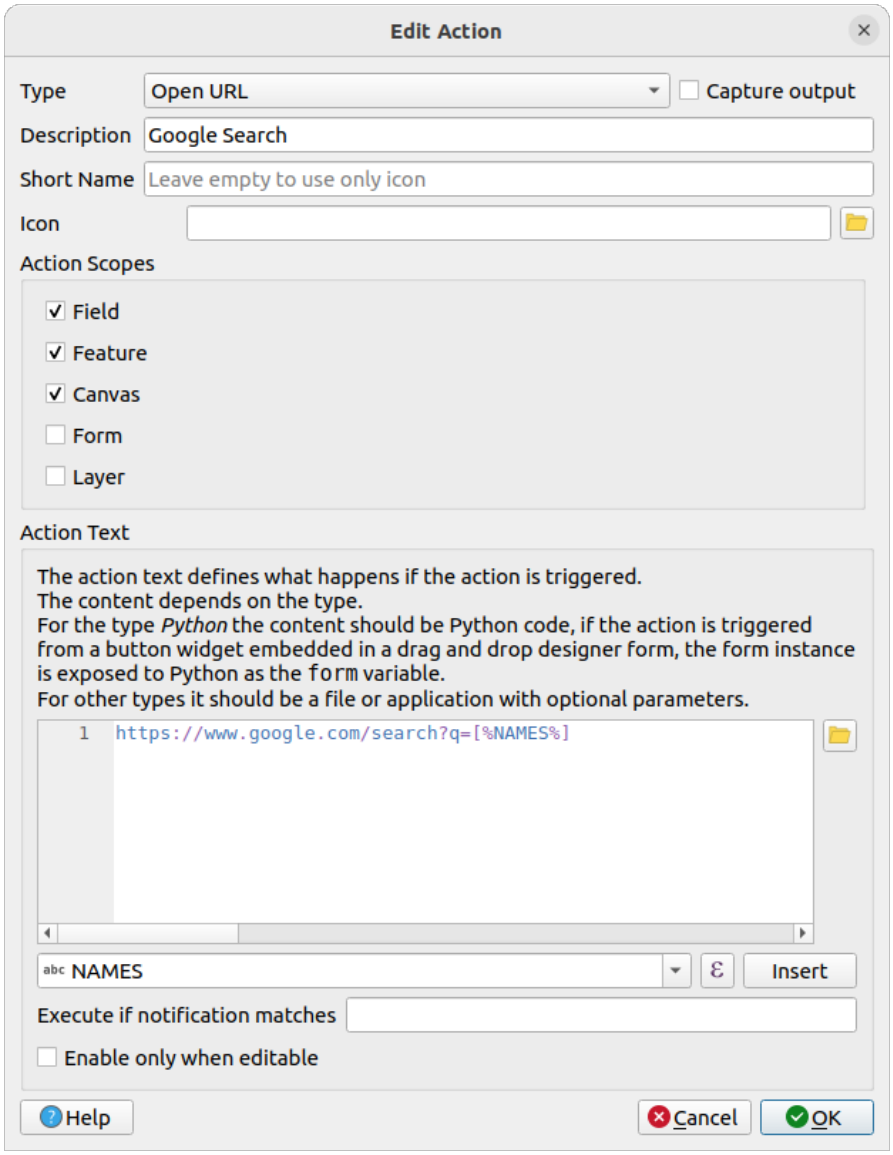


Fig. 12.59: Edit action dialog configured with the example

This completes the action, and it is ready to use.

Close the *Layer Properties* dialog and zoom in to an area of interest. Make sure the `lakes` layer is active and identify a lake. In the result box you'll now see that our action is visible:

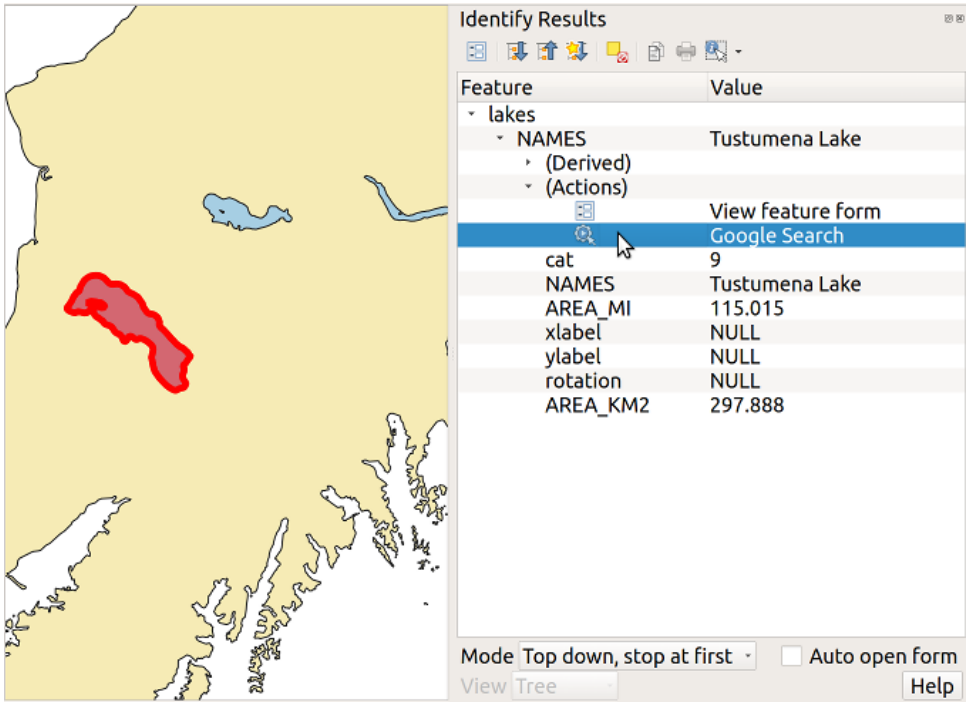


Fig. 12.60: Select feature and choose action

When we click on the action, it brings up our default browser and navigates to the URL https://www.google.com/search?q=Tustumena. It is also possible to add further attribute fields to the action. Therefore, you can add a + to the end of the action text, select another field and click on *Insert Field*. In this example, there is just no other field available that would make sense to search for.

You can define multiple actions for a layer, and each will show up in the *Identify Results* dialog.

You can also invoke actions from the attribute table by selecting a row and right-clicking, then choosing the action from the pop-up menu.

There are all kinds of uses for actions. For example, if you have a point layer containing locations of images or photos along with a file name, you could create an action to launch a viewer to display the image. You could also use actions to launch web-based reports for an attribute field or combination of fields, specifying them in the same way we did in our Google search example.

We can also make more complex examples, for instance, using **Python** actions.

Usually, when we create an action to open a file with an external application, we can use absolute paths, or eventually relative paths. In the second case, the path is relative to the location of the external program executable file. But what about if we need to use relative paths, relative to the selected layer (a file-based one, like Shapefile or SpatiaLite)? The following code will do the trick:

```
command = "firefox"
imagerelpath = "images_test/test_image.jpg"
layer = qgis.utils.iface.activeLayer()
import os.path
layerpath = layer.source() if layer.providerType() == 'ogr'
  else (qgis.core.QgsDataSourceURI(layer.source()).database()
    if layer.providerType() == 'spatialite' else None)
path = os.path.dirname(str(layerpath))
image = os.path.join(path,imagerelpath)
```

(continues on next page)

(continued from previous page)

```
import subprocess
subprocess.Popen( [command, image ] )
```

We just have to remember that the action is one of type *Python* and the *command* and *imagerelpath* variables must be changed to fit our needs.

But what about if the relative path needs to be relative to the (saved) project file? The code of the Python action would be:

```
command = "firefox"
imagerelpath = "images_test/test_image.jpg"
projectpath = qgis.core.QgsProject.instance().fileName()
import os.path
path = os.path.dirname(str(projectpath)) if projectpath != '' else None
image = os.path.join(path, imagerelpath)
import subprocess
subprocess.Popen( [command, image ] )
```

Another Python action example is the one that allows us to add new layers to the project. For instance, the following examples will add to the project respectively a vector and a raster. The names of the files to be added to the project and the names to be given to the layers are data driven (*filename* and *layername* are column names of the table of attributes of the vector where the action was created):

```
qgis.utils.iface.addVectorLayer('/yourpath/[% "filename" %].shp',
  '[% "layername" %]', 'ogr')
```

To add a raster (a TIF image in this example), it becomes:

```
qgis.utils.iface.addRasterLayer('/yourpath/[% "filename" %].tif',
  '[% "layername" %]')
```

12.1.13 Display Properties

The *Display* tab helps you configure fields to use for feature identification:

- The *Display name*: based on a field or an *expression*. By default, this is set to the first field in the layer if no field with `<name>` component exists. This is used as:
  - the label shown on top of the feature information in the *Identify tool* results
  - the field used in the *locator bar* when looking for features in all layers
  - the feature identifier in the attribute table *form view*
  - the feature identifier when the map or layout is exported to a layered output format such as Geospatial PDF
  - the map tip information, i.e. the message displayed in the map canvas when hovering over a feature of the active layer with the Show Map Tips icon pressed. Applicable when *Enable Map Tips* is active and no *HTML Map Tip* is set.
- *Enable Map Tips* controls whether to display map tips for the layer
- The *HTML Map Tip* provides a complex and full HTML text editor for map tips, mixing QGIS expressions and html styles and tags (multiline, fonts, images, hyperlink, tables, ...). You can check the result of your code sample in the *Preview* frame (also convenient for previewing the *Display name* output). Additionally, you can select and edit existing expressions using the *Insert/Edit Expression* button.

> **Note**
>
> Understanding the *Insert/Edit Expression* button behavior
>
> If you select some text within an expression (between "[%" and "%]"), or if no text is selected but the cursor is inside an expression, the whole expression will be automatically selected for editing. If the cursor or a selected text is outside an expression, the dialog opens with the selection.

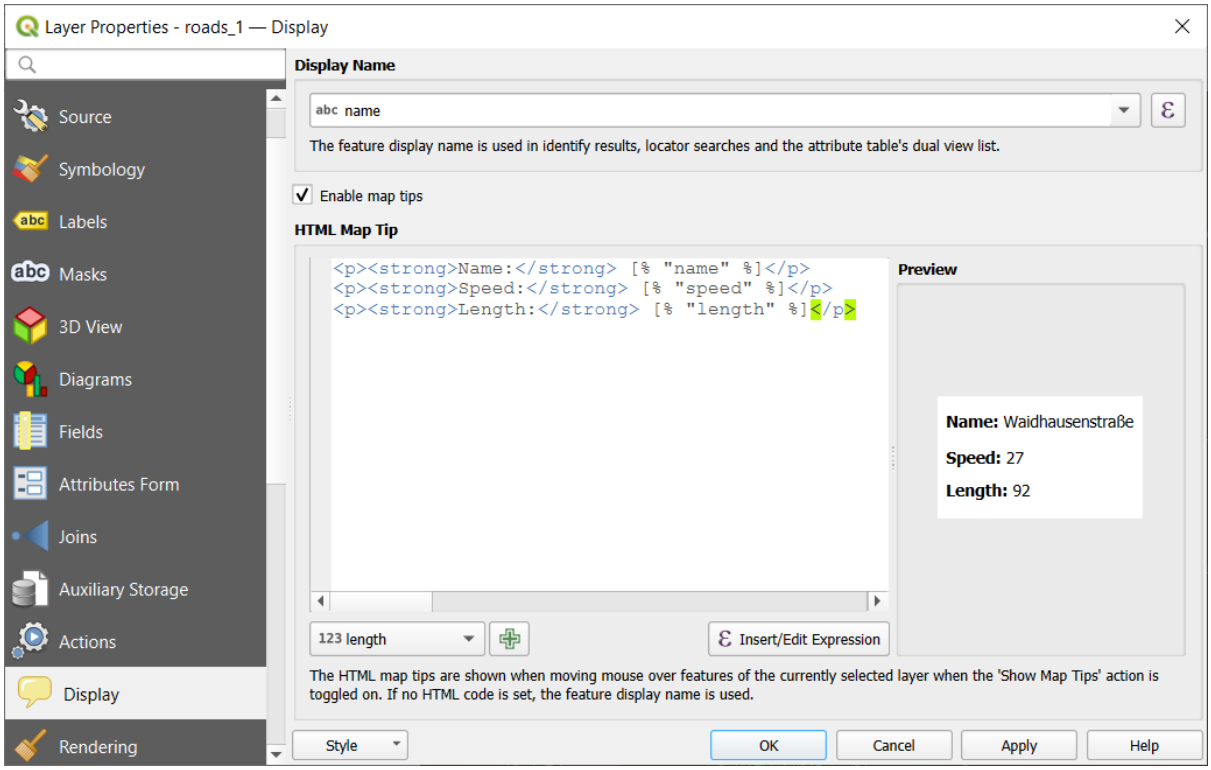


Fig. 12.61: HTML code for map tip

To display map tips:

1. Select the menu option *View* ► *Show Map Tips* or click on the Show Map Tips icon of the *Attributes Toolbar*.
2. Make sure that the layer you target is active and has the *Enable Map Tips* property checked.
3. Move over a feature, and the corresponding information will be displayed over.

Map tip is a cross-layer feature meaning that once activated, it stays on and applies to any map tip enabled layer in the project until it's toggled off.

Fig. 12.62: Map tip made with HTML code

12.1.14 Rendering Properties

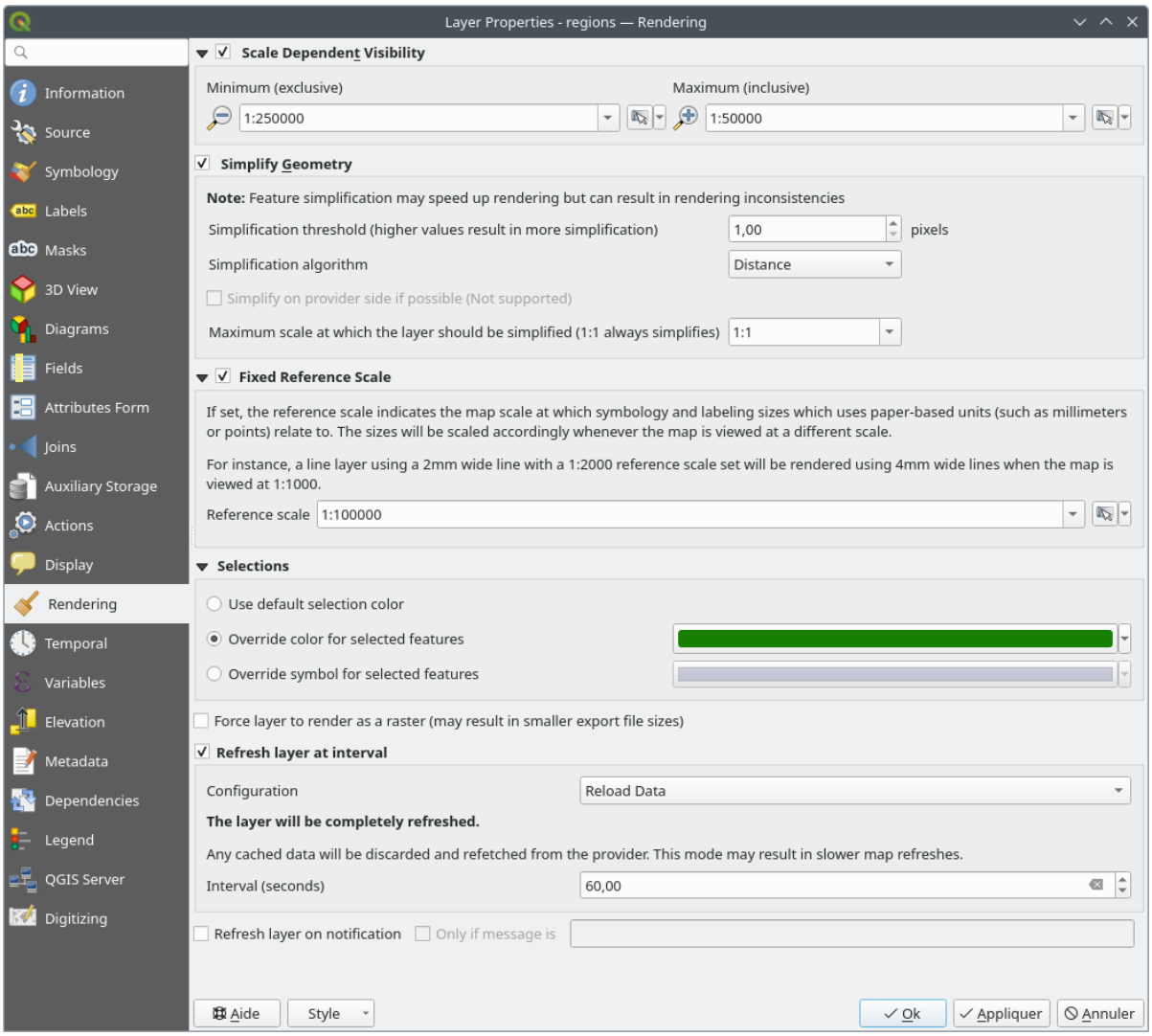


Fig. 12.63: Layer Rendering Properties dialog

The *Rendering* tab offers following properties:

- Under *Scale dependent visibility*, you can set the *Maximum (inclusive)* and *Minimum (exclusive)* scales, defining a range of scales in which features will be visible. Out of this range, they are hidden. The Set to current canvas scale button helps you use the current map canvas scale as boundary of the range visibility. See 可见比例选择器 for more information.

  > **Note**
  >
  > You can also activate scale dependent visibility on a layer from within the *Layers* panel: right-click on the layer and in the contextual menu, select *Set Layer Scale Visibility*.

- QGIS offers support for on-the-fly feature generalisation. This can improve rendering times when drawing many complex features at small scales. This feature can be enabled or disabled in the layer settings using the *Simplify geometry* option. There is also a global setting that enables generalisation by default for newly added layers (see *global simplification* for more information).

> **Note**
>
> Feature generalisation may introduce artefacts into your rendered output in some cases. These may include slivers between polygons and inaccurate rendering when using offset-based symbol layers.

- The *Fixed reference scale* indicates the map scale at which symbology and labeling sizes which uses paper-based units (such as millimeters or points) relate to. The sizes will be scaled accordingly whenever the map is viewed at a different scale.

  For instance, a line layer using a 2mm wide line with a 1:2000 *Reference scale* set will be rendered using 4mm wide lines when the map is viewed at 1:1000.

- The *Selections* group allows you to control whether a specific color or symbol should be used in place of the defaults (*Project properties* ► *General* ► *Selection color*) for a particular layer. This is useful to improve the visibility of selected features with certain symbology:

  - *Use default selection color*
  - *Override color for selected features*: e.g. if the layer uses a yellow color by default and the standard yellow selection is not visible.
  - *Override symbol for selected features*: e.g. if a line layer uses a thin symbol, and coloring the lines might not make them visible enough, overriding the symbol with a thicker line can help. Also, if the layer uses raster symbology or gradient fills/lines/shapeburst with color ramp symbology, then the default selection color is not applied at all; being able to set a specific simpler symbol to use for selected features in the layer can help.

- Rendering extremely detailed layers (e.g. polygon layers with a huge number of nodes), can cause layout exports in PDF/SVG format to be huge as all nodes are included in the exported file. This can also make the resultant file very slow to work with/open in other programs.

  Checking *Force layer to render as raster* forces these layers to be rasterised so that the exported files won't have to include all the nodes contained in these layers and the rendering is therefore sped up.

  You can also do this by forcing the layout to export as a raster, but that is an all-or-nothing solution, given that the rasterisation is applied to all layers. Alternatively, you can rely on geometry simplification in *layout export settings*.

- *Refresh layer at interval*: controls whether and how regular a layer can be refreshed. Available *Configuration* options are:

  - *Reload data*: the layer will be completely refreshed. Any cached data will be discarded and refetched from the provider. This mode may result in slower map refreshes.
  - *Redraw layer only*: this mode is useful for animation or when the layer's style will be updated at regular intervals. Canvas updates are deferred in order to avoid refreshing multiple times if more than one layer has an auto update interval set.

  It is also possible to set the *Interval (seconds)* between consecutive refreshments.

- Depending on the data provider (e.g. PostgreSQL), notifications can be sent to QGIS when changes are applied to the data source, out of QGIS. Use the *Refresh layer on notification* option to trigger an update. You can also limit the layer refresh to a specific message set in the *Only if message is* text box.

12.1.15 Temporal Properties

The *Temporal* tab provides options to control the rendering of the layer over time. Such dynamic rendering requires the *temporal navigation* to be enabled over the map canvas.

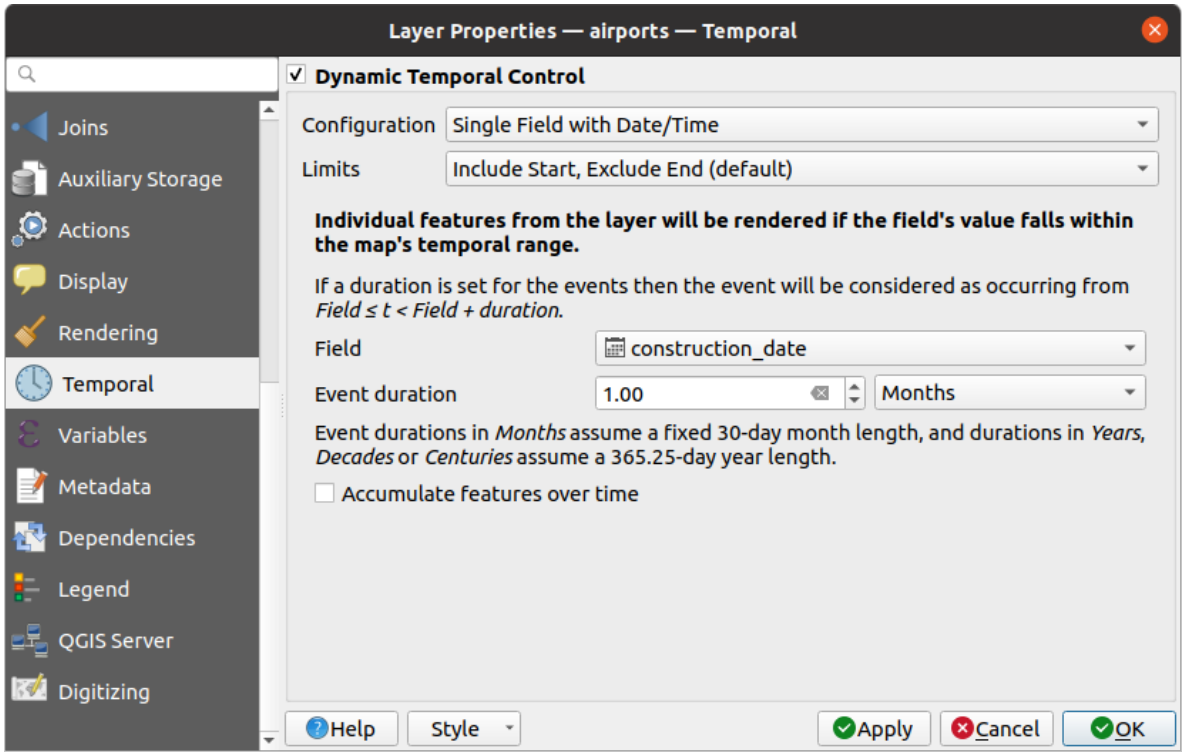


Fig. 12.64: Vector layer temporal properties dialog

Check the *Dynamic Temporal Control* option to configure the vector layer temporal rendering. Depending on the structure of your dataset, you may want to use one of the provided *Configuration* options:

- *Fixed time range*: all the features are rendered if the map canvas temporal frame overlaps the given *Start date* and *End date* range.
- *Single field with date/time*: features are rendered if their *Field*'s value falls within the map canvas temporal frame. An *Event duration* can be set. With checking the *Accumulate features over time* option, all features which occur before or within the map's temporal range will continue to be rendered. The event duration is thus ignored.
- *Separate fields for start and end date/time*: features are rendered if the range specified by their *Start field* and *End field* values overlaps the map canvas temporal.
- *Separate fields for start and event duration*: features are rendered if the range defined by their *Start field* and *Event duration field* values overlaps the map canvas temporal.
- *Start and end date/time from expressions*: features are rendered if the time range specified by the fields *Start expression* and *End expression* overlaps the map canvas temporal.
- *Redraw layer only*: the layer is redrawn at each new animation frame but no time-based filtering is applied to the features. It's useful when the layer uses time-based expression values for renderer settings (e.g. data-defined symbology).

It is also possible to set the *Limits* of the features time range as:

- *Include start, exclude end*
- *Include start, include end*

12.1.16 Variables Properties

The *Variables* tab lists all the variables available at the layer's level (which includes all global and project's variables).

It also allows the user to manage layer-level variables. Click the button to add a new custom layer-level variable. Likewise, select a custom layer-level variable from the list and click the button to remove it.

More information on variables usage in the General Tools 将值存储在变量中 section.

12.1.17 Elevation Properties

The *Elevation* tab provides options to control the layer elevation properties within a *3D map view* and its appearance in the *profile tool charts*. Specifically, you can set:

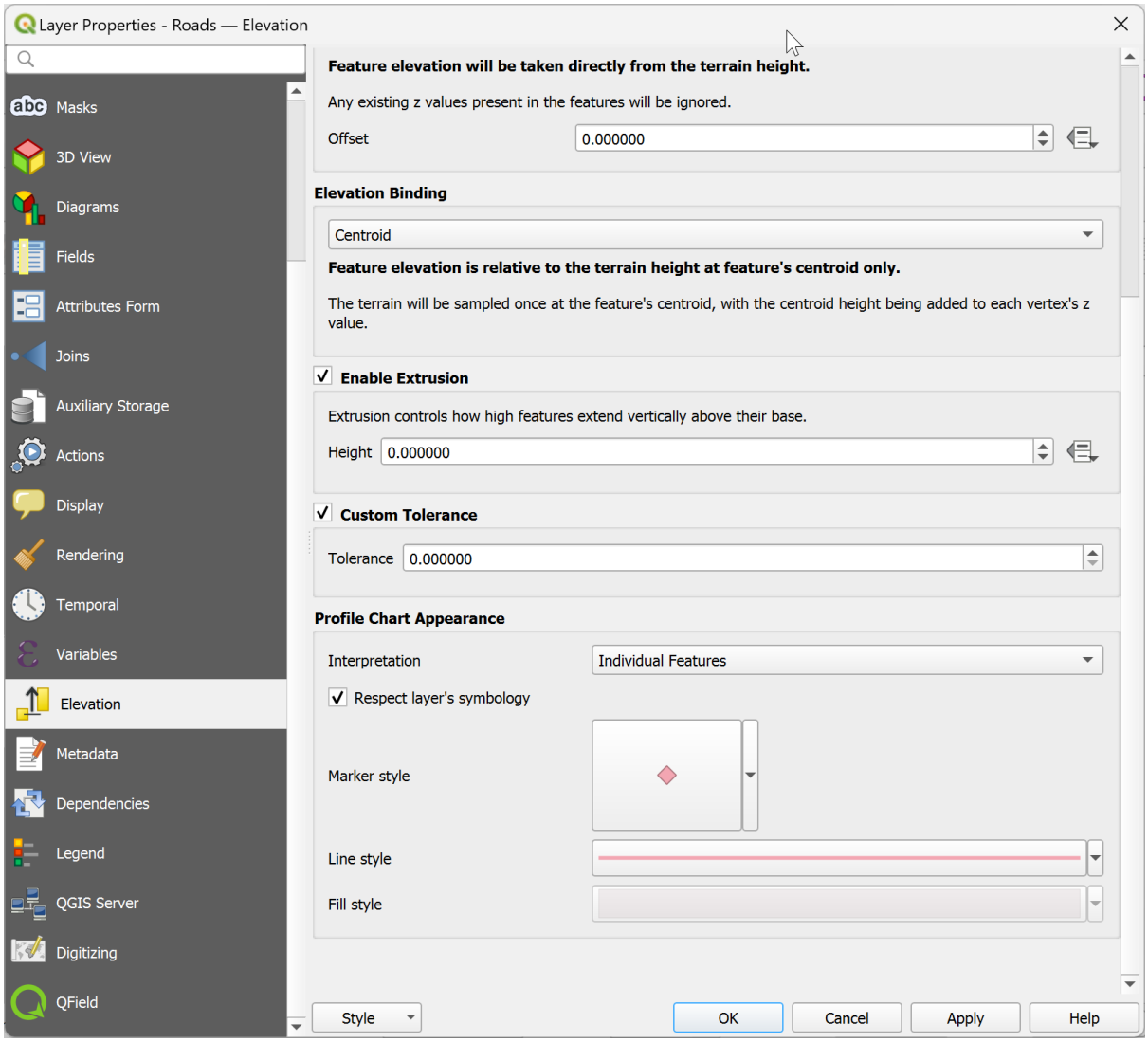


Fig. 12.65: Vector layer elevation properties dialog

- *Vertical Reference System*: If the CRS of your vector layer is a compound one (including a Z dimension), then the vertical CRS used for the layer will be the vertical component of the layer CRS. In this case, you cannot

manually set a different vertical CRS. If your layer CRS is horizontal (2D), then you can select a specific vertical CRS by clicking on the Select CRS. Vertical reference systems are supported for vector layers by:

  - *Elevation profiles*
  - *Identify Tool Results*
  - *3D map views*

- *Elevation Clamping*: defines how and whether the features altitude should be:
  - *Clamped to terrain*: takes elevation directly from the terrain height and ignores any existing Z values in the features. A data-defined *Offset* value from the terrain can also be filled.
  - *Relative to terrain*: any existing Z values in the features are added to the terrain height. A *Scale* factor followed by a data-defined *Offset* can be used to adjust the elevation. This option is not available for 2D geometry layers.
  - *Absolute*: ignores the terrain height and directly takes Z values from the features for the elevation. A *Scale* factor followed by a data-defined *Offset* can be used to adjust the elevation. For 2D geometry layers (with no Z values), a data-defined *Base height* can instead be set.
- *Enable extrusion*: you can set a *Height* to control how high features vertically extend above their base. This is convenient to indicate that a 2D geometry layers, e.g. a polygon building footprints layer, actually represents 3D objects.
- *Custom tolerance*: allows you to override the global sampling tolerance defined in the elevation profile settings, specifically for this vector layer. When enabled, you can set *Tolerance* that determines how the layer is sampled along the profile line. This is useful when you need finer or coarser sampling than the global default, for example, to improve performance or accuracy for complex geometries.
- *Elevation Binding*: only relevant when combining an *Elevation clamping* relying on the terrain with a line or polygon layer, this option controls how feature elevation is set relative to the terrain height. The terrain can be sampled:
  - at the feature's *Centroid*, with the centroid height being added to each vertex's z value
  - at every individual *Vertex* before being added to the vertex's z value
- *Profile Chart Appearance*: controls how features are rendered when drawing a profile chart. Two main *Interpretation* modes are available:
  - as *Individual features*: samples discrete positions where the cross section profile line intersects the vector features. That intersection can be represented as point, line or surface depending on the layer type and whether an extrusion is applied.

    With checking *Respect layer symbology*, features will be rendered on the profile chart with their corresponding *layer styling* (allowing e.g. categorized classes to be visible on the profile chart). If the profile symbol type does not match the layer's renderer symbol types, only the symbol color from the renderer is applied to the profile symbol.

    Depending on the layer settings, profile symbols can be represented with a custom style, using:

    * *Marker style*: for non-extruded point and line features, and for non-extruded polygon features touched by the profile line
    * *Line style*: for extruded point and line features, and for non-extruded polygon features intersected by the profile line
    * *Fill style*: for extruded polygon features
  - as *Continuous Surface (e.g. contours)*: the elevation chart will be rendered as a surface instead of separate features by joining the sampled elevation results into a continuous line. This can enhance visualisation and is designed for vector layers which represent a continuous elevation surface, e.g. contour lines or surveyed elevation points. The profile *Style* can be set as:
    * a *Line* with a specific *Line style*

    * an elevation surface rendered using a fill symbol either above (*Fill above*) or below (*Fill below*) the elevation curve line. The surface symbology is represented using:
        - a *Fill style*
        - and a *Limit*: the maximum (respectively minimum) altitude determining how high the fill surface will be

    Moreover, you can check *Show markers at sampled points* to make them visible over the interpretation line and assign them a *Marker style*.

12.1.18 元数据属性

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

12.1.19 Dependencies Properties

The *Dependencies* tab allows to declare data dependencies between layers. A data dependency occurs when a data modification in a layer, not by direct user manipulation, may modify data of other layers. This is the case for instance when geometry of a layer is updated by a database trigger or custom PyQGIS scripting after modification of another layer's geometry.

In the *Dependencies* tab, you can select any layers which may externally alter the data in the current layer. Correctly specifying dependent layers allows QGIS to invalidate caches for this layer when the dependent layers are altered.

12.1.20 Legend Properties

The *Legend* properties tab provides you with advanced settings for the *Layers panel* and/or the *print layout legend*. These options include:

- Depending on the symbology applied to the layer, you may end up with several entries in the legend, not necessarily readable/useful to display. The *Legend placeholder image* helps you *select an image* for replacement, displayed both in the *Layers* panel and the print layout legend.
- *Show label legend*: Displays overviews of the different label settings as entries in the legends. The *label style* is previewed along with the description.
- *Text on symbols*: In some cases it can be useful to add extra information to the symbols in the legend. With this frame, you can affect to any of the symbols used in the layer symbology a text that is displayed over the symbol, in both *Layers* panel and print layout legend. This mapping is done by typing each text next to the symbol in the table widget or filling the table using the *Set Labels from Expression* button. Text appearance is handled through the font and color selector widgets of the *Text Format* button.

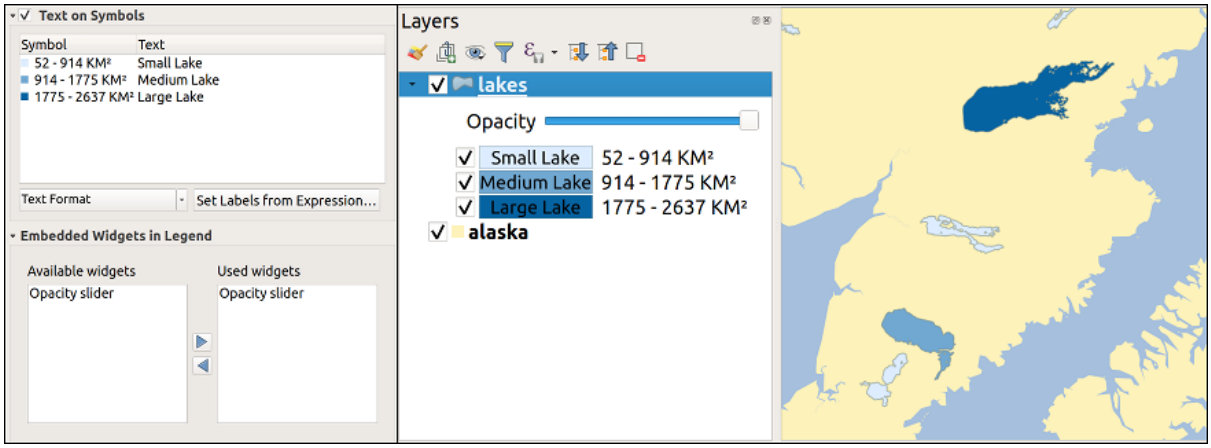


Fig. 12.66: Setting text on symbols (left) and its rendering in the *Layers* panel (right)

- a list of widgets you can embed within the layer tree in the Layers panel. The idea is to have a way to quickly access some actions that are often used with the layer (setup transparency, filtering, selection, style or other stuff...).

  By default, QGIS provides transparency widget but this can be extended by plugins registering their own widgets and assign custom actions to layers they manage.

12.1.21 QGIS Server Properties

The *QGIS Server* tab consists of *Description*, *Attribution*, *Metadata URL*, and *Legend URL* sections.

From the *Description* section, you can change the *Short name* used to reference the layer in requests (to learn more about short names, read services_basics_short_name). You can also add or edit a *Title*, an alternative *WFS Title* and *Abstract* for the layer, or define a *Keyword list* here. These keyword lists can be used in a metadata catalog. If you want to use a title from an XML metadata file, you have to fill in a link in the *Data URL* field.

Use *Attribution* to get attribute data from an XML metadata catalog.

In *Metadata URL*, you can add the general paths to the XML metadata catalog. This information will be saved in the QGIS project file for subsequent sessions and will be used for QGIS Server.

In the *Legend URL* section, you can provide the url of a legend image in the url field. You can use the Format drop-down option to apply the appropriate format of the image. Currently png, jpg and jpeg image formats are supported.

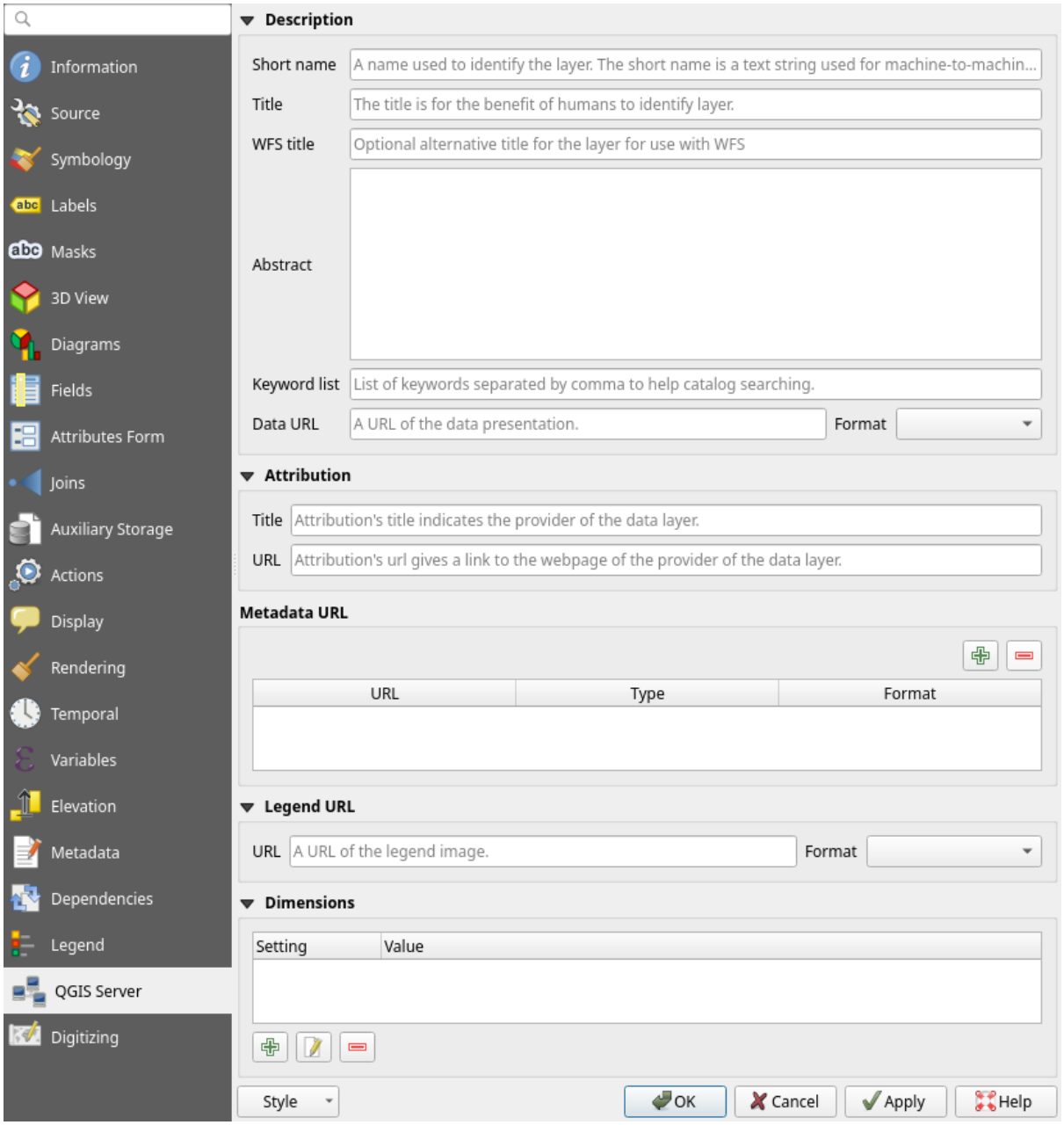


Fig. 12.67: QGIS Server tab in vector layers properties dialog

要了解有关 QGIS 服务器的更多信息，请阅读 QGIS-Server-manual 。

12.1.22 Digitizing Properties

The *Digitizing* tab gives access to options that help to ensure the quality of digitized geometries.

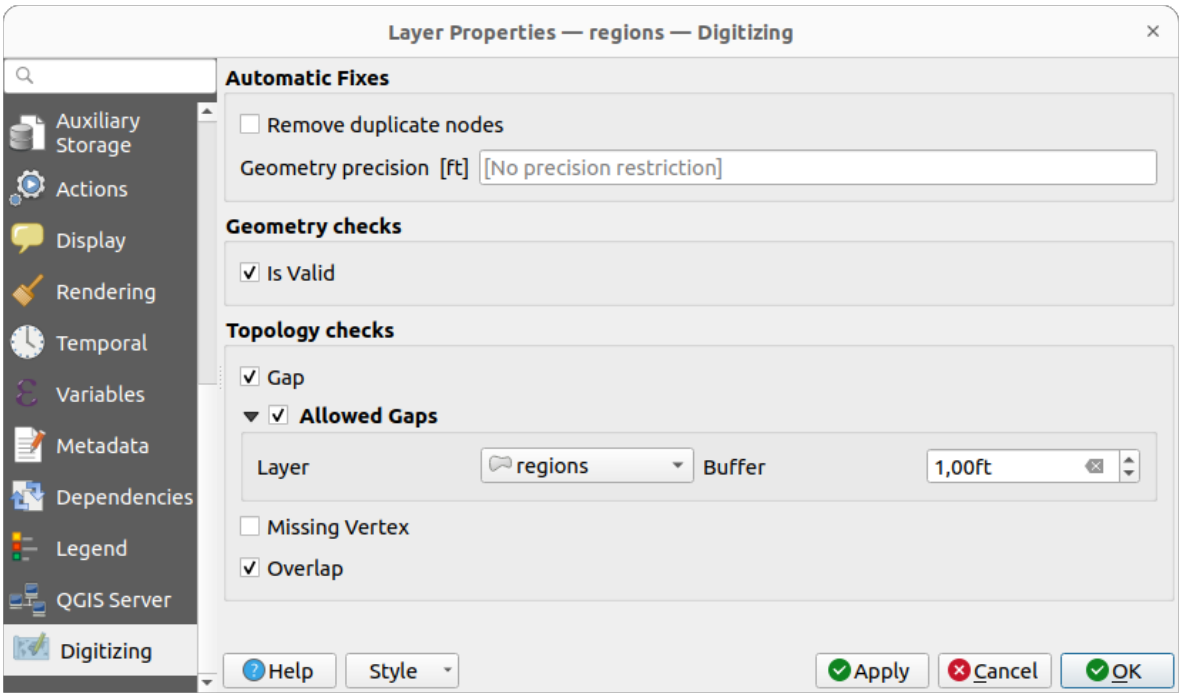


Fig. 12.68: The QGIS Digitizing tab in the vector layers properties dialog

Automatic Fixes

Options in the *Automatic Fixes* section will directly affect the vertices of any geometry which is added or modified. If the *Remove duplicate nodes* option is checked, any two subsequent vertices with exactly the same coordinates will be removed. If the *Geometry precision* is set, all vertices will be rounded to the closest multiple of the configured geometry precision. The rounding will happen in the layer coordinate reference system. Z and M values are not rounded. With many map tools, a grid is shown on the canvas while digitizing.

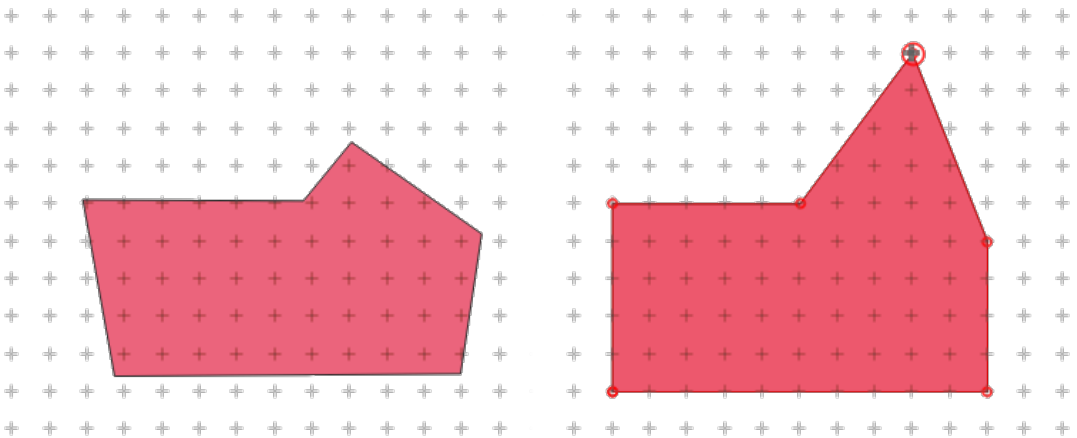


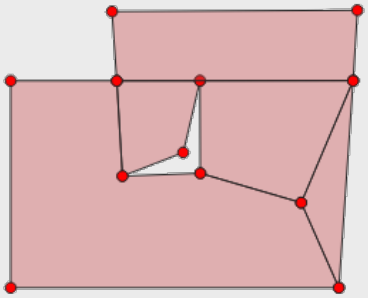
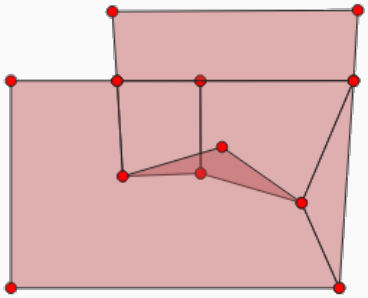
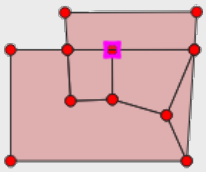
Fig. 12.69: Moving the top vertex snaps all the vertices to the grid

Geometry Checks

In the *Geometry checks* section, additional validations on a per geometry basis can be activated. Immediately after any geometry modification, failures in these checks are reported to the user in the *Geometry validation* panel. As long as a check is failing, it is not possible to save the layer. The *Is valid* check will run basic validity checks like self intersection on geometries.

Topology Checks

In the *Topology checks* section, additional topology validation checks can be activated. Topology checks will be executed when the user saves the layer. Check errors will be reported in the *Geometry validation* panel. As long as validation errors are present, the layer can not be saved. Topology checks are executed in the area of the bounding box of the modified features. Since other features may be present in the same area, topological errors concerning these features are reported as well as errors introduced in the current edit session.

Topology check option	Illustration
The *Gap* check will check for gaps between neighbouring polygons.	
The *Overlap* check will check for overlaps between neighbouring polygons.	
The *Missing vertex* check will check for shared boundaries of neighbouring polygons where one border misses a vertex which is present on the other one.	

Gap check exceptions

Sometimes it is desirable to keep gaps inside an area in a polygon layer that otherwise is fully covered by polygons. For example, a land use layer may have acceptable holes for lakes. It is possible to define areas that are ignored in the gap check. Since gaps inside these areas are allowed, we will refer to them as *Allowed Gaps* areas.

In the options for the gap checks under *Allowed Gaps*, an *Allowed Gaps layer* can be configured.

Whenever the gap check is executed, gaps which are covered by one or more polygons in the *Allowed Gaps Layer* are not reported as topology errors.

It is also possible to configure an additional *Buffer*. This buffer is applied to each polygon on the *Allowed Gaps Layer*. This makes it possible to make the tests less susceptible to small changes in the outlines at the borders of gaps.

When *Allowed Gaps* are enabled, an additional button (*Add Allowed Gap*) for detected gap errors is available in the geometry validation dock, where gaps are reported during digitizing. If the *Add Allowed Gap* button is pushed, a new polygon with the geometry of the detected gap is inserted into the *Allowed Gaps Layer*. This makes it possible to quickly flag gaps as allowed.

Geometry validation panel

The *Geometry Validation* panel is triggered when any of the abovementioned digitizing checks finds an error. The dialog provides you with the list of errors and their description, and you can to browse the list using the keyboard arrows or dedicated arrows.

You'll need to address all the issues before you can save edits to the layer. To do so:

1. Select an error, and it's possible to:
   - *Zoom to Feature(s)*
   - *Zoom to problem*
2. Pick the usual *digitizing tools* to fix the issue.

12.2 属性表操作

属性表展示的是所选图层中各个要素的属性信息。表中的每一行代表一个（几何或非几何）要素，每一列代表要素的某种属性。表中要素可以被选中，移动和编辑。

12.2.1 前言：空间与非空间属性表

QGIS allows you to load spatial and non-spatial layers. This currently includes tables supported by GDAL and delimited text, as well as the PostgreSQL, MS SQL Server, SpatiaLite and Oracle providers. All loaded layers are listed in the *Layers* panel. Whether a layer is spatially enabled or not determines whether you can interact with it on the map.

非空间表格可以使用属性表查看器进行浏览与编辑。不仅如此，它们还可以用于属性查找。例如，你可以在创建矢量图层的时候定义非空间表格的某一列为属性值本身，或者用来限定其取值范围。详见编辑部件的文档：ref:*vector_attributes_menu*

12.2.2 属性表界面介绍

点击对应图层的:ref:*label_legend*'按钮激活图层，选择:*menuselection:'Layer'*菜单下的 |*openTable*| :*menuselection:'Open Attribute Table'*打开图层属性表。你也可以右键单击图层，在下拉列表中选择 |*openTable*| :*menuselection:'Open Attribute Table*，或者在属性工具栏中点击 |openTable| *Open Attribute Table*'按钮。如果你想使用快捷键，按:*kbd:'F6'*打开属性表，:*kbd:'Shift+F6'*打开所选要素的属性表，:*kbd:'Ctrl+F6* 打开可见要素的属性表。

This will open a new window that displays the feature attributes for the layer (*figure_attributes_table*). According to the setting in *Settings* ► *Options* ► *Data sources* menu, the attribute table will open in a docked window or a regular window. The total number of features in the layer and the number of currently selected/filtered features are shown in the attribute table title, as well as if the layer is spatially limited.

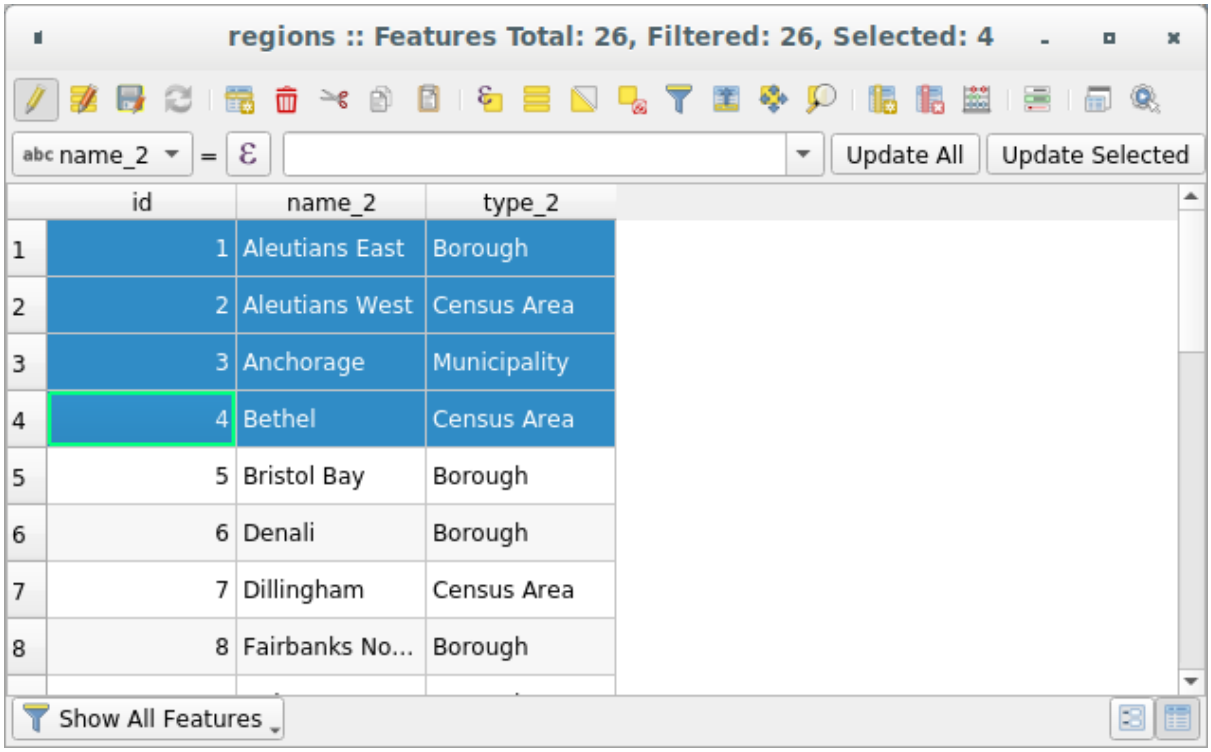


Fig. 12.70: Attribute Table for regions layer

The buttons at the top of the attribute table window provide the following functionality:

Table 12.1: 可用工具

Icon	标注	目标	Default Shortcut
	Toggle editing mode	Enable editing functionalities	`Ctrl+E`
	Toggle multi edit mode	Update multiple fields of many features	
	Save Edits	Save current modifications	
	Reload the table		
	Add feature	Add new geometryless feature	
	Delete selected features	Remove selected features from the layer	
	Cut selected features to clipboard		`Ctrl+X`
	Copy selected features to clipboard		`Ctrl+C`

continues on next page

Table 12.1 – continued from previous page

Icon	标注	目标	Default Shortcut
	Paste features from clipboard	Insert new features from copied ones	`Ctrl+V`
	Select features using an Expression		
	Select All	Select all features in the layer	`Ctrl+A`
	Invert selection	Invert the current selection in the layer	`Ctrl+R`
	Deselect all	Deselect all features in the current layer	`Ctrl+Shift+A`
	Filter/Select features using form		`Ctrl+F`
	Move selected to top	Move selected rows to the top of the table	
	Pan map to the selected rows		`Ctrl+P`
	Zoom map to the selected rows		`Ctrl+J`
	New field	Add a new field to the data source	`Ctrl+W`
	Delete field	Remove a field from the data source	
	Organize columns	Show/hide fields from the attribute table	
	Open field calculator	Update field for many features in a row	`Ctrl+I`
	Conditional formatting	Enable table formatting	
	Dock attribute table	Allows to dock/undock the attribute table	
	Actions	Lists the actions related to the layer	

Attention

Depending on the format of the data and the GDAL library built with your QGIS version, some tools may not be available.

Below these buttons is the Quick Field Calculation bar (enabled only in *edit mode*), which allows to quickly apply calculations to all or part of the features in the layer. This bar uses the same *expressions* as the Field Calculator (see *Editing attribute values*).

Table view vs Form view

QGIS provides two view modes to easily manipulate data in the attribute table:

- The Table view, displays values of multiple features in a tabular mode, each row representing a feature and each column a field. A right-click on the column header allows you to *configure the table display* while a right-click on a cell provides *interaction with the feature*.

  The attribute table supports `Shift+Mouse Wheel` scrolling in table view mode to switch between vertical and horizontal scrolling movements. This can also be achieved replacing the mouse with the trackpad on macOS.

- The Form view shows *feature identifiers* in a first panel and displays only the attributes of the clicked identifier in the second one. There is a pull-down menu at the top of the first panel where the ”identifier” can be specified using an attribute (*Column preview*) or an *Expression*. The pull-down also includes the last 10 expressions for re-use. Form view uses the layer fields configuration (see *Attributes Form Properties*).

  You can browse through the feature identifiers with the arrows on the bottom of the first panel. The features attributes update in the second panel as you go. It’s also possible to identify or move to the active feature in

the map canvas with pushing down any of the button at the bottom:

- Highlight current feature if visible in the map canvas
- Automatically pan to current feature
- Zoom to current feature

You can switch from one mode to the other by clicking the corresponding icon at the bottom right of the dialog.

You can also specify the *Default view* mode at the opening of the attribute table in *Settings* ► *Options* ► *Data Sources* menu. It can be 'Remember last view', 'Table view' or 'Form view'.

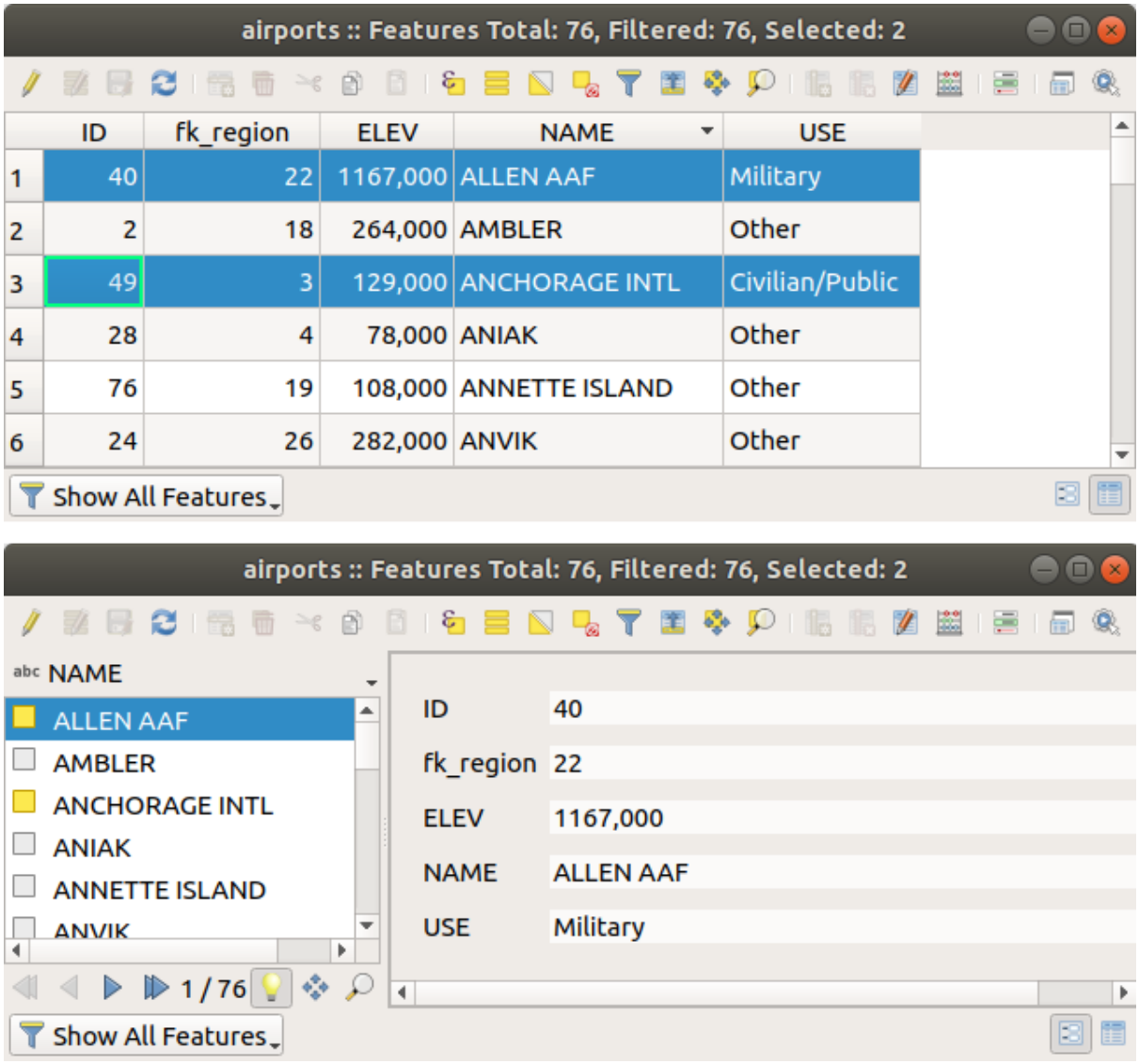


Fig. 12.71: Attribute table in table view (top) vs form view (bottom)

Configuring the columns

Right-click in a column header when in table view to have access to tools that help you control:

- the *column(s) size*
- the *column(s) visibility and order*
- the *sort order of the data*

Resizing columns widths

Columns width can be set through a right-click on the column header and select either:

- *Set width...* to enter the desired value. By default, the current value is displayed in the widget
- *Set all column widths...* to the same value
- *Autosize* to resize at the best fit the column.
- *Autosize all columns*

A column size can also be changed by dragging the boundary on the right of its heading. The new size of the column is maintained for the layer, and restored at the next opening of the attribute table.

In the *Data Source Settings*, you can choose to *Autosize all columns by default when opening attribute table*, which will make "Autosize All Columns" the default view every time attribute tables are opened in QGIS.

Hiding and organizing columns and enabling actions

By right-clicking in a column header, you can choose to *Hide column* from the attribute table (in "table view" mode).

For more advanced controls, press the Organize columns... button from the dialog toolbar or choose *Organize columns...* in a column header contextual menu.

In the new dialog, you can:

- check/uncheck columns you want to show or hide: a hidden column will disappear from every instance of the attribute table dialog until it is actively restored. It is also possible to:
  - choose *Show All* to display all the fields (columns) and actions in the table
  - choose *Hide All* to hide all the fields (columns) and actions in the table
  - use the *Toggle selection* to invert visibility of the current selection of columns. You can use *keyboard combination* for selecting multiple columns.
- drag-and-drop items to reorder the columns in the attribute table. Note that this change is for the table rendering and does not alter the fields order in the layer datasource
- add a new virtual *Actions* column that displays in each row a drop-down box or a button list of enabled actions. See *Actions Properties* for more information about actions.

Sorting rows

The rows can be sorted by any column, by clicking on the column header. A small arrow indicates the sort order (downward pointing means descending values from the top row down, upward pointing means ascending values from the top row down). You can also choose to sort the rows with the *Sort...* option of the column header context menu and write an expression. E.g. to sort the rows using multiple columns you can write `concat(col0, col1)`.

In form view, features identifier can be sorted using the *Sort by preview expression* option.

Note that sorting the rows only affects the table rendering and does not alter the features order in the layer datasource.

> **Tip**
>
> **Sorting based on columns of different types**
>
> Trying to sort an attribute table based on columns of string and numeric types may lead to unexpected result because of the `concat("USE", "ID")` expression returning string values (ie, `'Borough105' < 'Borough6'`). You can workaround this by using eg `concat("USE", lpad("ID", 3, 0))` which returns `'Borough105' > 'Borough006'`.

Formatting of table cells using conditions

Conditional formatting settings can be used to highlight in the attribute table features you may want to put a particular focus on, using custom conditions on feature's:

- geometry (e.g., identifying multi-parts features, small area ones or in a defined map extent...);
- or field value (e.g., comparing values to a threshold, identifying empty cells, duplicates, ...).

You can enable the conditional formatting panel clicking on Conditional formatting button at the top right of the attributes window in table view (not triggered in form view).

The new panel allows user to add new rules to format rendering of *Field* or *Full row*. Adding new rule opens a form to define:

- the name of the rule;
- a condition using any of the *expression builder* functions;
- the formatting: it can be choosen from a list of predefined formats or created based on properties like:
  - background and text colors;
  - use of icon;
  - bold, italic, underline, or strikeout;
  - font.

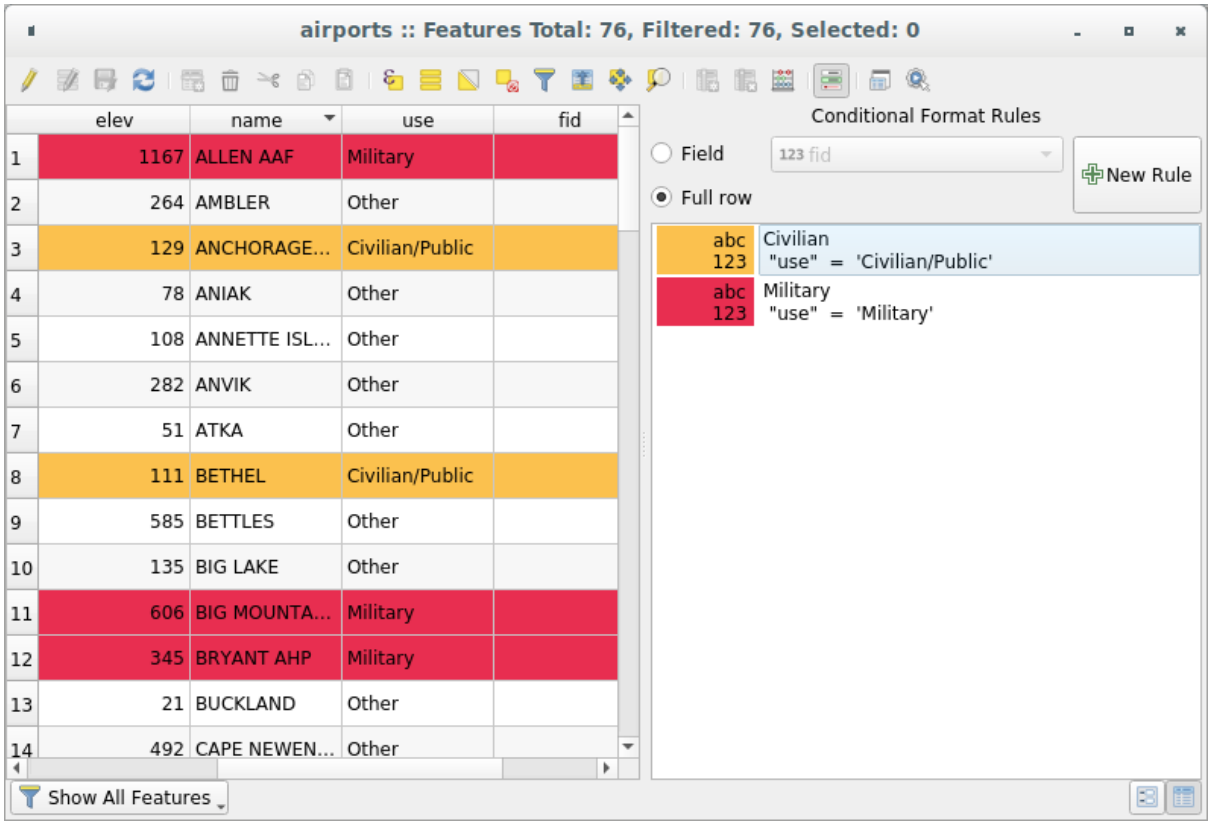


Fig. 12.72: Conditional Formatting of an attribute table

12.2.3 Interacting with features in an attribute table

选择要素

In table view, each row in the attribute table displays the attributes of a unique feature in the layer. Selecting a row selects the feature and likewise, selecting a feature in the map canvas (in case of geometry enabled layer) selects the row in the attribute table. If the set of features selected in the map canvas (or attribute table) is changed, then the selection is also updated in the attribute table (or map canvas) accordingly.

Rows can be selected by clicking on the row number on the left side of the row. **Multiple rows** can be marked by holding the `Ctrl` key. A **continuous selection** can be made by holding the `Shift` key and clicking on several row headers on the left side of the rows. All rows between the current cursor position and the clicked row are selected. Moving the cursor position in the attribute table, by clicking a cell in the table, does not change the row selection. Changing the selection in the main canvas does not move the cursor position in the attribute table.

In form view of the attribute table, features are by default identified in the left panel by the value of their displayed field (see *Display Properties*). This identifier can be replaced using the drop-down list at the top of the panel, either by selecting an existing field or using a custom expression. You can also choose to sort the list of features from the drop-down menu.

Click a value in the left panel to display the feature's attributes in the right one. To select a feature, you need to click inside the square symbol at the left of the identifier. By default, the symbol turns into yellow. Like in the table view, you can perform multiple feature selection using the keyboard combinations previously exposed.

Beyond selecting features with the mouse, you can perform automatic selection based on feature's attribute using tools available in the attribute table toolbar, such as (see section 自动选择 and subsequent for more information and use case):

- *Select By Expression...*

- *Select Features By Value...*
- *Deselect All Features from the Layer*
- *Select All Features*
- *Invert Feature Selection*.

It is also possible to *select features using forms*.

Filtering features

Once you have selected features in the attribute table, you may want to display only these records in the table. This can be easily done using the *Show Selected Features* item from the drop-down list at the bottom left of the attribute table dialog. This list offers the following filters:

- *Show All Features*
- *Show Selected Features* - same as using *Open Attribute Table (Selected Features)* from the *Layer* menu or the *Attributes Toolbar* or pressing `Shift+F6`
- *Show Features visible on map* - same as using *Open Attribute Table (Visible Features)* from the *Layer* menu or the *Attributes Toolbar* or pressing `Ctrl+F6`
- *Show Features with Failing Constraints* - features will be filtered to only show the ones which have failing *constraints*. Depending on whether the unmet constraint is hard or soft, failing field values are displayed in respectively dark or light orange cells.
- *Show Edited and New Features* - same as using *Open Attribute Table (Edited and New Features)* from the *Layer* menu or the *Attributes Toolbar*
- *Field Filter* - allows the user to filter based on value of a field: choose a column from a list, type or select a value and press `Enter` to filter. Then, only the features matching `num_field = value` or `string_field ilike '%value%'` expression are shown in the attribute table. You can check *Case sensitive* to be less permissive with strings.
- *Advanced filter (Expression)* - Opens the expression builder dialog. Within it, you can create *complex expressions* to match table rows. For example, you can filter the table using more than one field. When applied, the filter expression will show up at the bottom of the form.
- *Stored filter expressions* ►: a shortcut to *saved expressions* frequently used for filtering your attribute table.

It is also possible to *filter features using forms*.

> **Note**
>
> Filtering records out of the attribute table does not filter features out of the layer; they are simply momentaneously hidden from the table and can be accessed from the map canvas or by removing the filter. For filters that do hide features from the layer, use the *Query Builder*.

> **Tip**
>
> **Update datasource filtering with** `Show Features Visible on Map`

When for performance reasons, features shown in attribute table are spatially limited to the canvas extent at its opening (see *Data Source Options* for a how-to), selecting *Show Features Visible on Map* on a new canvas extent updates the spatial restriction.

Storing filter expressions

Expressions you use for attribute table filtering can be saved for further calls. When using *Field Filter* or *Advanced Filter (expression)* entries, the expression used is displayed in a text widget in the bottom of the attribute table dialog. Press the Save expression with text as name next to the box to save the expression in the project. Pressing the drop-down menu next to the button allows to save the expression with a custom name (*Save expression as...*). Once a saved expression is displayed, the button is triggered and its drop-down menu allows you to *Edit the expression* and name if any, or *Delete stored expression*.

Saved filter expressions are saved in the project and available through the *Stored filter expressions* menu of the attribute table. They are different from the *user expressions*, shared by all projects of the active user profile.

Filtering and selecting features using forms

Clicking the Filter/Select features using form or pressing `Ctrl+F` will make the attribute table dialog switch to form view and replace each widget with its search variant.

From this point onwards, this tool functionality is similar to the one described in 按值选择要素, where you can find descriptions of all operators and selecting modes.

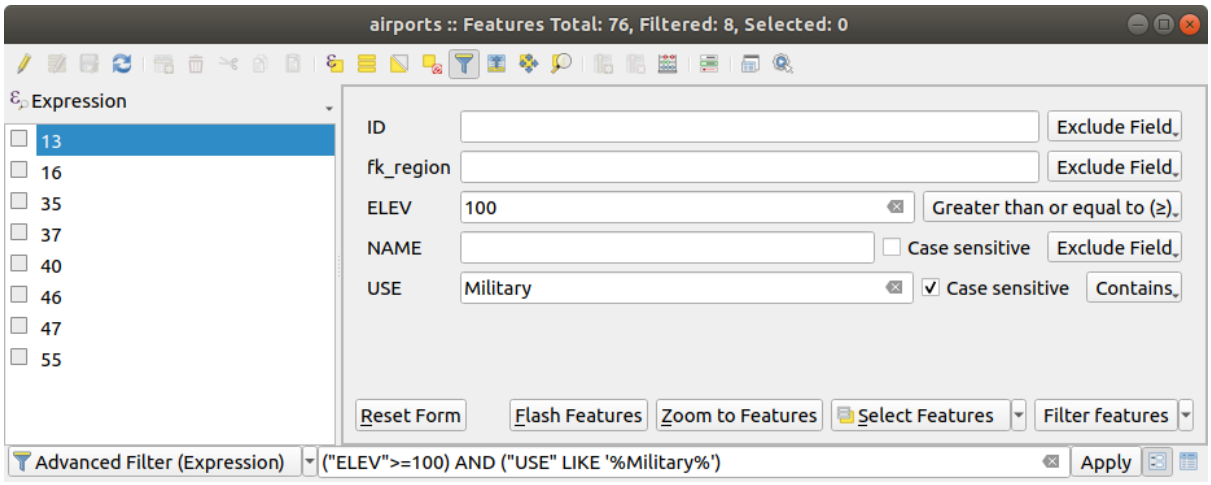


Fig. 12.73: Attribute table filtered by the filter form

When selecting / filtering features from the attribute table, there is a *Filter features* button that allows defining and refining filters. Its use triggers the *Advanced filter (Expression)* option and displays the corresponding filter expression in an editable text widget at the bottom of the form.

If there are already filtered features, you can refine the filter using the drop-down list next to the *Filter features* button. The options are:

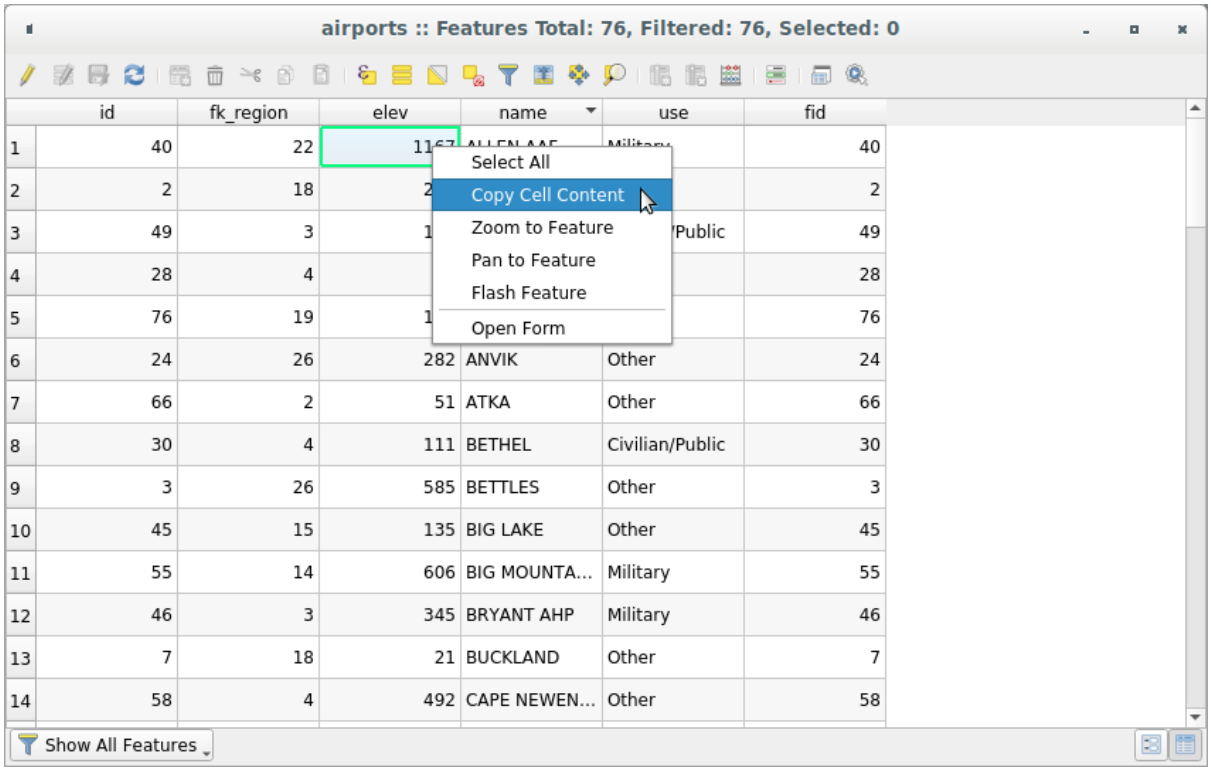
- *Filter within ("AND")*
- *Extend filter ("OR")*

To clear the filter, either select the *Show all features* option from the bottom left pull-down menu, or clear the expression and click *Apply* or press `Enter`.

More actions on features

Users have several possibilities to manipulate feature in an attribute table. Right-click in a cell and you can:

- *Select all* (`Ctrl+A`) the features;
- Copy the content of a cell in the clipboard with *Copy cell content*;
- *Zoom to feature* without having to select it beforehand;
- *Pan to feature* without having to select it beforehand;
- *Flash feature*, to highlight it in the map canvas;
- *Open form*: it toggles attribute table into form view with a focus on the clicked feature.
- Display a *list of actions*, previously enabled in the *Layer properties* ► *Actions* tab.



	id	fk_region	elev	name	use	fid
1	40	22	1167	ALLEN AAF	Military	40
2	2	18	2			2
3	49	3	1		Public	49
4	28	4				28
5	76	19	1			76
6	24	26	282	ANVIK	Other	24
7	66	2	51	ATKA	Other	66
8	30	4	111	BETHEL	Civilian/Public	30
9	3	26	585	BETTLES	Other	3
10	45	15	135	BIG LAKE	Other	45
11	55	14	606	BIG MOUNTA...	Military	55
12	46	3	345	BRYANT AHP	Military	46
13	7	18	21	BUCKLAND	Other	7
14	58	4	492	CAPE NEWEN...	Other	58

Fig. 12.74: Copy cell content button

If you want to use attribute data in external programs (such as Excel, LibreOffice, or a custom web application), select one or more row(s) and use the Copy selected rows to clipboard button or press `Ctrl+C`. Moreover, in *Settings* ► *Options* ► *Data Sources* menu you can define the format to paste to with the *Copy features as* option. More details at *Data Sources settings*.

12.2.4 Editing attribute values

In order to modify data in an attribute table, you should first toggle the layer into edit. Press the Toggle Editing button. Depending on the layer geometry type and the clipboard state, a few more tools are enabled in the attribute table top toolbar.

Editing attribute values can then be done by:

- typing the new value directly in the cell, whether the attribute table is in table or form view. Changes are hence done cell by cell, feature by feature;
- using the *field calculator*: update in a row a field that may already exist or to be created but for multiple features. It can be used to create virtual fields;
- using the quick field *calculation bar*: same as above but for only existing field;
- or using the *multi edit* mode: update in a row multiple fields for multiple features.

Putting the layer into edit mode will also allow you to Paste features from clipboard (`Ctrl+V`) Cut selected rows to clipboard (`Ctrl+X`) or Delete selected features. More details at *Cutting, Copying and Pasting Features*.

Using the Field Calculator

The Field Calculator button in the attribute table allows you to perform calculations on the basis of existing attribute values or defined functions, for instance, to calculate length or area of geometry features. The results can be used to update an existing field, or written to a new field (that can be a *virtual* one).

The field calculator is available on any layer that supports edit. When you click on the field calculator icon the dialog opens (see Fig. 12.75). If the layer is not in edit mode, a warning is displayed and using the field calculator will cause the layer to be put in edit mode before the calculation is made.

Based on the *Expression Builder* dialog, the field calculator dialog offers a complete interface to define an expression and apply it to an existing or a newly created field. To use the field calculator dialog, you must select whether you want to:

1. apply calculation on the whole layer or on selected features only
2. create a new field for the calculation or update an existing one.

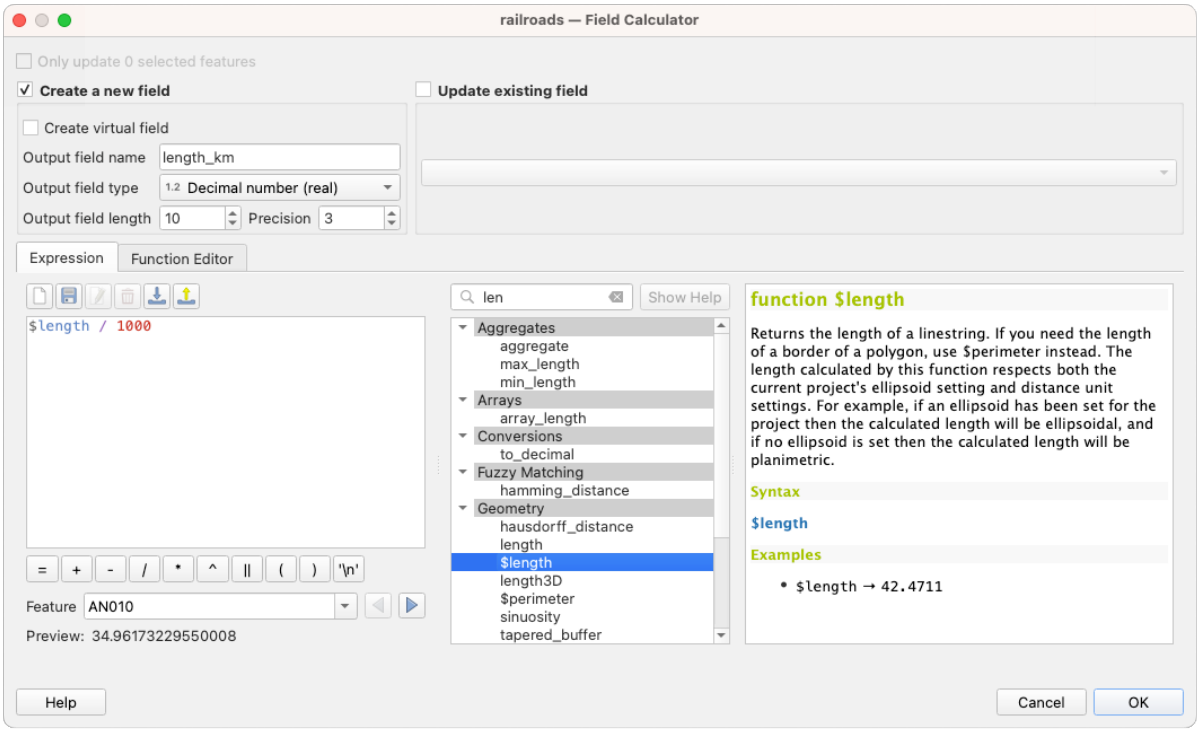


Fig. 12.75: Field Calculator

If you choose to add a new field, you need to enter a field name, a field type (integer, real, date or string) and if needed, the total field length and the field precision. For example, if you choose a field length of 10 and a field precision of 3, it means you have 7 digits before the dot, and 3 digits for the decimal part.

A short example illustrates how field calculator works when using the *Expression* tab. We want to calculate the length in km of the `railroads` layer from the QGIS sample dataset:

1. Load the shapefile `railroads.shp` in QGIS and press Open Attribute Table.
2. Click on Toggle editing mode and open the Field Calculator dialog.
3. Select the *Create a new field* checkbox to save the calculations into a new field.
4. Set *Output field name* to `length_km`
5. Select `Decimal number (real)` as *Output field type*
6. Set the *Output field length* to `10` and the *Precision* to `3`
7. Double click on `$length` in the *Geometry* group to add the length of the geometry into the Field calculator expression box (you will begin to see a preview of the output, up to 60 characters, below the expression box updating in real-time as the expression is assembled).
8. Complete the expression by typing `/ 1000` in the Field calculator expression box and click *OK*.
9. You can now find a new *length_km* field in the attribute table.

Creating a Virtual Field

A virtual field is a field based on an expression calculated on the fly, meaning that its value is automatically updated as soon as an underlying parameter changes. The expression applies to all the features in the layer and is set once; you no longer need to recalculate the field each time underlying values change. For example, you may want to use a virtual field if you need area to be evaluated as you digitize features or to automatically calculate a duration between dates that may change (e.g., using `now()` function).

Creating a virtual field is done through the *Field calculator* dialog and follows the *same procedure* as for regular fields. Simply remember to check the *Create virtual field* option and use a field type compatible with the data your expression would generate.

Editing a virtual field is done through the *Fields* tab of the layer properties dialog (see *Fields Properties*). The expression defining the field is exposed in the *Comment* column, and pressing the button next to it opens an expression editor window for update.

> **Note**
>
> **Use of Virtual Fields**
>
> - A field can be set virtual only at its creation.
> - Virtual fields are not permanent in the layer attributes, meaning that they're only saved and available in the project file they've been created.

Using the Quick Field Calculation Bar

While Field calculator is always available, the quick field calculation bar on top of the attribute table is only visible if the layer is in edit mode. Thanks to the expression engine, it offers a quicker access to edit an already existing field:

1. Select the field to update in the drop-down list.
2. Fill the textbox with a value, an expression you directly write or build using the expression button.
3. Click on *Update All*, *Update Selected* or *Update Filtered* button according to your need.

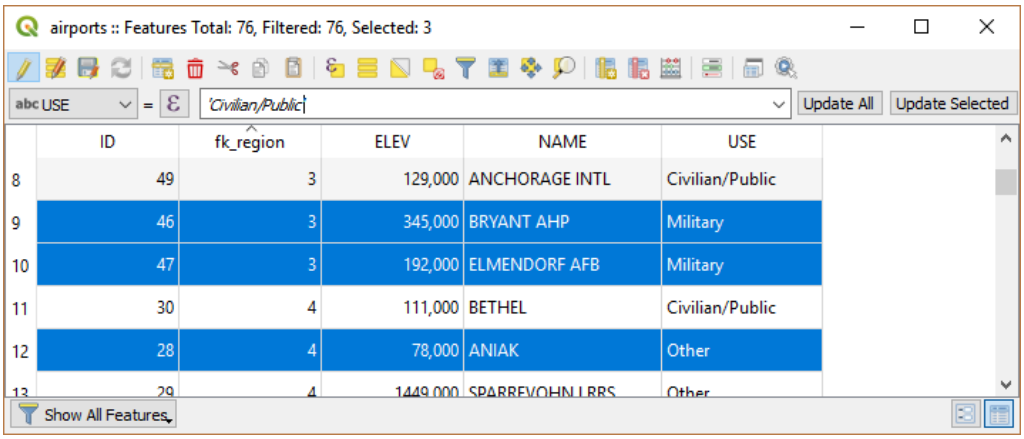


Fig. 12.76: Quick Field Calculation Bar

Editing multiple fields

Unlike the previous tools, multi edit mode allows multiple attributes of different features to be edited simultaneously. When the layer is toggled to edit, multi edit capabilities are accessible:

- using the Toggle multi edit mode button from the toolbar inside the attribute table dialog;
- or selecting *Edit* ► *Modify attributes of selected features* menu.

> **Note**
>
> Unlike the tool from the attribute table, hitting the *Edit* ► *Modify Attributes of Selected Features* option provides you with a modal dialog to fill attributes changes. Hence, features selection is required before execution.

In order to edit multiple fields in a row:

1. Select the features you want to edit.
2. From the attribute table toolbar, click the button. This will toggle the dialog to its form view. Feature selection could also be made at this step.
3. At the right side of the attribute table, fields (and values) of selected features are shown. New widgets appear next to each field allowing for display of the current multi edit state:
   - The field contains different values for selected features. It's shown empty and each feature will keep its original value. You can reset the value of the field from the drop-down list of the widget.
   - All selected features have the same value for this field and the value displayed in the form will be kept.
   - The field has been edited and the entered value will be applied to all the selected features. A message appears at the top of the dialog, inviting you to either apply or reset your modification.

   Clicking any of these widgets allows you to either set the current value for the field or reset to original value, meaning that you can roll back changes on a field-by-field basis.

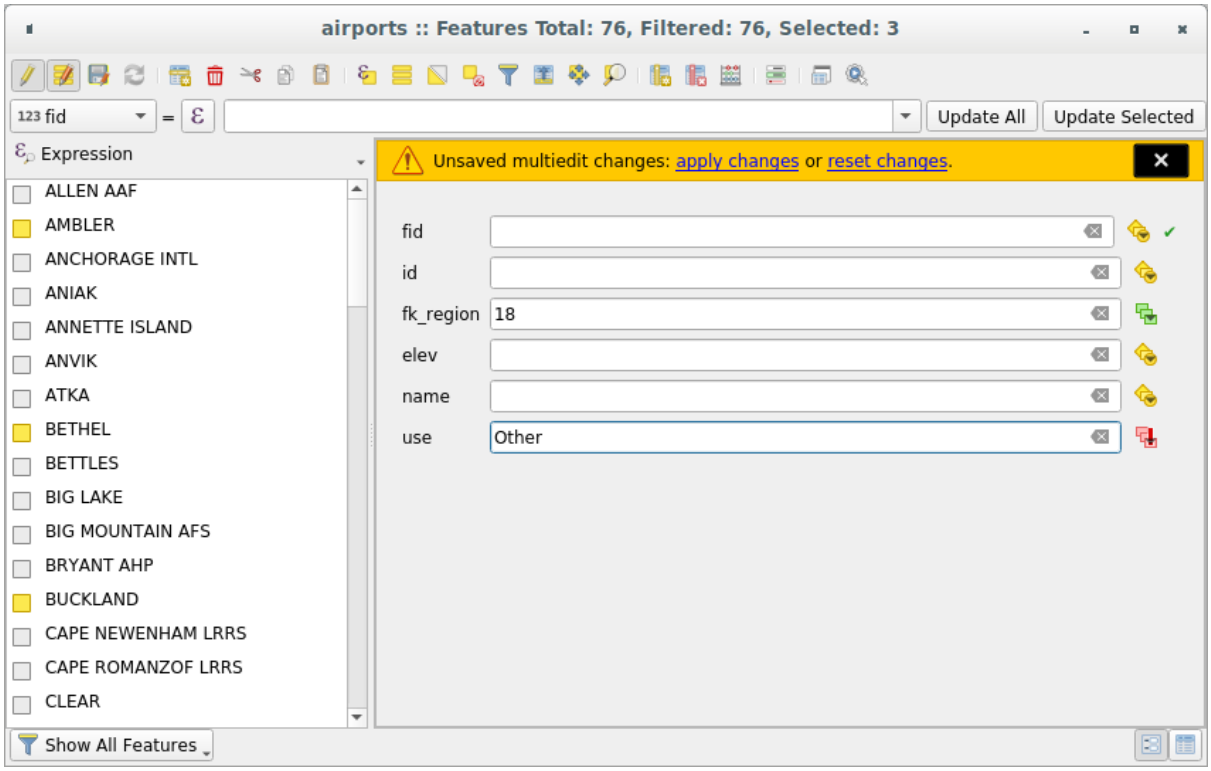


Fig. 12.77: Editing fields of multiple features

4. Make the changes to the fields you want.
5. Click on **Apply changes** in the upper message text or any other feature in the left panel.

Changes will apply to **all selected features**. If no feature is selected, the whole table is updated with your changes. Modifications are made as a single edit command. So pressing Undo will rollback the attribute changes for all selected features at once.

> **Note**
>
> Multi edit mode is only available for auto generated and drag and drop forms (see *Customizing a form for your data*); it is not supported by custom ui forms.

12.2.5 Exploring features attributes through the Identify Tool

The *Identify features* tool can be used to display all attributes of a feature in the map canvas. It is a quick way to view and verify all data without having to search for it in the attribute table.

To use the *Identify features* tool for vector layers, follow these steps:

1. Select the vector layer in the Layers panel.
2. Click on the *Identify features* tool in the toolbar or press `Ctrl+Shift+I`.
3. Click on a feature in the map view.

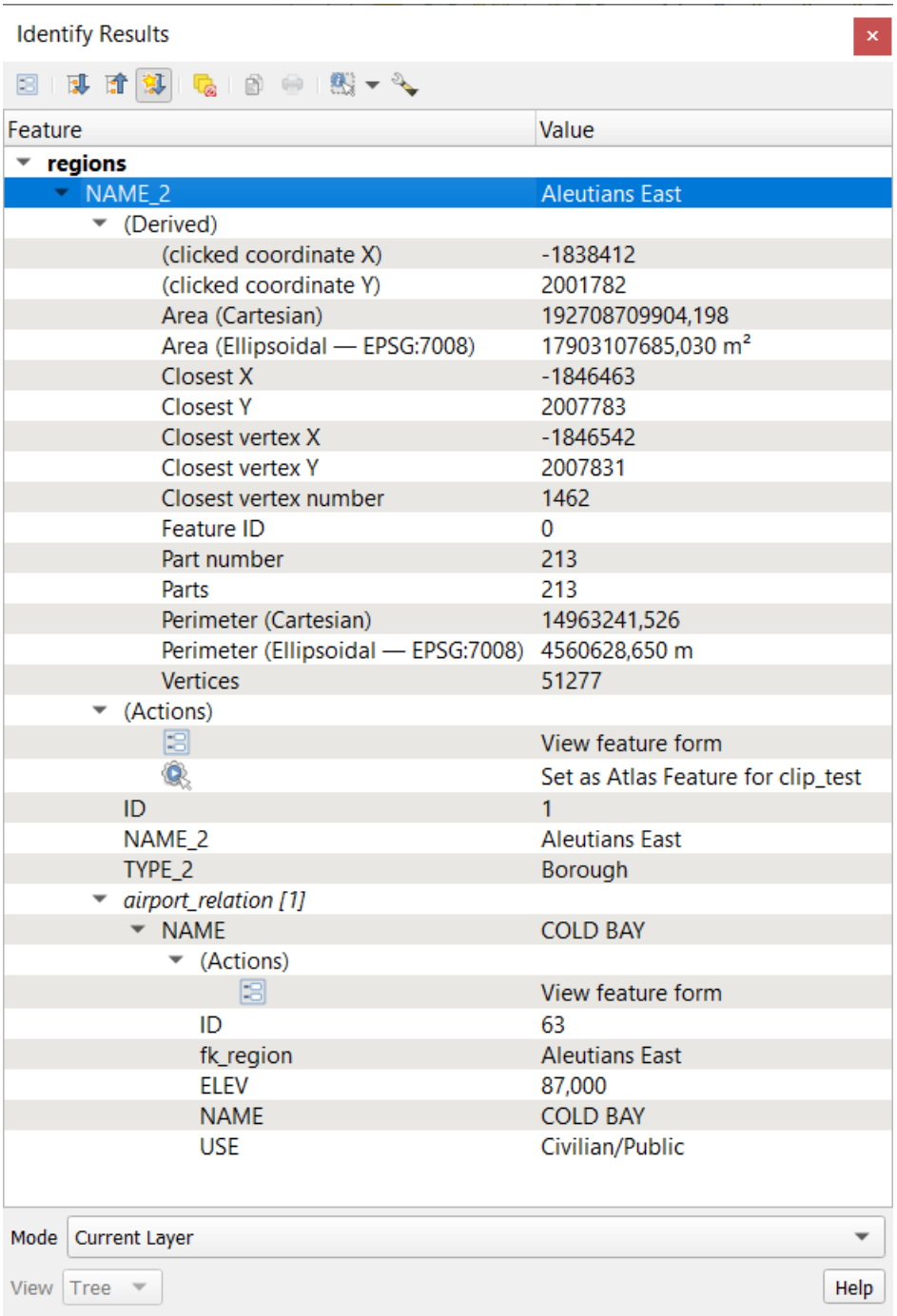


Fig. 12.78: Identify Results dialog

The *Identify results* panel will display different features information in a tree view. There are two columns in the panel: *Feature* and *Value*. The first item is the name of the layer and its children are its identified feature(s). Each feature is identified by the *display name* field along with its value. Other information about the feature follows:

- **Derived** section - those are the information calculated or derived from other information in the layer. For example, the area of a polygon or the length of a line. General information that can be found in this section:
  - Depending on the geometry type, the cartesian measurements of length, perimeter or area in the layer's CRS units. For 3D line vectors, the cartesian line length is available.
  - Depending on the geometry type and if an ellipsoid is set in the *Project Properties* dialog (*General* ► *Measurements*), ellipsoidal values of length, perimeter, or area using the specified units.

  - The count of geometry parts in the feature and the number of the part clicked.
  - The count of vertices in the feature.
  - Coordinate information, using the project properties *Coordinates display* settings:
    * `X` and `Y` coordinate values of the clicked point
    * the number of the closest vertex to the clicked point
    * `X` and `Y` coordinate values of the closest vertex (and `Z`/`M` if applicable)
    * if you click on a curved segment, the radius of that section is also displayed.
    * if both the vector layer and the project have vertical datums set and they differ, the `Z` value will be displayed for both datums.

- **Actions**: Actions can be added to the identify feature windows. The action is run by clicking on the action label. By default, only one action is added, namely `View feature form` for editing. You can define more actions in the layer's properties dialog (see *Actions Properties*).
- **数据属性**: 这是已单击要素的属性字段和值的列表。

  Note

  Links in the feature's attributes are clickable from the *Identify Results* panel and will open in your default web browser.

- When a vector layer has defined *relations*, the *Identify Results* panel can display both **referenced** and **referencing** related features. To view these relations, ensure that the *Show relations* option is enabled in the *Identify Settings*. Available information for each related feature include:
  - the name of the relation and the linked layer
  - the entry in the referenced or referencing field, identifying the feature
  - the activated actions
  - the list of attribute fields and values

  You can expand related features to explore connected records through multiple levels, including many-to-many (n:m) and polymorphic relations. Only nodes you explicitly expand are loaded, preventing excessive nesting.

  To focus on a specific related record, right-click it and choose *Identify Feature*. This re-centers the identify results on the selected feature, starting a new identification tree from it and effectively limiting the visible nesting depth. Features already shown in ancestor nodes are automatically omitted to avoid duplicates or circular relations.

12.3 编辑

QGIS has various capabilities for editing OGR, SpatiaLite, PostgreSQL, MS SQL Server and Oracle Spatial vector layers and tables. They can be of 2D or 3D geometry type.

Note

编辑 GRASS 图层具有不同的步骤 - 详细信息请参阅*Digitizing and editing a GRASS vector layer* 。

Attention

并发编辑

QGIS 不会跟踪其他人是否与您同时编辑同一要素，最终结果取决于最后保存编辑的人。

Tip

验证编辑

可以在 图层属性 ► 数字化选项卡中基于图层激活连续验证。详细信息请参阅*Digitizing Properties* 。

12.3.1 设置捕捉容差和搜索半径

在 设置 ► 选项... ► 数字化菜单中，QGIS 提供了许多参数来配置编辑工具的默认操作。更多信息请参阅数字化设置 。

为了优化和准确编辑矢量图层几何图形，需要为要素顶点设置合适的捕捉容差和搜索半径的值。捕捉组提供了相关选项，即捕捉容差和搜索半径的处理。

- 捕捉容差：当添加新顶点或移动现有顶点时，捕捉容差是 QGIS 用于搜索您试图连接到最近顶点或线段的距离。如果不在捕捉容差范围内，QGIS 将把顶点放置在您松开鼠标按钮的位置，而不是将其捕捉到现有顶点或线段。

  容差设置将影响所有使用捕捉的工具，并且默认会应用于新建图层和工程。但是，它可以在图层级别被覆写 (请参阅捕捉和数字化选项) 。

- 搜索半径: 顶点编辑的搜索半径是 QGIS 在单击地图时用于 搜索要选择的顶点的距离。如果不在搜索半径内，QGIS 将无法找到并选择任何顶点进行编辑。

捕捉容差和搜索半径是以 地图单位或 像素进行设置的。您可能需要进行尝试才能最优设置。如果指定的容差太大，尤其是在处理大量非常邻近的顶点时，QGIS 可能会捕捉到错误的顶点。搜索半径越小，就越难找到您想要移动的对象。

12.3.2 捕捉和数字化选项

全局捕捉和数字化设置 (捕捉模式、容差值和单位...) 优先于在工程中从 工程 ► 捕捉选项... 菜单的设置。在 捕捉和数字化选项中，还可以配置一些其他属性 (捕捉图层、比例限制、拓扑⋯)。捕捉工具栏可以访问其中大部分功能。

默认情况下，在点击 启用捕捉 按钮或按 S 键之前，捕捉在工程中处于禁用状态。捕捉模式、容差值和单位也可以在此工具栏中进行配置。

捕捉属性

有三个选项可用于选择要捕捉到的图层：

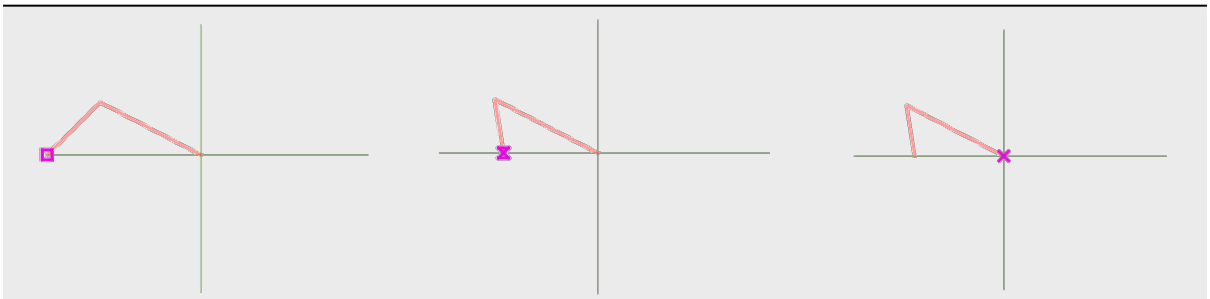
- 所有图层: 工程中所有可见图层的快速设置，以便指针捕捉到所有顶点和/或线段。在多数情况下，使用这种捕捉模式就足够了，但在将其用于具有许多矢量图层的工程时要小心，因为这可能会影响性能。
- 活动图层: 只作用于活动图层，这是确保正在编辑的图层内拓扑一致性的一种便捷方式。
- 高级配置: 允许您在图层的基础上启用和调整捕捉模式、容差和单位、重叠和捕捉比例 (请参阅 Fig. 12.79) 。如果需要编辑一个图层并将其顶点捕捉到另一个图层，请确保选中目标图层，并将捕捉容差增大到更高值。捕捉不会作用在捕捉选项对话框中未勾选的图层。

When moving or creating vertex, you can opt for the following snapping modes:

- *Vertex*
- *Segment*: snaps along a line or a polygon perimeter. If topological editing is enabled, then a new vertex is added at the snapping location.
- *Area*: guarantees that the snap point lies anywhere on a polygon's area, not necessarily on its boundary
- *Centroid*: snaps to the centroid of the geometry of a feature. In case of a multipart geometry, the target point may be distinct from the existing parts.
- *Middle of Segments* on line or polygon feature
- *Line Endpoints*: snaps to the first or last vertex of every part of a line or polygon feature.

QGIS 将根据 捕捉类型显示不同的 捕捉图标：

Table 12.2: 捕捉图标



捕捉到顶点：方框图标	捕捉到线段：沙漏图标	捕捉到相交点：十字图标

请注意，可以在全局设置的 数字化选项卡中更改这些图标的颜色。

容差值可以使用工程的 地图单位或 像素。选择 像素的优点是在不同的地图比例下保持捕捉不变。10 到 12 像素通常是一个不错的值，但这取决于屏幕的 DPI。使用地图单位可以使容差与实际范围的距离相关。例如，如果元素之间的距离较近，此选项可用于确保添加的顶点不会彼此太近。

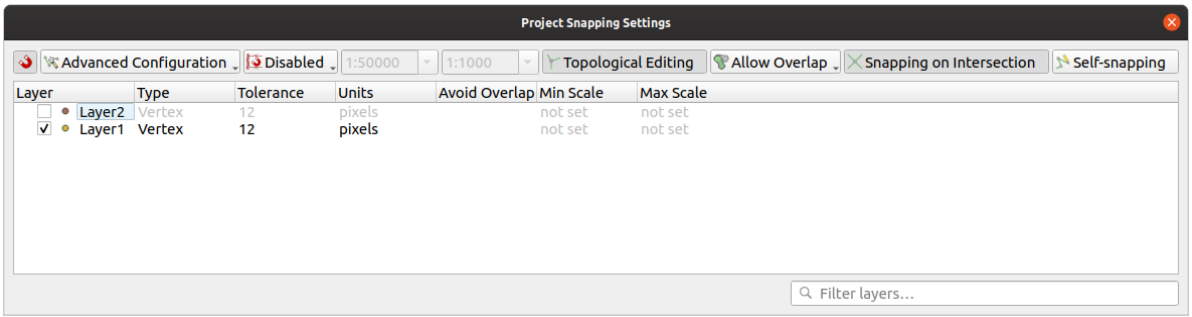


Fig. 12.79: 捕捉选项 (高级配置模式)

Note

默认仅捕捉可见要素 (显示样式的要素，符号化为" 无符号" 的图层除外)。可以通过勾选 设置 ► 选项 ► 数字化选项卡下的 对不可见的要素 (未在地图画布中显示) 也启用捕捉来启用不可见要素的捕捉。

> **Tip**
>
> **默认启用捕捉**
>
> 您可以在 设置 ► 选项 ► 数字化选项卡中将捕捉设置为默认在所有新建工程中启用。还可以设置默认捕捉模式、容差值和单位，这将填充 捕捉选项对话框。

启用相交处捕捉

另一个可用的选项是使用 相交处捕捉，这使您可以捕捉到启用捕捉的图层的几何图形相交点，即使相交处没有顶点。

将捕捉限制在比例范围内

在某些情况下，捕捉可能会变得非常迟钝。这通常是由某些图层中的要素数量引起的，这些要素需要大量索引来计算和维护。某些参数仅在地图视图处于相关比例范围内才启用捕捉。这允许仅执行与绘制比例捕捉相关的价值高的索引计算。

捕捉的比例限制在 工程 ► 捕捉选项... 中配置。将捕捉限制为缩放仅在 高级配置模式下可用。

要将捕捉限制在比例范围内，有三种可用模式：

- *禁用*: 无论当前地图比例是多少，都会启用捕捉。这是默认模式。
- *全局*: 捕捉是有限的，仅当地图的当前比例介于全局最小值和全局最大值之间时才会启用。选择此模式时，将有两个控件用于配置启用捕捉的比例范围。
- *每个图层*: 捕捉比例范围限制是为每个图层定义的。选择此模式时，将有两列可用于配置每个图层的最小和最大比例。

请注意，最小和最大比例遵循 QGIS 约定：最小比例是” 缩小” 比例的最小值，而最大比例是” 放大” 比例的最大值。设置为”0” 或” 未设置” 的最小或最大比例被视为非限制性。

自捕捉

自捕捉 选项允许捕捉到正在编辑的几何图形。结合高级数字化面板 ，提供了一种相对于之前边或顶点对新边进行数字化的便捷方式。自捕捉可能会导致无效的几何图形，请谨慎使用。

在自定义网格上捕捉

还可以在图层属性对话框的 数字化选项卡中基于图层自定义捕捉距离。通过设置 几何图形精度距离，可以启用在地图画布处于相干比例时可见的点缀网格进行显示。然后对网格的点进行捕捉：添加或修改的几何图形的所有顶点将自动捕捉到网格的最近节点。更多信息，请参阅*Digitizing Properties* 。

12.3.3 拓扑编辑

除了这些捕捉选项之外，捕捉选项... 对话框 (工程 ► 捕捉选项) 和 捕捉工具栏允许您启用 / 禁用一些其他拓扑功能。

启用拓扑编辑

拓扑编辑 按钮有助于编辑和维护具有公共边界的要素。启用此选项后，QGIS 将' 检测' 共享边界。移动公共顶点/线段时，QGIS 也会在相邻要素的几何图形中移动它们。

拓扑编辑适用于来自不同图层的要素，只要图层可见且处于编辑模式即可。

在具有 Z 或 M 值的图层中，拓扑编辑将根据用于连接的边的值对顶点的 Z 或 M 值进行插值。

重叠控制

重叠可防止绘制与选定图层中现有要素相重叠的新要素，从而加快邻接多边形的数字化。可以通过重叠工具进行控制，并有三种模式可用：

1. 允许重叠 (默认)
2. 在活动图层避免重叠: 防止与正在编辑的图层中的其他要素发生重叠。数字化新几何图形，使其与相邻几何图形重叠，QGIS 将剪切新几何图形的重叠部分，并将其捕捉到现有要素的边界。其优势是不必数字化边界上的公共顶点。
3. 遵循高级配置: 允许在 高级配置视图模式中以图层为基础设置重叠的相关参数。

> **Note**
>
> 如果新几何图形完全被现有几何图形覆盖，会将其被清除，QGIS 显示一条错误信息。

> **Warning**
>
> **请谨慎使用** 避免重叠 **选项**
>
> 由于此选项将剪切任何多边形图层的新重叠几何图形，因此如果您在不再需要时忘记取消勾选，可能会得到意外的几何图形。

自动追踪

通常，使用捕捉地图工具 (添加要素、添加部件、添加环、重塑和分割) 时，需要单击要素的每个顶点。使用自动追踪模式，可以加快数字化处理，因为在数字化过程中不再需要手动放置所有顶点：

1. 通过按下图标或按 T 键启用 追踪 工具 (在 捕捉工具栏中) 。
2. 捕捉到 沿其追踪要素的顶点或线段。
3. 将鼠标移动到要捕捉的另一个顶点或线段上，数字化橡皮筋表示从捕捉的最后一个点到当前位置的路径，而不是通常的直线。该工具也适用于曲线几何图形。

   QGIS 实际是使用底层要素拓扑来构建两点之间的最短路径。追踪要求在可追踪图层中激活捕捉以构建路径。数字化时还应捕捉到现有顶点或线段，并确保两个节点可通过现有要素边进行拓扑连接，否则 QGIS 将无法进行连接，从而追踪单一直线。
4. 单击，QGIS 将中间顶点放置在显示的路径之后。

展开 启用追踪 图标，并设置 偏移选项以数字化平行于要素的路径，而不是沿着要素进行追踪。正值将新图形移动到追踪方向的左侧，负值则相反。

Note

为实现最佳追踪请调整地图比例或捕捉设置

如果地图显示的要素过多，将禁用追踪，避免可能过长的追踪结构准备和大量内存占用。放大或禁用某些图层后，将再次启用追踪。

Note

不添加拓扑点

即使启用了 *拓扑编辑*，该工具也不会向现有多边形几何图形添加点。如果在编辑的图层激活了几何图形精度，则生成的几何图形可能不会完全遵循现有几何图形。

Tip

按 ** :kbd:‘T‘ ** 键可快速启用或禁用自动追踪

通过按 `T` 键，可以随时启用/禁用追踪 (即使在数字化要素时也是如此)，因此可以在启用追踪时数字化要素的某些部分，而在禁用追踪的情况下数字化其他部分。禁用跟踪时，工具的行为恢复到正常状态。

Tip

将追踪转换为曲线几何图形

通过使用 `设置 ► 选项 ► 数字化 ► 追踪`，可以在数字化的同时创建曲线几何图形。请参阅数字化选项 。

12.3.4 数字化现有图层

默认情况下，QGIS 以只读方式加载图层。这是一种保护措施，可以避免在鼠标滑动时意外编辑图层。但是，只要数据提供程序支持，并且基础数据源是可写的 (其文件不是只读属性)，就可以选择编辑任何图层 (请参阅探索数据格式与字段) 。

Tip

约束工程中图层的编辑权限

From the *Project* ► *Properties...* ► *Data Sources* ► *Layers Capabilities* table, you can choose to set any layer read-only regardless the provider permission. This can be a handy way, in a multi-users environment to avoid unauthorized users to mistakenly edit layers (e.g., Shapefile), hence potentially corrupt data. Note that this setting only applies inside the current project.

In general, tools for editing vector layers are divided into a digitizing and an advanced digitizing toolbar, described in section *Advanced digitizing*. You can select and unselect both under *View* ► *Toolbars* ►.

Using the basic digitizing tools, you can perform the following functions:

Table 12.3: 矢量图层基本编辑工具栏

工具	目标	工具	目标
	Access to save, rollback or cancel changes in all or selected layers simultaneously		Turn on or off edit status of selected layer(s) based on the active layer status
	Save edits to the active layer		
	Digitize using straight segments		Digitize using curve lines
	Enable freehand digitizing		Digitize polygon of regular shape
	Add new record		Add Feature: Capture Point
	Add Feature: Capture Line		Add Feature: Capture Polygon
	Vertex Tool (All Layers)		Vertex Tool (Current Layer)
Show Vertex Editor	Set whether the vertex editor panel should auto-open		Modify the attributes of all selected features simultaneously
	Delete Selected features from the active layer		Cut Features from the active layer
	Copy selected Features from the active layer		Paste Features into the active layer
	Undo changes in the active layer		Redo changes in active layer

Note that while using any of the digitizing tools, you can still *zoom or pan* in the map canvas without losing the focus on the tool.

All editing sessions start by choosing the Toggle editing option found in the context menu of a given layer, from the attribute table dialog, the digitizing toolbar or the *Layer* menu.

Once the layer is in edit mode, additional tool buttons on the editing toolbar will become available and markers will appear at the vertices of all features unless *Show markers only for selected features* option under *Settings* ► *Options...* ► *Digitizing* menu is checked.

Tip

Save Regularly

Remember to Save Layer Edits regularly. This will also check that your data source can accept all the changes.

Geometry editing techniques

When a geometry drawing tool (mainly the ones that add, split, reshape features) is enabled for a line or polygon based layer, you can select the technique for adding new vertices:

- The Digitize with Segment: draws straight segment whose start and end points are defined by left clicks.
- The Digitize with Curve: draws curve line based on three consecutive nodes defined by left clicks (start, point along the arc, end). If the geometry type does not support curves, then consecutive smaller segments are used to approximate the curvature.
- The Stream Digitizing: draws lines in freehand mode, i.e. nodes are added following cursor movement in the map canvas and a *Streaming Tolerance*. The streaming tolerance defines the spacing between consecutive

vertices. Currently, the only supported unit is pixels (`px`). Only the starting left click and the ending right click are necessary in this mode.

- The Digitize Shape: triggers tools on the *Shape Digitizing Toolbar* to draw a polygon of a regular shape.

The selected technique remains while switching among the digitizing tools. You can combine any of the first three methods while drawing the same geometry.

Adding Features

Depending on the layer type, you can use the Add Record, Add Point Feature, Add Line Feature or Add Polygon Feature icons on the toolbar to add new features into the current layer.

To add a geometryless feature, click on the Add Record button and you can enter attributes in the feature form that opens.

To create features with the spatially enabled tools, you first digitize the geometry then enter its attributes. To digitize the geometry:

1. (Optional as it is the default) Select the Digitize With Segment geometry drawing method
2. Left-click on the map area to create the first point of your new feature. For point features, this should be enough and trigger, if required, the feature form to fill in their attributes.
3. For line or polygon geometries, keep on left-clicking for each additional point you wish to capture. You can rely on the *snapping to features* options, the *snap-to-grid* or the *advanced digitizing* panel to accurately position each vertex.

   Along with drawing straight segments between nodes you click one by one, lines and polygons can be:

   - *traced automatically*, accelerating the digitization. This will create consecutive straight lines between the vertices you place, following existing features.
   - free-hand digitized, pressing `R` or activating Stream Digitizing.
   - drawn as curve, pressing `Ctrl+Shift+G` or activating Digitize with Curve.

   Note

   While digitizing line or polygon geometries, you can switch back and forth between the geometry drawing methods, allowing you to create features mixing straight segments, free-hand ones and curved parts.

4. Press `Delete` or `Backspace` key to revert the last node(s) you may wrongly add.
5. When you have finished adding points, right-click anywhere on the map area to confirm you have finished entering the geometry of that feature.

   Tip

   Customize the digitizing rubber band

   While capturing polygon, the by-default red rubber band can hide underlying features or places you'd like to capture a point. This can be fixed by setting a lower opacity (or alpha channel) to the rubber band's *Fill Color* in *Settings* ► *Options* ► *Digitizing* menu. You can also avoid the use of the rubber band by checking *Don't update rubber band during node editing*.

6. For line feature pressing `Shift` + right-click will close the line automatically.

7. The attribute window will appear, allowing you to enter the information for the new feature. Fig. 12.80 shows setting attributes for a fictitious new river. However, in the *Digitizing* menu under the *Settings* ► *Options* menu, you can also:

   - *Suppress attributes pop-up windows after each created feature* to avoid the form opening;
   - or *Reuse last entered attribute values* to have fields automatically filled at the opening of the form and just have to type changing values.

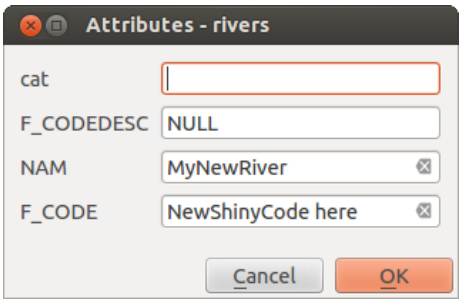


Fig. 12.80: Enter Attribute Values Dialog after digitizing a new vector feature

顶点工具

QGIS provides two tools to interact with vector features vertices:

- Vertex Tool (Current Layer): only overlaid features in the active layer (in the *Layers* panel) are affected
- Vertex Tool (All Layers): any overlaid features in all editable layers are affected. This allows you to edit features without switching the active layer or edit multiple layers at once (e.g., country and their regions boundaries)

For any editable vector layer, the vertex tools provide manipulation capabilities of feature vertices similar to CAD programs. It is possible to select multiple vertices at once and to move, add or delete them altogether. The vertex tools also support the topological editing feature. They are selection persistent, so when some operation is done, selection stays active for this feature and tool.

> **Tip**
>
> **Drawing a series of new vertices**
>
> The vertex tool does not support *automatic tracing* and is optimized for editing individual vertices and moving or deleting an arbitrary selection of multiple vertices. Try using the Reshape Features tool when you need to replace or insert a series of new vertices, especially if you want to trace other features.

It is important to set the property *Settings* ► *Options* ► *Digitizing* ► *Search Radius:* 1,00 to a number greater than zero. Otherwise, QGIS will not be able to tell which vertex is being edited and will display a warning.

> **Tip**
>
> **Vertex Markers**
>
> QGIS supports different kinds of vertex markers: 'Semi-transparent circle', 'Cross' and 'None'. To change the marker style, choose *Options* from the *Settings* menu, click on the *Digitizing* tab and select the appropriate entry.

Basic operations

Given a layer in edit mode, start by activating the vertex tool. Red circles will appear when hovering vertices.

- **Selecting vertices**: You can select vertices by:
  - Clicking on them one at a time holding `Shift` key pressed
  - Click-and-dragging a rectangle surrounding the target vertices
  - Drawing a polygon surrounding the target vertices: Hold `Alt` and click using the vertex tool to start digitizing a polygon. Each subsequent click adds a new vertex to the rubberband polygon. `Backspace` or `Delete` removes last added rubberband vertex. `Esc` cancels the polygon selection mode, as also does backspacing/deleting all of the rubberband's vertices. Right click finalizes the polygon digitizing and selects all vertices within the rubberband polygon.

  When a vertex is selected, its color changes to blue. To add more vertices to the current selection, hold down the `Shift` key while proceeding as above. To remove vertices from the selection, hold down `Ctrl`.

  > **Tip**
  >
  > **Feature selection bounds vertex tool**
  >
  > Vertices can be selected accross different features (or layers). If you are looking for vertices of a specific feature in a crowded place, first select that feature. Then draw the rectangle or polygon selector with the vertex tool around the vertices: only the selected feature's vertices are selected.
  >
  > This is also the case if you display the feature in the *vertex editor* panel.

- **Batch vertex selection mode**: The batch selection mode can be activated by pressing `Shift+R`. Select a first node with one single click, and then hover **without clicking** another vertex. This will dynamically select all the nodes in between using the shortest path (for polygons).

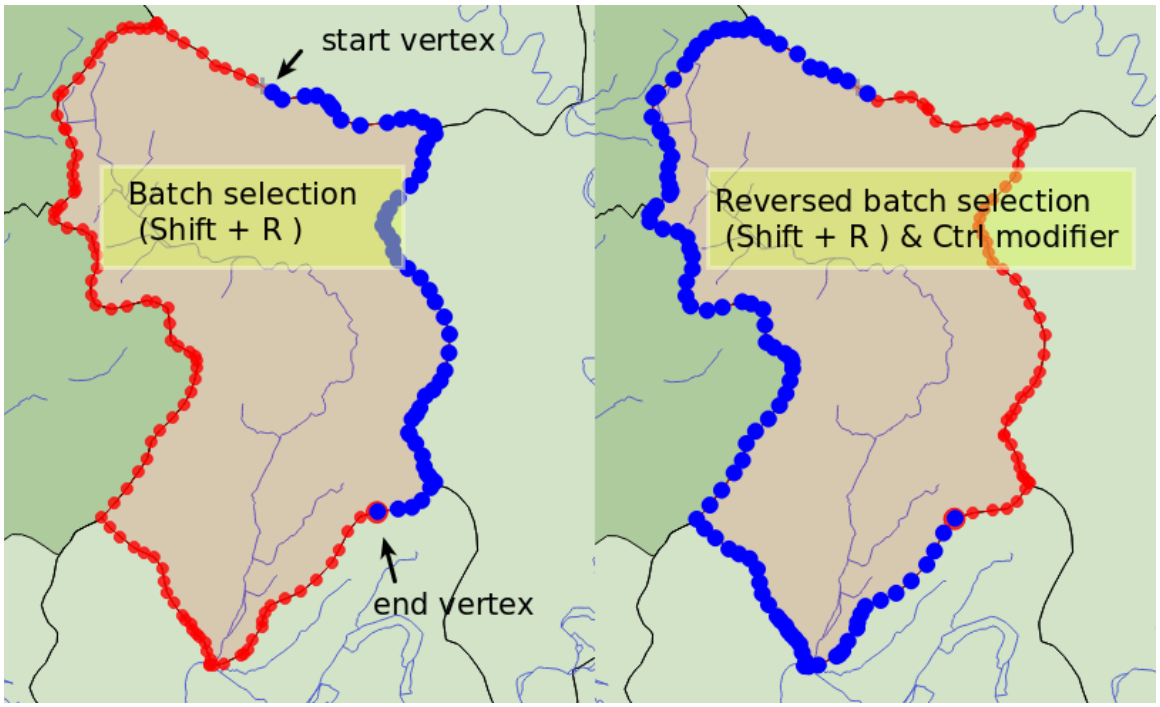


Fig. 12.81: Batch vertex selection using `Shift+R`

Press `Ctrl` will invert the selection, selecting the longest path along the feature boundary. Ending your node selection with a second click, or pressing `Esc` will escape the batch mode.

- **Adding vertices**: To add a vertex to a line or polygon geometry, hold `Shift` and double-click the place on the segment.

  When hovering a segment, a virtual new node appears on the center. Click on it, move the cursor to a new location and click again to add a new vertex. For lines, a virtual node is also proposed at both extremities: click on it, do subsequent clicks and finish with a right-click; this allows to easily extend an existing line.

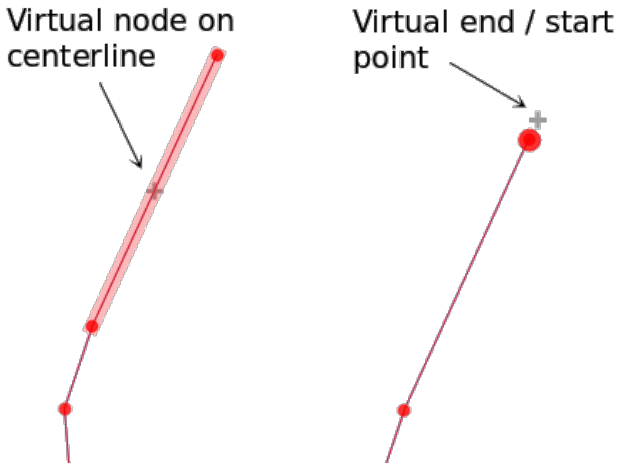


Fig. 12.82: Virtual nodes for adding vertices

- **Deleting vertices**: Select the vertices and click the `Delete` key. Deleting all the vertices of a feature generates, if compatible with the datasource, a geometryless feature. Note that this doesn't delete the complete feature, just the geometry part. To delete a complete feature use the Delete Selected tool.
- **Moving vertices**: Select all the vertices you want to move, click on a selected vertex or edge, and click on the desired new location. You can use the *snapping to feature capabilities* and the *Advanced Digitizing Panel* constraints for distance, angles, exact X and Y location before the second click. All the selected vertices will be translated.

  However, if the *snap-to-grid* option is enabled, selected vertices are snapped to the closest grid intersection to their translated position. Unselected vertices are also moved to their closest grid intersection. There is no simple translation.

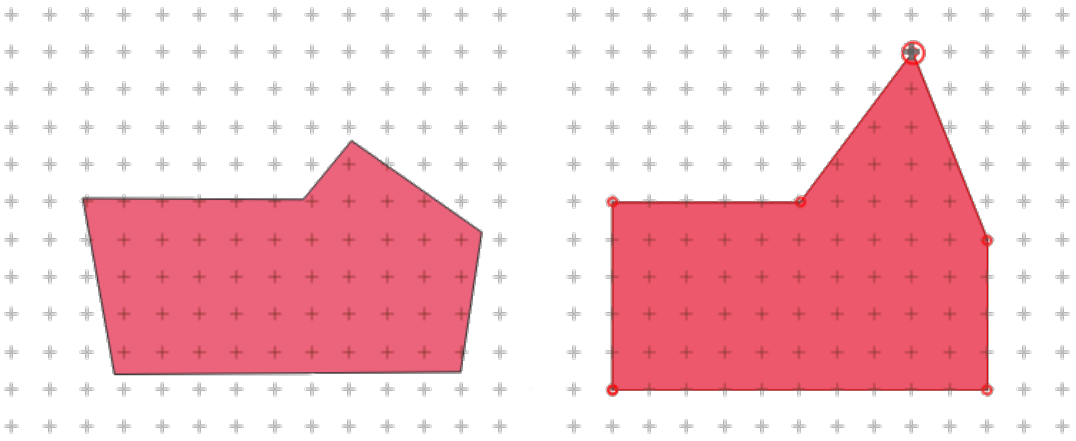


Fig. 12.83: Moving the top vertex snaps all the vertices to the grid

- **Converting adjacent segments to/from curve**: Select the center vertex of the segment you want to convert, hit the `O` letter key. If the vertex was in a curve, the curve is converted into straight lines. If the vertex was

between two straight lines, they are converted into a curve. A first or a last vertex of a line can't be converted to a center vertex curve. The layer must be compatible with curve geometry type.

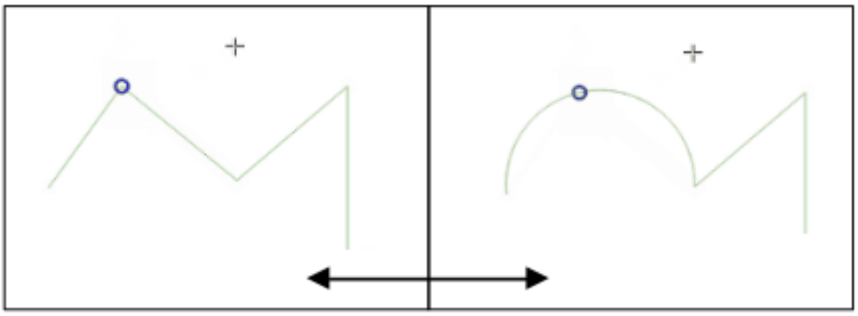


Fig. 12.84: Switch from curve to straight lines with O letter

Each change made with the vertex tool is stored as a separate entry in the *Undo* dialog. Remember that all operations support topological editing when this is turned on. On-the-fly projection is also supported.

The Vertex Editor Panel

With enabling a vertex tool, you also open the *Vertex Editor* panel. Right-clicking over a feature fills the panel with the list of all the vertices of the feature with their *x*, *y* (*z*, *m* if applicable) coordinates and *r* (for the radius, in case of circular geometry). The feature is also made exclusive for editing, meaning that the edit of any other features is disabled:

- Selecting a row in the table does select the corresponding vertex in the map canvas, and vice versa.
- Clicking or dragging over the map canvas will only select or move vertices and segments of that feature
- Change a coordinate in the table and the vertex position is updated. This is a convenient way to edit Z coordinate or M value on vertices.
- You can also select multiple rows and delete them altogether.
- New vertices can only be added to the bound feature

If you do not want the *Vertex Editor* panel to immediately show each time you interact with vertex tools (and potentially hide other panels or disturb panels placement), uncheck the *Auto-open table* entry in the ☰ Options menu at the top of the panel. You can then also close the panel. To reopen the panel, you would need to right-click over a panel or toolbar and select it in the list or tick the *Show vertex editor* entry in the *Digitizing toolbar*.

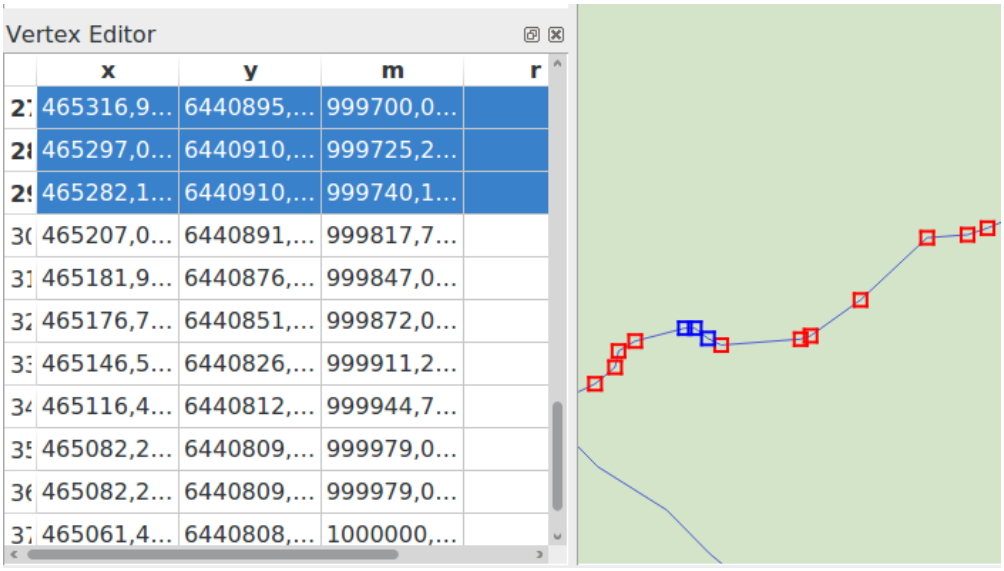


Fig. 12.85: Vertex editor panel showing selected nodes

Rules of Z coordinate or M value assignment

Digitizing 3D vector features or features with M value is not that different from (X,Y) 2D layers'. Tools and options described in this chapter are still available and help you place the vertex or point in a planar environment. Then you may need to handle the Z coordinate (or M value) assignment:

- By default, QGIS will assign to new vertices the *Default Z value* (respectively *Default M value*) set in the *Settings* ► *Options* ► *Digitizing* tab. If the *Advanced Digitizing Panel* is in use, then the value is taken from its *z* (respectively *m*) widget.
- When snapping to a vertex, the new or moved vertex takes the snapped one's Z or M value.
- When snapping to a segment while the topological editing is on, then the new vertex Z or M value is interpolated along the segment.
- If the *z* (respectively *m*) widget of the *Advanced Digitizing Panel* is locked, then its value is applied to the vertex, taking precedence over any snapped vertex or segment Z or M value.

To edit Z or M values of an existing feature, you can use the *Vertex editor panel*. To create features with custom Z or M values you may want to rely on the *Advanced Digitizing Panel*.

Cutting, Copying and Pasting Features

Selected features can be cut, copied and pasted between layers in the same QGIS project, as long as destination layers are set to Toggle editing beforehand.

Tip

Transform polygon into line and vice-versa using copy/paste

Copy a line feature and paste it in a polygon layer: QGIS pastes in the target layer a polygon whose boundary corresponds to the closed geometry of the line feature. This is a quick way to generate different geometries of the same data.

Features can also be pasted to external applications as text. That is, the features are represented in CSV format, with the geometry data appearing in the OGC Well-Known Text (WKT) format. WKT and GeoJSON features from outside QGIS can also be pasted to a layer within QGIS.

When would the copy and paste function come in handy? Well, it turns out that you can edit more than one layer at a time and copy/paste features between layers. Why would we want to do this? Say we need to do some work on a new layer but only need one or two lakes, not the 5,000 on our `big_lakes` layer. We can create a new layer and use copy/paste to plop the needed lakes into it.

As an example, we will copy some lakes to a new layer:

1. Load the layer you want to copy from (source layer)
2. Load or create the layer you want to copy to (target layer)
3. Start editing for target layer
4. Make the source layer active by clicking on it in the legend
5. Use the Select Features by area or single click tool to select the feature(s) on the source layer
6. Click on the Copy Features tool
7. Make the destination layer active by clicking on it in the legend
8. Click on the Paste Features tool
9. Stop editing and save the changes

What happens if the source and target layers have different schemas (field names and types are not the same)? QGIS populates what matches and ignores the rest. If you don't care about the attributes being copied to the target layer, it doesn't matter how you design the fields and data types. If you want to make sure everything - the feature and its attributes - gets copied, make sure the schemas match.

When you paste features into another layer, QGIS checks the attribute values against the *constraints* defined in the destination layer (for example, `NOT NULL`, `Unique`, or expression constraints). If one or more pasted features contain invalid attribute values that do not meet these constraints, a dialog will appear, listing all affected features and fields. When pasting a single feature, you can decide whether to:

- Edit the invalid values directly in the dialog. The corrected values will be used when pasting.
- *Paste Anyway*: the feature is pasted as-is, even with invalid values.

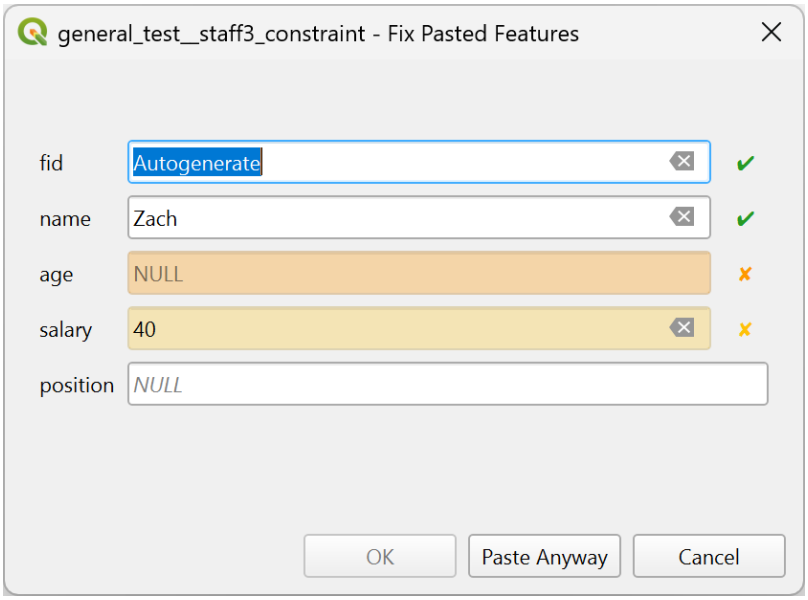


Fig. 12.86: Dialog for handling invalid attribute values when pasting a single feature

When pasting multiple features, the dialog lists all invalid fields across all features, allowing bulk review and correction. You can decide per field or per feature whether to:

- Edit the invalid values directly in the dialog. The corrected values will be used when pasting.
- *Discard All*: no features pasted.
- *Discard All Invalid*: pasted only features with valid attribute values.
- *Paste All (Including Invalid)*: all features are pasted as-is, even with invalid values.
- *Skip*: skip the specific feature, allowing proceeding with the rest.

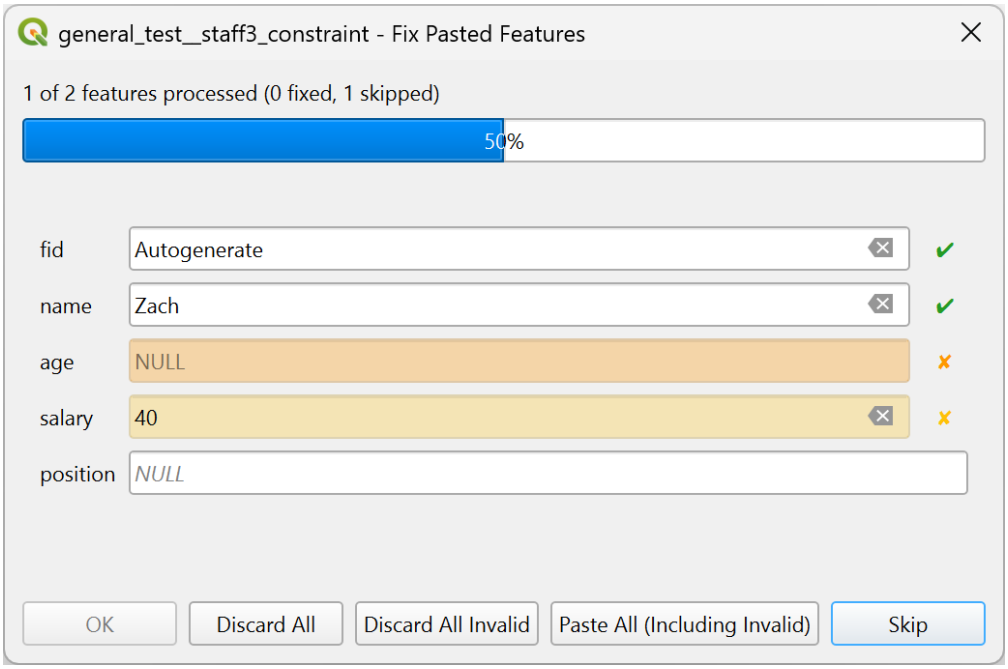


Fig. 12.87: Dialog for handling invalid attribute values when pasting multiple features

Note

Congruency of Pasted Features

If your source and destination layers use the same projection, then the pasted features will have geometry identical to the source layer. However, if the destination layer is a different projection, then QGIS cannot guarantee the geometry is identical. This is simply because there are small rounding-off errors involved when converting between projections.

Tip

Copy string attribute into another

If you have created a new column in your attribute table with type 'string' and want to paste values from another attribute column that has a greater length the length of the column size will be extended to the same amount. This is because the GDAL Shapefile driver knows to auto-extend string and integer fields to dynamically accommodate for the length of the data to be inserted.

Deleting Selected Features

If we want to delete an entire feature (attribute and geometry), we can do that by first selecting the geometry using the regular Select Features by area or single click tool. Selection can also be done from the attribute table. Once you have the selection set, press `Delete` or `Backspace` key or use the Delete Selected tool to delete the features. Multiple selected features can be deleted at once.

The Cut Features tool on the digitizing toolbar can also be used to delete features. This effectively deletes the feature but also places it on a "spatial clipboard". So, we cut the feature to delete. We could then use the Paste Features tool to put it back, giving us a one-level undo capability. Cut, copy, and paste work on the currently selected features, meaning we can operate on more than one at a time.

Undo and Redo

The Undo and Redo tools allows you to undo or redo vector editing operations. There is also a dockable widget, which shows all operations in the undo/redo history (see Fig. 12.88). This widget is not displayed by default; it can be displayed by right-clicking on the toolbar and activating the *Undo/Redo Panel* checkbox. The Undo/Redo capability is however active, even if the widget is not displayed.

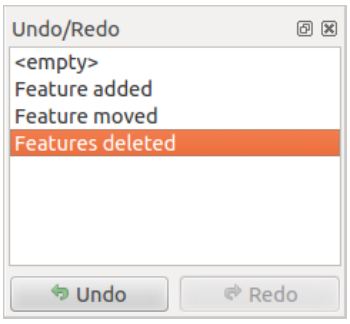


Fig. 12.88: Redo and Undo digitizing steps

When Undo is hit or `Ctrl+Z` (or `Cmd+Z`) pressed, the state of all features and attributes are reverted to the state before the reverted operation happened. Changes other than normal vector editing operations (for example, changes done by a plugin) may or may not be reverted, depending on how the changes were performed.

To use the undo/redo history widget, simply click to select an operation in the history list. All features will be reverted to the state they were in after the selected operation.

Saving Edited Layers

When a layer is in editing mode, any changes remain in the memory of QGIS. Therefore, they are not committed/saved immediately to the data source or disk. If you want to save edits to the current layer but want to continue editing without leaving the editing mode, you can click the Save Layer Edits button. When you turn editing mode off with Toggle editing (or quit QGIS for that matter), you are also asked if you want to save your changes or discard them.

If the changes cannot be saved (e.g., disk full, or the attributes have values that are out of range), the QGIS in-memory state is preserved. This allows you to adjust your edits and try again.

Tip

Data Integrity

It is always a good idea to back up your data source before you start editing. While the authors of QGIS have made every effort to preserve the integrity of your data, we offer no warranty in this regard.

Saving multiple layers at once

This feature allows the digitization of multiple layers. Choose *Save for Selected Layers* to save all changes you made in multiple layers. You also have the opportunity to *Rollback for Selected Layers*, so that the digitization may be withdrawn for all selected layers. If you want to stop editing the selected layers, *Cancel for Selected Layer(s)* is an easy way.

The same functions are available for editing all layers of the project.

Tip

Use transaction group to edit, save or rollback multiple layers changes at once

When working with layers from the same PostGreSQL database, activate the *Automatically create transaction groups where possible* option in *Project* ► *Properties...* ► *Data Sources* to sync their behavior (enter or exit the edit mode, save or rollback changes at the same time).

12.3.5 Advanced digitizing

Table 12.4: Vector layer advanced editing toolbar

Icon	目标	Icon	目标
	Enable Advanced Digitizing Tools		
	Move Feature(s)		Copy and Move Feature(s)
	Rotate Feature(s)		Simplify Feature
	Scale Feature		
	Add Ring		Add Part
	Fill Ring		Swap direction
	Delete Ring		Delete Part
	Offset Curve		Reshape Features
	Split Parts		Split Features
	合并（merge）选中要素的属性		合并（merge）选中的要素
	Rotate Point Symbols		Offset Point Symbols
	Trim or Extend Feature		

Move Feature(s)

The Move Feature(s) tool allows you to move existing features:

1. Select the feature(s) to move.
2. Click on the map canvas to indicate the origin point of the displacement; you can rely on snapping capabilities to select an accurate point.

   You can also take advantages of the *advanced digitizing constraints* to accurately set the origin point coordinates. In that case:

   1. First click on the button to enable the panel.
   2. Type `x` and enter the corresponding value for the origin point you'd like to use. Then press the button next to the option to lock the value.
   3. Do the same for the `y` coordinate.
   4. Click on the map canvas and your origin point is placed at the indicated coordinates.
3. Move over the map canvas to indicate the destination point of the displacement, still using snapping mode or, as above, use the advanced digitizing panel which would provide complementary `distance` and `angle` placement constraints to place the end point of the translation.
4. Click on the map canvas: the whole features are moved to new location.

Likewise, you can create a translated copy of the feature(s) using the Copy and Move Feature(s) tool.

> **Note**
>
> If no feature is selected when you first click on the map canvas with any of the *Move Feature(s)* or *Copy and Move Feature(s)* tools, then only the feature under the mouse is affected by the action. So, if you want to move several features, they should be selected first.

Rotate Feature(s)

Use the Rotate Feature(s) tool to rotate one or multiple features in the map canvas:

1. Press the Rotate Feature(s) icon
2. Then click on the feature to rotate. The feature's centroid is referenced as rotation center, a preview of the rotated feature is displayed and a widget opens showing the current *Rotation* angle.
3. Click on the map canvas when you are satisfied with the new placement or manually enter the rotation angle in the text box. You can also use the *Snap to* ° box to constrain the rotation values.
4. If you want to rotate several features at once, they shall be selected first, and the rotation is by default around the centroid of their combined geometries.

You can also use an anchor point different from the default feature centroid: press the `Ctrl` button, click on the map canvas and that point will be used as the new rotation center.

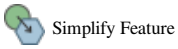
If you hold `Shift` before clicking on the map, the rotation will be done in 45 degree steps, which can be modified afterwards in the user input widget.

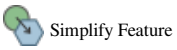
To abort feature rotation, press the `ESC` button or click on the Rotate Feature(s) icon.

Scale Feature

The tool is similar to the Rotate feature. Though instead of performing a rotation of selected features, it rescales their geometry. The change is performed in relation to the anchor point and the scale ratio can be manually specified in the widget that appears in the upper corner of the canvas.

Simplify Feature

The  tool allows you to interactively reshape a line or polygon geometry by reducing or densifying the number of vertices, as long as the geometry remains valid:

1. Select the  tool.
2. Click on the feature or drag a rectangle over the features.
3. A dialog pops up allowing you to define the *Method* to apply, ie whether you would like to:
   - *simplify the geometry*, meaning less vertices than the original. Available methods are `Simplify by distance`, `Simplify by snapping to grid` or `simplify by area (Visvalingam)`. You'd then need to indicate the value of *Tolerance* in `Layer units`, `Pixels` or `map units` to use for simplification. The higher the tolerance is the more vertices can be deleted.
   - or *densify the geometries* with new vertices thanks to the `Smooth` option: for each existing vertex, two vertices are placed on each of the segments originated from it, at an *Offset* distance representing the percentage of the segment length. You can also set the number of *Iterations* the placement would be processed: the more iterations, the more vertices and smoother is the feature.

   Settings that you used will be saved when leaving a project or an edit session. So you can go back to the same parameters the next time you simplify a feature.
4. A summary of the modifications that would apply is shown at the bottom of the dialog, listing number of features and number of vertices (before and after the operation and the ratio the change represents). Also, in the map canvas, the expected geometry is displayed over the existing one, using the rubberband color.
5. When the expected geometry fits your needs, click *OK* to apply the modification. Otherwise, to abort the operation, you can either press *Cancel* or right-click in the map canvas.

> **Note**
>
> Unlike the feature simplification option in *Settings* ► *Options* ► *Rendering* menu which simplifies the geometry just for rendering, the tool permanently modifies feature's geometry in data source.

Add Part

You can to a selected feature generating a multipoint, multiline or multipolygon feature. The new part must be digitized outside the existing one which should be selected beforehand.

The can also be used to add a geometry to a geometryless feature. First, select the feature in the attribute table and digitize the new geometry with the tool.

> **Note**
>
> **Order of vertices in polygon parts**

Unlike the OGC standards, QGIS doesn't constrain vertices of the exterior boundary of a polygon feature to be ordered counterclockwise. Thus, you can find both directions in a layer. However, every parts of the same multipolygon feature will have their outer vertices ordered following the same direction.

You can however use the *Force right-hand-rule* algorithm to constrain features of a layer to have vertices of their outer boundaries ordered in the clockwise direction.

Delete Part

The Delete Part tool allows you to delete parts from multifeatures (e.g., to delete polygons from a multi-polygon feature). This tool works with all multi-part geometries: point, line and polygon. Furthermore, it can be used to totally remove the geometric component of a feature. To delete a part, simply click within the target part.

Add Ring

You can create ring polygons using the Add Ring icon in the toolbar. This means that inside an existing area, it is possible to digitize further polygons that will occur as a 'hole', so only the area between the boundaries of the outer and inner polygons remains as a ring polygon.

To add a ring:

1. Select the feature(s) to modify
2. Activate the Add Ring tool
3. Draw a polygon within the selected geometries, using the aforementioned *techniques*. A hole appears in the selected geometries.
4. If no geometry is selected when the ring is drawn, then a hole is added to each of the polygons the ring is drawn over.

Note

Order of vertices in polygon rings

Unlike the OGC standards, QGIS doesn't constrain vertices of the exterior boundary of a polygon feature to be ordered counterclockwise. Thus, you can find both directions in a layer. However, every rings of the same (multi)polygon feature will have their vertices ordered in the opposite direction to the outer boundary's.

You can however use the *Force right-hand-rule* algorithm to constrain features of a layer to have vertices of their outer boundaries ordered in the clockwise direction, and vertices of their interior rings ordered in the counter-clockwise direction.

Fill Ring

The Fill Ring tool helps you create polygon feature that totally falls within another one without any overlapping area; that is the new feature covers a hole within the existing one. To create such a feature:

1. Select the Fill Ring tool.
2. Draw a new polygon over the existing feature: QGIS adds a ring to its geometry (like if you used the Add Ring tool) and creates a new feature whose geometry matches the ring (like if you *traced* over the interior boundaries with the Add polygon feature tool).

3. Or alternatively, if the ring already exists on the feature, place the mouse over the ring and left-click while pressing `Shift`: a new feature filling the hole is drawn at that place.

   The *Feature Attributes* form of the new feature opens, pre-filled with values of the ”parent” feature and/or *fields constraints*.

Delete Ring

The Delete Ring tool allows you to delete rings within an existing polygon, by clicking inside the hole. This tool only works with polygon and multi-polygon features. It doesn't change anything when it is used on the outer ring of the polygon.

Reshape Features

You can reshape line and polygon features using the Reshape Features tool on the toolbar. For lines, it replaces the line part from the first to the last intersection with the original line.

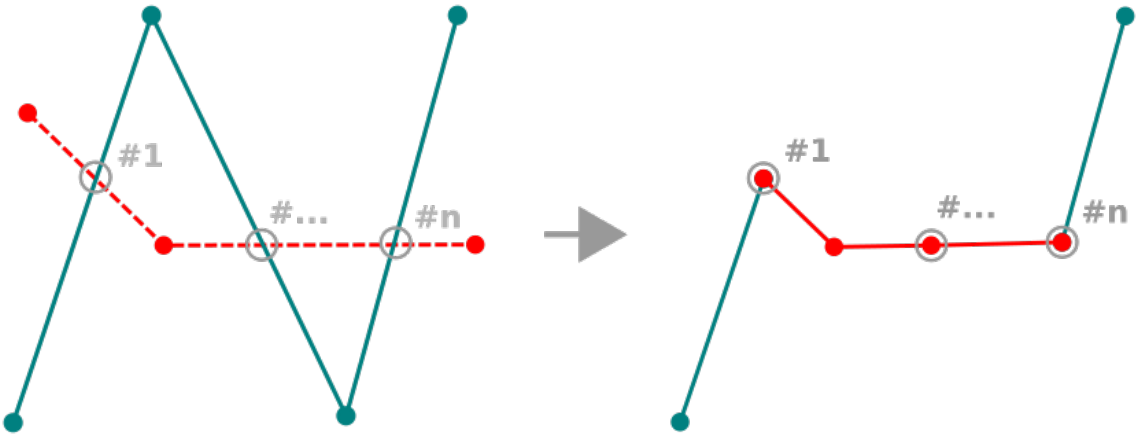


Fig. 12.89: Reshape line

Tip

Extend linestring geometries with reshape tool

Use the Reshape Features tool to extend existing linestring geometries: snap to the first or last vertex of the line and draw a new one. Validate and the feature's geometry becomes the combination of the two lines.

For polygons, it will reshape the polygon's boundary. For it to work, the reshape tool's line must cross the polygon's boundary at least twice. To draw the line, click on the map canvas to add vertexes. To finish it, just right-click. Like with the lines, only the segment between the first and the last intersections is considered. The reshape line's segments that are inside the polygon will result in cropping it, where the ones outside the polygon will extend it.

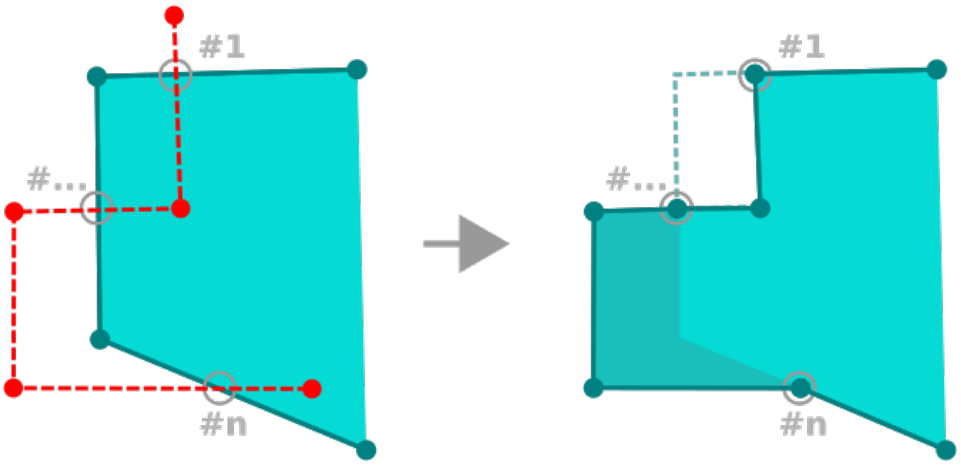


Fig. 12.90: Reshape polygon

With polygons, reshaping can sometimes lead to unintended results. It is mainly useful to replace smaller parts of a polygon, not for major overhauls, and the reshape line is not allowed to cross several polygon rings, as this would generate an invalid polygon.

Note

The reshape tool may alter the starting position of a polygon ring or a closed line. So, the point that is represented 'twice' will not be the same any more. This may not be a problem for most applications, but it is something to consider.

Offset Curves

The Offset Curve tool creates parallel shifts of line layers. The tool can be applied to the edited layer (the geometries are modified) or also to background layers (in which case it creates copies of the lines / rings and adds them to the edited layer). It is thus ideally suited for the creation of distance line layers. The *User Input* dialog pops-up, showing the displacement distance and other settings.

To create a shift of a line or polygon layer, you must first go into editing mode and activate the Offset Curve tool. Then click on a feature to shift it. Move the mouse and click where wanted or enter the desired distance in the user input widget. Holding `Ctrl` during the 2nd click will make an offset copy. Your changes may then be saved with the Save Layer Edits tool.

For geometries on background layers make sure that snapping is on and hold `Ctrl` to select the geometry from the background. Also hold `Ctrl` when doing the second click. Geometries will be converted to the target layer geometry type.

QGIS options dialog (Digitizing tab then **Curve offset tools** section) or the icon in the *User Input* dialog allows you to configure *some parameters* like **Join style**, **Quadrant segments**, **Miter limit** and **End cap style**.

Reverse Line

Changing the direction of a line geometry can be useful for cartographical purposes (e.g., for orienting symbols or labels that follow the line direction) or when preparing for network analysis where the direction of flow or travel matters.

This tool works only on line layers. For multipart lines, you can reverse individual parts without affecting the other parts of the feature.

To change a line direction:

1. Select a line layer in the *Layers* panel and make it editable.
2. Activate the reverse line tool by clicking Reverse line in the *Advanced Digitizing* toolbar.
3. Move the pointer over a line feature. The part under the cursor is highlighted.
4. Click on the line: the direction of the line (or part) is reversed.
5. You can reverse other lines by clicking on them.

> **Tip**
>
> For bulk line direction changes, use the *Reverse line direction* processing algorithm which can reverse all lines in a layer or a selection at once.

Split Features

Use the Split Features tool to split a feature into two or more new and independent features, ie. each geometry corresponding to a new row in the attribute table.

To split line or polygon features:

1. Select the Split Features tool.
2. Draw a line across the feature(s) you want to split. If a selection is active, only selected features are split. The original feature is then assigned the biggest geometry resulting from the splitting, and new features are created for the remaining parts. Fields of the features are filled/updated according to the datasource provider rules or their *splitting policy*.
3. You can then as usual modify any of the attributes of any resulting feature.

> **Tip**
>
> **Split a polyline into new features in one-click**
>
> Using the Split Features tool, snap and click on an existing vertex of a polyline feature to split that feature into two new features.

Split parts

In QGIS it is possible to split the parts of a multi part feature so that the number of parts is increased. Just draw a line across the part you want to split using the Split Parts icon.

> **Tip**
>
> **Split a polyline into new parts in one-click**
>
> Using the Split Parts tool, snap and click on an existing vertex of a polyline feature to split the feature into two new polylines belonging to the same feature.

Merge selected features

The Merge Selected Features tool allows you to create a new feature by merging existing ones: their geometries are merged to generate a new one. If features don't have common boundaries, a multipolygon/multipolyline/multipoint feature is created.

1. First, select the features you'd like to combine.
2. Then press the Merge Selected Features button.
3. In the new dialog, the *Merge* line at the bottom of the table shows the attributes of the resulting feature. If any *Merge policy* have been defined, they are automatically applied to populate this row. However, you can override these values manually. Modify the values in the following ways:
   - manually replacing the value in the corresponding cell;
   - selecting a row in the table and pressing *Take attributes from selected feature* to use the values of this initial feature;
   - pressing the *Take attributes from the largest geometry* to use the attributes from the longest line feature, the largest polygon, or the multipoints with the most parts;
   - pressing *Skip all fields* to use empty attributes;
   - expanding the drop down menu at the top of the table, select any of the above options to apply to the corresponding field only. There, you can also choose to aggregate the initial features attributes (Minimum, Maximum, Median, Sum, Count, Concatenation... depending on the type of the field. see 统计摘要面板 for the full list of functions).

   > **Note**
   >
   > If the layer has default values or clauses present on fields, these are used as the initial value for the merged feature.

4. Press *OK* to apply the modifications. A single (multi)feature is created in the layer, replacing the previously selected ones.

Merge attributes of selected features

The Merge Attributes of Selected Features tool allows you to apply same attributes to features without merging their boundaries. The dialog is the same as the `Merge Selected Features` tool's except that unlike that tool, selected objects are kept with their geometry while some of their attributes are made identical.

Rotate Point Symbols

The Rotate Point Symbols allows you to individually change the rotation of point symbols in the map canvas.

1. First, you need to indicate the field to store the rotation value in. This is made by assigning a field to the symbol *data-defined* rotation property:
   1. In the *Layer Properties* ► *Symbology* dialog, browse to the symbol editor dialog.
   2. Click the *Data-defined override* widget near the *Rotation* option of the top *Marker* level (preferably) of the symbol layers.
   3. Choose a field in the *Field Type* combobox. Values of this field are hence used to rotate each feature's symbol accordingly.

      You can also check the Store data in project entry to generate an *auxiliary data storage* field to control the rotation value.

> **Note**
>
> **Make sure that the same field is assigned to all the symbol layers**
>
> Setting the data-defined rotation field at the topmost level of the symbol tree automatically propagates it to all the symbol layers, a prerequisite to perform graphical symbol rotation with the *Rotate Point Symbols* tool. Indeed, if a symbol layer does not have the same field attached to its rotation property, the tool will not work.

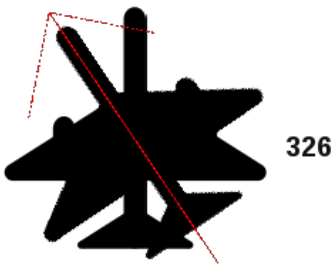


Fig. 12.91: Rotating a point symbol

2. Then click on a point symbol in the map canvas with the Rotate Point Symbols tool
3. Move the mouse around. A red arrow with the rotation value will be visualized (see Fig. 12.91). If you hold the `Ctrl` key while moving, the rotation will be done in 15 degree steps.
4. When you get the expected angle value, click again. The symbol is rendered with this new rotation and the associated field is updated accordingly.

   You can right-click to abort symbol rotation.

Offset Point Symbols

The Offset Point Symbols allows you to interactively change the rendered position of point symbols in the map canvas. This tool behaves like the Rotate Point Symbols tool except that it requires you to connect a field to the data-defined *Offset (X,Y)* property of each layer of the symbol. The field will then be populated with the offset coordinates for the features whose symbol is moved in the map canvas.

1. Associate a field to the data-defined widget of the *Offset (X,Y)* property of the symbol. If the symbol is made with many layers, you may want to assign the field to each of them
2. Select the Offset Point Symbols tool
3. Click a point symbol
4. Move to a new location
5. Click again. The symbol is moved to the new place. Offset values from the original position are stored in the linked field.

   You can right-click to abort symbol offset.

> **Note**
>
> The Offset Point Symbols tool doesn't move the point feature itself; you should use the Vertex Tool (Current Layer) or Move Feature tool for this purpose.

Trim/Extend Feature

The Trim/Extend tool allows you to shorten or lengthen segments of a (multi)line or (multi)polygon geometry to converge with a selected segment (the cutting line). This results in a modified geometry with a vertex snapped to a reference segment or in its prolongation. Depending on how the selected geometries are placed in relation to each other, the tool will either:

- **Trim**: removes parts of the line segment or polygon boundary, beyond the cutting line
- **Extend**: extends polygon boundaries or line segments so that they can snap to the cutting line.

In order to trim or extend existing geometries:

1. Select the Trim/Extend tool
2. Click the reference segment, i.e., the segment with respect to which you want to extend or trim another segment. It doesn't need to be on the active layer. A red dotted line appears, visually extending the reference segment across the map canvas according to the active layer' s CRS.
3. Hover over the target segment, i.e., the one you want to trim or extend. It does not need to be the last segment of the geometry, but has to be on the active layer. QGIS displays a preview of the feature's geometry with the segment limited to its intersection with the highlighted red dotted extending line.
   1. If an extension of the segment is required, click anywhere on the segment.
   2. In the case of a trim, click the part that should remain.
   3. In either case, the feature's geometry is updated accordingly. When both segments are in 3D, the tool performs an interpolation on the limit segment to get the Z value.
4. If you need to use the same reference segment for trimming or extending many features:
   1. Press `Shift` while selecting the reference segment.
   2. Click consecutively on the segments to modify and each will be trimmed or extended accordingly.

> **Note**
>
> Snapping is automatically enabled when this tool is activated. Your original snapping settings will be restored once the tool is deactivated.

> **Attention**
>
> Pay attention to the modified geometry while using the Trim/Extend tool. Depending on the inputs, it can create invalid geometries, potentially resulting in failure at layer saving.

12.3.6 Shape digitizing

The *Shape Digitizing* toolbar offers a set of tools to draw lines or polygons features of regular shape. It is synchronized with the Digitize Shape *geometry drawing method* you can select on the *Digitizing Toolbar*. To use it:

1. Display the toolbar: *View* ► *Toolbars* ► *Shape Digitizing*
2. Select a tool that creates or modifies the shape of a geometry, e.g. Add line feature, Add polygon feature, Add part, Add ring, Reshape Features, ...
3. The Digitize with segment button on the *Digitizing Toolbar* is enabled. The first time, you may need to switch it to the Digitize Shape in order to enable tools on the *Shape Digitizing* toolbar.
4. Pick a shape digitizing tool and draw.

Circular string by radius

The Circular string by radius button allows to add line or polygon features with a circular geometry, given two nodes on the curve and a radius:

1. Left click twice to place the two points on the geometry.
2. A *Radius* widget in the top right corner of the map canvas displays current radius (corresponding to distance between the points). Edit that field to the value you want.
3. An overview of the arcs matching these constraints is displayed while moving around the cursor. Right-click to validate when the expected arc is shown.
4. Add a new point to start shaping another arc.

> **Note**
>
> **Curved geometries are stored as such only in compatible data provider**
>
> Although QGIS allows to digitize curved geometries within any editable data format, you need to be using a data provider (e.g. PostgreSQL, Oracle Spatial, memory layer, GML or WFS) that supports curves to have features stored as curved, otherwise QGIS segmentizes the circular arcs.

Draw Circles

There is a set of tools for drawing circles. The tools are described below.

Circles are converted into circular strings. Therefore, as explained in *Circular string by radius*, if allowed by the data provider, it will be saved as a curved geometry, if not, QGIS will segmentize the circular arcs.

- Circle from 2 points: The two points define the diameter and the orientation of the circle. (Left-click, right-click)
- Circle from 3 points: Draws a circle from three known points on the circle. (Left-click, left-click, right-click)
- Circle by a center point and another point: Draws a circle with a given center and a point on the circle (Left-click, right-click). When used with the *The Advanced Digitizing panel* this tool can become a "Add circle from center and radius" tool by setting and locking the distance value after first click.
- Circle from 3 tangents: Draws a circle that is tangential to three segments. **Note that you must activate snapping to segments** (See 设置捕捉容差和搜索半径). Click on a segment to add a tangent. If two tangents are parallel, the coordinates of the click on the first parallel tangent are used to determine the positioning of the circle. If three tangents are parallel, an error message appears and the input is cleared. (Left-click, left-click, right-click)
- Circle from 2 tangents and a point: Similar to circle from 3 tangents, except that you have to select two tangents, enter a radius and select the desired center.

Draw Ellipses

There is a set of tools for drawing ellipses. The tools are described below.

Ellipses cannot be converted as circular strings, so they will always be segmented.

- Ellipse from center and two points: Draws an ellipse with a given center, major axis and minor axis. (Left-click, left-click, right-click)
- Ellipse from center and a point: Draws an ellipse into a bounding box with the center and a corner. (Left-click, right-click)
- Ellipse from extent: Draws an ellipse into a bounding box with two opposite corners. (Left-click, right-click)
- Ellipse from foci: Draws an ellipse by 2 points for foci and a point on the ellipse. (Left-click, left-click, right-click)

Draw Rectangles

There is a set of tools for drawing rectangles. The tools are described below.

- Rectangle from center and a point: Draws a rectangle from the center and a corner. (Left-click, right-click)
- Rectangle from extent: Draws a rectangle from two opposite corners. (Left-click, right-click)
- Rectangle from 3 points (distance): Draws an oriented rectangle from three points. The first and second points determine the length and angle of the first edge. The third point determines the length of the other edge. One can use *The Advanced Digitizing panel* to set the length of the edges. (Left-click, left-click, right-click)
- Rectangle from 3 points (projected): Same as the preceding tool, but the length of the second edge is computed from the projection of the third point on the first edge. (Left-click, left-click, right-click)

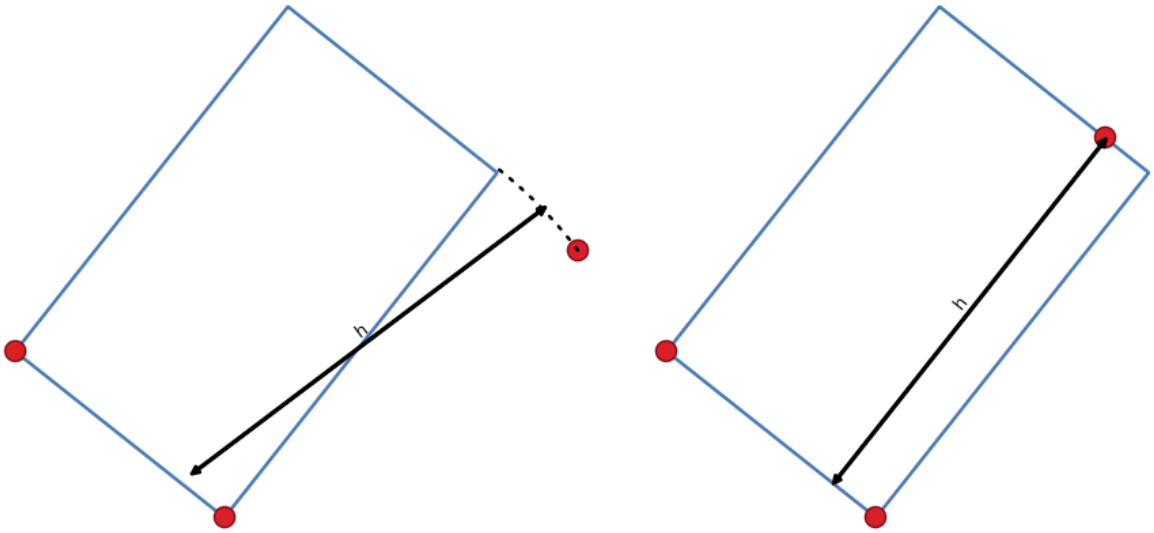


Fig. 12.92: Draw rectangle from 3 points using distance (right) and projected (left)

Draw Regular Polygons

There is a set of tools for drawing regular polygons. The tools are described below. Left-click to place the first point. A dialog appears, where you can set the number of polygon edges. Right-click to finish the regular polygon.

- Regular polygon from two points: Draws a regular polygon where the two points determine the length and angle of the first edge.
- Regular polygon from center and a point: Draws a regular polygon from the provided center point. The second point determines the angle and distance to the middle of an edge.
- Regular polygon from center and a corner: Same as the preceding tool, but the second point determines the angle and distante to a vertex.

12.3.7 The Advanced Digitizing panel

When capturing, reshaping, splitting new or existing geometries you also have the possibility to use the Advanced Digitizing panel. You can digitize lines exactly parallel or perpendicular to a particular angle or lock lines to specific angles. Furthermore, you can make a precise definition of your new geometry by entering X and Y coordinates as well as Z for 3D features, or M values.

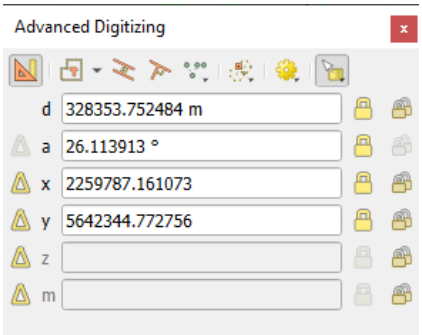


Fig. 12.93: The Advanced Digitizing panel

The *Advanced Digitizing* panel can be opened either with a right-click on the toolbar, from *View* ► *Panels* ► menu or pressing `Ctrl+4`. Once the panel is visible, click the Enable advanced digitizing tools button to activate the set of tools.

> **Note**
>
> The tools are not enabled if the map view is in geographic coordinates.

The aim of the Advanced Digitizing tool is to lock coordinates, lengths, and angles when moving the mouse during the digitalizing in the map canvas.

You can also create constraints with relative or absolute reference. Relative reference means that the next vertex constraints' values will be relative to the previous vertex or segment.

The toolbar

At the top of the *Digitizing panel*, you find the following buttons:

- Enable advanced digitizing tools
- Construction mode: allows to capture the clicks' positions to reuse as reference points to lock distance, angle, X, Y, Z or M relative values. More details at *Construction mode*.
- Parallel to draw a line parallel to an existing one (more at *Parallel and perpendicular lines*)
- Perpendicular to draw a line perpendicular to an existing one (more at *Parallel and perpendicular lines*)
- Construction Tools provides a couple of options that constrain the vertices placement based on extrapolated coordinates of existing elements:
  - *Line Extension*: hover over a segment and you get a purple dotted line extending the segment across the map canvas. You can snap the vertex anywhere on this virtual line.
  - *X/Y Point*: hover over a vertex and you get a purple dotted line along its X or Y coordinate, across the map canvas. You can snap the vertex anywhere on this virtual line. It is even possible to hover over two different vertices, generating virtual coordinate lines for both, and snap to their intersection.
- 2-circle Point Intersection: allows you to digitize a point or vertex at the intersection of two circles. (more at *2-circle point intersection*).
- Snap to common angles: when moving the cursor, displays a virtual line that you can snap to to add the next vertex. The snapping line is defined by the last added vertex and an (absolute or relative to previous segment) angle from a preset list (following steps of 0.1°, 0.5°, 1°, 5°, 10°, 15°, 18°, 22.5°, 30°, 45° or 90°). Choose *Do not snap to common angles* to disable this feature.

  Snapping to features can be used along with snapping to common angles for accurate digitizing. For a fine-grained control on how the target element to snap to is retained, you can indicate whether to prioritize snapping to features over common angles, and vice-versa under the *Snapping priority* entry. You can switch from one method to the other during the digitizing operation, and this avoids disabling any of the snapping options in the meantime. Press `N` (or `Shift+N`) during a digitizing operation to cycle through the angles list.
- Floater settings: if the *Show floater* item is checked, a contextual menu with digitizing information follows the cursor during digitizing. The values can be accessed using the *panel's shortcuts*, edited and Locked after validation (pressing `Enter`). The type of information to display can be selected in the bottom part of the menu:
  - *Show distance*
  - *Show angle*

- *Show XY coordinates*
- *Show Z value*
- *Show M value*
- *Show bearing/azimuth*
- *Show common snapping angle*

Below the toolbar, you will find a number of text boxes whose value reflects by default the position or movement of the cursor in the map canvas. Editing these values helps you constrain the position of the items you edit:

- *d* for the distance from a reference position, usually the last edited vertex
- *a* for the angle (absolute or relative) from a reference position, usually the last edited segment
- *x* for the X coordinate of the pointer
- *y* for the Y coordinate of the pointer
- *z* for the default Z value or the Z coordinate of the vertex or segment under the pointer
- *m* for the default M value or the M value of the vertex or segment under the pointer

Keyboard shortcuts

To speed up the use of Advanced Digitizing Panel, there are a couple of keyboard shortcuts available:

Table 12.5: Keyboard shortcuts of the Advanced Digitizing Panel tools

Key	Simple	`Ctrl+` or `Alt+`	`Shift+`
`D`	Set distance	Lock distance	
`A`	Set angle	Lock angle	Toggle relative angle to last segment
`X`	Set X coordinate	Lock X coordinate	Toggle relative X to last vertex
`Y`	Set Y coordinate	Lock Y coordinate	Toggle relative Y to last vertex
`Z`	Set Z coordinate	Lock Z coordinate	Toggle relative Z to last vertex
`M`	Set M value	Lock M value	Toggle relative M to last vertex
`C`	Toggle construction mode		
`P`	Toggle perpendicular and parallel modes		

Note

Z coordinate and M value options are available only if compatible with the layer geometry dimension.

Absolute reference digitizing

When drawing a new geometry from scratch, it is very useful to have the possibility to start digitizing vertexes at given coordinates.

For example, to add a new feature to a polygonal layer, click the button. You can enter the exact coordinates where you want to start editing the feature, i.e.:

1. Click the *x* text box (or use the `X` keyboard shortcut).
2. Type the X coordinate value you want and press `Enter` or click the button to their right to lock the mouse to the X axis on the map canvas.
3. Click the *y* text box (or use the `Y` keyboard shortcut).

4. Type the Y coordinate value you want and press `Enter` or click the button to their right to lock the mouse to the Y axis on the map canvas.
5. If the layer has Z coordinate or M values, the corresponding *z* or *m* widget is enabled and displays its default value, as set in *Settings* ► *Options* ► *Digitizing* tab.
    1. Click the *z* or *m* text box (or use respectively the `Z` or `M` keyboard shortcut).
    2. Type the coordinate value you want and press `Enter` or click the button to their right to lock the value in the widget.

> **Note**
>
> Read *Rules of Z coordinate or M value assignment* for details on how Z coordinate and M values are automatically determined from existing features.

6. Two blue dotted lines and a green cross identify the exact coordinates you entered. Click on the map canvas to add a vertex at the green cross position.

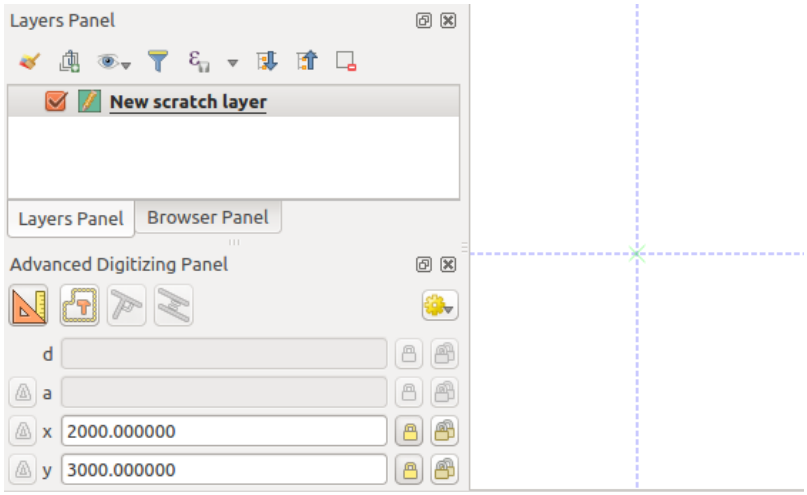


Fig. 12.94: Start drawing at given coordinates

7. You can proceed as above, adding a new set of coordinates for the next vertex, or switch to another *mode of digitizing* (e.g. segment, curve or stream).
8. If you want to draw a segment of a given length:
    1. Click the *d (distance)* text box (keyboard shortcut `D`)
    2. Type the distance value (in map units)
    3. Press `Enter` or click the button on the right to lock the mouse in the map canvas to the length of the segment. In the map canvas, the latest vertex is surrounded by a circle whose radius is the value entered in the distance text box. A cross on the circle shows the position of the next vertex if you click.

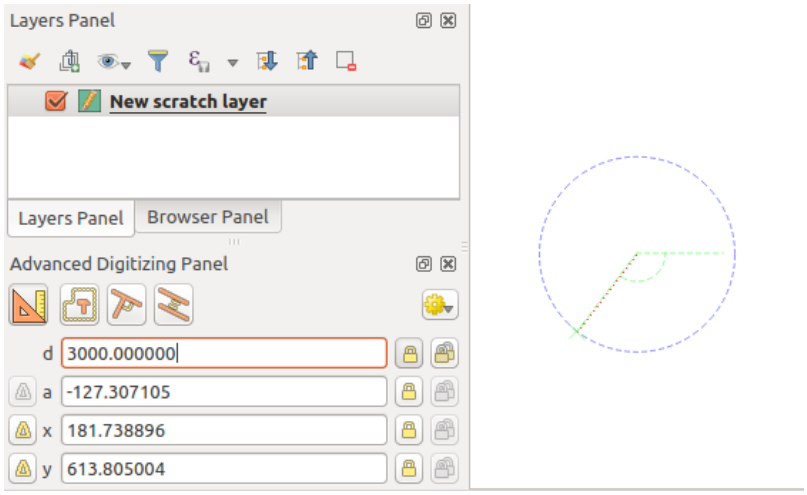


Fig. 12.95: Fixed length segment

9. You can also constrain the vertex position, setting the angle of the segment. As described before:
   1. Click the *a (angle)* text box (keyboard shortcut `A`)
   2. Type the angle value (in degrees)
   3. Press `Enter` or click the button on the right to lock it. A line going through the latest vertex and rotated based on the set angle appears in the map canvas and a cross on it shows the next vertex position if you click.

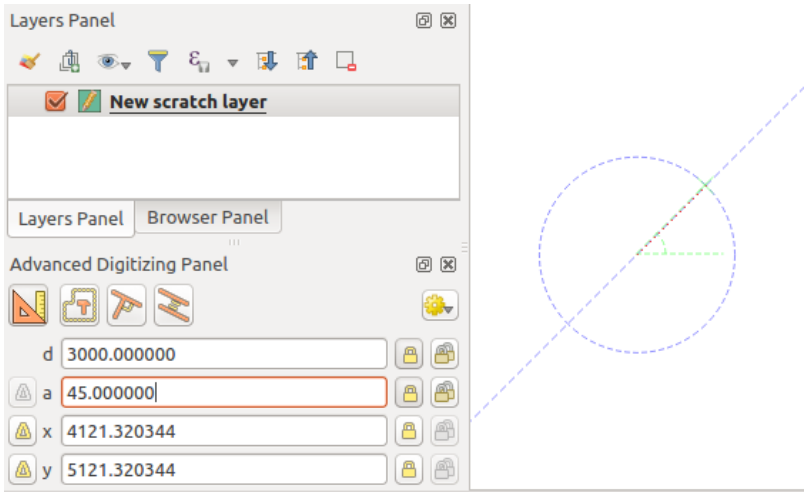


Fig. 12.96: Fixed angle segment

Hint

Pressing `Ctrl+<key>` or `Alt+<key>` automatically locks the target property and puts its value into edit. Modify, press `Enter` and you are done. Combined with the Toggle floater, this can be a real time saver, with keyboard digitizing.

Relative reference digitizing

Instead of using absolute values of angles or coordinates, you can also use values relative to the last digitized vertex or segment.

For angles, you can click the button on the left of the *a* text box (or press `Shift+A`) to toggle relative angles to the previous segment. With that option on, angles are measured between the last segment and the mouse pointer.

For coordinates, click the buttons to the left of the *x*, *y*, *z* or *m* text boxes (or press `Shift+<key>`) to toggle relative coordinates to the previous vertex. With these options on, coordinates measurement will consider the last vertex to be the origin of the set coordinates.

Continuous lock

Both in absolute or relative reference digitizing, angle, distance, X, Y, Z and M constraints can be locked continuously by clicking the *Continuous lock* buttons. Using continuous lock allows you to digitize several points or vertexes using the same constraints.

Construction mode

You can enable and disable *construction mode* by clicking on the Construction mode icon or with the `C` keyboard shortcut. While in construction mode, clicking the map canvas won't add new vertexes, but will capture the clicks' positions so that you can use them as reference points to then lock distance, angle or X, Y, Z, M relative values.

In the Construction mode you will find a drop-down menu where you can choose to:

- *Record Construction Guides*: All construction steps are rendered as dashed lines. Each step that you perform is visually represented, allowing you to trace the construction process. The guides are displayed as long as *Advanced Digitizing* remains active.
- *Show Construction Guides*: Allows you to make construction guides visible or hidden on the canvas. When enabled, all active guides are displayed, offering better spatial orientation and precision during the digitizing process.
- *Snap to Visible Construction Guides*: The guides are snap-able, allowing you to start new construction steps from any point along the existing guides.
- Choose *Clear Construction Guides* to remove all the guides from the canvas.

As an example, the construction mode can be used to draw some point at an exact distance from an existing point.

With an existing point in the map canvas and the snapping mode correctly activated, you can easily draw other points at given distances and angles from it. In addition to the button, you have to activate also the *construction mode* by clicking the Construction mode icon or with the `C` keyboard shortcut.

Click next to the point from which you want to calculate the distance and click on the *d* box (`D` shortcut) type the desired distance and press `Enter` to lock the mouse position in the map canvas:

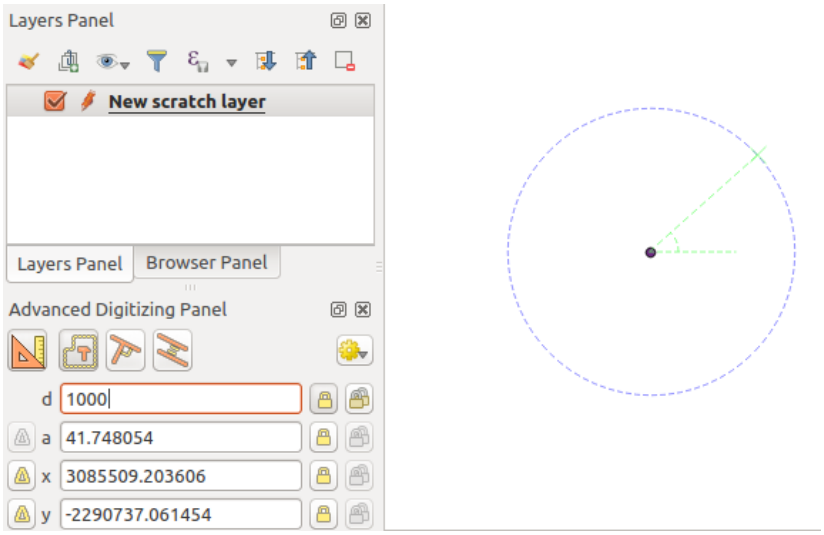


Fig. 12.97: Distance from point

Before adding the new point, press `C` to exit the construction mode. Now, you can click on the map canvas, and the point will be placed at the distance entered.

You can also use the angle constraint to, for example, create another point at the same distance of the original one, but at a particular angle from the newly added point. Click the Construction mode icon or with the `C` keyboard shortcut to enter construction mode. Click the recently added point, and then the other one to set a direction segment. Then, click on the *d* text box (`D` shortcut) type the desired distance and press `Enter`. Click the *a* text box (`A` shortcut) type the angle you want and press `Enter`. The mouse position will be locked both in distance and angle.

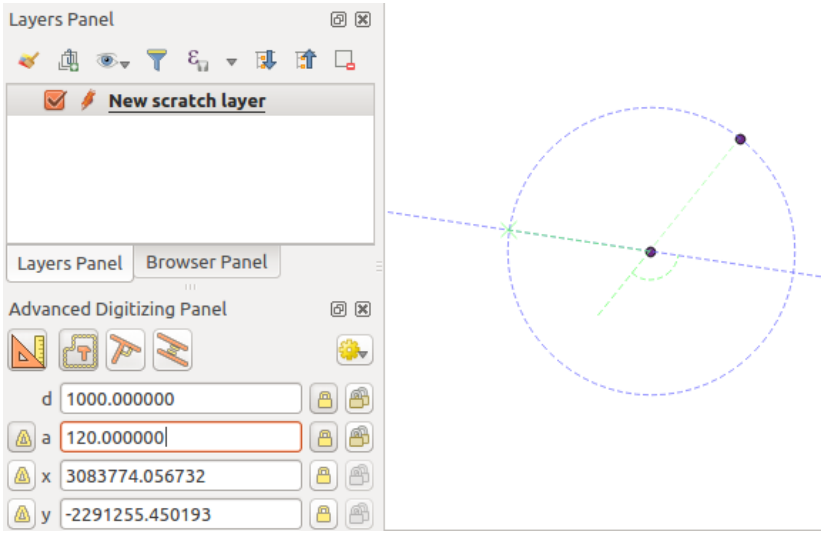


Fig. 12.98: Distance and angle from points

Before adding the new point, press `C` to exit the construction mode. Now, you can click on the map canvas, and the point will be placed at the distance and angle entered. Repeating the process, several points can be added.

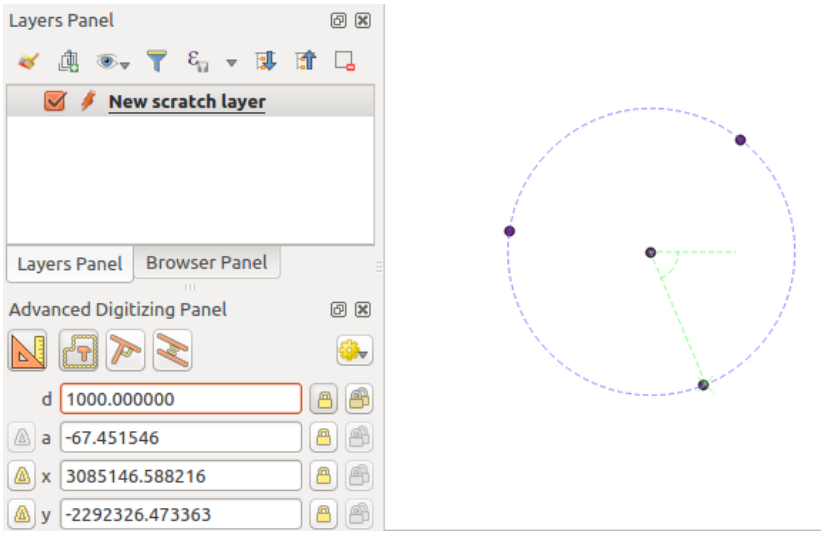


Fig. 12.99: Points at given distance and angle

Parallel and perpendicular lines

All the tools described above can be combined with the Perpendicular and Parallel tools. These two tools allow drawing segments perfectly perpendicular or parallel to another segment. The target segment can be on another layer, another feature within the layer or the feature being digitized (requires *self-snapping option*).

To draw a *perpendicular* segment:

1. First add one of the segment vertices.
2. Click the Perpendicular icon (keyboard shortcut `P`) to activate it.
3. Click on the segment that you want to be perpendicular to.
4. A virtual dotted line perpendicular to the segment through the previous vertex appears. The angle property is locked, constraining the next vertex on that line and, a cross indicates the projected position of the cursor on the line. Click to place the new vertex.

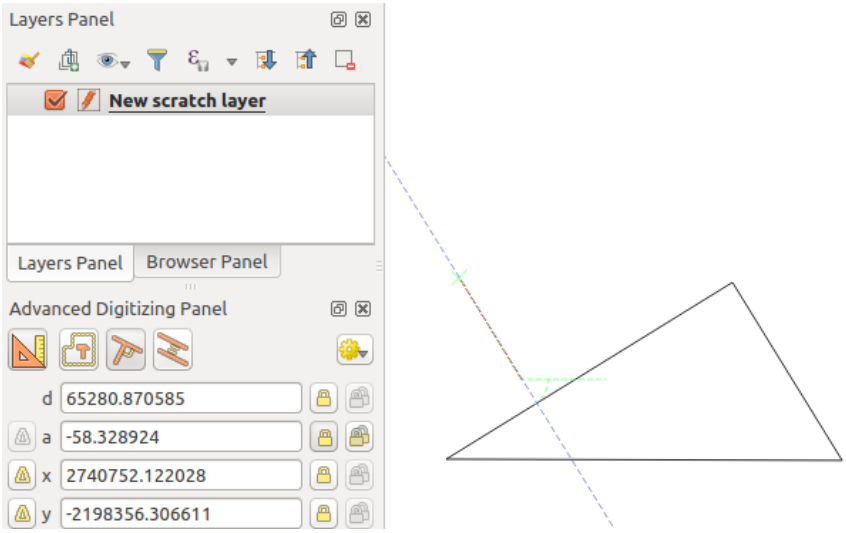


Fig. 12.100: Perpendicular digitizing

To draw a *parallel* segment, the steps are the same except that you need to click on the Parallel icon (keyboard shortcut `P` twice).

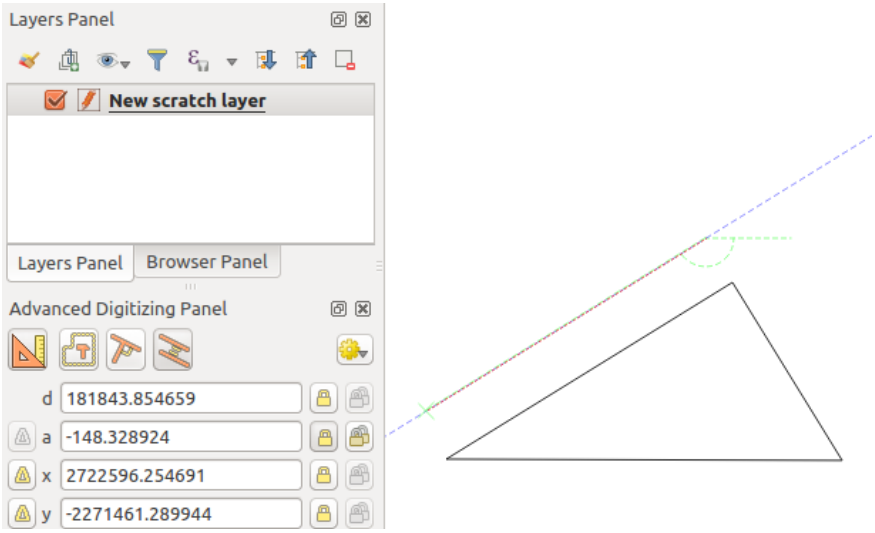


Fig. 12.101: Parallel digitizing

These two tools just find the right angle of the perpendicular and parallel angle and lock this parameter during your editing. Unlock the angle parameter to cancel their use in the middle of the process.

2-circle point intersection

To add a vertex at the intersection of two circles, follow these steps:

1. Click the 2-circle Point Intersection icon.
2. A dialog will open where you can define the parameters for *Circle #1* and *Circle #2*.
3. Click on the map canvas, and the tool will automatically calculate the *X* and *Y* coordinates for the centers of both circles.
4. Enter the distance *d* for each circle.
5. The tool will calculate and display the two intersection points of the circles.
6. Click one of the intersection points to add the new vertex.

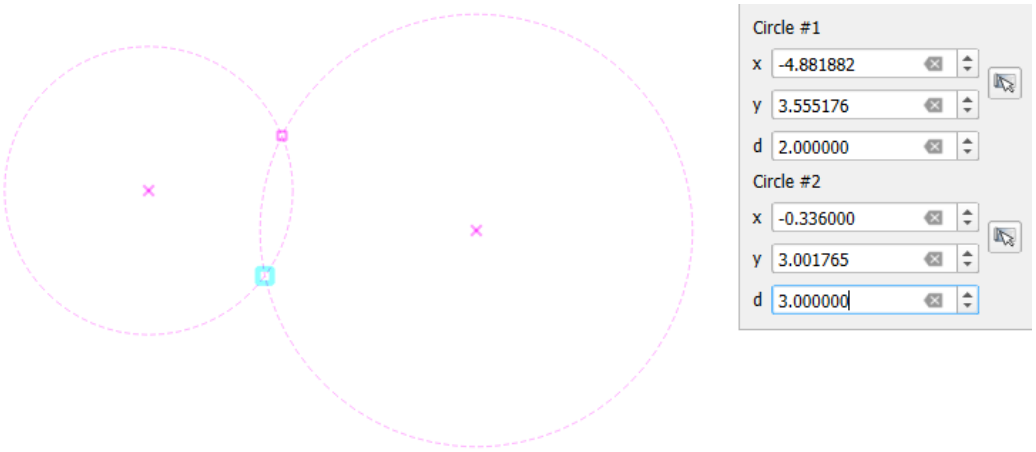


Fig. 12.102: 2-circle Point Intersection

12.3.8 The Processing in-place layer modifier

The *Processing menu* provides access to a large set of tools to analyze and create new features based on the properties of the input features or their relations with other features (within the same layer or not). While the common behavior is to create new layers as outputs, some algorithms also allow modifications to the input layer. This is a handy way to automate multiple features modification using advanced and complex operations.

To edit features in-place:

1. Select the layer to edit in the *Layers* panel.
2. Select the concerned features. You can skip this step, in which case the modification will apply to the whole layer.
3. Press the Edit Features In-Place button at the top of the *Processing toolbox*. The list of algorithms is filtered, showing only those compatible with in-place modifications, i.e.:
   - They work at the feature source and not at the layer level.
   - They do not change the layer structure, e.g. adding or removing fields.
   - They do not change the geometry type, e.g. from line to point layer.

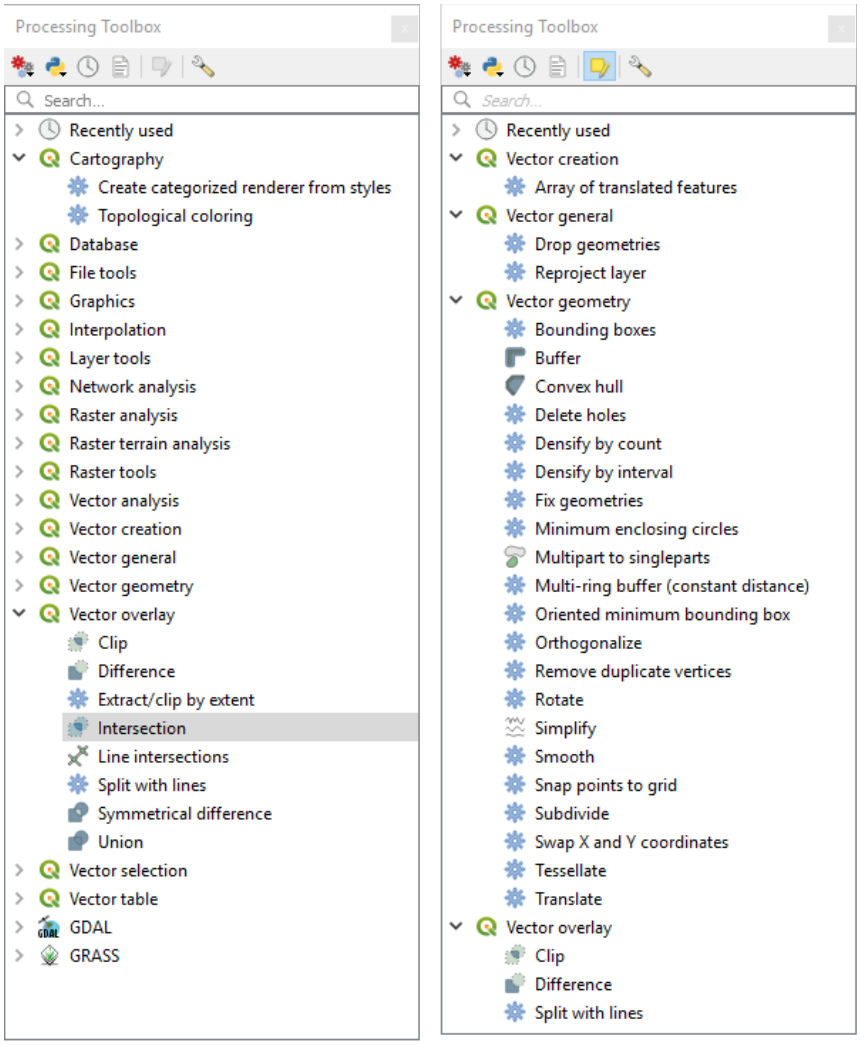


Fig. 12.103: Processing algorithms: all (left) vs polygon in-place editors (right)

4. Find the algorithm you'd like to run and double-click it.

> **Note**
>
> If the algorithm does not need any additional user-set parameters (excluding the usual input and output layer parameters), then the algorithm is run immediately without any dialog popup.

   1. If parameters other than the usual input or output layers are needed, the algorithm dialog pops up. Fill in the required information.
   2. Click *Modify Selected Features* or *Modify All Features* depending on whether there's an active selection.

   Changes are applied to the layer and placed in the edit buffer: the layer is indeed toggled to editing mode with unsaved modification as indicated by the icon next to the layer name.

5. As usual, press Save layer edits to commit the changes in the layer. You can also press Undo to rollback the whole modification.

12.4 Connecting and Editing Data Across Layers

Ability to connect data from different layers is one of the duties of a GIS software. Such a connection can be based on the spatial relationship between the features, or on their shared attributes. QGIS provides tools to handle any of these associations, such as:

- Processing algorithms that can create a new layer as a result of the connection, namely *Join attributes by location*, *Join attributes by nearest*, *Join attributes by field value*, ...
- SQL queries to create a new layer from the *DB Manager* or as a *virtual layer*
- *Joins properties* or *relations settings* that temporarily extend attributes of features in a given layer, with those of features in another layer based on some matching attribute(s).

  Joins and relations are technical concepts borrowed from databases to get the most out of data stored in tables by combining their contents. The idea is that features (rows) of different layers (tables) can be associated to each other. The number of rows which are matching each other can be of any value (zero, one, many).

12.4.1 Joining features between two layers

Joins in QGIS allow you to associate features of the current layer to features from another loaded vector layer. Whether they are spatially enabled and the type of geometry do not matter. The join is based on an attribute that is shared by the layers, in a one-to-one relationship.

To create a join on a layer (identified below as `target layer`):

1. Go to the layer *Properties* ► *Joins* tab
2. Click the Add new join button. The *Add vector join* dialog appears.
3. Select the *Join layer* you want to connect with the target vector layer
4. Specify the *Join field* (from the `join layer`) and the *Target field* (from the `target layer`). These are the fields that are used to find matching feature in both layers hence they should have values in common.
5. Press *OK* and a summary of selected parameters is added to the *Join* panel.

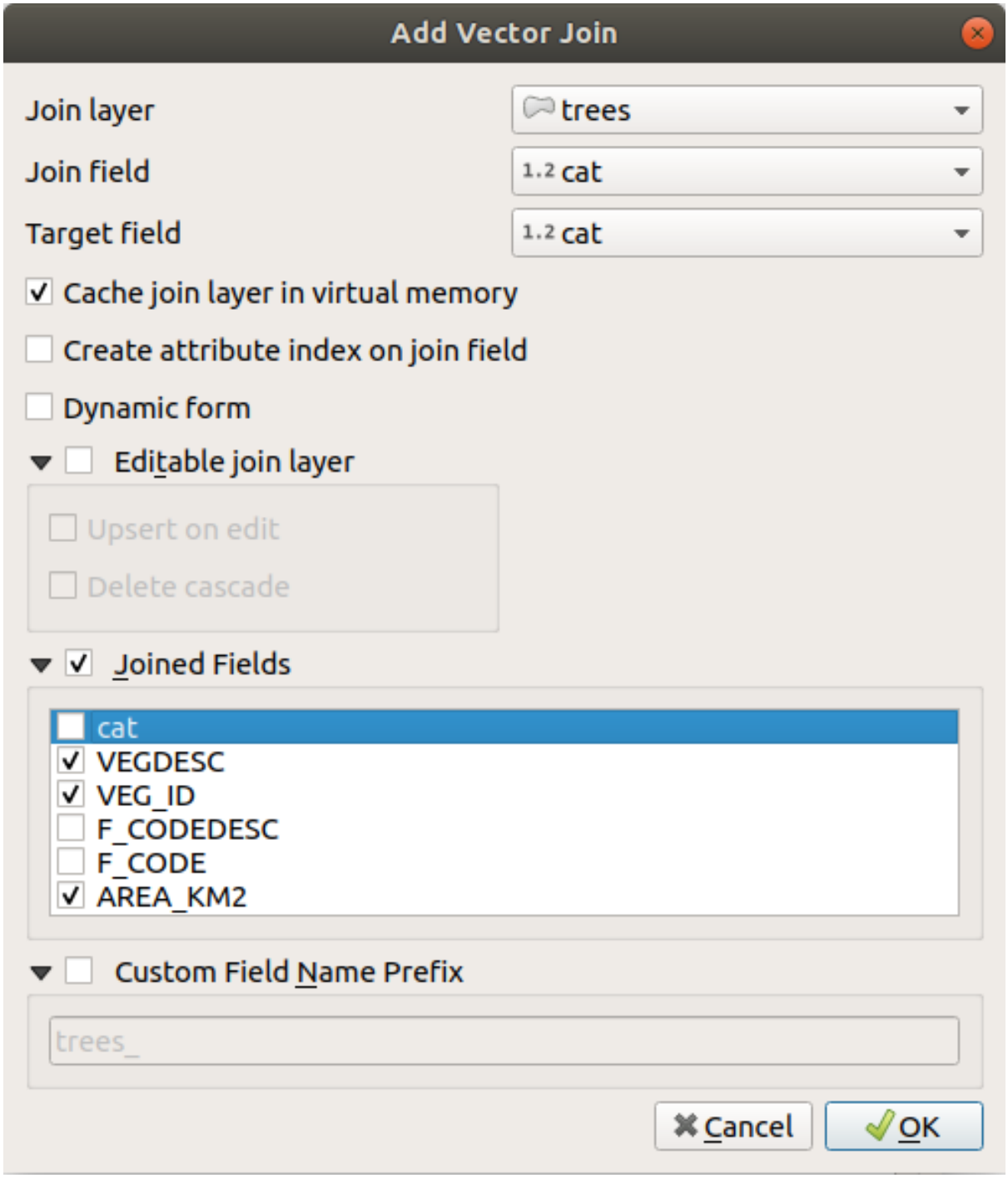


Fig. 12.104: Join an attribute table to an existing vector layer

The steps above will create a join, where **ALL** the attributes of the **first matching feature** in the join layer is added to the target layer's feature. The following logic is used to pair features during a join process:

- All the features in the target layer are returned, regardless they have a match
- If the target field contains duplicate values, these features are assigned the same feature from the join layer.
- If the join field contains duplicate matching values, only the first fetched feature is picked.

Note

Joins in QGIS are based on a single field matching so most of the times, you would want to make sure that values in the matchable fields are unique.

QGIS provides some more options to tweak the join:

- *Cache join layer in virtual memory*: allows you to cache values in memory (without geometries) from the joined layer in order to speed up lookups.
- *Create attribute index on the join field* to speed up lookups
- *Dynamic form*: helps to synchronize join fields on the fly, according to the *Target field*. This way, constraints for join fields are also correctly updated. Note that it's deactivated by default because it may be very time consuming if you have a lot of features or a myriad of joins.
- If the target layer is editable, then some icons will be displayed in the attribute table next to fields, in order to inform about their status:
  - : the join layer is not configured to be editable. If you want to be able to edit join features from the target attribute table, then you have to check the option *Editable join layer*.
  - : the join layer is well configured to be editable, but its current status is read only.
  - : the join layer is editable, but synchronization mechanisms are not activated. If you want to automatically add a feature in the join layer when a feature is created in the target layer, then you have to check the option *Upsert on edit*. Symmetrically, the option *Delete cascade* may be activated if you want to automatically delete join features.
- *Joined fields*: instead of adding all the fields from the joined layer, you can specify a subset.
- *Custom field name prefix* for joined fields, in order to avoid name collision

12.4.2 Setting relations between multiple layers

Unlike joins that define a one-to-one link between features across two layers, relations help you build interconnections between multiple features across two or more layers. As such, relations are project level settings and are set in *Project* ► *Properties* ► *Relations* tab. From there, you can:

- *Add relation* whose type can be:
  - *one to many relation*
  - *many to many relation*
  - *polymorphic relation* that you can add or edit with the dedicated tools in the action drop-down menu.

> **Note**
>
> There is no simple way yet to edit a non-polymorphic relation once it has been created. Only the name can be edited with a double-click. For any other parameters of such a relation you will have to remove and recreate it.

- *Discover relations*: QGIS is able to discover existing relations from supported database formats (PostgreSQL, GeoPackage, ESRI File Geodatabase, ...). This can be a good way to ease the relations definition.
- *Remove relation*

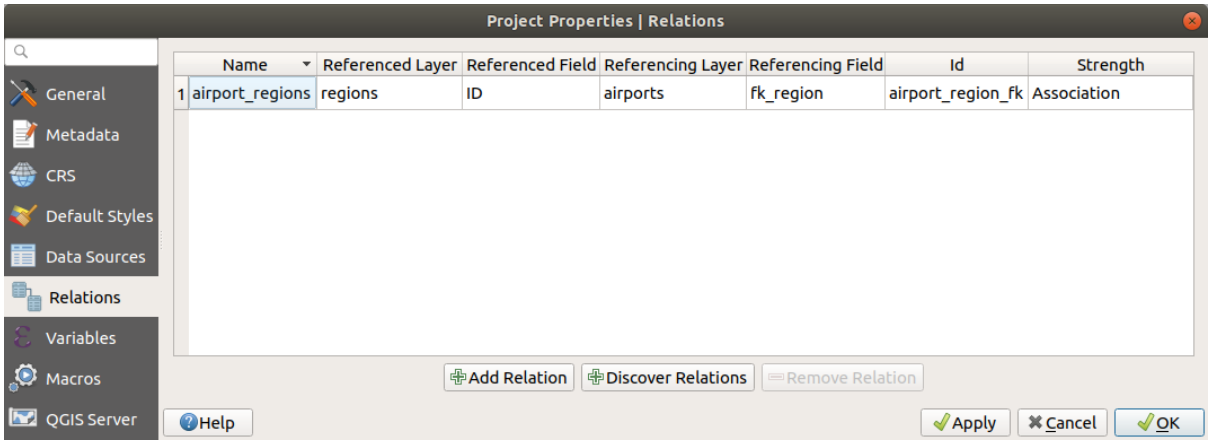


Fig. 12.105: Relations tab

One to many (1-N) relations

As an example you have a layer with all regions of Alaska (polygon) which provides some attributes about its name and region type and a unique id (which acts as primary key).

Then you get another point layer or table with information about airports that are located in the regions and you also want to keep track of these. If you want to add them to the regions layer, you need to create a one to many relation using foreign keys, because there are several airports in most regions.

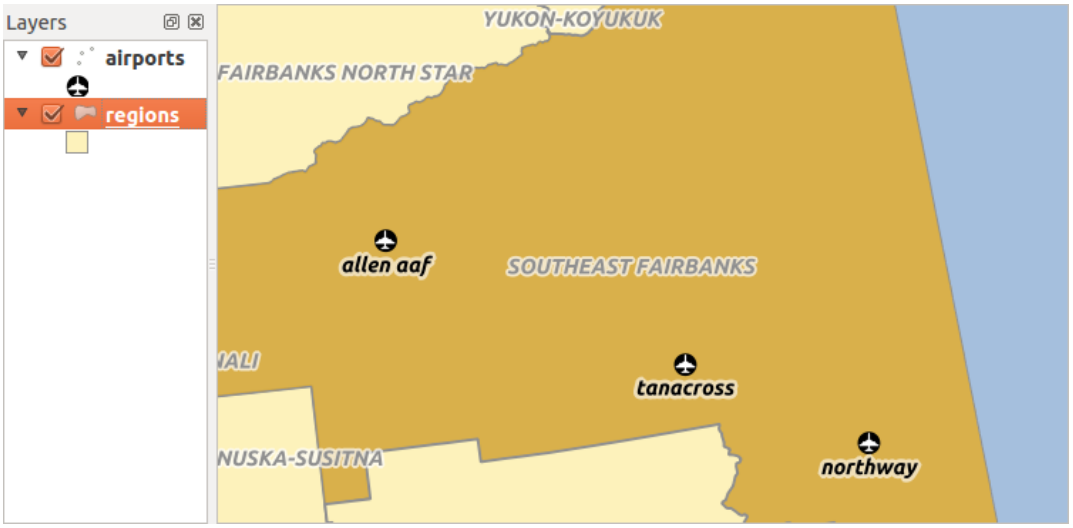


Fig. 12.106: Alaska region with airports

Layers and keys

QGIS makes no difference between a table and a vector layer. Basically, a vector layer is a table with a geometry. So you can add your table as a vector layer. To demonstrate the 1-n relation, you can load the `regions` and `airports` layers in the sample dataset. In practice, each airport belongs to exactly one region while each region can have any number of airports (a typical one to many relation).

In addition to the attributes describing the airports, the aiports layer has another field `fk_region` which acts as a foreign key (if you have a database, you will probably want to define a constraint on it). This `fk_region` field will always contain an id of a region. It can be seen like a pointer to the region it belongs to.

All you have to do is to tell QGIS the relation between the layers so that you can design a custom edit form for editing and QGIS takes care of the setup. It works with different providers (so you can also use it with shape and csv files).

Defining 1-N relations

The first thing we are going to do is to let QGIS know about the relations between the layers. This is done in *Project* ► *Properties...*. Open the *Relations* tab and click on *Add Relation*.

- **Name** is going to be used as a title. It should be a human readable string describing what the relation is used for. We will just call say **airport_relation** in this case.
- **Referenced Layer (Parent)** also considered as parent layer, is the one with the primary key, pointed to, so here it is the `regions` layer. You need to define the primary key of the referenced layer, so it is `ID`.
- **Referencing Layer (Child)** also considered as child layer, is the one with the foreign key field on it. In our case, this is the `airports` layer. For this layer you need to add a referencing field which points to the other layer, so this is `fk_region`.

> **Note**
>
> Sometimes, you need more than a single field to uniquely identify features in a layer. Creating a relation with such a layer requires a **composite key**, i.e. more than a single pair of matching fields. Use the Add new field pair as part of a composite foreign key button to add as many pairs as necessary.

- **Id** will be used for internal purposes and has to be unique. You may need it to build *custom forms*. If you leave it empty, one will be generated for you but you can assign one yourself to get one that is easier to handle
- **Relationship strength** sets the strength of the relation between the parent and the child layer. The default *Association* type means that the parent layer is *simply* linked to the child one while the *Composition* type allows you to duplicate also the child features when duplicating the parent ones and on deleting a feature the children are deleted as well, resulting in cascade over all levels (means children of children of... are deleted as well).

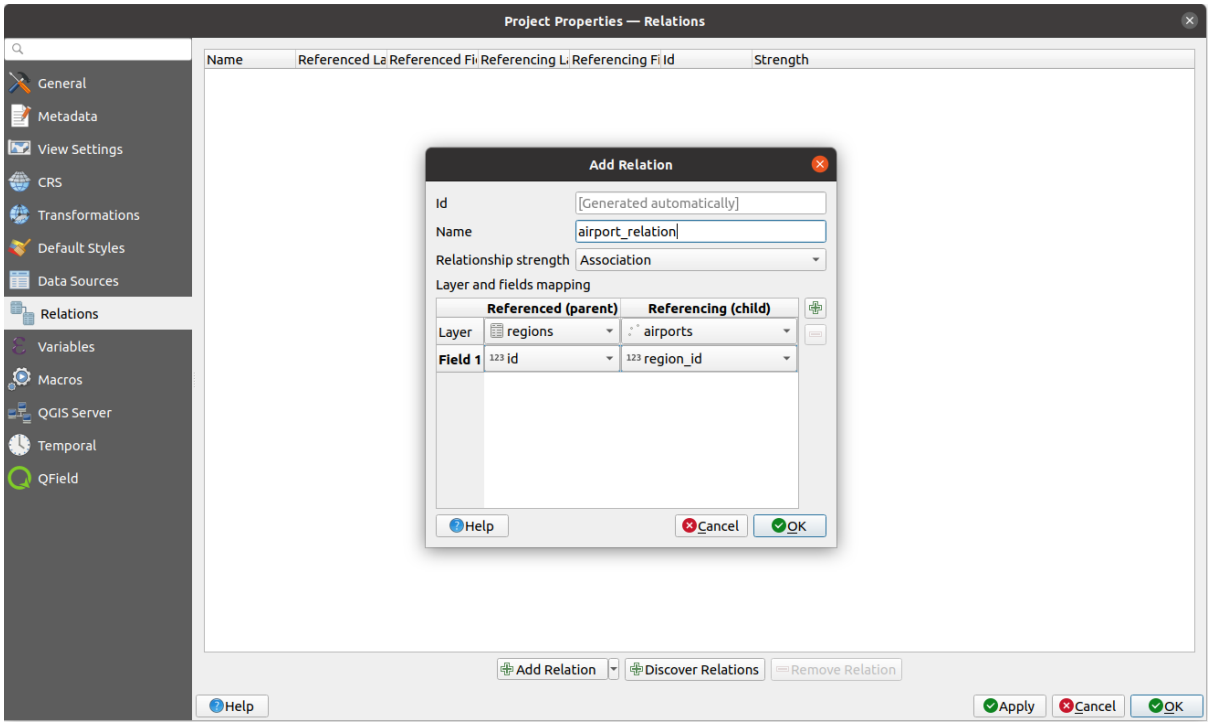


Fig. 12.107: Adding a relation between regions and airports layers

From the *Relations* tab, you can also press the *Discover Relation* button to fetch the relations available from the providers of the loaded layers. This is possible for layers stored in data providers like PostgreSQL or SpatiaLite.

Forms for 1-N relations

Now that QGIS knows about the relation, it will be used to improve the forms it generates. As we did not change the default form method (autogenerated), it will just add a new widget in our form. So let's select the layer region in the legend and use the identify tool. Depending on your settings, the form might open directly or you will have to choose to open it in the identification dialog under actions.

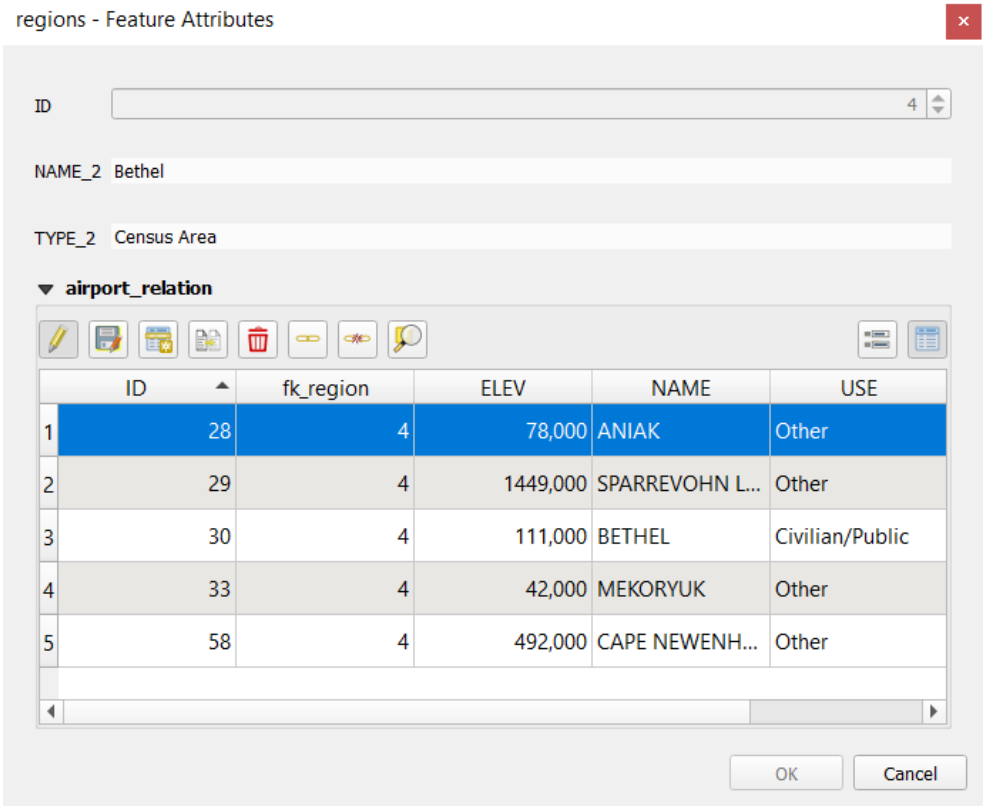


Fig. 12.108: Identification dialog regions with relation to airports

As you can see, the airports assigned to this particular region are all shown in a table. And there are also some buttons available. Let's review them shortly:

- The button is for toggling the edit mode. Be aware that it toggles the edit mode of the airport layer, although we are in the feature form of a feature from the region layer. But the table is representing features of the airport layer.
- The button is for saving all the edits in the child layer (airport).
- The button lets you digitize the airport geometry in the map canvas and assigns the new feature to the current region by default. Note that the icon will change according to the geometry type.
- The button adds a new record to the airport layer attribute table and assigns the new feature to the current region by default. The geometry can be drawn later with the *Add part* digitizing tool.
- The button allows you to copy and paste one or more child features within the child layer. They can later be assigned to a different parent feature or have their attributes modified.
- The button deletes the selected airport(s) permanently.
- The symbol opens a new dialog where you can select any existing airport which will then be assigned to the current region. This may be handy if you created the airport on the wrong region by accident.

- The symbol unlinks the selected airport(s) from the current region, leaving them unassigned (the foreign key is set to NULL) effectively.
- With the button you can zoom the map to the selected child features.
- The two buttons and to the right switch between the *table view and form view* of the related child features.

If you use the *Drag and Drop Designer* for the regions feature, you can select which tools are available. You can even decide whether to open a new form when a new feature is added using *Force hide form on add feature* option. Be aware that this option implies that not null attributes must take a valid default value to work correctly.

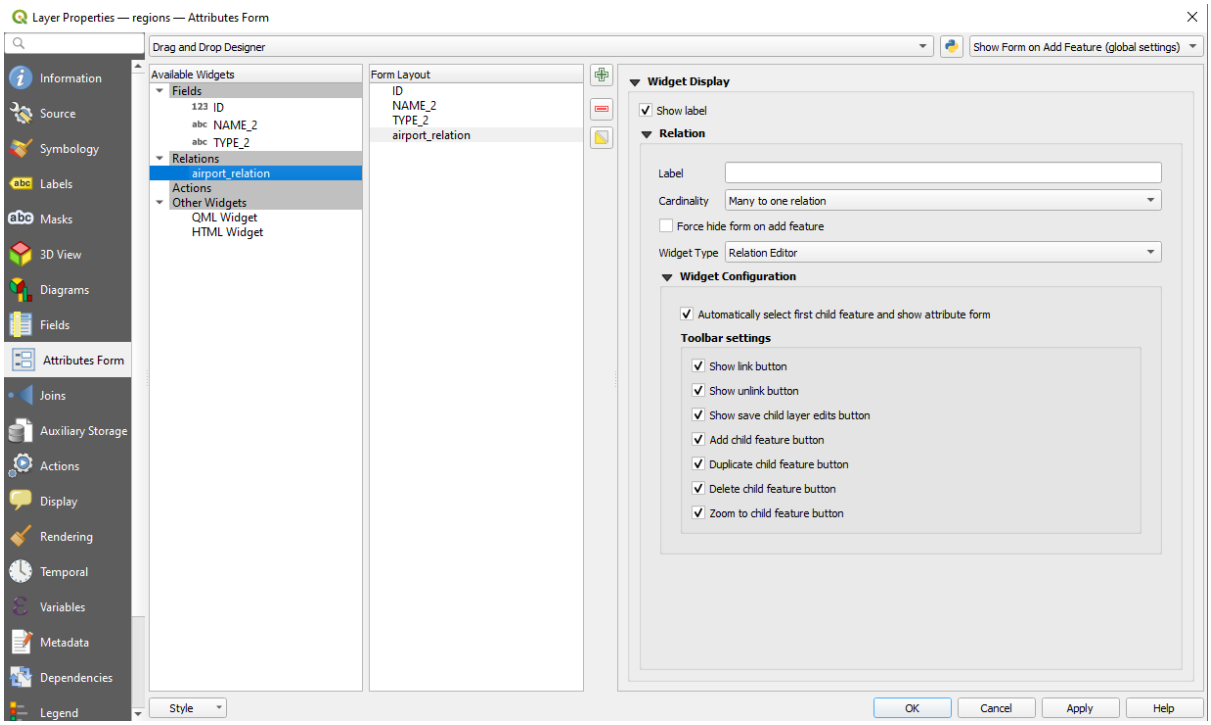


Fig. 12.109: Drag and Drop Designer for configure regions-airports relation tools

In the above example the referencing layer has geometries (so it isn't just an alphanumeric table) so the above steps will create an entry in the layer attribute table that has no corresponding geometric feature. To add the geometry:

1. Choose *Open Attribute Table* for the referencing layer.
2. Select the record that has been added previously within the feature form of the referenced layer.
3. Use the Add Part digitizing tool to attach a geometry to the selected attributes table record.

If you work on the airport table, the widget Relation Reference is automatically set up for the `fk_region` field (the one used to create the relation), see *Relation Reference widget*.

In the airport form you will see the button at the right side of the `fk_region` field: if you click on the button the form of the region layer will be opened. This widget allows you to easily and quickly open the forms of the linked parent features.

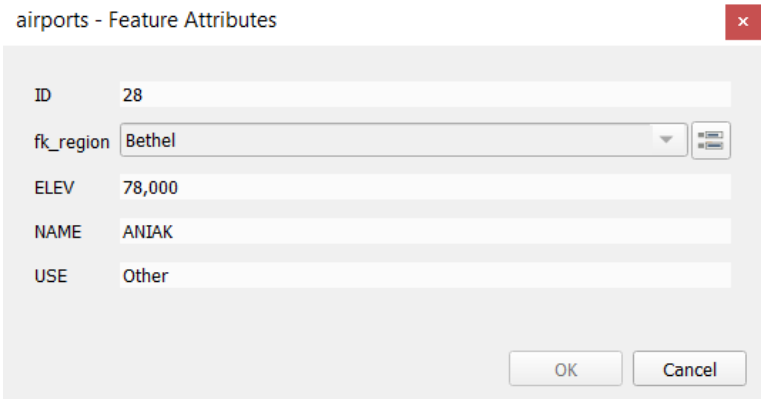
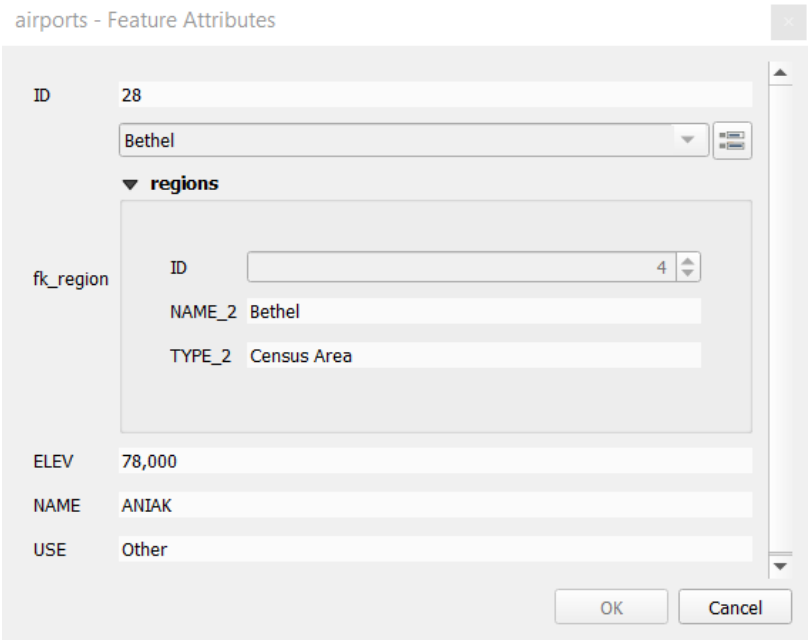


Fig. 12.110: Identification dialog airport with relation to regions

The Relation Reference widget has also an option to embed the form of the parent layer within the child one. It is available in the *Properties* ► *Attributes Form* menu of the airport layer: select the `fk_region` field and check the `Show embedded form` option.

If you look at the feature dialog now, you will see that the form of the region is embedded inside the airports form and will even have a combobox, which allows you to assign the current airport to another region.



Moreover if you toggle the editing mode of the airport layer, the `fk_region` field has also an autocompleter function: while typing you will see all the values of the `id` field of the region layer. Here it is possible to digitize a polygon for the region layer using the button if you chose the option `Allow adding new features` in the *Properties* ► *Attributes Form* menu of the airport layer.

The child layer can also be used in the 按值选择要素 tool in order to select features of the parent layer based on attributes of their children.

In Fig. 12.111, all the regions where the mean altitude of the airports is greater than 500 meters above sea level are selected.

You will find that many different aggregation functions are available in the form.

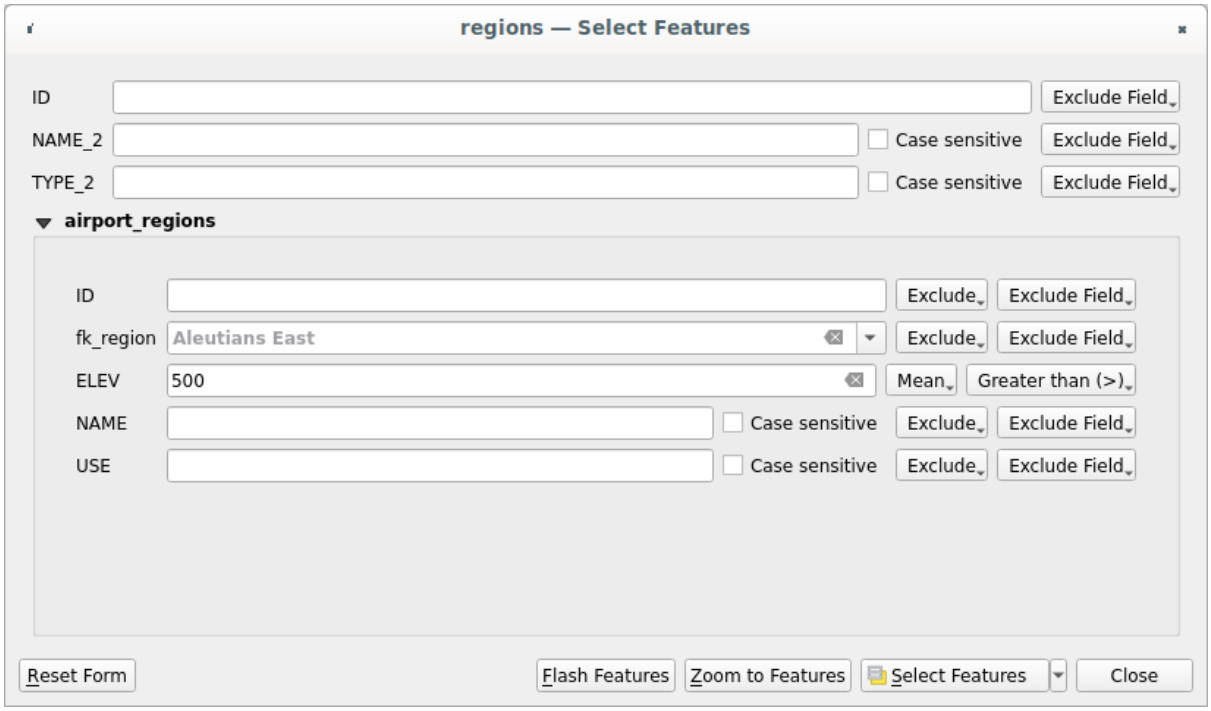


Fig. 12.111: Select parent features with child values

Many-to-many (N-M) relations

N-M relations are many-to-many relations between two tables. For instance, the `airports` and `airlines` layers: an airport receives several airline companies and an airline company flies to several airports.

This SQL code creates the three tables we need for an N-M relationship in a PostgreSQL schema named *locations*. You can run the code using the *Database* ► *DB Manager*⋯ for PostgreSQL or external tools such as pgAdmin. The airports table stores the `airports` layer and the airline table stores the `airlines` layer. In both tables few fields are used for clarity. The *tricky* part is the `airports_airlines` table. We need it to list all airlines for all airports (or vice versa). This kind of table is known as a *pivot table*. The *constraints* in this table force that an airport can be associated with an airline only if both already exist in their layers.

```
CREATE SCHEMA locations;

CREATE TABLE locations.airports
(
   id serial NOT NULL,
   geom geometry(Point, 4326) NOT NULL,
   airport_name text NOT NULL,
   CONSTRAINT airports_pkey PRIMARY KEY (id)
);

CREATE INDEX airports_geom_idx ON locations.airports USING gist (geom);

CREATE TABLE locations.airlines
(
   id serial NOT NULL,
   geom geometry(Point, 4326) NOT NULL,
   airline_name text NOT NULL,
   CONSTRAINT airlines_pkey PRIMARY KEY (id)
);

CREATE INDEX airlines_geom_idx ON locations.airlines USING gist (geom);
```

(continues on next page)

(continued from previous page)

```
CREATE TABLE locations.airports_airlines
(
   id serial NOT NULL,
   airport_fk integer NOT NULL,
   airline_fk integer NOT NULL,
   CONSTRAINT airports_airlines_pkey PRIMARY KEY (id),
   CONSTRAINT airports_airlines_airport_fk_fkey FOREIGN KEY (airport_fk)
      REFERENCES locations.airports (id)
      ON DELETE CASCADE
      ON UPDATE CASCADE
      DEFERRABLE INITIALLY DEFERRED,
   CONSTRAINT airports_airlines_airline_fk_fkey FOREIGN KEY (airline_fk)
      REFERENCES locations.airlines (id)
      ON DELETE CASCADE
      ON UPDATE CASCADE
      DEFERRABLE INITIALLY DEFERRED
 );
```

Instead of PostgreSQL you can also use GeoPackage. In this case, the three tables can be created manually using the *Database* ► *DB Manager*···. In GeoPackage there are no schemas so the *locations* prefix is not needed.

Foreign key constraints in `airports_airlines` table can´t be created using *Table* ► *Create Table*··· or *Table* ► *Edit Table*··· so they should be created using *Database* ► *SQL Window*···. GeoPackage doesn't support *ADD CONSTRAINT* statements so the `airports_airlines` table should be created in two steps:

1. Set up the table only with the `id` field using *Table* ► *Create Table*···
2. Using *Database* ► *SQL Window*···, type and execute this SQL code:

```
ALTER TABLE airports_airlines
   ADD COLUMN airport_fk INTEGER
   REFERENCES airports (id)
   ON DELETE CASCADE
   ON UPDATE CASCADE
   DEFERRABLE INITIALLY DEFERRED;

ALTER TABLE airports_airlines
   ADD COLUMN airline_fk INTEGER
   REFERENCES airlines (id)
   ON DELETE CASCADE
   ON UPDATE CASCADE
   DEFERRABLE INITIALLY DEFERRED;
```

Then in QGIS, you should set up two *one-to-many relations* as explained above:

- a relation between `airlines` table and the pivot table;
- and a second one between `airports` table and the pivot table.

An easier way to do it (only for PostgreSQL) is using the *Discover Relations* in *Project* ► *Properties* ► *Relations*. QGIS will automatically read all relations in your database and you only have to select the two you need. Remember to load the three tables in the QGIS project first.

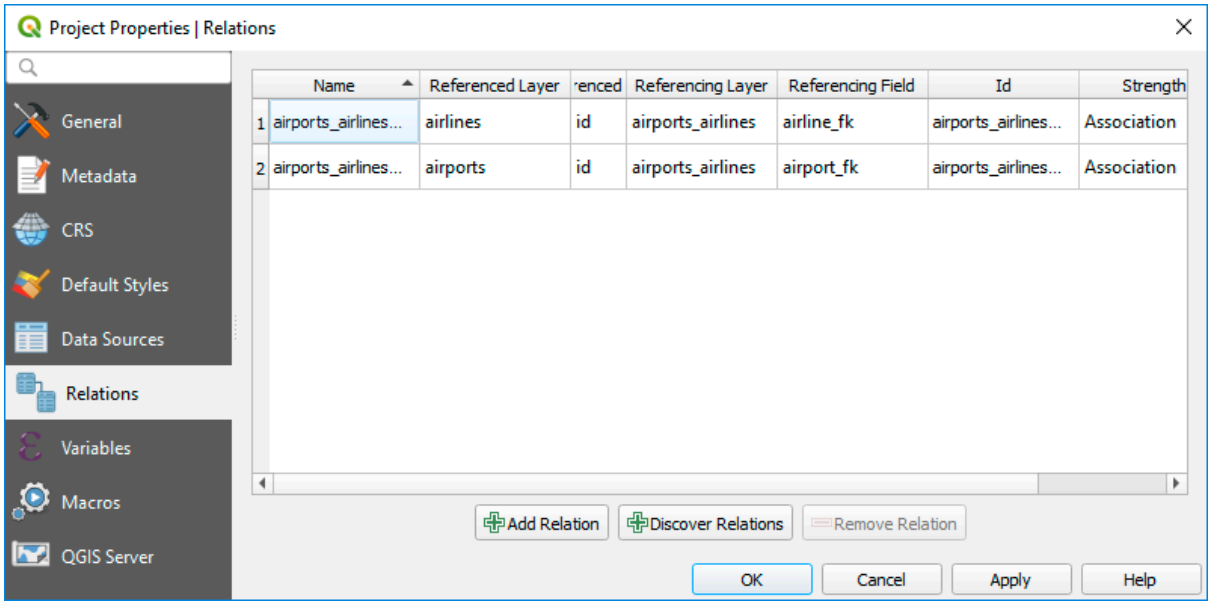


Fig. 12.112: Relations and autodiscover

In case you want to remove an `airport` or an `airline`, QGIS won't remove the associated record(s) in `airports_airlines` table. This task will be made by the database if we specify the right *constraints* in the pivot table creation as in the current example.

Note

Combining N-M relation with automatic transaction group

You should enable the transaction mode in *Project Properties* ► *Data Sources* ► when working on such context. QGIS should be able to add or update row(s) in all tables (airlines, airports and the pivot tables).

Finally we have to select the right cardinality in the *Layer Properties* ► *Attributes Form* for the `airports` and `airlines` layers. For the first one we should choose the **airlines (id)** option and for the second one the **airports (id)** option.

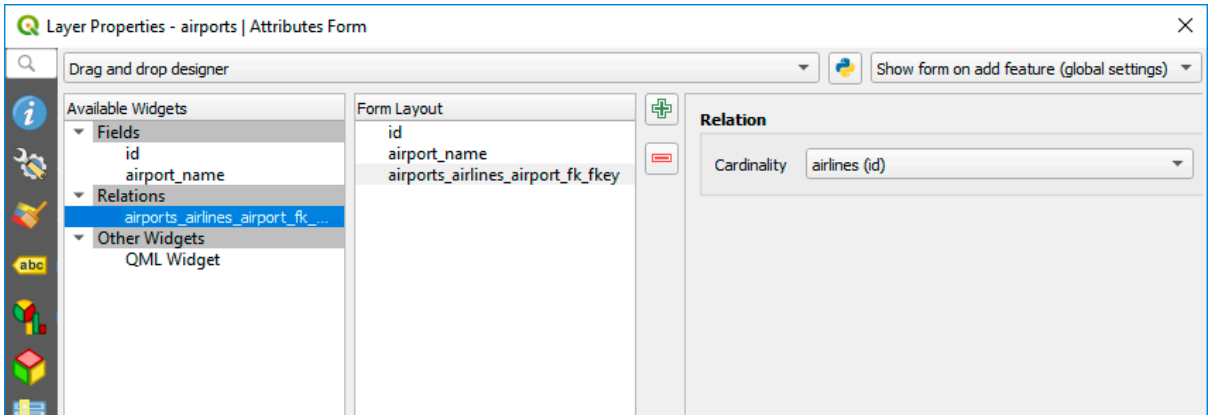


Fig. 12.113: Set relationship cardinality

Now you can associate an airport with an airline (or an airline with an airport) using *Add child feature* or *Link existing child feature* in the subforms. A record will automatically be inserted in the `airports_airlines` table.

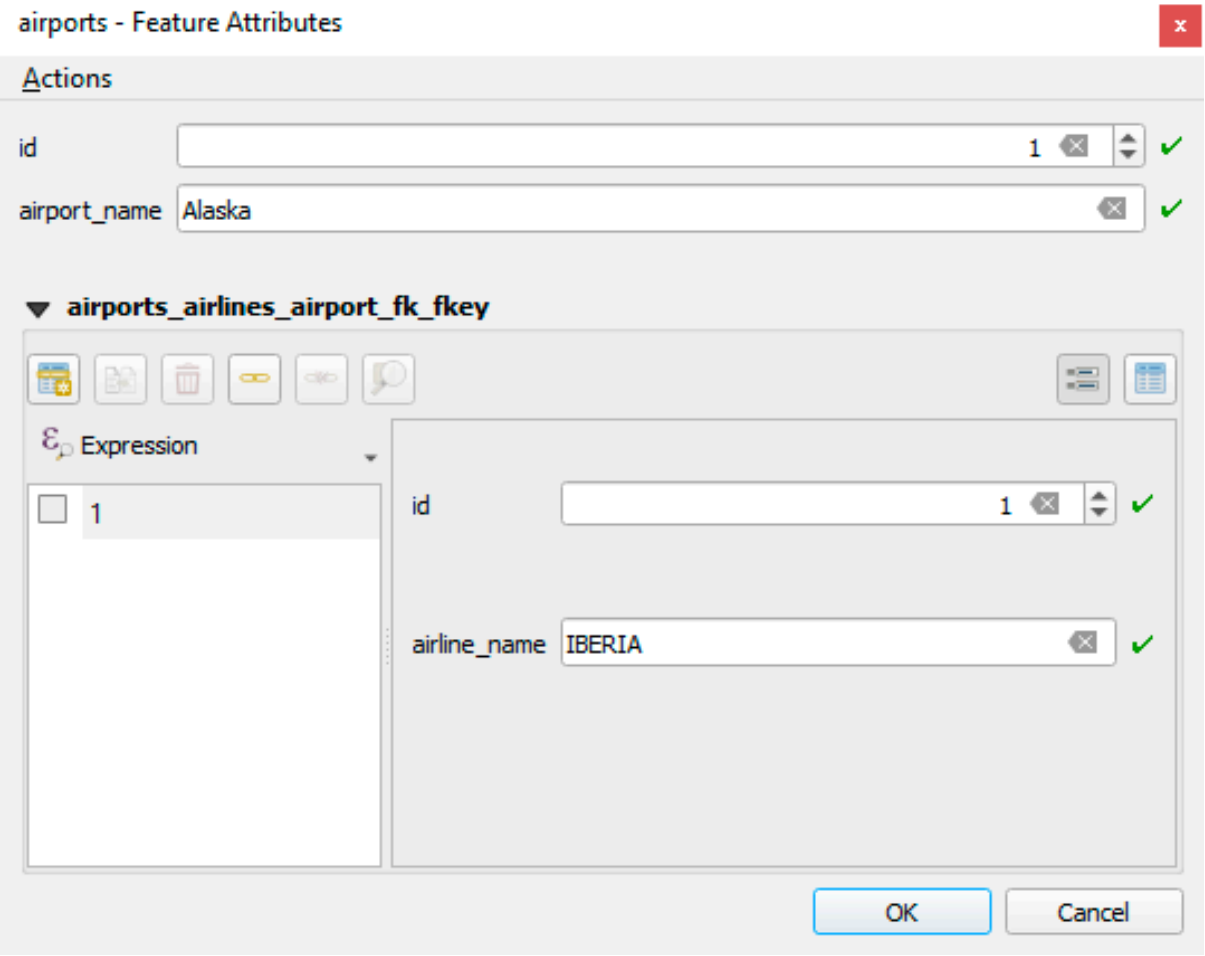


Fig. 12.114: N-M relationship between airports and airlines

Note

Using **Many to one relation** cardinality

Sometimes hiding the pivot table in an N-M relationship is not desirable. Mainly because there are attributes in the relationship that can only have values when a relationship is established. If your tables have a geometry field, it could be interesting to activate the *On map identification* option (*Layer Properties* ► *Attributes Form* ► *Available widgets* ► *Fields*) for the foreign key fields in the pivot table.

Note

Pivot table primary key

Avoid using multiple fields in the primary key in a pivot table. QGIS assumes a single primary key so a constraint like `constraint airports_airlines_pkey primary key (airport_fk, airline_fk)` will not work.

Polymorphic relations

The purpose

Polymorphic relations are special case of 1-N relations, where a single referencing (document) layer contains the features for multiple referenced layers. This differs from normal relations which require different referencing layer for each referenced layer. A single referencing (document) layer is achieved by adding an adiditonal `layer_field` column in the referencing (document) layer that stores information to identify the referenced layer. In its most simple form, the referencing (document) layer will just insert the layer name of the referenced layer into this field.

To be more precise, a polymorphic relation is a set of normal relations having the same referencing layer but having the referenced layer dynamically defined. The polymorphic setting of the layer is solved by using an expression which has to match some properties of the referenced layer like the table name, layer id, layer name.

Imagine we are going to the park and want to take pictures of different species of `plants` and `animals` we see there. Each plant or animal has multiple pictures associated with it, so if we use the normal 1:N relations to store pictures, we would need two separate tables, `animal_images` and `plant_images`. This might not be a problem for 2 tables, but imagine if we want to take separate pictures for mushrooms, birds etc.

Polymorphic relations solve this problem as all the referencing features are stored in the same table `documents`. For each feature the referenced layer is stored in the `referenced_layer` field and the referenced feature id in the `referenced_id` field.

Defining polymorphic relations

First, let QGIS know about the polymorphic relations between the layers. This is done in *Project* ► *Properties...*.

Open the *Relations* tab and click on the little down arrow next to the *Add Relation* button, so you can select the *Add Polymorphic Relation* option from the newly appeared dropdown.

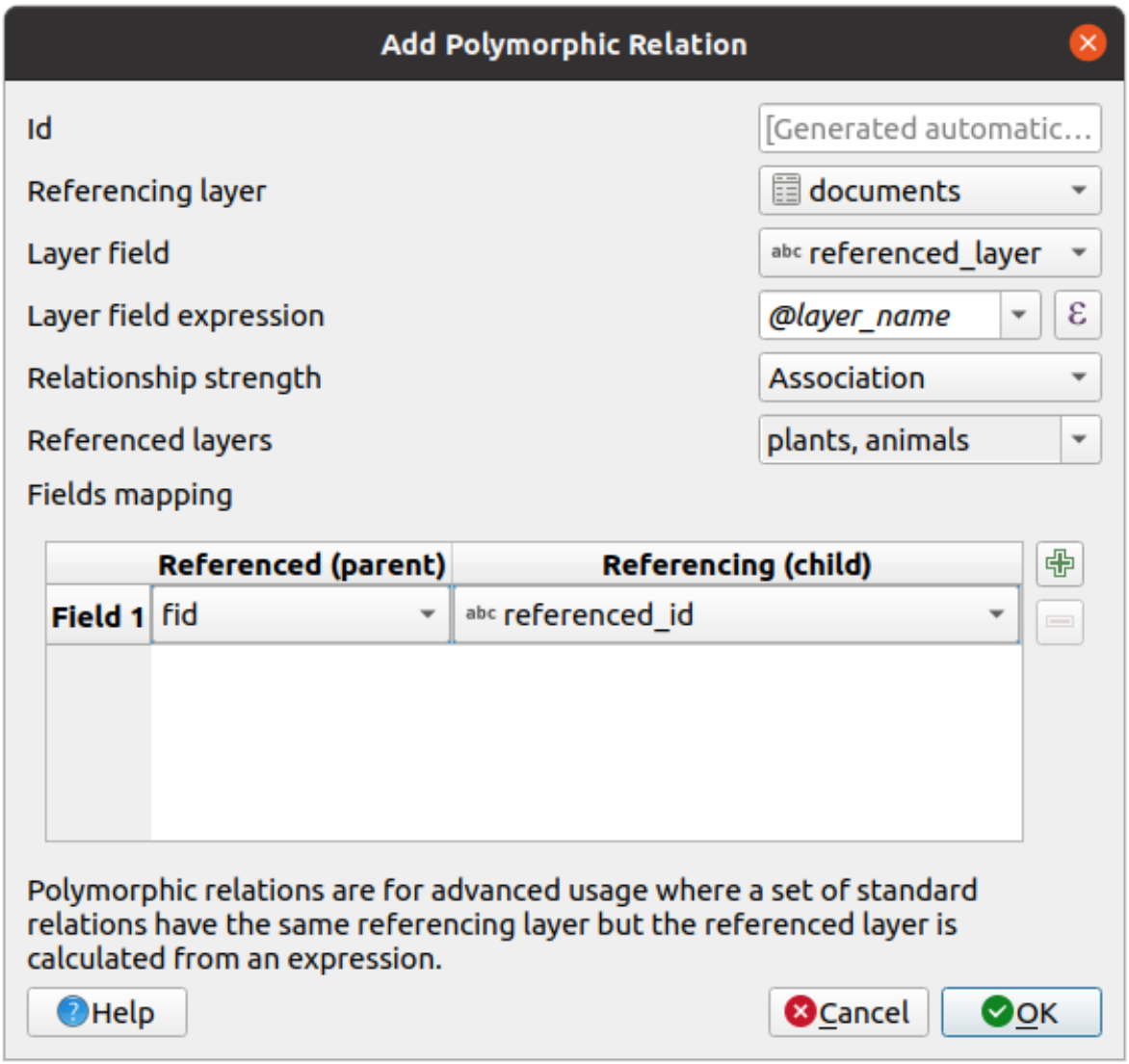


Fig. 12.115: Adding a polymorphic relation using `documents` layer as referencing and `animals` and `plants` as referenced layers.

- **Id** will be used for internal purposes and has to be unique. You may need it to build *custom forms*. If you leave it empty, one will be generated for you but you can assign one yourself to get one that is easier to handle
- **Referencing Layer (Child)** also considered as child layer, is the one with the foreign key field on it. In our case, this is the `documents` layer. For this layer you need to add a referencing field which points to the other layer, so this is `referenced_id`.

> **Note**
>
> Sometimes, you need more than a single field to uniquely identify features in a layer. Creating a relation with such a layer requires a **composite key**, i.e. more than a single pair of matching fields. Use the Add new field pair as part of a composite foreign key button to add as many pairs as necessary.

- **Layer Field** is the field in the referencing table that stores the result of the evaluated layer expression which is the referencing table that this feature belongs to. In our example, this would be the `referenced_layer` field.
- **Layer expression** evaluates to a unique identifier of the layer. This can be the layer name `@layer_name`,

the layer id `@layer_id`, the layer's table name `decode_uri(@layer, 'table')` or anything that can uniquely identifies a layer.

- **Relationship strength** sets the strength of the generated relations between the parent and the child layer. The default *Association* type means that the parent layer is *simply* linked to the child one while the *Composition* type allows you to duplicate also the child features when duplicating the parent ones and on deleting a feature the children are deleted as well, resulting in cascade over all levels (means children of children of... are deleted as well).
- **Referenced Layers** also considered as parent layers, are those with the primary key, pointed to, so here they would be `plants` and `animals` layers. You need to define the primary key of the referenced layers from the dropdown, so it is `fid`. Note that the definition of a valid primary key requires all the referenced layers to have a field with that name. If there is no such field you cannot save a polymorphic relation.

Once added, the polymorphic relation can be edited via the *Edit Polymorphic Relation* menu entry.

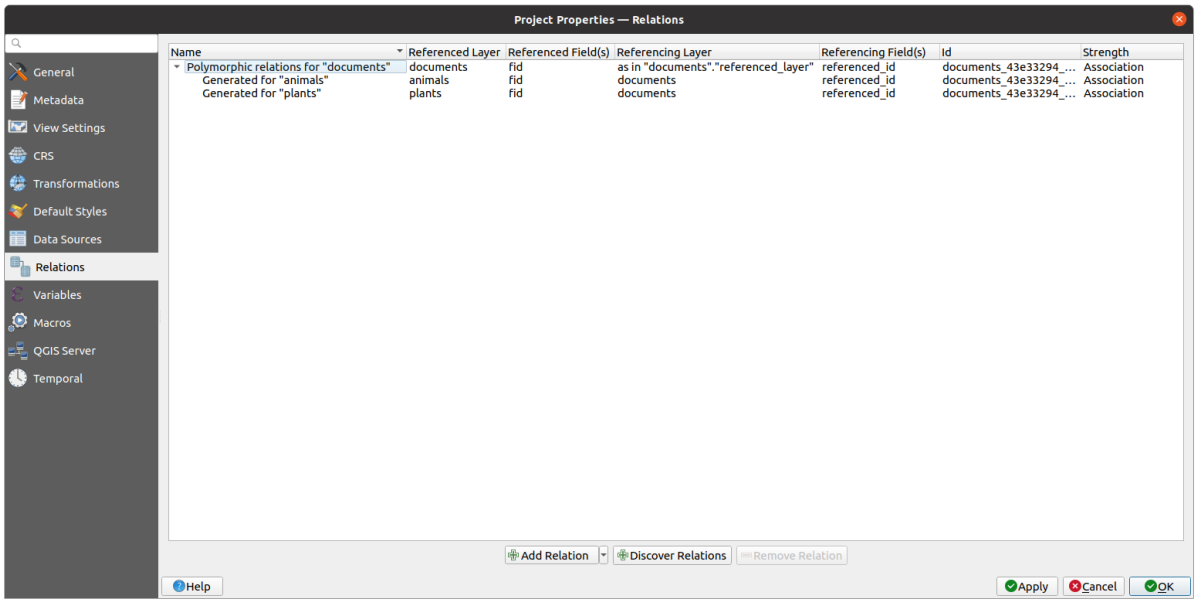


Fig. 12.116: Preview of the newly created polymorphic relation and its child relations for animals and plants.

The example above uses the following database schema:

```
CREATE SCHEMA park;

CREATE TABLE park.animals
(
   fid serial NOT NULL,
   geom geometry(Point, 4326) NOT NULL,
   animal_species text NOT NULL,
   CONSTRAINT animals_pkey PRIMARY KEY (fid)
);

CREATE INDEX animals_geom_idx ON park.animals USING gist (geom);

CREATE TABLE park.plants
(
   fid serial NOT NULL,
   geom geometry(Point, 4326) NOT NULL,
   plant_species text NOT NULL,
   CONSTRAINT plants_pkey PRIMARY KEY (fid)
);
```

(continues on next page)

(continued from previous page)

```
CREATE INDEX plants_geom_idx ON park.plants USING gist (geom);

CREATE TABLE park.documents
(
   fid serial NOT NULL,
   referenced_layer text NOT NULL,
   referenced_id integer NOT NULL,
   image_filename text NOT NULL,
   CONSTRAINT documents_pkey PRIMARY KEY (fid)
);
```

Once relations are established, the *Identify Features* result panel will display related features, allowing you to view associated data directly.

Dataset-stored Relationships

Other than the above mentioned relationships that are created in the project properties and apply only in the active project, QGIS can also read database-level relations, i.e., connections defined between layers in the dataset. This is supported for database providers such as PostgreSQL, SpatiaLite, GeoPackage, ESRI FileGeodatabase. When such parent and child layers are loaded in a project, QGIS can automatically detect the relation settings and fill appropriately the *Project* ► *Project properties* ► *Relations* tab. This can also be triggered using the *Discover relations* button.

For some data types (e.g., GeoPackage, SpatiaLite or ESRI FileGeodatabase), QGIS also allows to create and edit the relations between the tables in the database. This can be managed through the *Browser Panel*. From the *Browser panel*, right-click on the database entry, select *New relationship...*, and in the new dialog, fill the following options:

- *Name*: set the name of the new relationship.
- *Cardinality*: define the cardinality of the relationship.
- *Strength*: specify the strength of the relationship.
- Under the *Tables* group, set your *Parent* and *Child* tables.
- Under the *Fields* group, set your *Parent field* and *Child field*.
- Under the *Advanced* section, depending on the database type, you can set:
  - *Related table type*:
    - *features*: for standard feature tables.
    - *media*: for tables storing media files.
    - *simple_attributes*: for tables with simple attribute data.
    - *attributes*: for tables with complex attribute data.
    - *tiles*: for tables storing tile data.
  - *Forward label*: a name to uniquely identify the relationship when navigating from the parent table to the child table.
  - *Reverse label*: a name to uniquely identify the relationship when navigating from the child table to the parent table.

Applying the dialog will add a *Relationships* node under the database entry, with the new relationship as child item.

Editing or deleting existing relationships can be done by right-clicking on the relationship in the Browser Panel.

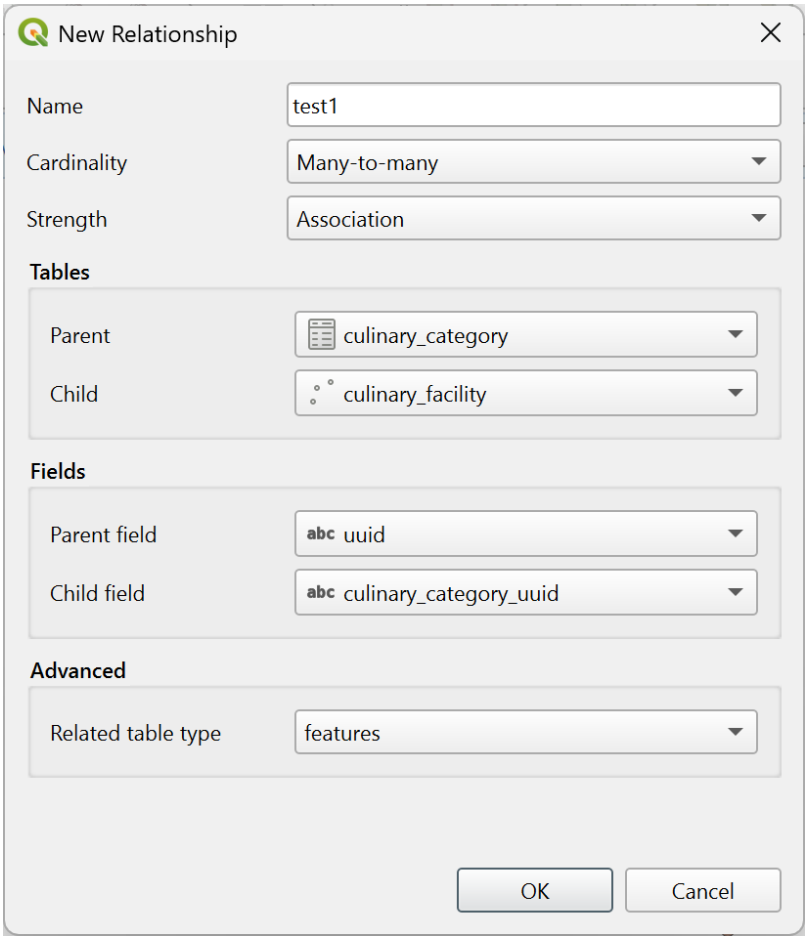


Fig. 12.117: Managing dataset-stored relationships in the Browser Panel

12.4.3 Storing and fetching an external resource

A field may target a resource stored on an external storage system. Attribute forms can be configured so they act as a client to an external storage system in order to store and fetch those resources, on users demand, directly from the forms.

Configuring an external storage

In order to setup an external storage, you have to first configure it from the vector *attribute form properties* and select the *Attachment* widget.

Fig. 12.118: Editing a WebDAV external storage for a given field

From the *Attachment* widget, you have to first select the *Storage type*:

- *Select Existing File*: The target URL already exists. When you select a resource, no store operation is achieved, the attribute is simply updated with the URL.
- *Simple Copy*: Stores a copy of the resource on a file disk destination (which could be a local or network shared file system) and the attribute is updated with the path to the copy.
- *WebDAV Storage*: The resource is pushed to a HTTP server supporting the WebDAV protocol and the attribute is updated with its URL. Nextcloud, Pydio or other file hosting software support this protocol.
- *AWS S3*: The resource is pushed to a server supporting AWS Simple Storage Service protocol and the attribute is updated with its URL. Amazon Web Service and MinIO hosting software support this protocol.

Then, you have to set up the *Store URL* parameter, which provides the URL to be used when a new resource needs to be stored. It's possible to set up an expression using the *data defined override widget* in order to have specific values according to feature attributes.

The variable **@selected_file_path** could be used in that expression and represent the absolute file path of the user selected file (using the file selector or drag'n drop).

Note

Using the **WebDAV** or **AWS S3** external storage, if the URL ends with a "/", it is considered as a folder and the selected file name will be appended to get the final URL.

If the external storage system needs to, it's possible to configure an *authentication*.

Note

To use the **AWS S3** external storage, you must use an **AWS S3** authentication type.

Using an external storage

Once configured, you can select a local file using the button ... when editing a feature's attribute. Depending on the configured *storage type*, the file will be stored on the external storage system (except if *Select existing file* has been selected) and the field will be updated with the new resource URL.

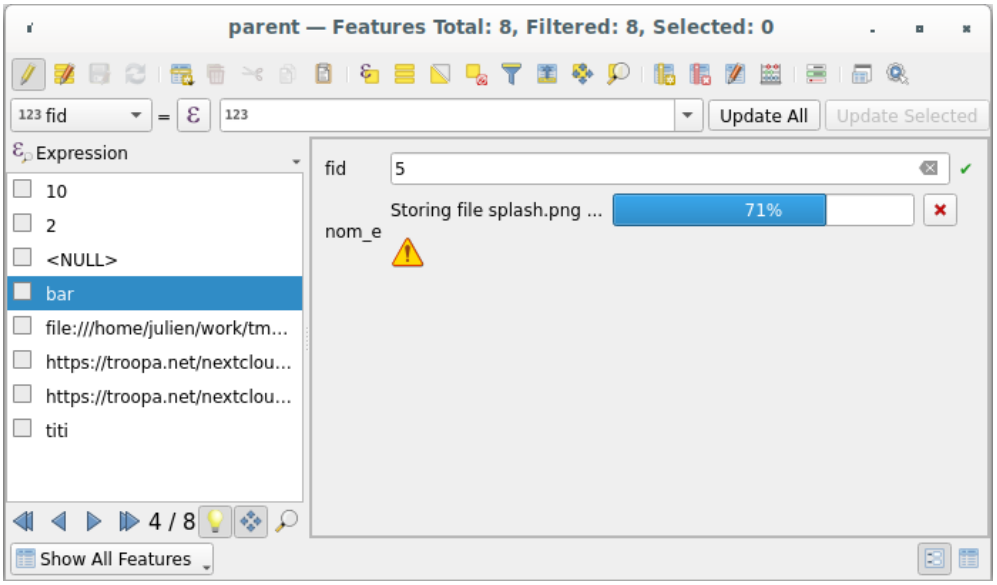


Fig. 12.119: Storing a file to a WebDAV external storage

> **Note**
>
> User can also achieve the same result if he drags and drops a file on the whole attachment widget.

Use the Cancel button to abort the storing process. It's possible to configure a viewer using the *Integrated document viewer* so the resource will be automatically fetched from the external storage system and displayed directly below the URL. The above icon indicates that the resource cannot be fetched from the external storage system. In that case, more details might appear in the 日志消息面板.

CHAPTER 13

使用栅格数据

13.1 Raster Properties Dialog

Raster data is made up of pixels (or cells), and each pixel has a value. It is commonly used to store various types of data, including:

- Imagery, such as satellite images, digital aerial photographs, scanned maps
- Elevation data, such as digital elevation models (DEMs), digital terrain models (DTMs)
- Other types of data, such as land cover, soil types, rainfall and many others.

Raster data can be stored in several supported formats, including GeoTIFF, ERDAS Imagine, ArcInfo ASCII GRID, PostgreSQL Raster and others. See more at 打开数据.

To view and set the properties for a raster layer, double click on the layer name in the map legend, or right click on the layer name and choose *Properties* from the context menu. This will open the *Raster Layer Properties* dialog.

There are several tabs in the dialog:

Information	*Source*	*Symbology* ([1])
Transparency ([1])	*Labels* ([1])	*Histogram* ([1])
Rendering	*Temporal*	*Pyramids*
Elevation	*Metadata*	*Legend*
Display	*Attribute Tables*	*QGIS Server*
External plugins ([2]) tabs		

[1] Also available in the *Layer styling panel*
[2] *External plugins* you install can optionally add tabs to this dialog. Those are not presented in this document. Refer to their documentation.

> **Tip**
>
> **Live update rendering**
>
> The 图层样式面板 provides you with some of the common features of the Layer properties dialog and is a good modeless widget that you can use to speed up the configuration of the layer styles and view your changes on the map canvas.

> **Note**
>
> Because properties (symbology, label, actions, default values, forms...) of embedded layers (see 从外部工程嵌入图层) are pulled from the original project file, and to avoid changes that may break this behavior, the layer properties dialog is made unavailable for these layers.

13.1.1 信息属性

The *Information* tab is read-only and represents an interesting place to quickly grab summarized information and metadata for the current layer. Provided information are:

- 常规：例如工程中的名称、源路径、辅助文件列表、上次保存的时间和大小、所使用的提供程序
- custom properties, used to store in the active project additional information about the layer. Default custom properties include *Identify/format*, which influences how the results from using the *Identify features* tool over a raster layer are formatted. More properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- based on the provider of the layer: extent, width and height, data type, GDAL driver, bands statistics
- 坐标参照系：名称、单位、方法、精度、参照 (即静态还是动态)
- read from layer properties: data type, extent, width/height, compression, pixel size, statistics on bands, number of columns, rows and no-data values of the raster...
- picked from the *filled metadata*: access, extents, links, contacts, history...

13.1.2 源属性

The *Source* tab displays basic information about the selected raster, including:

- 图层名显示在:guilabel:'图层面板 '中；
- the *Coordinate Reference System*: Displays the layer's *Coordinate Reference System (CRS)*. You can change the layer's CRS, by selecting a recently used one in the drop-down list or clicking on the Select CRS button (see 坐标参照系选择器). Use this process only if the layer CRS is a wrong or not specified. If you wish to reproject your data, use a reprojection algorithm from Processing or *Save it as new dataset*.
- Depending on the data provider, a *Layer source* group indicates the path to the source of the dataset and allows for replacing the loaded layer:
  - When the layer is stored as file on disk, edit the path shown in the text box or press ... Browse to select another file on the disk
  - When the layer is provided by an ArcGIS MapServer service, it is possible to modify its *authentication settings*, while keeping unchanged the details for connecting to the service.

– For an XYZ remote layer, it is possible to modify at the layer level any of its *connection details* (URL, authentication settings, zoom levels, resolution, ...). Modifications are applied to the layer without altering the original connection settings.

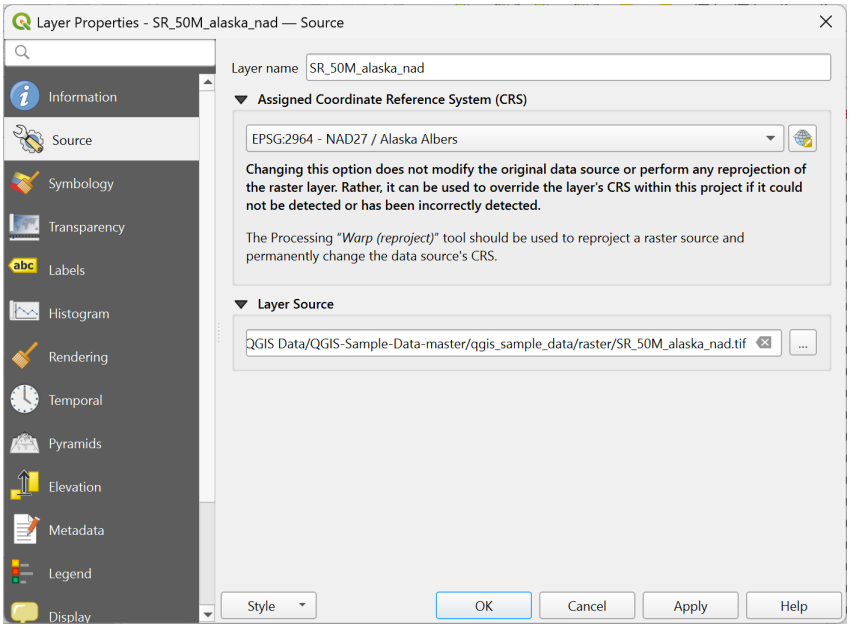


Fig. 13.1: Raster Layer Properties - Source Dialog

13.1.3 Symbology Properties

The raster layer symbology tab is made of three different sections:

- The *Band rendering* where you can control the renderer type to use
- The *Layer rendering* to apply effects on rendered data
- The *Resampling* methods to optimize rendering on map

Band rendering

QGIS offers many different *Render types*. The choice of renderer depends on the data type and the information you'd like to highlight.

1. *Multiband color* - if the file comes with several bands (e.g. a satellite image with several bands).
2. *Paletted/Unique values* - for single band files that come with an indexed palette (e.g. a digital topographic map) or for general use of palettes for rendering raster layers.
3. *Singleband gray* - (one band of) the image will be rendered as gray. QGIS will choose this renderer if the file is neither multiband nor paletted (e.g. a shaded relief map).
4. *Singleband pseudocolor* - this renderer can be used for files with a continuous palette or color map (e.g. an elevation map).
5. *Single color* - the raster layer will be rendered with a single color.
6. *Hillshade* - Creates hillshade from a band.
7. *Contours* - Generates contours on the fly for a source raster band.

Multiband color

With the multiband color renderer, three selected bands from the image will be used as the red, green or blue component of the color image. QGIS automatically fetches *Min* and *Max* values for each band of the raster and scales the coloring accordingly. You can control the value ranges in the *Min/Max Value Settings* section.

A *Contrast enhancement* method can be applied to the values: 'No enhancement', 'Stretch to MinMax', 'Stretch and clip to MinMax' and 'Clip to min max'.

Note

Contrast enhancement

When adding GRASS rasters, the option *Contrast enhancement* will always be set automatically to *stretch to min max*, even if this is set to another value in the QGIS general options.

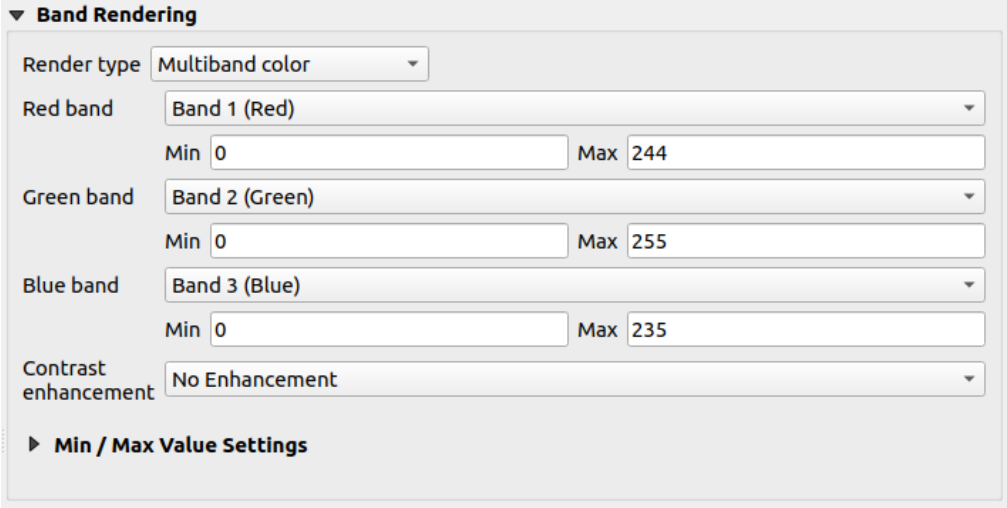


Fig. 13.2: Raster Symbology - Multiband color rendering

Tip

Viewing a Single Band of a Multiband Raster

If you want to view a single band of a multiband image (for example, Red), you might think you would set the Green and Blue bands to *Not Set*. But the preferred way of doing this is to set the image type to *Singleband gray*, and then select Red as the *Gray band* to use.

Paletted/Unique values

This is the standard render option for singleband files that include a color table, where a certain color is assigned to each pixel value. In that case, the palette is rendered automatically.

It can be used for all kinds of raster bands, assigning a color to each unique raster value.

If you want to change a color, just double-click on the color and the *Select color* dialog appears.

It is also possible to assign labels to the colors. The label will then appear in the legend of the raster layer.

Right-clicking over selected rows in the color table shows a contextual menu to:

- *Change Color...* for the selection

- *Change Opacity...* for the selection
- *Change Label...* for the selection

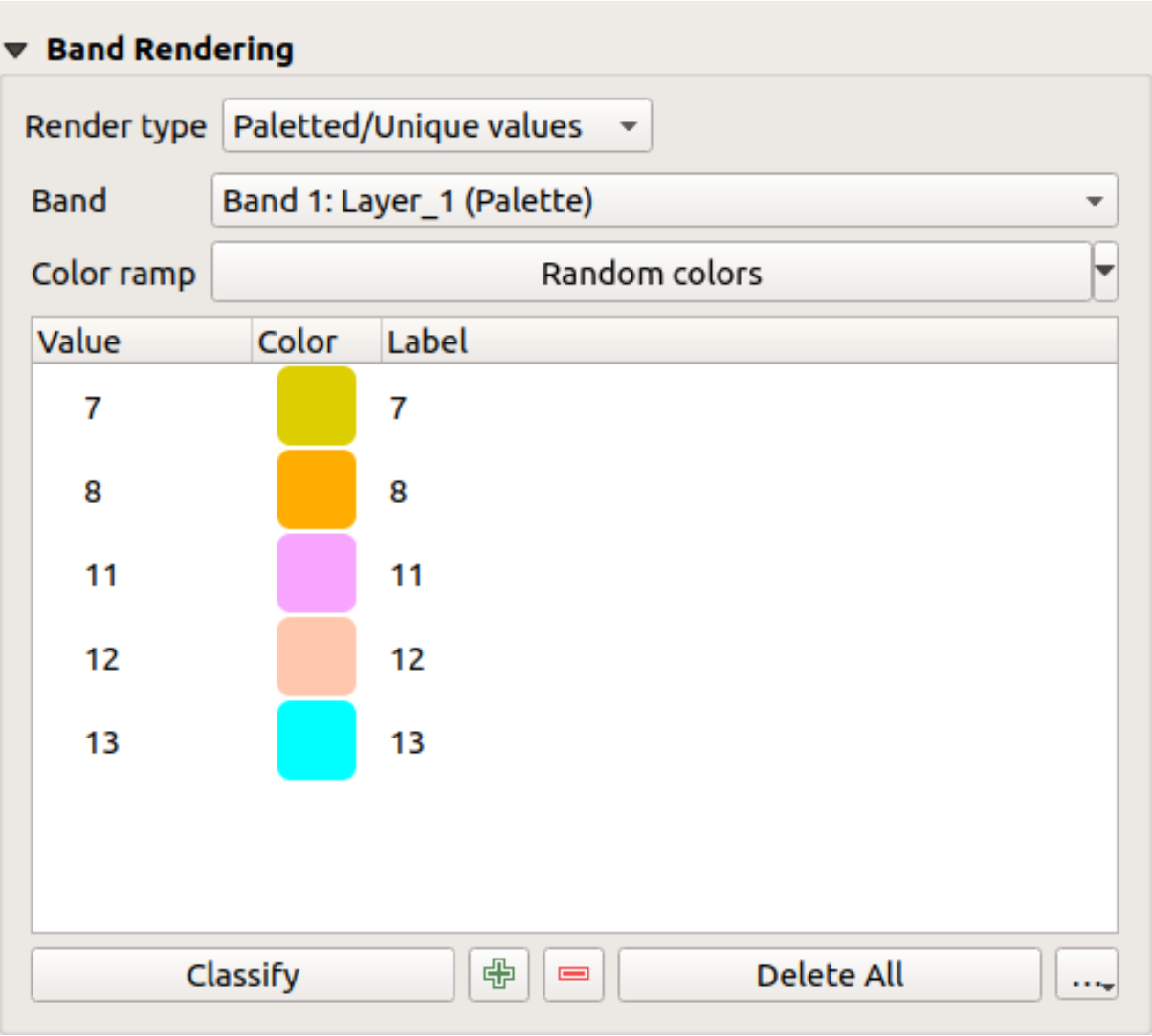


Fig. 13.3: Raster Symbology - Paletted unique value rendering

The pulldown menu, that opens when clicking the ... (Advanced options) button below the color map to the right, offers color map loading (*Load Color Map from File...*) and exporting (*Export Color Map to File...*), and loading of classes (*Load Classes from Layer*).

Singleband gray

This renderer allows you to render a layer using only one band with a *Color gradient*: 'Black to white' or 'White to black'. You can change the range of values to color (*Min* and *Max*) in the *Min/Max Value Settings*.

A *Contrast enhancement* method can be applied to the values: 'No enhancement', 'Stretch to MinMax', 'Stretch and clip to MinMax' and 'Clip to min max'.

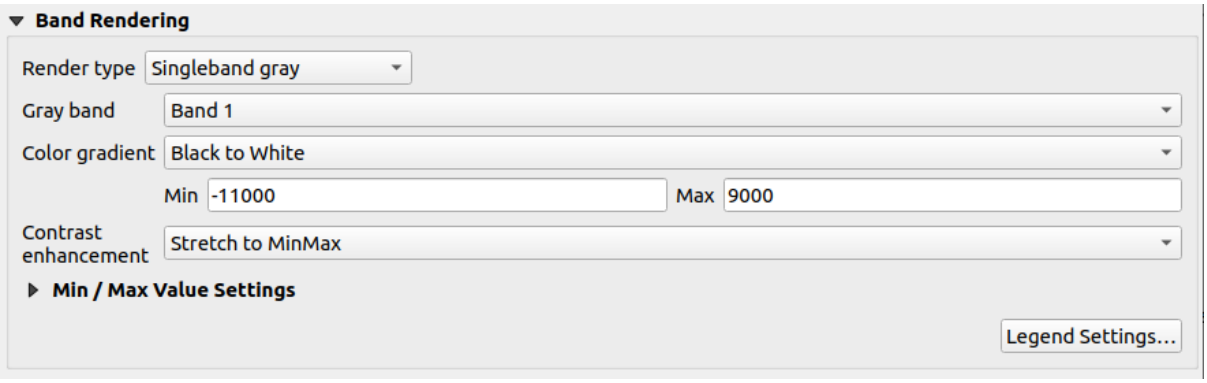


Fig. 13.4: Raster Symbology - Singleband gray rendering

Pixels are assigned a color based on the selected color gradient and the layer's legend (in the *Layers* panel and the layout *legend item*) is displayed using a continuous color ramp. Press *Legend settings...* if you wish to tweak the settings. More details at *Customize raster legend*.

Singleband pseudocolor

This is a render option for single-band files that include a continuous palette. You can also create color maps for a band of a multiband raster.

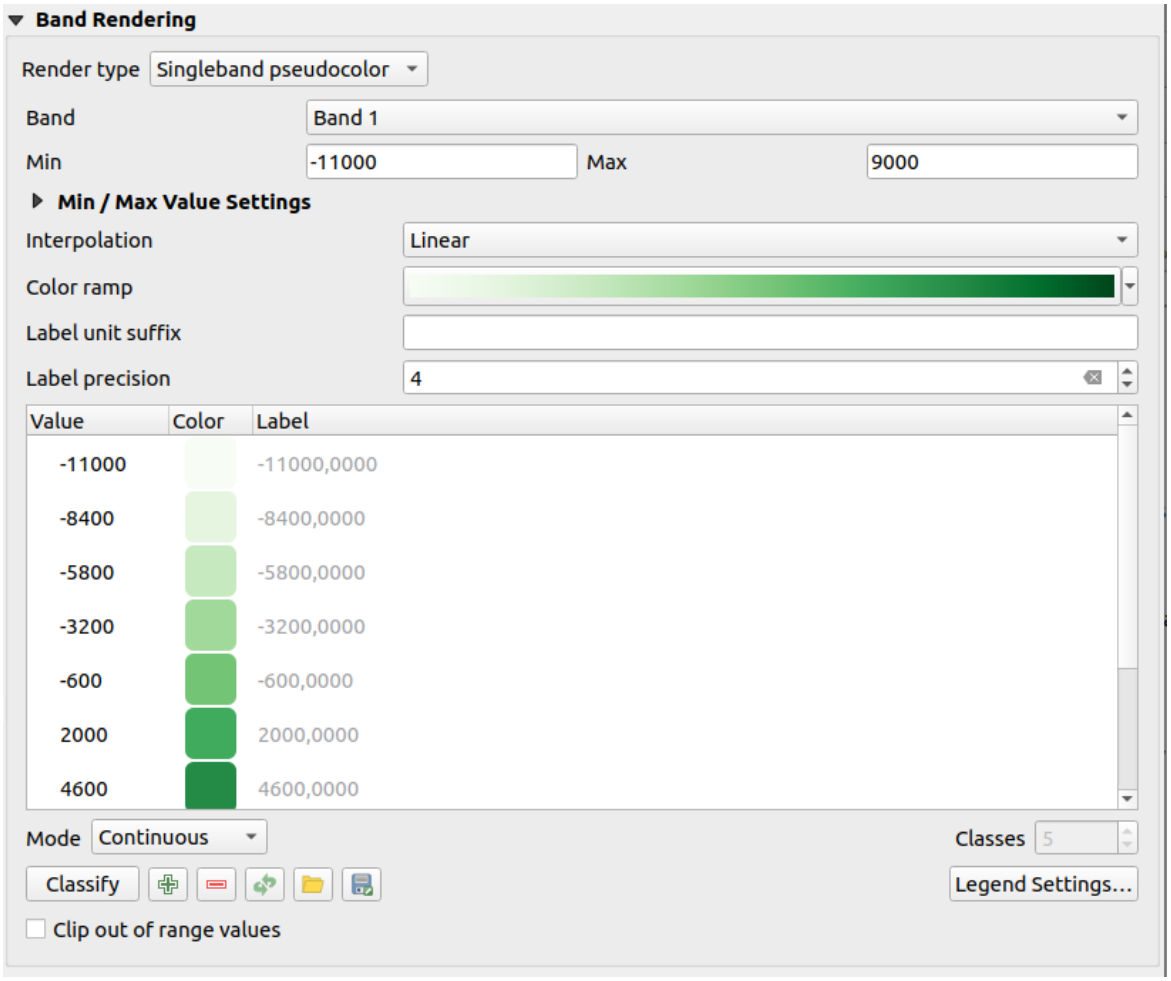


Fig. 13.5: Raster Symbology - Singleband pseudocolor rendering

Using a *Band* of the layer and a *values range*, you can now interpolate and assign representation color to pixels within classes. More at *Color ramp shader classification*.

Pixels are assigned a color based on the selected color ramp and the layer's legend (in the *Layers* panel and the layout *legend item*) is displayed using a continuous color ramp. Press *Legend settings...* if you wish to tweak the settings or instead use a legend with separated classes (and colors). More details at *Customize raster legend*.

Single color

This renderer allows you to render a raster layer using *Single color*. This type of renderer is useful when you want to display a raster layer uniformly, without any variation in color based on pixel values.

The single color renderer can be used with both single-band and multiband raster layers. When used with multiband rasters, you can select which band to apply the single color to, effectively displaying that specific band uniformly across the entire layer.

Fig. 13.6: Raster Symbology - Single color rendering

山体阴影

Render a band of the raster layer using hillshading.

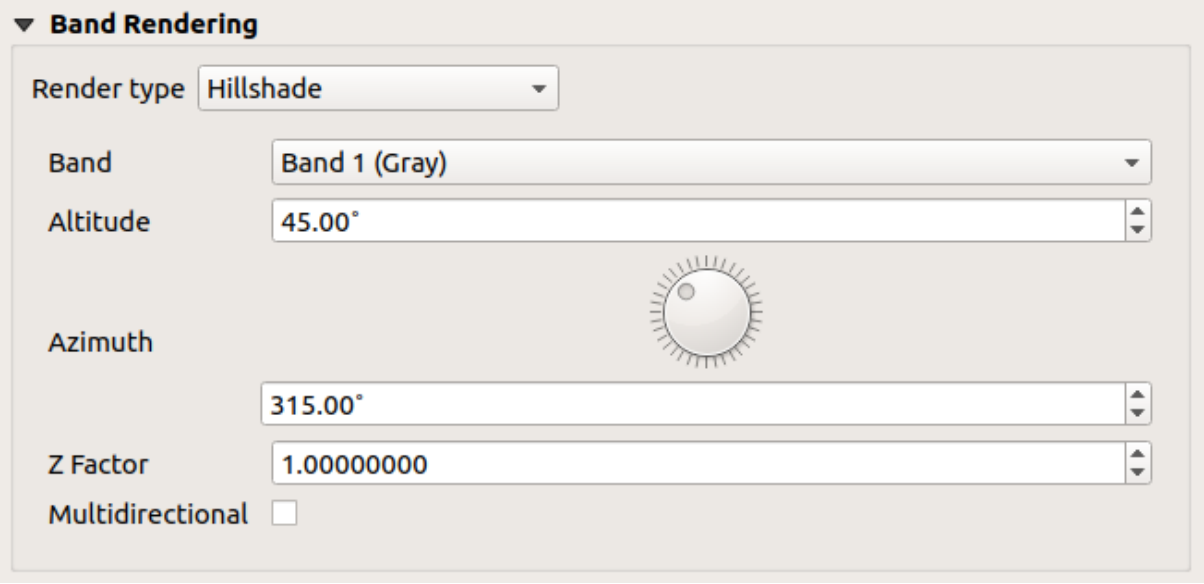


Fig. 13.7: Raster Symbology - Hillshade rendering

Options:

- *Band*: The raster band to use.
- *Altitude*: The elevation angle of the light source (default is `45°`).

- *Azimuth*: The azimuth of the light source (default is `315°`).
- *Z Factor*: Scaling factor for the values of the raster band (default is `1`).
- *Multidirectional*: Specify if multidirectional hillshading is to be used (default is `off`).

Contours

This renderer draws contour lines that are calculated on the fly from the source raster band.

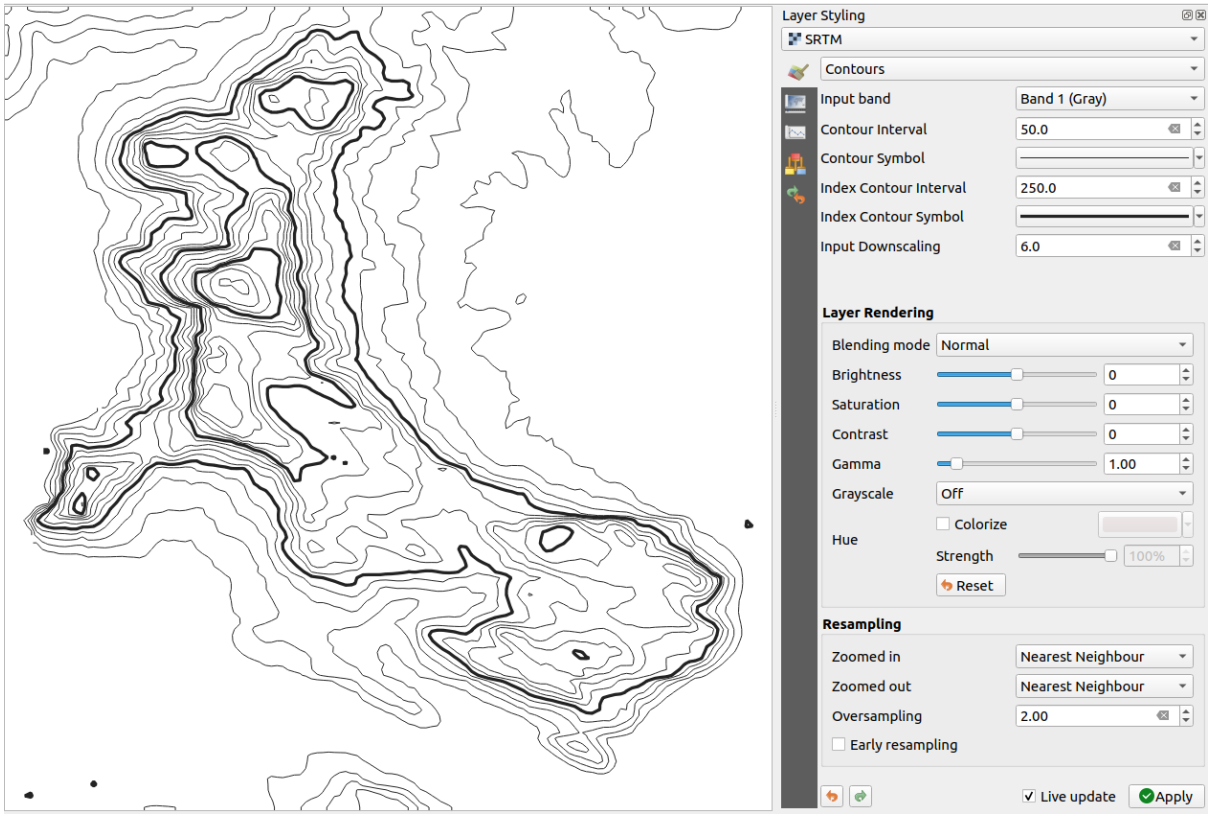


Fig. 13.8: Raster Symbology - Contours rendering

Options:

- *Input band*: the raster band to use.
- *Contour interval*: the distance between two consecutive contour lines
- *Contour symbol*: the *symbol* to apply to the common contour lines.
- *Index contour interval*: the distance between two consecutive **index contours**, that is the lines shown in a distinctive manner for ease of identification, being commonly printed more heavily than other contour lines and generally labeled with a value along its course.
- *Index contour symbol*: the symbol to apply to the index contour lines
- *Input downscaling*: Indicates by how much the renderer will scale down the request to the data provider (default is `4.0`).

  For example, if you generate contour lines on input raster block with the same size as the output raster block, the generated lines would contain too much detail. This detail can be reduced by the "downscale" factor, requesting lower resolution of the source raster. For a raster block 1000x500 with downscale 10, the renderer will request raster 100x50 from provider. Higher downscale makes contour lines more simplified (at the expense of losing some detail).

Setting the min and max values

By default, QGIS reports the *Min* and *Max* values of the band(s) of the raster. A few very low and/or high values can have a negative impact on the rendering of the raster. The *Min/Max Value Settings* frame helps you control the rendering.

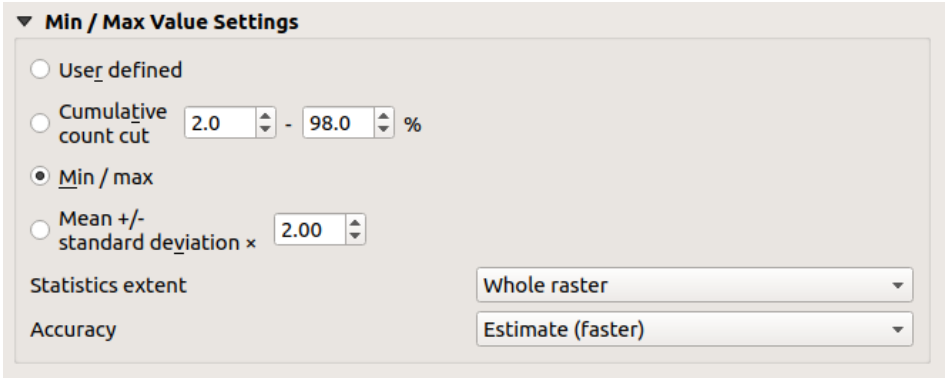


Fig. 13.9: Raster Symbology - Min and Max Value Settings

可用的选项包括:

- *User defined*: The default *Min* and *Max* values of the band(s) can be overridden
- *Cumulative count cut*: Removes outliers. The standard range of values is `2%` to `98%`, but it can be adapted manually.
- *Min / max*: Uses the whole range of values in the image band.
- *Mean +/- standard deviation x*: Creates a color table that only considers values within the standard deviation or within multiple standard deviations. This is useful when you have one or two cells with abnormally high values in a raster layer that impact the rendering of the raster negatively.

Calculations of the min and max values of the bands are made based on the:

- *Statistics extent*: it can be *Whole raster*, *Current canvas* or *Updated canvas*. *Updated canvas* means that min/max values used for the rendering will change with the canvas extent (dynamic stretching).
- *Accuracy*, which can be either *Estimate (faster)* or *Actual (slower)*.

> **Note**
>
> For some settings, you may need to press the *Apply* button of the layer properties dialog in order to display the actual min and max values in the widgets.

Color ramp shader classification

This method can be used to classify and represent scalar dataset (raster or mesh contour) based on their values. Given a *color ramp* and a number of classes, it generates intermediate color map entries for class limits. Each color is mapped with a value interpolated from a range of values and according to a classification mode. The scalar dataset elements are then assigned their color based on their class.

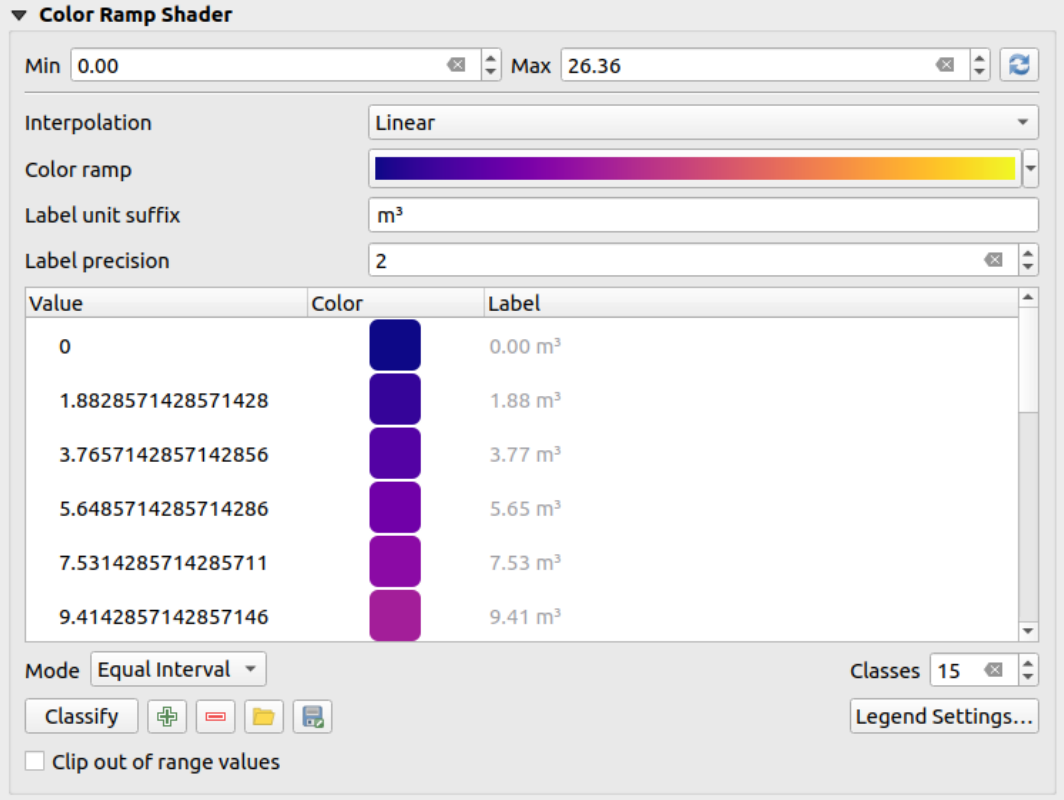


Fig. 13.10: Classifying a dataset with a color ramp shader

1. A *Min* and *Max* values must be defined and used to interpolate classes bounds. By default QGIS detects them from the dataset but they can be modified.
2. The *Interpolation* entry defines how scalar elements are assigned their color :
   - *Discrete* (a <= symbol appears in the header of the *Value* column): The color is taken from the closest color map entry with equal or higher value
   - *Linear*: The color is linearly interpolated from the color map entries above and below the pixel value, meaning that to each dataset value corresponds a unique color
   - *Exact* (a = symbol appears in the header of the *Value* column): Only pixels with value equal to a color map entry are applied a color; others are not rendered.
3. The *Color ramp* widget helps you select the color ramp to assign to the dataset. As usual with *this widget*, you can create a new one and edit or save the currently selected one. The name of the color ramp will be saved in the configuration.
4. The *Label unit suffix* adds a label after the value in the legend, and the *Label precision* controls the number of decimals to display.
5. The classification *Mode* helps you define how values are distributed across the classes:
   - *Equal interval*: Provided the *Number of classes*, limits values are defined so that the classes all have the same magnitude.
   - *Continuous*: Classes number and color are fetched from the color ramp stops; limits values are set following stops distribution in the color ramp.
   - *Quantile*: Provided the *Number of classes*, limits values are defined so that the classes have the same number of elements. Not available with *mesh layers*.
6. You can then *Classify* or tweak the classes:
   - The button Add values manually adds a value to the table.

- The button Remove selected row deletes selected values from the table.
- Double clicking in the *Value* column lets you modify the class value.
- Double clicking in the *Color* column opens the dialog *Change color*, where you can select a color to apply for that value.
- Double clicking in the *Label* column to modify the label of the class, but this value won't be displayed when you use the identify feature tool.
- Right-clicking over selected rows in the color table shows a contextual menu to *Change Color...* and *Change Opacity...* for the selection.

You can use the buttons Load color map from file or Export color map to file to load an existing color table or to save the color table for later use.

7. With linear *Interpolation*, you can also configure:

   - *Clip out of range values*: By default, the linear method assigns the first class (respectively the last class) color to values in the dataset that are lower than the set *Min* (respectively greater than the set *Max*) value. Check this setting if you do not want to render those values.
   - *Legend settings*, for display in the *Layers* panel and the layout *legend item*. More details at *Customize raster legend*.

Customize raster legend

When applying a color ramp to a raster or a mesh layer, you may want to display a legend showing the classification. By default, QGIS displays a continuous color ramp with min and max values in the *Layers* panel and the layout *legend item*. This can be customized using the *Legend settings* button in the classification widget.

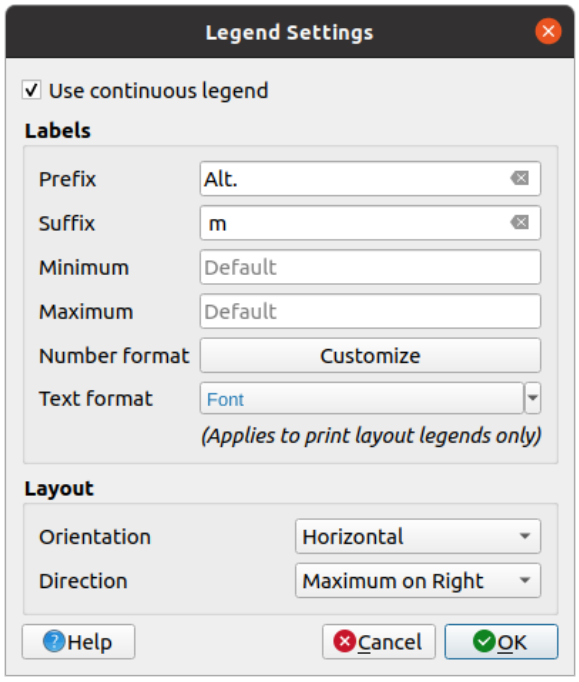


Fig. 13.11: Modifying a raster legend

In this dialog, you can set whether to *Use continuous legend*: if unchecked, the legend displays separated colors corresponding to the different classes applied. This option is not available for raster *singleband gray* symbology.

Checking the *Use continuous legend* allows you to configure both the labels and layout properties of the legend.

Labels

- Add a *Prefix* and a *Suffix* to the labels
- Modify the *Minimum* and a *Maximum* values to show in the legend
- *Customize* the *Number format*
- *Customize* the *Text format* to use in the print layout legend.

Layout

- Control the *Orientation* of the legend color ramp; it can be **Vertical** or **Horizontal**
- Control the *Direction* of the values depending on the orientation:
  - If vertical, you can display the **Maximum on top** or the **Minimum on top**
  - If horizontal, you can display the **Maximum on right** or the **Minimum on right**

Layer rendering

Over the symbology type applied to the layer band(s), you can achieve special rendering effects for the whole raster file(s):

- Use one of the blending modes (see *Blending Modes*)
- Set custom *Brightness*, *Saturation*, *Gamma* and *Contrast* to colors.
- With the *Invert colors*, the layer is rendered with opposite colors. Handy, for example, to switch out-of-the box OpenStreetMap tiles to dark mode.
- Turn the layer to *Grayscale* option either 'By lightness', 'By luminosity' or 'By average'.
- *Colorize* and adjust the *Strength* of *Hue* in the color table

Press *Reset* to remove any custom changes to the layer rendering.

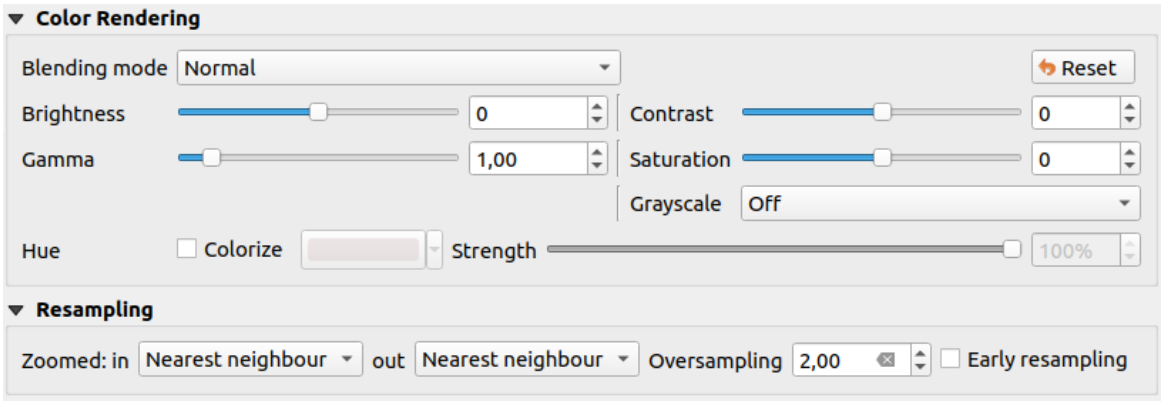


Fig. 13.12: Raster Symbology - Layer rendering and Resampling settings

Resampling

The *Resampling* option has effect when you zoom in and out of an image. Resampling modes can optimize the appearance of the map. They calculate a new gray value matrix through a geometric transformation.

When applying the 'Nearest neighbour' method, the map can get a pixelated structure when zooming in. This appearance can be improved by using the 'Bilinear (2x2 kernel)' or 'Cubic (4x4 kernel)' method, which cause sharp edges to be blurred. The effect is a smoother image. This method can be applied to for instance digital topographic raster maps.

Early resampling: allows to calculate the raster rendering at the provider level where the resolution of the source is known, and ensures a better zoom in rendering with QGIS custom styling. Really convenient for tile rasters loaded using an *interpretation method*.

13.1.4 Transparency Properties

QGIS provides capabilities to set the *Transparency* level of a raster layer.

Use the *Global opacity* slider to set to what extent the underlying layers (if any) should be visible through the current raster layer. This is very useful if you overlay raster layers (e.g., a shaded relief map overlayed by a classified raster map). This will make the look of the map more three dimensional. The opacity of the raster can be data-defined, and vary e.g. depending on the visibility of another layer, by temporal variables, on different pages of an atlas, ...

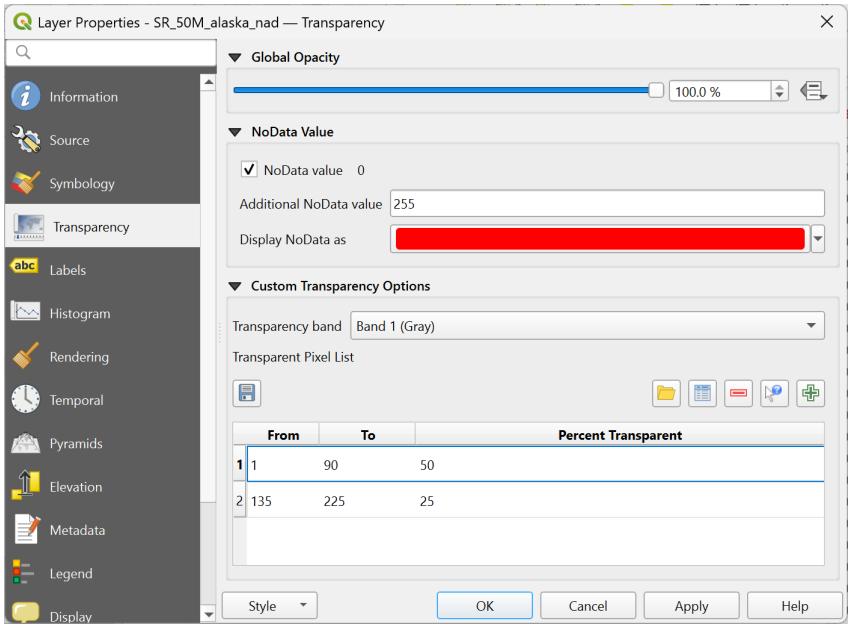


Fig. 13.13: Raster Transparency

With *No data value* QGIS reports the original source no data value (if defined) which you can consider as is in the rendering. Additionally, you can enter a raster value that should be treated as an *Additional no data value*. The *Display no data as* color selector allows you to apply a custom color to no data pixels, instead of the default transparent rendering.

An even more flexible way to customize the transparency is available in the *Custom transparency options* section:

- Use *Transparency band* to apply transparency for an entire band.
- Provide a list of pixels to make transparent with corresponding levels of transparency:

  1. Click the Add values manually button. A new row will appear in the pixel list.

2. For single-band based symbology (e.g. DEMs), enter the **From** and **To** values and adjust the **Percent Transparent** to apply.
3. For multiband based symbology (e.g. RGB images) enter the **Red**, **Green** and **Blue** values of the pixel and adjust the **Percent Transparent** to apply. QGIS supports **Tolerance** for pixel values, when defining transparency. This means that pixels with color close to the specified RGB values can also be made transparent. Note that this feature applies only to multiband rasters.
4. Alternatively, you can fetch the pixel values directly from the raster using the Add values from display button. Then enter the transparency value.
5. Repeat the steps to adjust more values with custom transparency.
6. Press the *Apply* button and have a look at the map.

As you can see, it is quite easy to set custom transparency, but it can be quite a lot of work. Therefore, you can use the button Export to file to save your transparency list to a file. The button Import from file loads your transparency settings and applies them to the current raster layer.

13.1.5 Labels Properties

The *Labels* properties provides you with all the needed and appropriate capabilities to configure smart labeling on raster layers. This dialog can also be accessed from the *Layer Styling* panel.

At the top of the dialog, you have:

- a combobox for selecting the appropriate labeling method for the active layer
- the Configure project labeling rules button: helps you control interactions between labels and features across the layers in the project. More details at *Configuring project labeling rules*.
- the Automated placement settings (applies to all layers) button: configure general properties on label placement and conflicts resolution. More details at *Setting the automated placement engine*.

The first step is to choose the labeling method from the drop-down list. Available methods are:

- *No labels*: the default value, showing no labels from the layer
- *Label with pixel values*: Show labels on the map using a band.

Using the *Label with pixel values* option, the following dialog opens.

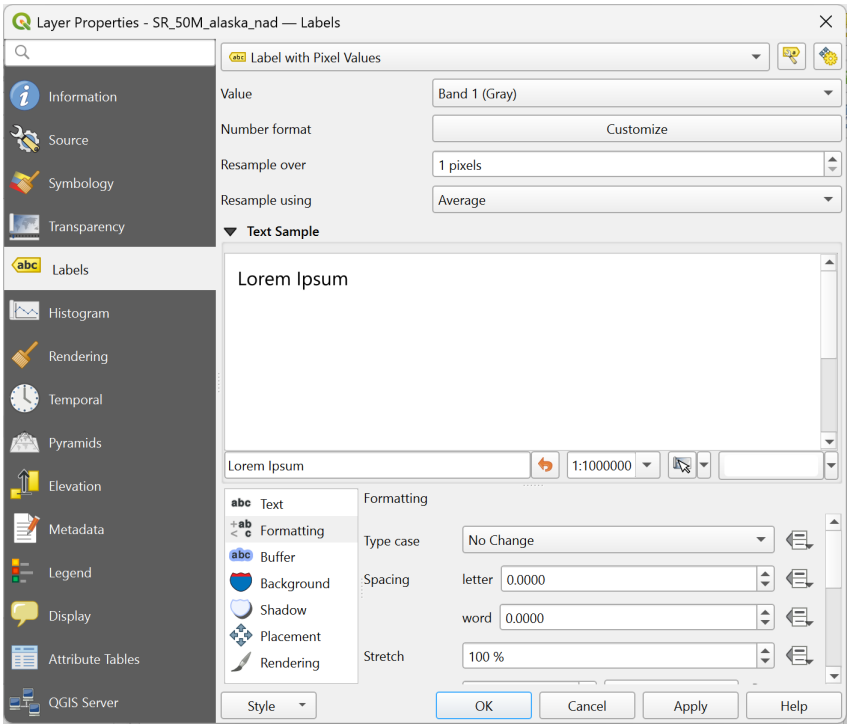


Fig. 13.14: Raster layer labeling settings

At the top of the dialog:

- A *Value* drop-down list allows you to select the band to take the values from
- Press *Customize* to configure a *proper number formatting*
- By default, the displayed value represents individual pixel band value. With *Resample over*, you can compute the value from the neighbouring pixels (setting `2` means `2*2=4` pixels) using a statistical method set in the *Resample using* widget.

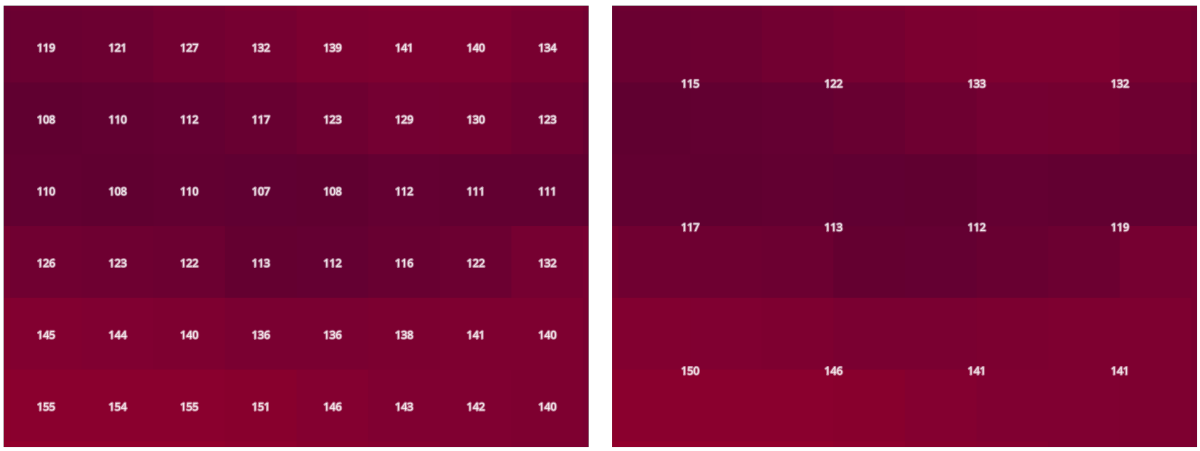


Fig. 13.15: Pixels labeled using various resampling options

Below are displayed options to customize the labels, under various tabs:

- abc *Text*
- *Formatting*
- abc *Buffer*

- *Background*
- *Shadow*
- *Placement*
- *Rendering*

Description of how to set each property is exposed at 设置标注.

13.1.6 Histogram Properties

The *Histogram* tab allows you to view the distribution of the values in your raster. The histogram is generated when you press the *Compute Histogram* button. All existing bands will be displayed together. You can save the histogram as an image with the button.

At the bottom of the histogram, you can select a raster band in the drop-down menu and *Set min/max style for* it. The *Prefs/Actions* drop-down menu gives you advanced options to customize the histogram:

- With the *Visibility* option, you can display histograms for individual bands. You will need to select the option *Show selected band*.
- The *Min/max options* allow you to 'Always show min/max markers', to 'Zoom to min/max' and to 'Update style to min/max'.
- The *Actions* option allows you to 'Reset' or 'Recompute histogram' after you have changed the min or max values of the band(s).

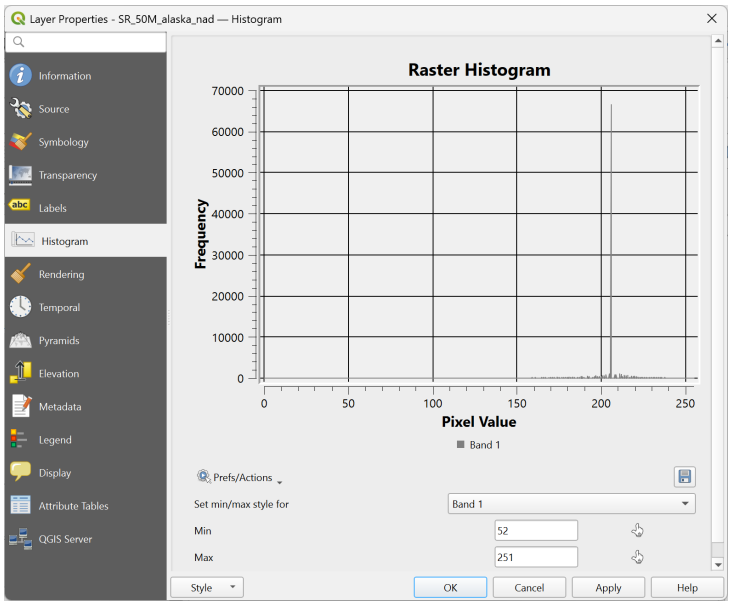


Fig. 13.16: Raster Histogram

13.1.7 Rendering Properties

In the *Rendering* tab, it's possible to:

- set *Scale dependent visibility* for the layer: You can set the *Maximum (inclusive)* and *Minimum (exclusive)* scales, defining a range of scales in which the layer will be visible. It will be hidden outside this range. The Set to current canvas scale button helps you use the current map canvas scale as a boundary. See 可见比例选择器 for more information.

  Note

  You can also activate scale dependent visibility on a layer from within the *Layers* panel: right-click on the layer and in the contextual menu, select *Set Layer Scale Visibility*.

- *Refresh layer at interval*: controls whether and how regular a layer can be refreshed. Available *Configuration* options are:
  - *Reload data*: the layer will be completely refreshed. Any cached data will be discarded and refetched from the provider. This mode may result in slower map refreshes.
  - *Redraw layer only*: this mode is useful for animation or when the layer's style will be updated at regular intervals. Canvas updates are deferred in order to avoid refreshing multiple times if more than one layer has an auto update interval set.

  It is also possible to set the *Interval (seconds)* between consecutive refreshments.

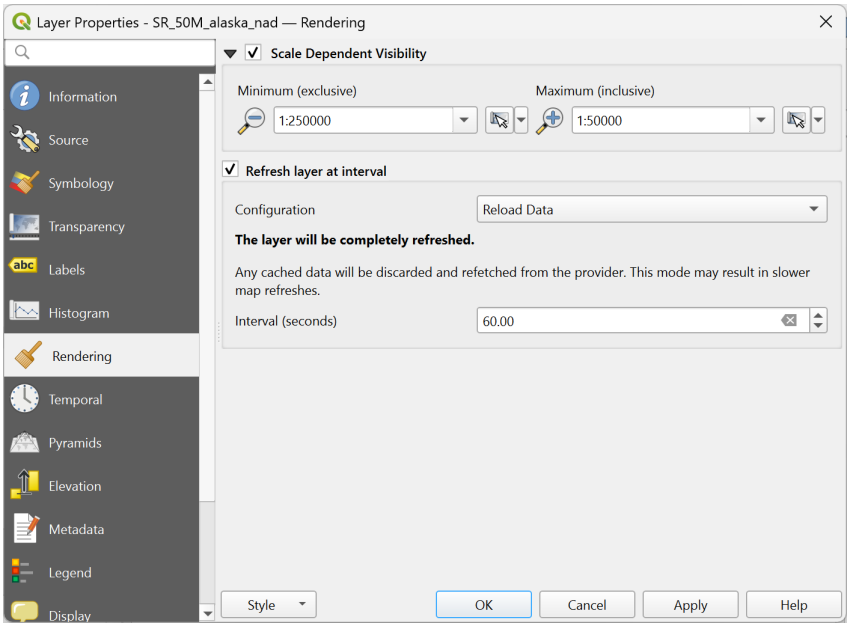


Fig. 13.17: Raster Rendering Properties

13.1.8 Temporal Properties

The *Temporal* tab provides options to control the rendering of the layer over time. Such dynamic rendering requires the *temporal navigation* to be enabled over the map canvas.

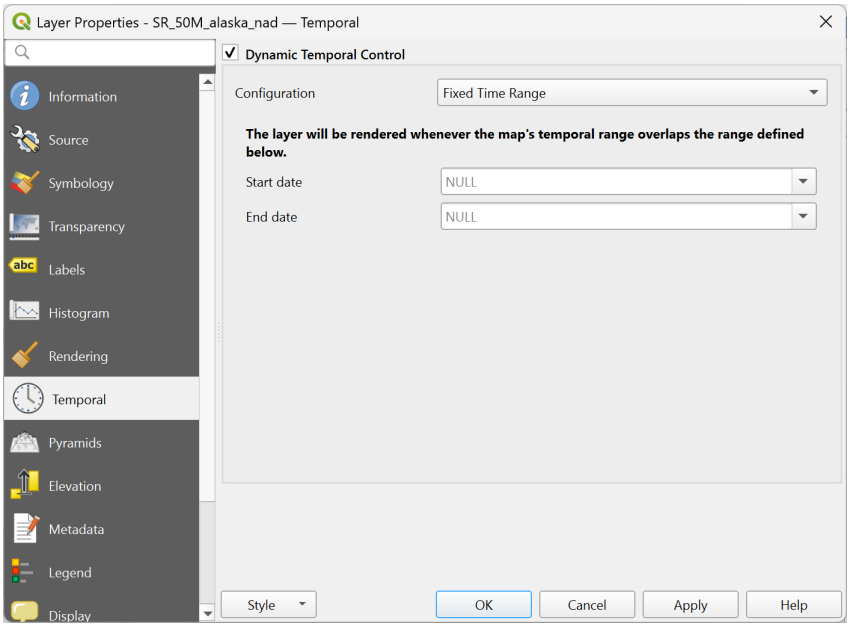


Fig. 13.18: Raster Temporal Properties

Check the *Dynamic Temporal Control* option and set whether the layer redraw should be:

- *Automatic*: the rendering is controlled by the underlying data provider if it suppports temporal data handling. E.g. this can be used with WMS-T layers or PostgreSQL rasters.
- *Fixed Date/Time*: only show the raster layer at a single, specific date/time. This avoids having to enter the same value for both the start and end of the temporal range.
- *Fixed time range*: only show the raster layer if the animation time is within a *Start date* and *End date* range.
- *Fixed Time Range Per Band*: only shows a band when the current animation time is between its *Begin* and *End* date range. This option allows you to either manually set these time ranges for each band or use the button to automatically generate datetime values, enabling detailed temporal analysis and visualization. This mode is particularly useful for working with raster layers where each band corresponds to a specific time period, such as NetCDF files.
- *Represents Temporal Values*: interprets each pixel in the raster layer as a datetime value. When this temporal mode is active, pixels that do not fall within the temporal range specified in the render context will be hidden, ensuring that only temporally relevant data is displayed. This mode is effective for:
  - Analyzing land use changes, like observing deforestation patterns.
  - Studying flooding by comparing water coverage across different times.
  - Evaluating movement costs in terrain analysis, for example, using GRASS GIS's r.walk tool to calculate travel costs across a landscape.
- *Redraw layer only*: the layer is redrawn at each new animation frame. It's useful when the layer uses time-based expression values for renderer settings (e.g. data-defined renderer opacity, to fade in/out a raster layer).

13.1.9 Pyramids Properties

High resolution raster layers can slow navigation in QGIS. By creating lower resolution copies of the data (pyramids), performance can be considerably improved, as QGIS selects the most suitable resolution to use depending on the zoom level.

You must have write access in the directory where the original data is stored to build pyramids.

From the *Resolutions* list, select resolutions at which you want to create pyramid levels by clicking on them.

If you choose **Internal (if possible)** from the *Overview format* drop-down menu, QGIS tries to build pyramids internally.

> **Note**
>
> Please note that building pyramids may alter the original data file, and once created they cannot be removed. If you wish to preserve a 'non-pyramided' version of your raster, make a backup copy prior to pyramid building.

If you choose **External** and **External (Erdas Imagine)** the pyramids will be created in a file next to the original raster with the same name and a `.ovr` extension.

Several *Resampling methods* can be used for pyramid calculation:

- 最邻近采样分析
- 平均
- Gauss
- 立方体
- Cubic Spline
- Laczos
- Mode
- None

Finally, click *Build Pyramids* to start the process.

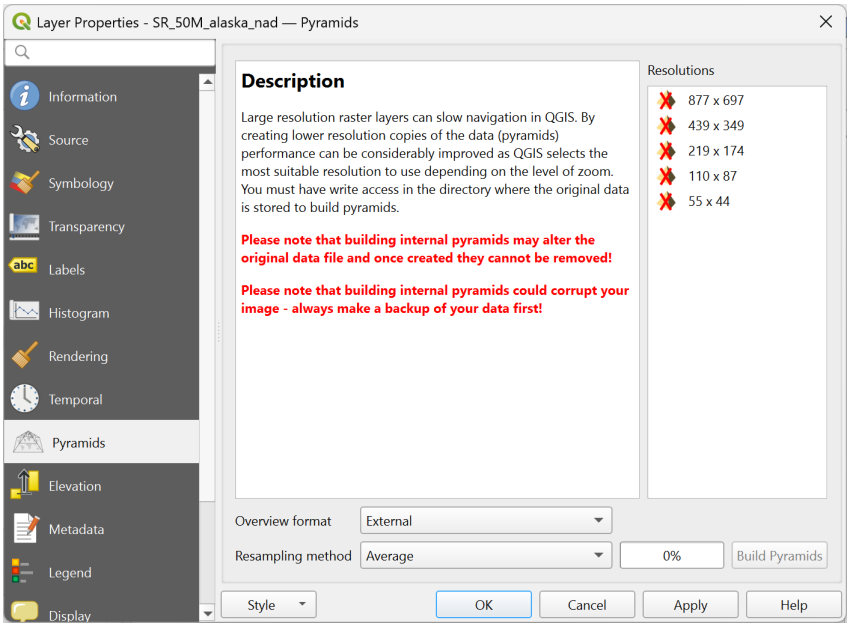


Fig. 13.19: Raster Pyramids

13.1.10 Elevation Properties

The *Elevation* tab provides options to control the layer elevation properties within a *3D map view* and its appearance in the *profile tool charts*. Specifically, you can choose to *Disable* this configuration if the layer does not contain elevation data or you can set:

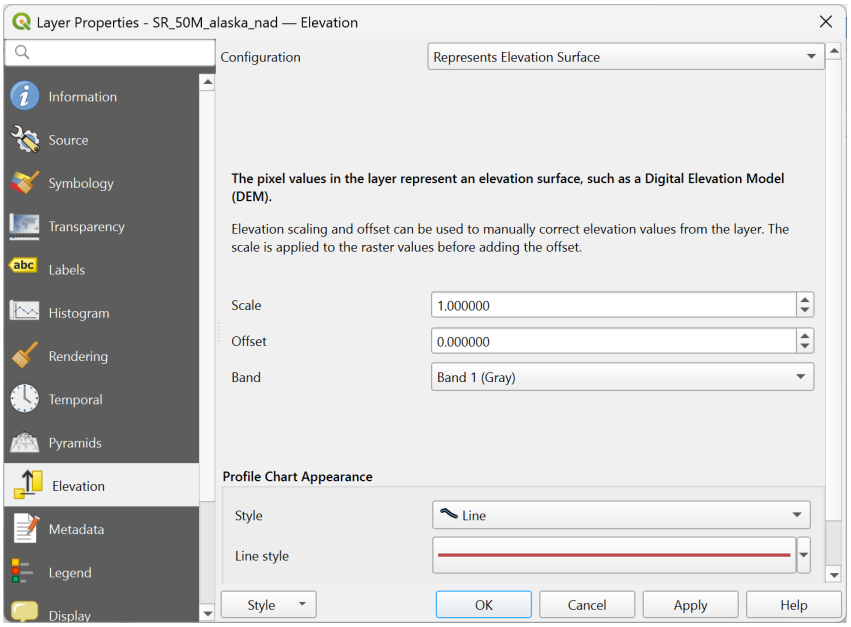


Fig. 13.20: Raster Elevation Properties

- *Represents Elevation Surface*: whether the raster layer represents a height surface (e.g DEM) and the pixel values should be interpreted as elevations. Choose this option if you want to display a raster in an *elevation profile view*. You will also need to fill in the *Band* to pick values from and can apply a *Scale* factor and an *Offset*.
- *Fixed Elevation Range*: The raster layer (or selected raster band) is associated with a fixed elevation range. This mode can be used when a layer has a single fixed elevation or a range (slice) of elevation values. If a range is specified, pixels will be extruded over this range. You can set the *Lower* and *Upper* elevation range values for the layer, and specify whether the lower or upper *Limits* are inclusive or exclusive.
- *Fixed Elevation Range Per Band*: Each band in the raster can have a fixed elevation range associated with it. This is designed for data sources that expose elevation-related data in bands, such as NetCDF files. For example, a raster with temperature data at different ocean depths. When rendering, the uppermost matching band will be selected and used for the layer's data. This feature is exposed as a user-editable table for raster bands with lower and upper values. Users can either populate the lower and upper values manually or use an *Expression* to auto-fill all band values based on expression. The expression-based fill allows you to design expressions that extract useful information from band names. For example, extracting the depth value from a band name like "Band 001: depth=-5500 (meters)".
- *Dynamic Elevation Range Per Band*: This mode calculates elevation ranges for raster bands dynamically using QGIS expressions. It's ideal for datasets where elevation values follow a consistent pattern across bands (like equally spaced vertical layers), eliminating the need to manually assign fixed elevations. Instead of entering individual values for each band, you define expressions for the *Lower* and *Upper* elevation bounds. These expressions can use variables like `@band`, `@band_name`, or `@band_description` automatically computing the elevation range based on each band's properties.
- *Profile Chart Appearance*: controls the rendering of the raster elevation data in the profile chart. The profile *Style* can be set as:
  - a *Line* with a specific *Line style*

- an elevation surface rendered using a fill symbol either above (*Fill above*) or below (*Fill below*) the elevation curve line. The surface symbology is represented using:
  - a *Fill style*
  - and a *Limit*: the maximum (respectively minimum) altitude determining how high the fill surface will be

13.1.11 元数据属性

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

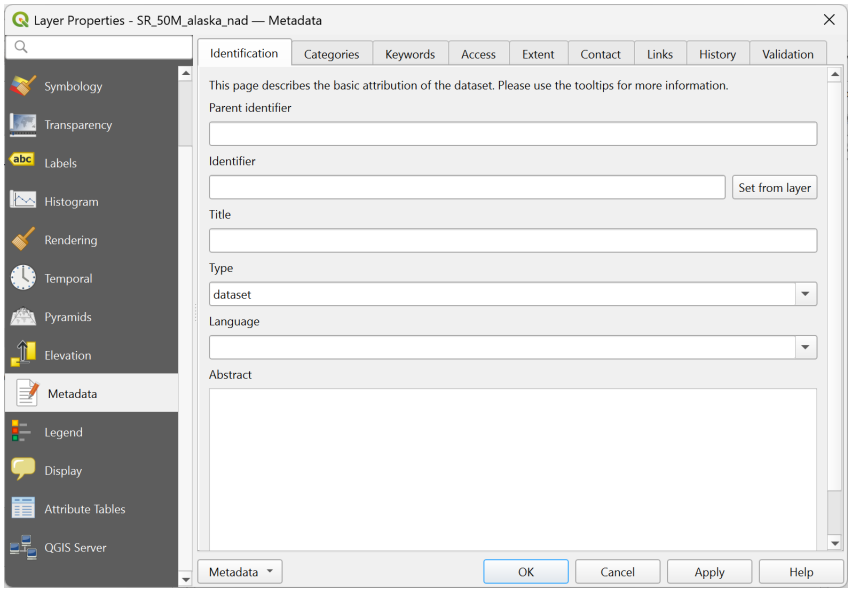


Fig. 13.21: Raster Metadata

13.1.12 Legend Properties

The *Legend* tab provides you with advanced settings for the *Layers panel* and/or the *print layout legend*. These options include:

- Depending on the symbology applied to the layer, you may end up with several entries in the legend, not necessarily readable/useful to display. The *Legend placeholder image* helps you *select an image* for replacement, displayed both in the *Layers* panel and the print layout legend.
- The *Embedded widgets in Legend* provides you with a list of widgets you can embed within the layer tree in the Layers panel. The idea is to have a way to quickly access some actions that are often used with the layer (setup transparency, filtering, selection, style or other stuff...).

  By default, QGIS provides a transparency widget but this can be extended by plugins that register their own widgets and assign custom actions to layers they manage.

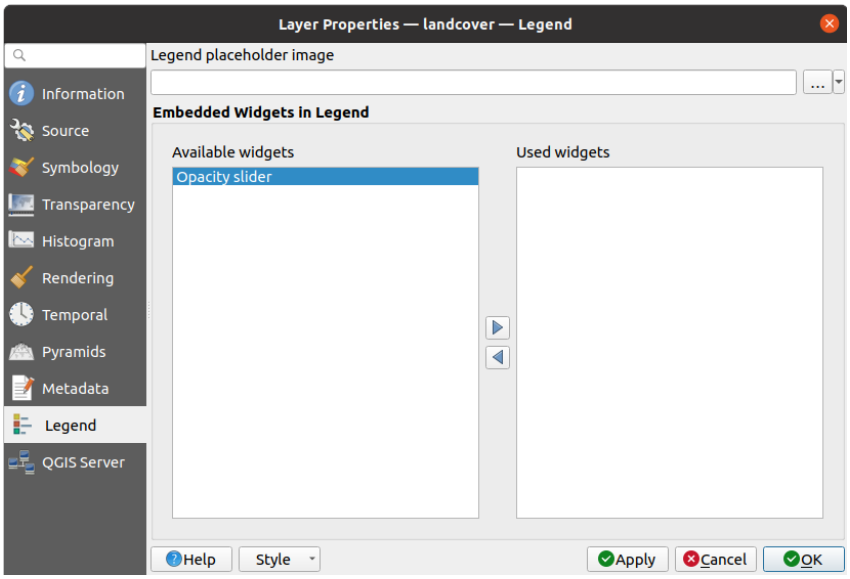


Fig. 13.22: Raster Legend

13.1.13 Display Properties

The *Display* tab helps you configure HTML map tips to use for pixels identification:

- *Enable Map Tips* controls whether to display map tips for the layer
- The *HTML Map Tip* provides a complex and full HTML text editor for map tips, mixing QGIS expressions and html styles and tags (multiline, fonts, images, hyperlink, tables, ...). You can check the result of your code sample in the *Preview* frame. You can also select and edit existing expressions using the *Insert/Edit Expression* button.

  You might look for expressions located in *Rasters* group or the `@layer_cursor_point` variable in the *Expressions* dialog.

> **Note**
>
> **Understanding the *Insert/Edit Expression* button behavior**
>
> If you select some text within an expression (between "[%" and "%]"), or if no text is selected but the cursor is inside an expression, the whole expression will be automatically selected for editing. If the cursor or a selected text is outside an expression, the dialog opens with the selection.

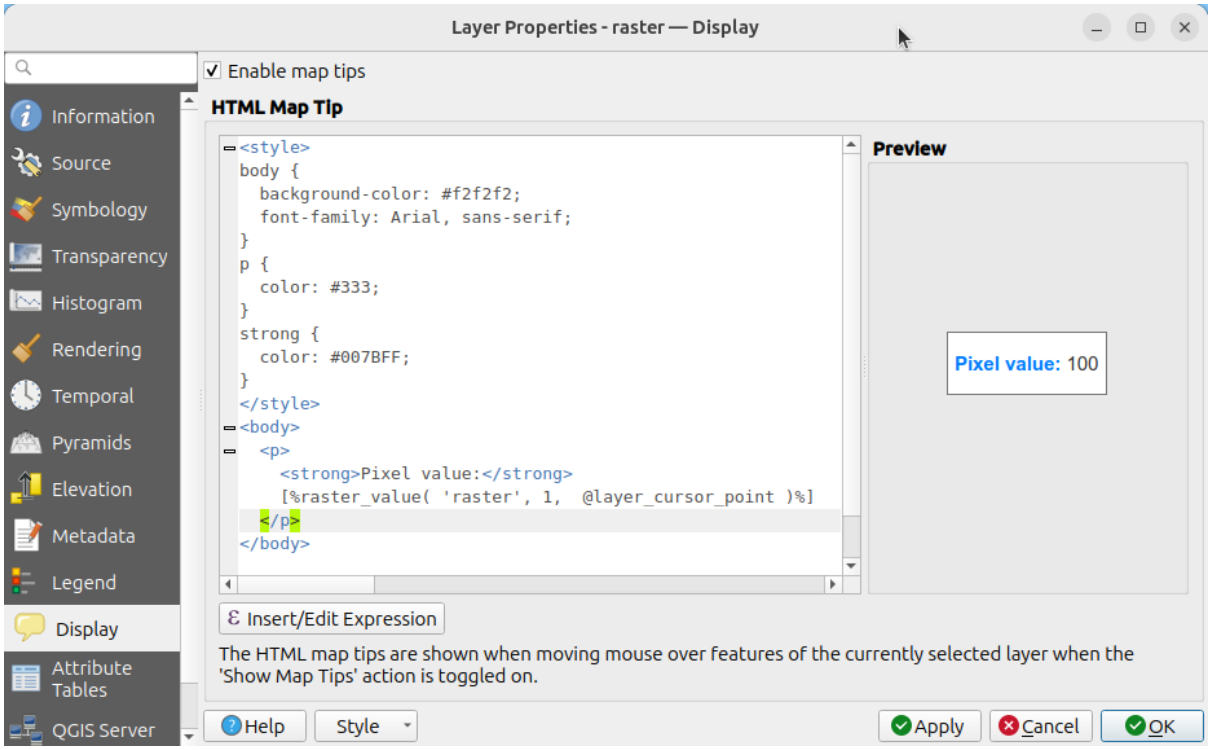


Fig. 13.23: Map tips with raster layer

To display map tips:

1. Select the menu option *View* ► *Show Map Tips* or click on the Show Map Tips icon of the *Attributes Toolbar*.
2. Make sure that the layer you target is active and has the *Enable Map Tips* property checked.
3. Move over a pixel, and the corresponding information will be displayed over.

Map tip is a cross-layer feature meaning that once activated, it stays on and applies to any map tip enabled layer in the project until it is toggled off.

13.1.14 QGIS Server Properties

The *QGIS Server* tab helps you configure settings of the data when published by QGIS Server. The configuration concerns:

- *Description*: provides information to describe the data, such as *Short name*, *Title*, *Abstract*, a *Keyword list*, and a *Data URL* whose *Format* can be in `text/html`, `text/plain` or `application/pdf`.
- *Attribution*: a *Title* and *URL* to identify who provides the data
- *Metadata URL*: a list of *URL* for the metadata that can be of `FGDC` or `TC211` *Type*, and in `text/plain` or `text/xml` *Format*
- *Legend URL*: a *URL* for the legend, in either `image/png` or `image/jpeg` *Format*

> **Note**
>
> When the raster layer you want to publish is already provided by a web service, further *properties* are available for setting.

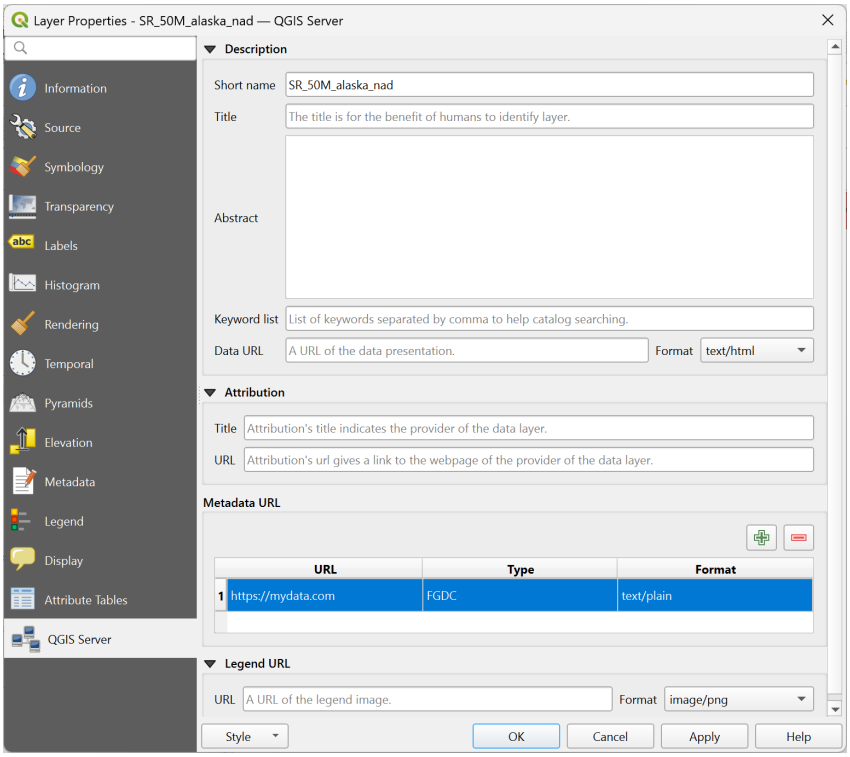


Fig. 13.24: QGIS Server in Raster Properties

13.1.15 Attribute Tables

Similar to vector layers, raster layers in QGIS can have associated Raster Attribute Tables (RATs) that store additional information about raster pixel values.

A Raster Attribute Table links raster values (or value ranges) to descriptive attributes such as class names, pixel counts, and colors. RATs are commonly used for classified rasters, color tables, and histogram information, helping QGIS interpret how raster values should be displayed and classified.

The *Attribute Tables* tab allows you to view and edit raster attribute data. To access the attribute table of a raster layer, right-click the layer in the layers panel, open *Properties...* and select *Attribute Tables*.

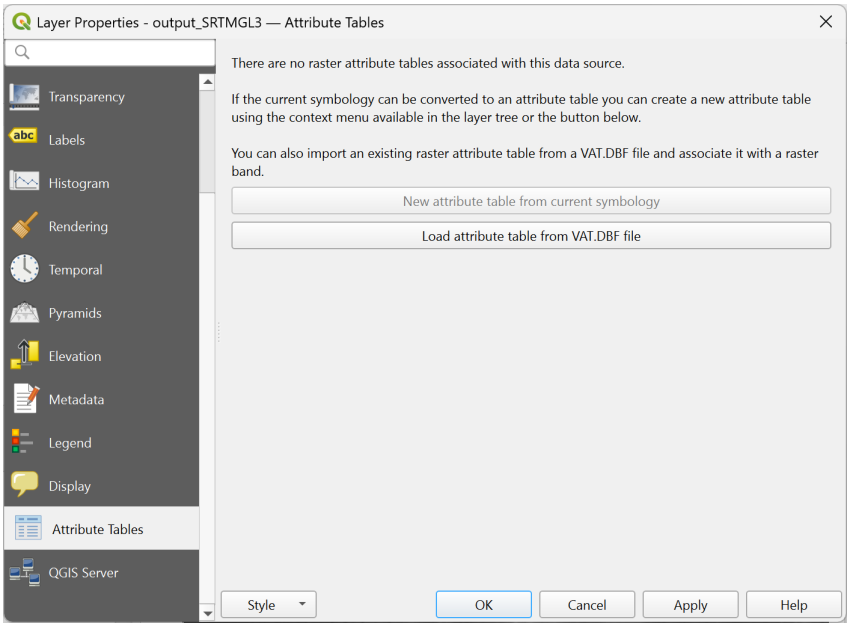


Fig. 13.25: Accessing the raster attribute table

When opening the *Attribute Tables* for a raster layer that does not have an associated Raster Attribute Table, QGIS displays a notice informing that no attribute table is currently linked to the data source. In this state, the interface provides two options to define and associate a new attribute table:

- *New attribute table from current symbology*: will appear if your raster layer has a classified symbology applied. Otherwise, this option is disabled. Applying classified symbology also triggers an option in the layer context menu to *Create Raster Attribute Table*. Click this button and set the following options in the *New Raster Attribute Table* dialog.
  - *Managed by the data provider*: the attribute table will be saved and managed by the data provider (if supported), overwriting any existing table for the raster band used by the current style. Depending on the data provider, the attribute table will be embedded in the main raster file or saved into a sidecar file managed by the data provider.
  - *Sidecar VAT.DBF file*: saves the attribute table into a sidecar VAT.DBF file. The resulting file will not be associated with any particular band.

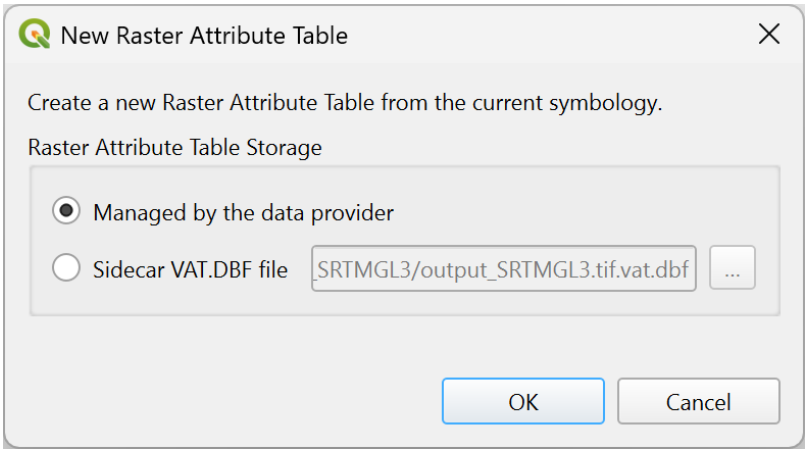


Fig. 13.26: Create new attribute table from current symbology

- *Load attribute table from VAT.DBF file*: retrieves the attribute table from an external DBF file. *Load Raster Attribute Table* dialog will appear to let you browse and select the DBF file and associate it with the raster

band. The action also can be done by right-clicking the raster layer in the layers panel and selecting *Load Raster Attribute Table from VAT.DBF*. *Open the newly loaded attribute table* allows you to directly open the attribute table editor after loading the table.

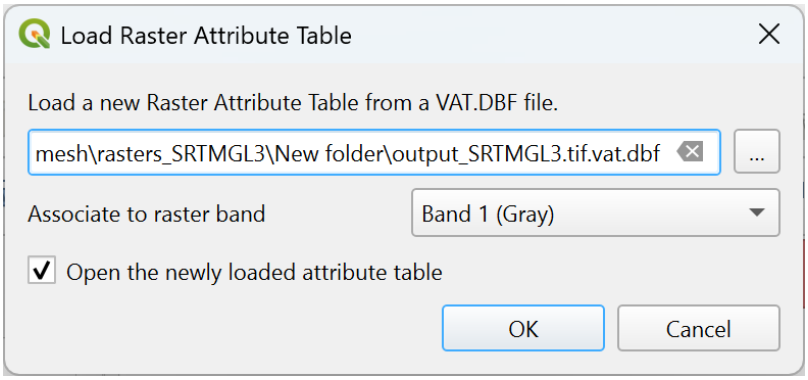


Fig. 13.27: Load attribute table from VAT.DBF file

Editing raster attribute tables

When set, a raster attribute table can be accessed from the *Attribute tables* tab of the layer properties dialog or by selecting *Open raster attribute table* from the layer contextual menu in *Layers* panel. This dialog shows a table whose rows correspond to the different pixel values or value ranges classified in a given band of the raster layer. Depending on the type of classification, a row is described by:

- for an exact values classification (e.g., paletted/unique values): the band name and the applied *Color*. Additional fields of the *Red*, *Green*, *Blue* and *Opacity* of the color are displayed.
- for range values classification (e.g., singleband pseudocolor): the range of values defined by their *Min*, *Max*, the associated label (*Class*) and color ramp interpolated over the values (*Color*). Additional fields of the minimum and maximum of *Red*, *Green*, *Blue* and *Opacity* of the color ramp are displayed.

You can also customize the raster attribute table by adding new fields for classification and group different value ranges accordingly, modifying value ranges and their associated colors or labels, adding new rows, ... The new attribute table can be used to update the current layer symbology.

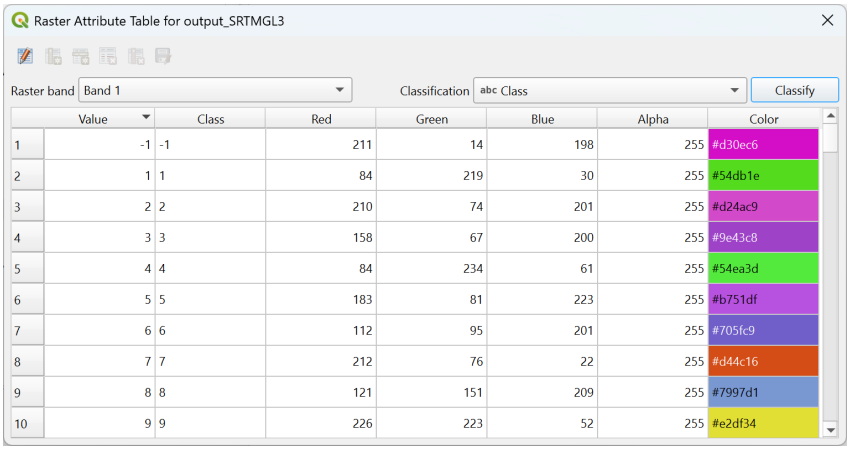


Fig. 13.28: Raster Attribute Table

At the top of the dialog, a set of tools allows to edit and save changes to the attribute table:

- Edit Attribute Table turns the table into edit mode. If no specific row or field is selected, only the Add Column… option is available.

- Add Column... opens the *Add Column* dialog to add a new column to the raster attribute table. The dialog lets you define:
  - *Column Type*, which can be:
    - *Standard column*
    - *Color RGBA*
    - *Color ramp (RGBA minimum and RGBA maximum)*
  - *Column Definition*, including:
    - *Name* of the column (mandatory)
    - *Usage*, which can be:
      - *Pixel Count*: stores the number of pixels belonging to a given value or value range.
      - *Generic*: general-purpose column with no predefined meaning.
      - *Name*: stores the class name or label associated with a raster value or range.
    - *Data type*, which can be:
      - *String*
      - *Integer*
      - *Long Integer*
      - *Double*
  - *Insertion point*, defining whether the new column is inserted *Before* or *After* a selected existing column.

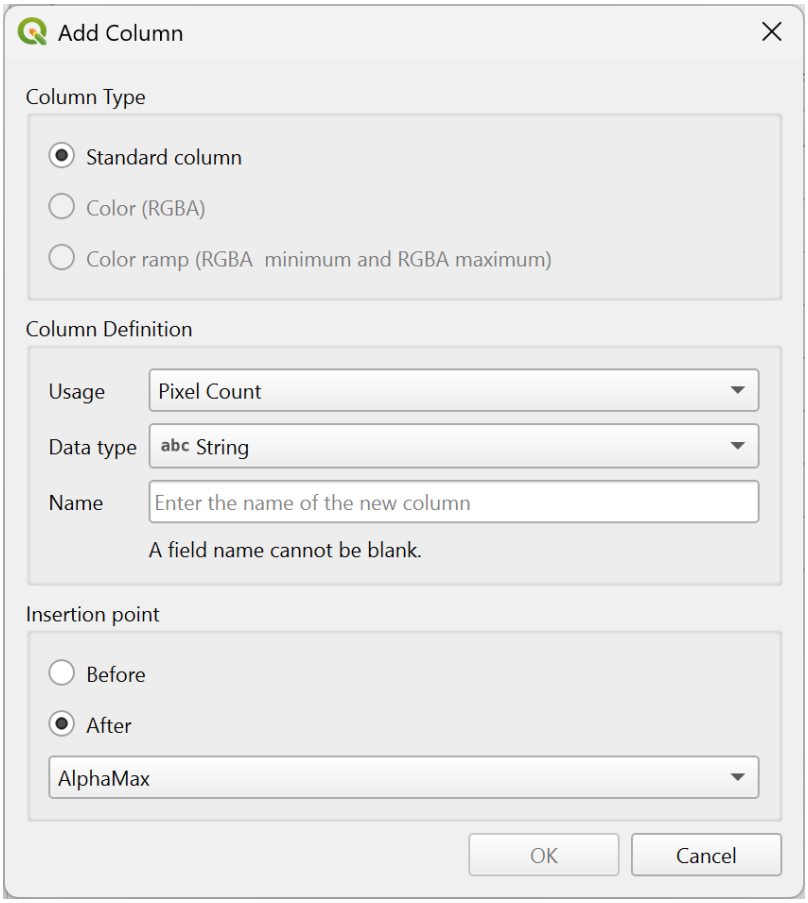


Fig. 13.29: Add Column dialog

- *Add Row...* opens the *Add Row* dialog to add a new row to the raster attribute table. Choose the *Insertion point* as *Before current row* or *After current row*.
- Remove Row removes the selected row from the raster attribute table.
- Remove Column removes the selected column from the raster attribute table.
- Save Changes saves any modifications made to the raster attribute table.

The next row provides options for controlling the displayed information in the table:

- *Raster band*: allows selecting which raster band the attribute table applies to. Raster layers may contain multiple bands (e.g., multispectral data). Each band can have its own Raster Attribute Table.
- *Classification*: allows selecting the fields used for classifying the raster layer. Keep in mind that the existing symbology for the raster will be replaced by a new symbology from the attribute table and any unsaved changes to the current symbology will be lost. Make sure to save your symbology style if needed before applying a classification.

13.1.16 Identify raster cells

The *identify features* tool allows you to get information about specific points in a raster layer.

To use the *Identify features* tool:

1. Select the raster layer in the Layers panel.
2. Click on the *Identify features* tool in the toolbar or press `Ctrl+Shift+I`.
3. Click on the point in the raster layer that you want to identify. The pixel will get highlighted.

The Identify Results panel will open in its default `Tree` view and display information about the clicked point. Formatting of the results vary depending on the provider of the layer. For example:

- For a local raster layer: below the name of the layer, you have on the left the band(s) of the clicked pixel, and on the right their respective value.
- For a remote layer such as WMS, a *Format* menu allows you to select whether the information should be displayed as *HTML*, *Feature* or *Text*.

These values can also be rendered (from the *View* menu located at the bottom of the panel) in:

- a `Table` view - organizes the information about the identified features and their values in a table.
- a `Graph` view - organizes the information about the identified features and their values in a graph.

Under the pixel attributes, you will find the *Derived* information, such as:

- 单击点的 `X` 和 `Y` 坐标值
- Column and row of the point clicked (pixel) when compatible

13.2 栅格分析

13.2.1 Raster Calculator

The *Raster Calculator* in the *Raster* menu allows you to perform calculations on the basis of existing raster pixel values (see Fig. 13.30). The results are written to a new raster layer in a GDAL-supported format.

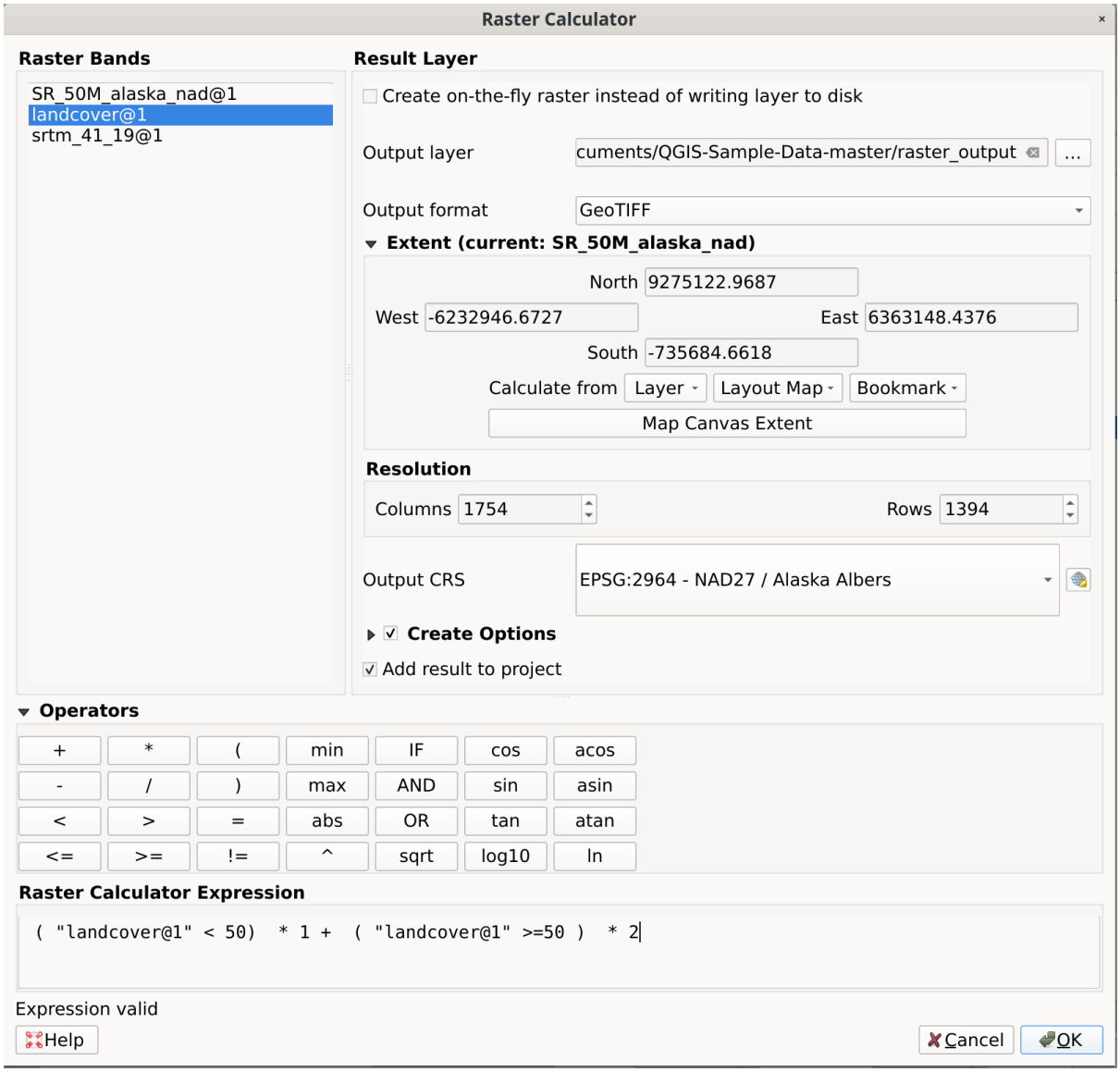


Fig. 13.30: Raster Calculator

The *Raster bands* list contains all loaded raster layers that can be used. To add a raster to the raster calculator expression field, double click its name in the Fields list. You can then use the operators to construct calculation expressions, or you can just type them into the box.

In the *Result layer* section, you will need to define an output layer. You can:

- *Create on-the-fly raster instead of writing layer to disk*:
  - If unchecked, the output is stored on the disk as a new plain file. An *Output layer* path and an *Output format* are required.
  - If checked, a virtual raster layer, i.e. a raster layer defined by its URI and whose pixels are calculated on-the-fly, is created. It's not a new file on disk; the virtual layer is still connected to the rasters used in the calculation meaning that deleting or moving these rasters would break it. A *Layer name* can be provided, otherwise the calculation expression is used as such. Removing the virtual layer from the project deletes it, and it can be made persistent in file using the layer *Export* ► *Save as...* contextual menu.
- Define the *Spatial extent* of the calculation using the *extent selector* widget.
- Set the *Resolution* of the layer using columns and rows number. If the input layer has a different resolution, the values will be resampled with the nearest neighbor algorithm.

- Enable the *Create Options* to define additional raster creation options. These options let you control how the output file is structured and compressed, including format, compression method, and other driver-specific settings. See more at *Raster driver options*.
- With the *Add result to project* checkbox, the result layer will automatically be added to the legend area and can be visualized. Checked by default for virtual rasters.

The *Operators* section contains all available operators. To add an operator to the raster calculator expression box, click the appropriate button. Mathematical calculations (+, −, *, ...) and trigonometric functions (`sin`, `cos`, `tan`, ...) are available. Conditional expressions (=, !=, <, >=, ...) return either 0 for false or 1 for true, and therefore can be used with other operators and functions.

See also

Raster calculator and *Raster calculator (virtual)* algorithms

Raster calculator expression

The dialog

The *Raster calculator expression* dialog provides means to write expressions for pixels calculations between a set of raster layers.

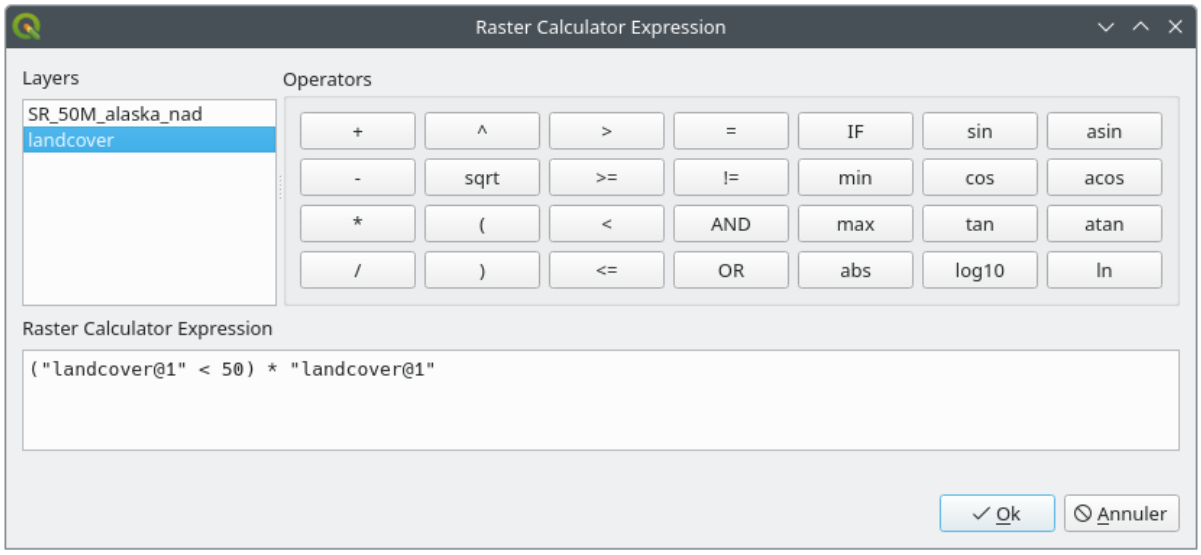


Fig. 13.31: Raster Expression Calculator

- *Layers*: Shows the list of all raster layers loaded in the legend. These can be used to fill the expression box (double click to add). Raster layers are referred by their name and the number of the band: `layer_name@band_number`. For instance, the first band from a layer named `DEM` will be referred as `DEM@1`.
- *Operators*: contains a number of calculation operators for pixels manipulation:
  - Arithmetical: +, −, *, `sqrt`, `abs`, `ln`, ...
  - Trigonometric: `sin`, `cos`, `tan`, ...
  - Comparison: =, !=, <, >=, ...
  - Logical: `IF`, `AND`, `OR`, `(`, `)`
  - Statistical: `min`, `max`

To add an operator to the raster calculator expression box, click the appropriate button.

- *Raster calculator expression* is the area in which the expression is composed

Examples

Convert elevation values from meters to feet

Creating an elevation raster in feet from a raster in meters, you need to use the conversion factor for meters to feet: 3.28. The expression is:

```
"elevation@1" * 3.28
```

Using a mask

If you want to mask out parts of a raster -- say, for instance, because you are only interested in elevations above 0 meters -- you can use the following expression to create a mask and apply the result to a raster in one step.

```
("elevation@1" >= 0) * "elevation@1"
```

In other words, for every cell greater than or equal to 0 the conditional expression evaluates to 1, which keeps the original value by multiplying it by 1. Otherwise the conditional expression evaluates to 0, which sets the raster value to 0. This creates the mask on the fly.

Classify a Raster

If you want to classify a raster -- say, for instance into two elevation classes, you can use the following expression to create a raster with two values 1 and 2 in one step.

```
("elevation@1" < 50) * 1 + ("elevation@1" >= 50) * 2
```

In other words, for every cell less than 50 set its value to 1. For every cell greater than or equal 50 set its value to 2.

Or you can use the `IF` operator.

```
if ( elevation@1 < 50 , 1 , 2 )
```

CHAPTER 14

Working with Mesh Data

14.1 What's a mesh?

A mesh is an unstructured grid usually with temporal and other components. The spatial component contains a collection of vertices, edges and/or faces, in 2D or 3D space:

- **vertices** - XY(Z) points (in the layer's coordinate reference system)
- **edges** - connect pairs of vertices
- **faces** - a face is a set of edges forming a closed shape - typically a triangle or a quadrilateral (quad), rarely polygons with more vertices

Relying on the above, mesh layers can thus have different types of structure:

- 1D Meshes: consist of vertices and edges. An edge connects two vertices and can have assigned data (scalars or vectors) on it. The 1D mesh network can be for example used for modelling of an urban drainage system.
- 2D meshes: consist of faces with triangles, regular or unstructured quads.
- 3D layered meshes: consist of multiple stacked 2D unstructured meshes each extruded in the vertical direction (levels) by means of a vertical coordinate. The vertices and faces have the same topology in each vertical level. The mesh definition (vertical level extrusion) could in general change in time. The data is usually defined in volume centres or by some parametric function.

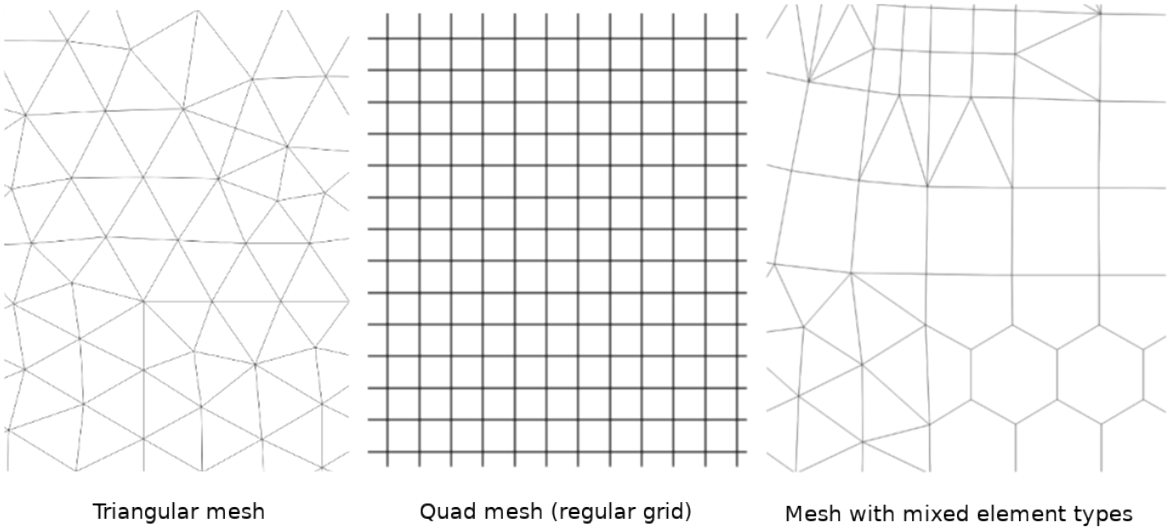


Fig. 14.1: Different mesh types

Mesh provides information about the spatial structure. In addition, the mesh can have datasets (groups) that assign a value to every vertex. For example, having a triangular mesh with numbered vertices as shown in the image below:

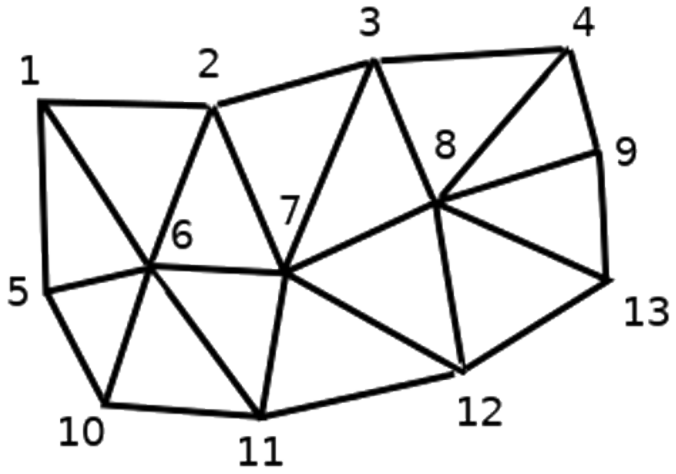


Fig. 14.2: Triangular grid with numbered vertices

Each vertex can store different datasets (typically multiple quantities), and those datasets can also have a temporal dimension. Thus, a single file may contain multiple datasets.

The following table gives an idea about the information that can be stored in mesh datasets. Table columns represent indices of mesh vertices, each row represents one dataset. Datasets can have different datatypes. In this case, it stores wind velocity at 10m at a particular moments in time (t1, t2, t3).

In a similar way, the mesh dataset can also store vector values for each vertex. For example, wind direction vector at the given time stamps:

Table 14.1: Example of mesh dataset

10 metre wind	1	2	3	...
10 metre speed at time=t1	17251	24918	32858	...
10 metre speed at time=t2	19168	23001	36418	...
10 metre speed at time=t3	21085	30668	17251	...
...	...	...	...	...
10m wind direction time=t1	[20,2]	[20,3]	[20,4.5]	...
10m wind direction time=t2	[21,3]	[21,4]	[21,5.5]	...
10m wind direction time=t3	[22,4]	[22,5]	[22,6.5]	...
...	...	...	...	...

We can visualize the data by assigning colors to values (similarly to how it is done with *Singleband pseudocolor* raster rendering) and interpolating data between vertices according to the mesh topology. It is common that some quantities are 2D vectors rather than being simple scalar values (e.g. wind direction). For such quantities it is desirable to display arrows indicating the directions.

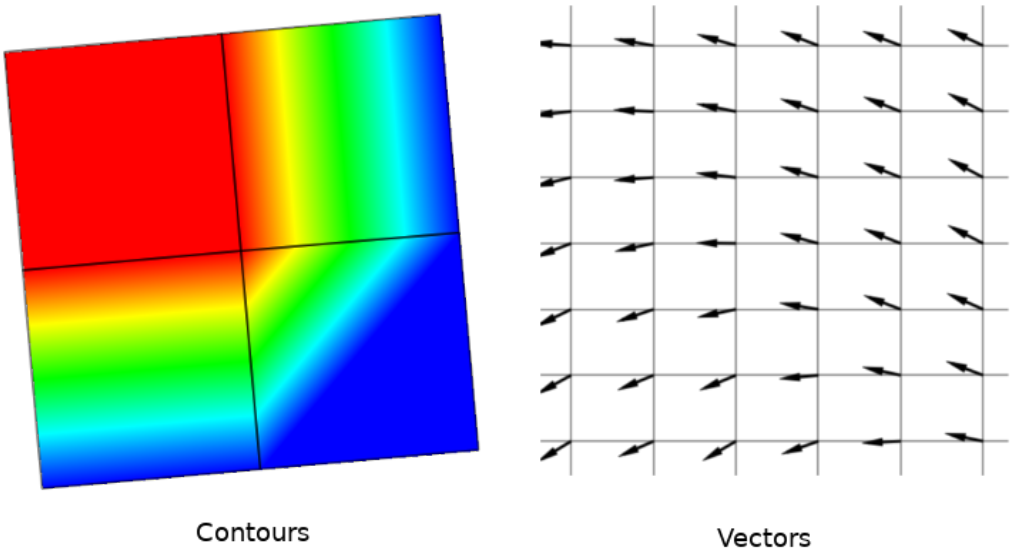


Fig. 14.3: Possible visualisation of mesh data

14.2 Supported formats

QGIS accesses mesh data using the MDAL drivers, and natively supports a variety of formats. Whether QGIS can edit a mesh layer depends on the format and the mesh structure type.

To load a mesh dataset into QGIS, use the *Mesh* tab in the *Data Source Manager* dialog. Read 加载网孔图层 for more details.

14.3 Mesh Dataset Properties

The *Layer Properties* dialog for a mesh layer provides general settings to manage dataset groups of the layer and their rendering (active dataset groups, symbology, 2D and 3D rendering). It also provides information about the layer.

要访问 图层属性对话框，请执行以下操作：

- 在 图层面板，双击该图层或右键单击并从弹出式菜单选择 属性... ；
- 选择图层后，转到 图层 ► 图层属性... 菜单。

The mesh *Layer Properties* dialog provides the following sections:

Table 14.2: Tabs of the Mesh Layer Properties

Information	*Source*	*Symbology* [1]
Labels [1]	*3D View* [1]	*Rendering*
Temporal	*Elevation*	*Metadata*

Note

Most of the properties of a mesh layer can be saved to or loaded from a `.qml` using the *Style* menu at the bottom of the dialog. More details at 管理自定义样式.

[1] Also available in the *Layer styling panel*

14.3.1 信息属性

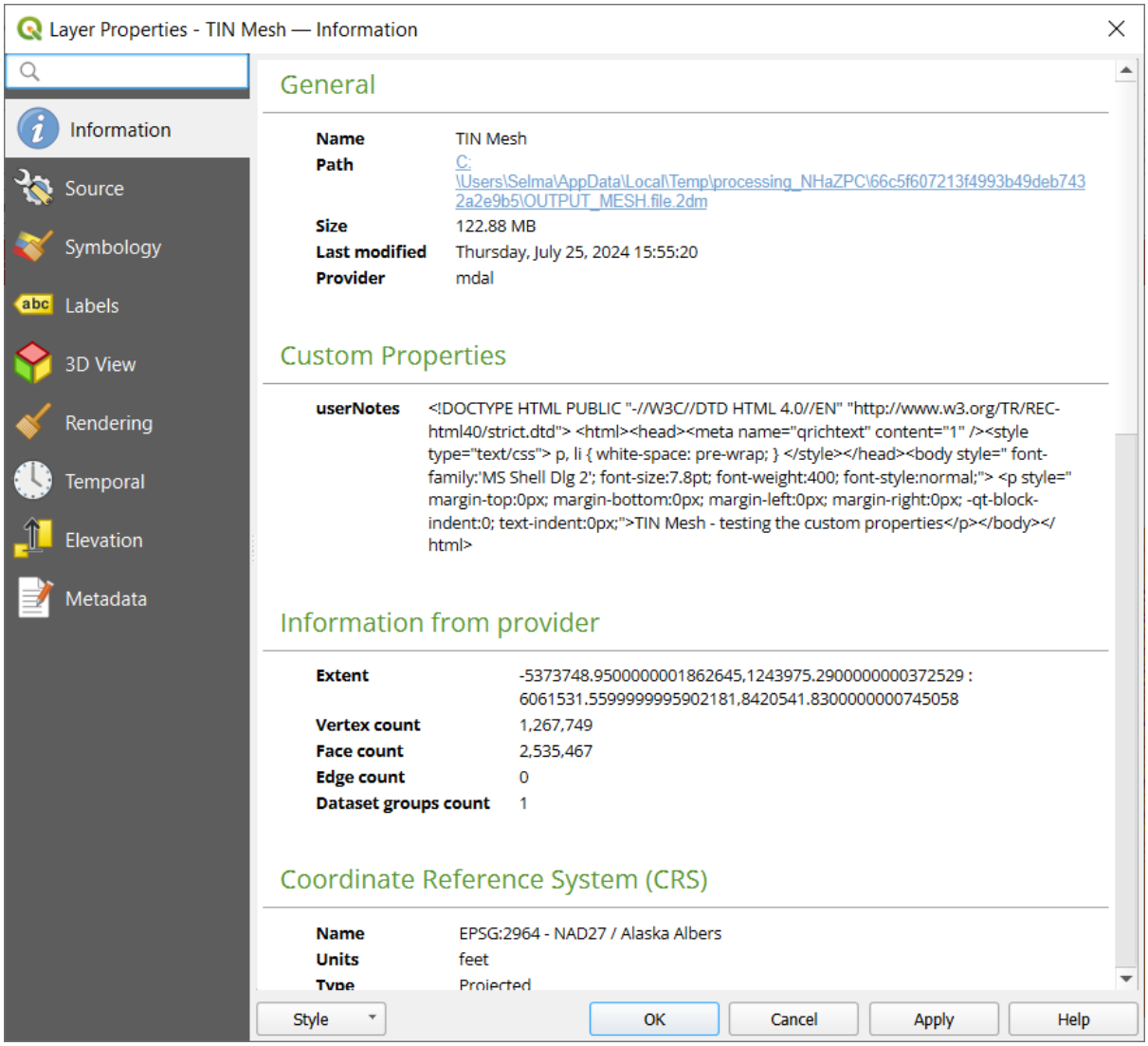


Fig. 14.4: Mesh Layer Information Properties

The *Information* tab is read-only and represents an interesting place to quickly grab summarized information and metadata on the current layer. Provided information are:

- general such as name in the project, source path, list of auxiliary files, last save time and size, the used provider
- custom properties, used to store in the active project additional information about the layer. Default custom properties may include *layer notes*. More properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- based on the provider of the layer: extent, vertex, face, edges and/or dataset groups count
- 坐标参照系：名称、单位、方法、精度、参照 (即静态还是动态)
- extracted from filled *metadata*: access, extents, links, contacts, history...

14.3.2 源属性

The *Source* tab displays basic information about the selected mesh, including:

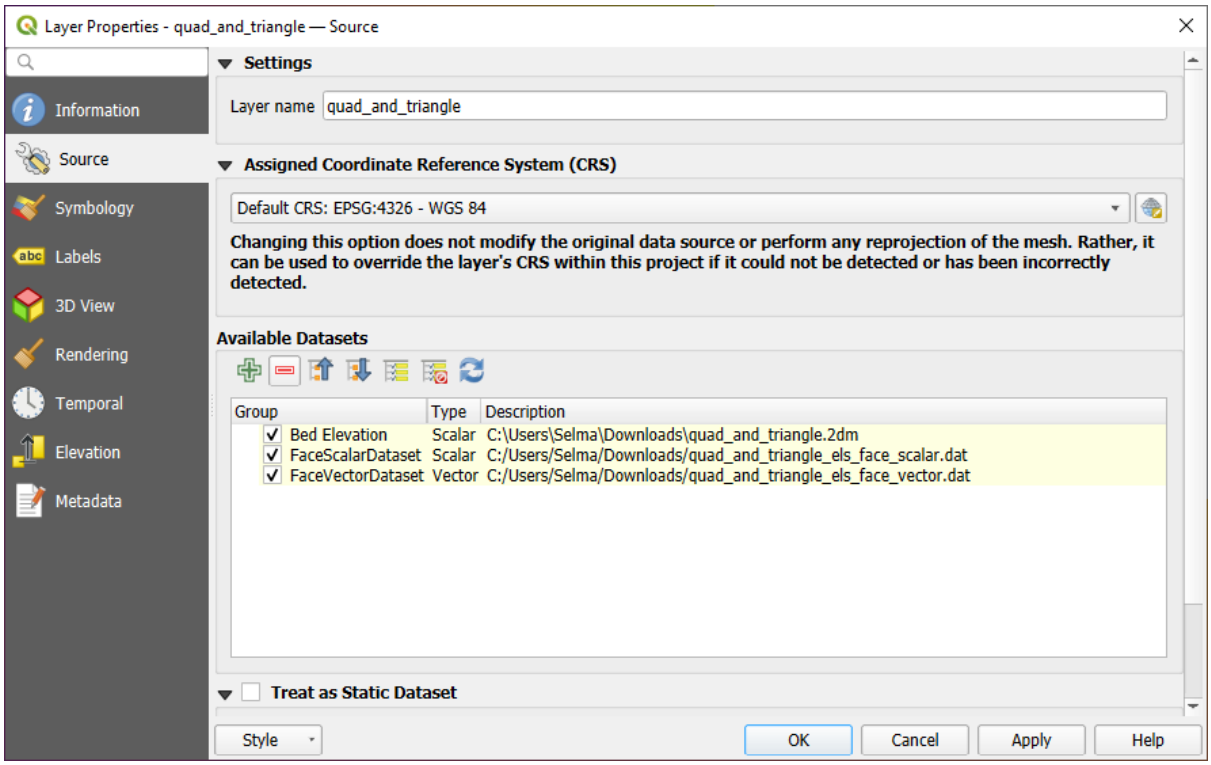


Fig. 14.5: Mesh Layer Source Properties

- the layer name to display in the *Layers* panel
- setting the Coordinate Reference System: Displays the layer's *Assigned Coordinate Reference System (CRS)*. You can change the layer's CRS by selecting a recently used one in the drop-down list or clicking on *Select CRS* button (see 坐标参照系选择器). Use this process only if the CRS applied to the layer is wrong or if none was applied.
- The *Available datasets* frame lists all the dataset groups (and subgroups) in the mesh layer, with their type and description in a tree view. Both regular datasets (i.e. their data is stored in the file) and virtual datasets (which are *calculated on the fly*) are listed.
  - Use the Assign Extra Dataset to Mesh button to add more groups to the current mesh layer. You can add dataset group to a mesh layer with the same name, but not from the same URI. Dataset group names are automatically renamed to "Original Name_Number".
  - Use Remove Extra Dataset from Mesh to remove additional datasets groups from the mesh layer. Note that only dataset groups not associated with the mesh source file can be removed.
  - Collapse all and Expand all the dataset tree, in case of embedded groups
  - If you are interested in few datasets, you can uncheck the others and make them unavailable in the project
  - Double-click over a name and you can rename the dataset.
  - Reset to defaults: checks all the groups and renames them back to their original name in the provider.
  - Right-click over a virtual dataset group and you can:
    - *Remove dataset group* from the project

    * *Save dataset group as...* a file on disk, to any supported format. The new file is kept assigned to the current mesh layer in the project.

- Checking the *Treat as static dataset* group allows to ignore the *map temporal navigation* properties while rendering the mesh layer. For each active dataset group (as selected in *Symbology* ► *Datasets* tab), you can:
  - set to *None*: the dataset group is not displayed at all
  - *Display dataset*: e.g., for the "bed elevation" dataset which is not time aware
  - extract a particular date time: the dataset matching the provided time is rendered and stays fixed during map navigation.

14.3.3 Symbology Properties

Click the *Symbology* button to activate the dialog. Symbology properties are divided into several tabs:

- *Datasets*
- *Contours*
- *Vectors*
- *Rendering*
- *Stacked mesh averaging method*

Datasets

The tab Datasets is the main place to control and set which datasets will be used for the layer. It presents the following items:

- *Groups* available in the mesh dataset, with whether they provide:
  - scalar dataset
  - or vector dataset: by default, each vector dataset has a scalar dataset representing its magnitude automatically generated.

  Click on the icon next to the dataset name to select the group and type of data to represent.
- *Selected dataset group(s) metadata*, with details on:
  - the mesh type: edges or faces
  - the data type: vertices, edges, faces or volume
  - whether it's of vector type or not
  - the original name in the mesh layer
  - the unit, if applicable
- *blending mode* available for the selected datasets.

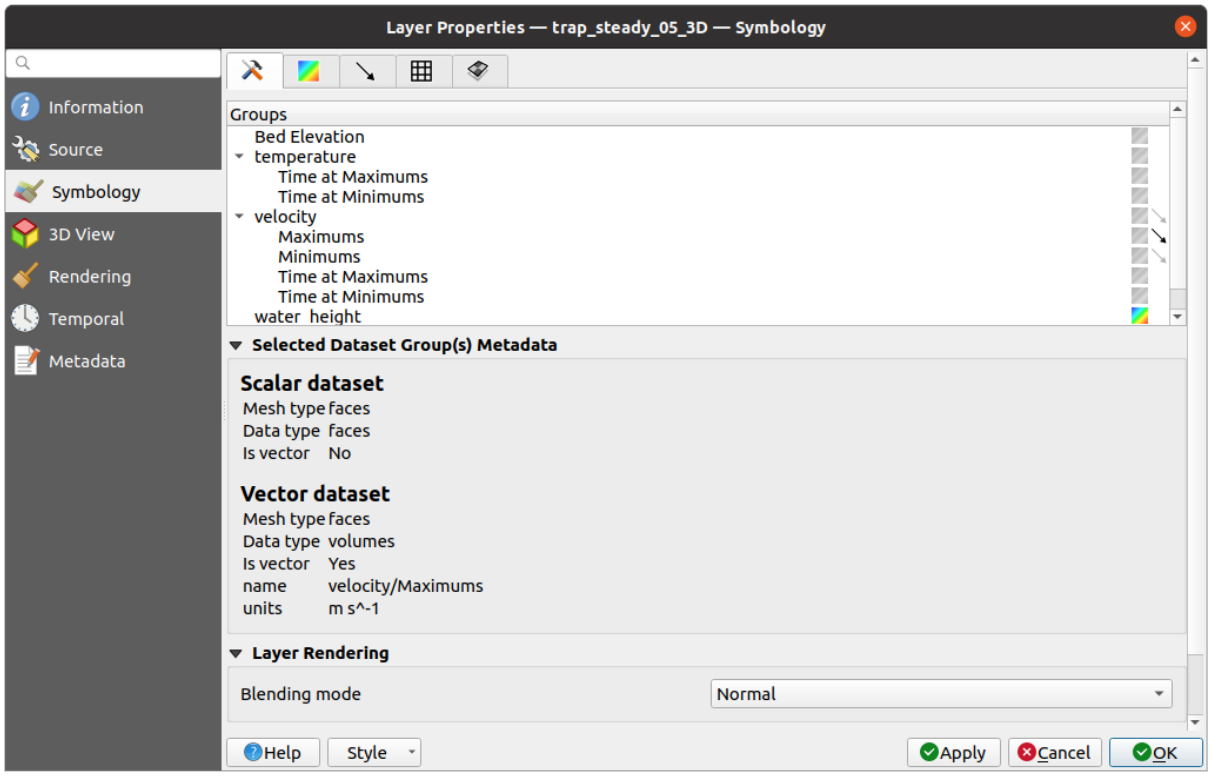


Fig. 14.6: Mesh Layer Datasets

You can apply symbology to the selected vector and/or scalar group using the next tabs.

Contours Symbology

> **Note**
>
> The Contours tab can be activated only if a scalar dataset has been selected in the *Datasets* tab.

In the Contours tab you can see and change the current visualization options of contours for the selected group, as shown in Fig. 14.7 below:

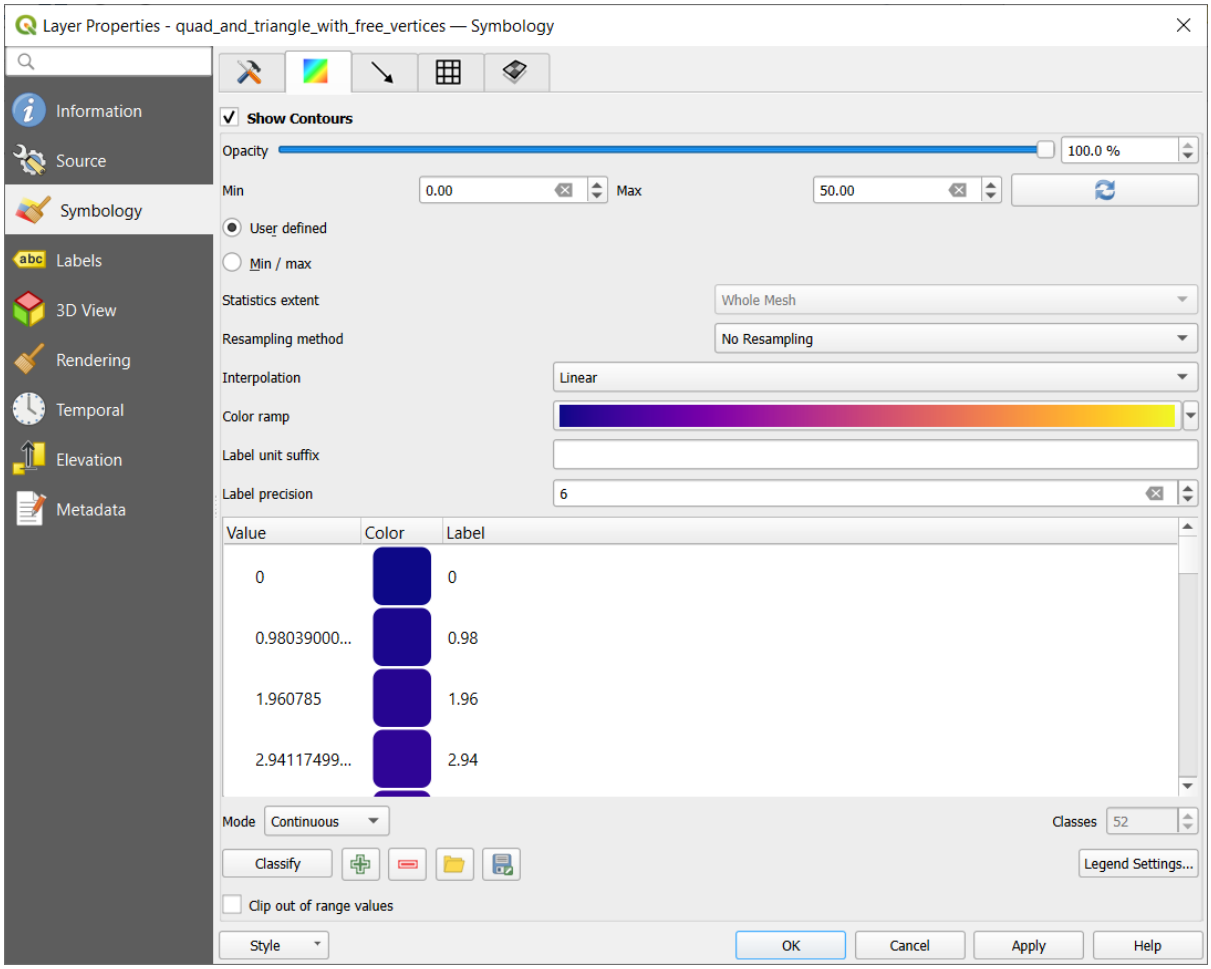


Fig. 14.7: Styling Contours in a Mesh Layer

- For 1D mesh, set the *Stroke width* of the edges. This can be a fixed size for the whole dataset, or vary along the geometry (more details with the *interpolated line renderer*)
- Use the slider or the spinbox to set the *Opacity* of the current group, if of a 2D mesh type.
- *User Defined* allows you to enter the range of values you want to represent on the current group: use Load to fetch the min and max values of the current group or enter custom values if you want to exclude some.
- Select *Min/Max* to set the renderer's minimum and maximum values based on the chosen extent.
- *Statictics extent* can be *Whole mesh*, *Current canvas* or *Updated canvas*. *Updated canvas* means that min/max values used for the rendering will change with the canvas extent (dynamic stretching).
- For 2D/3D meshes, select the *Resampling method* to interpolate the values on the surrounding vertices to the faces (or from the surrounding faces to the vertices) using the *Neighbour average* method. Depending on whether the dataset is defined on the vertices (respectively on the faces), QGIS defaults this setting to *None* (respectively *Neighbour average*) method in order to use values on vertices and keep the default rendering smooth.
- Classify the dataset using the *color ramp shader* classification.

Vectors Symbology

Note

The Vectors tab can be activated only if a vector dataset has been selected in the *Datasets* tab. Click on the Vectors icon on the right side of the *Datasets* tab.

In the Vectors tab you can see and change the current visualization options of vectors for the selected group, as shown in Fig. 14.8:

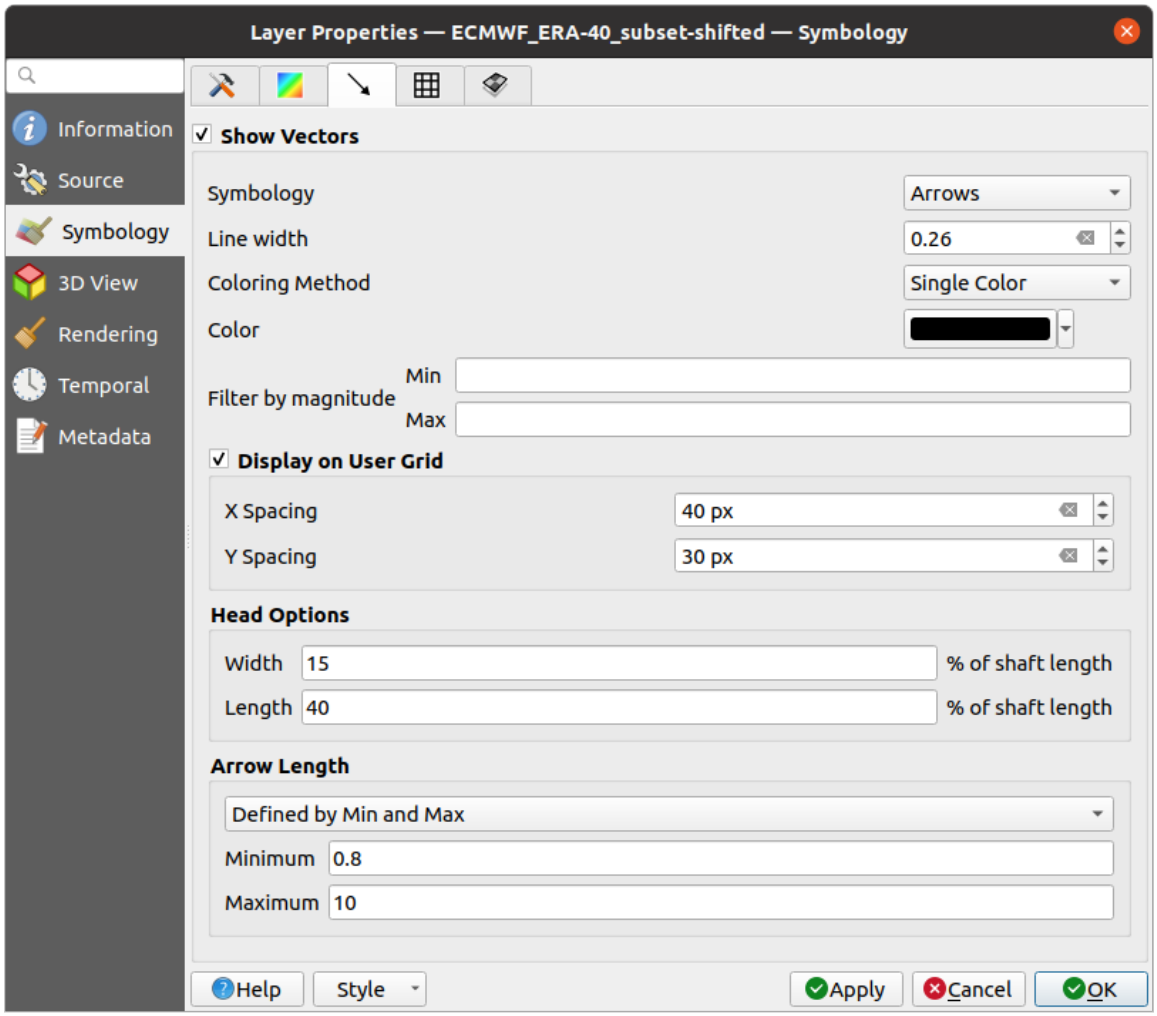


Fig. 14.8: Styling Vectors in a Mesh Layer with arrows

Mesh vector dataset can be styled using various types of *Symbology*:

- **Arrows**: vectors are represented with arrows at the same place as they are defined in the raw dataset (i.e. on the nodes or center of elements) or on a user-defined grid (hence, they are evenly distributed). The arrow length is proportional to the magnitude of the arrow as defined in the raw data but can be scaled by various methods.
- **Streamlines**: vectors are represented with streamlines seeded from start points. The seeding points can start from the vertices of the mesh, from a user grid or randomly.
- **Traces**: a nicer animation of the streamlines, the kind of effect you get when you randomly throws sand in the water and see where the sand items flows.

- **Wind barbs**: vectors are represented with wind barbs, a common way to represent wind speed and direction.

Available properties depend on the selected symbology as shown in the following table.

Table 14.3: Availability and meaning of the vectors symbology properties

Label, Description and Properties	Arrow	Streamlines	Traces	Wind Barbs
Line width: Width of the vector representation	☑	☑	☑	☑
Coloring method: • a *Single color* assigned to all vectors • or a variable color based on vectors magnitude, using a *Color ramp shader*	☑	☑	☑	☑
Filter by magnitude: Only vectors whose length for the selected dataset falls between a *Min* and *Max* range are displayed	☑	☑		☑
Display on user grid: Places the vector on a grid with custom *X spacing* and *Y spacing* and interpolates their length based on neighbours	☑	☑		☑
Head options: *Length* and *Width* of the arrow head, as a percentage of its shaft length	☑			
Arrow length: • **Defined by Min and Max**: You specify the minimum and maximum length for the arrows, QGIS will interpolate their size based on the underlying vector's magnitude • **Scale to magnitude**: arrow length is proportional to their vector's magnitude • **Fixed**: all the vectors are shown with the same length	☑			
Streamlines seeding method: • **On mesh/grid**: relies on the user grid to display the vectors • **Randomly**: vector placement is randomly done with respect to a certain density		☑		
Particles count: The amount of "sand" you want to throw into visualisation			☑	
Max tail length: The time until the particle fades out			☑	
Wind Barbs: • **Length**: Specifies the length of the wind barbs. Choose the units from a drop-down menu. • **Data Units**: Specifies the units of wind speed data. Wind barbs encode the wind in knots (nautical miles per hour). Therefore, you should select the units that your data are in, or use a custom *Multiplier* to convert your data units to knots. This ensures that the wind speed is accurately represented.				☑

Rendering

In the tab Rendering tab, QGIS offers possibilities to display and customize the mesh structure. *Line width* and *Line color* can be set to represent:

- the edges for 1D meshes
- For 2D meshes:
  - *Native mesh rendering*: shows original faces and edges from the layer
  - *Triangular mesh rendering*: adds more edges and displays the faces as triangles

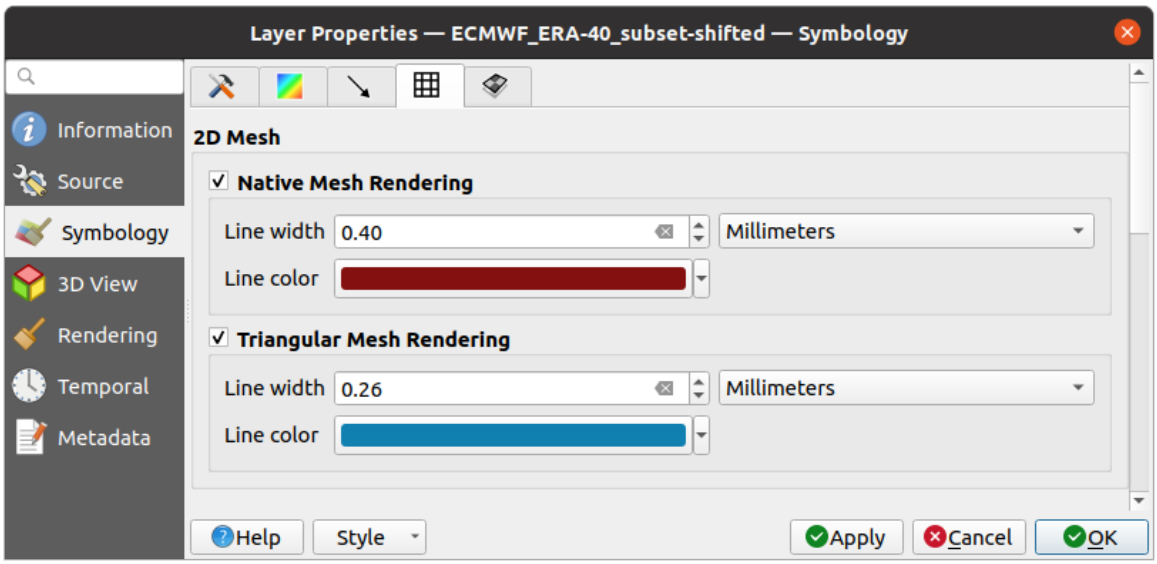


Fig. 14.9: 2D Mesh Rendering

Stacked mesh averaging method

3D layered meshes consist of multiple stacked 2D unstructured meshes each extruded in the vertical direction (`levels`) by means of a vertical coordinate. The vertices and faces have the same topology in each vertical level. Values are usually stored on the volumes that are regularly stacked over base 2d mesh. In order to visualise them on 2D canvas, you need to convert values on volumes (3d) to values on faces (2d) that can be shown in mesh layer. The Stacked mesh averaging method provides different averaging/interpolation methods to handle this.

You can select the method to derive the 2D datasets and corresponding parameters (level index, depth or height values). For each method, an example of application is shown in the dialog but you can read more on the methods at https://fvwiki.tuflow.com/Depth_Averaging_Results.

14.3.4 Labels Properties

The *Labels* tab offers you dynamic labeling of mesh vertices and native mesh faces based on geometric properties and custom expressions. This dialog can also be accessed from the *Layer Styling* panel.

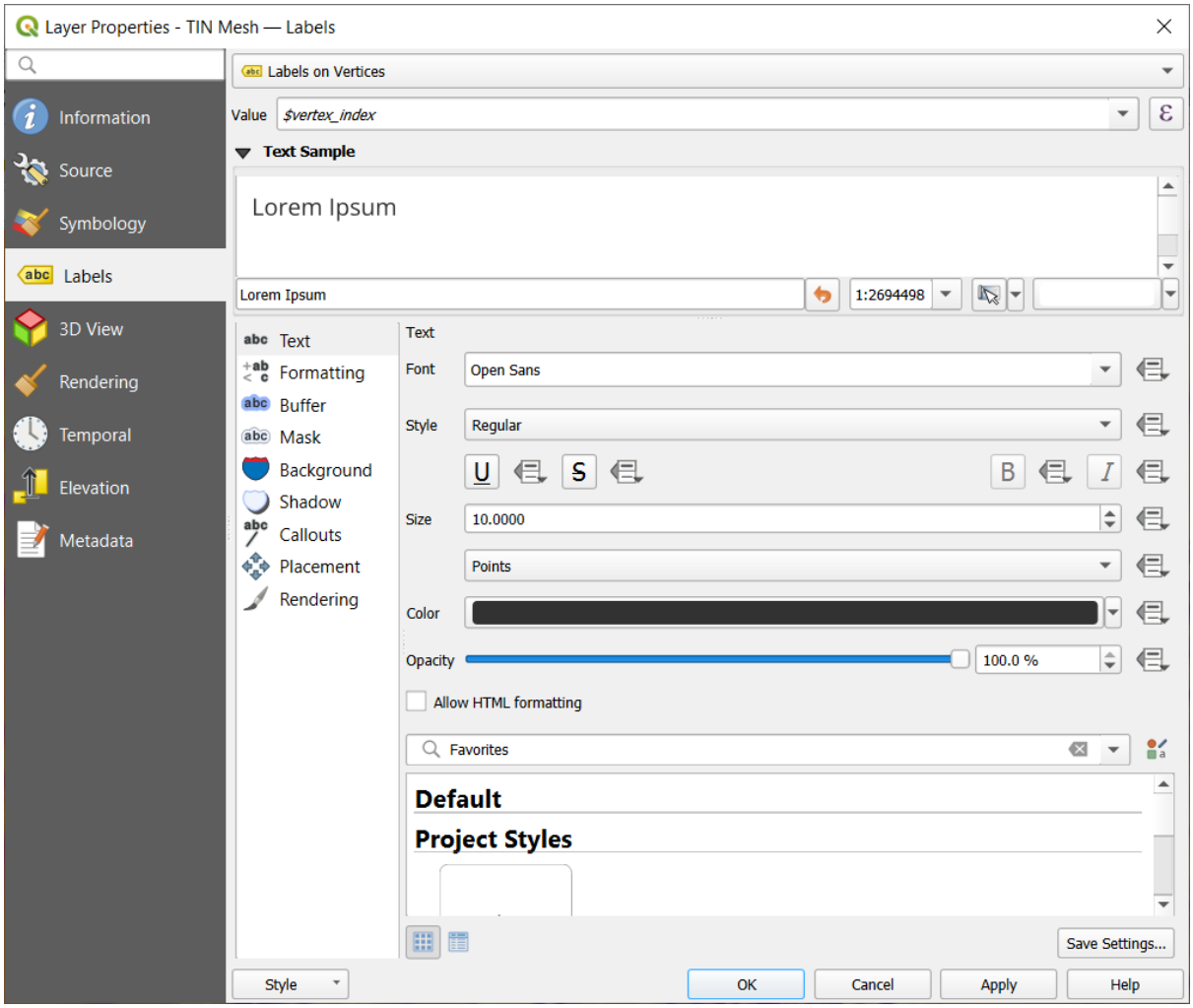


Fig. 14.10: Mesh Labels Properties

Choose the labeling method from the drop-down list. Available methods are:

- *No labels*: the default value, showing no labels from the layer.
- *Labels on vertices*: labels the mesh vertices as a point feature.
- *Labels on faces*: labels the mesh faces as a polygon feature.

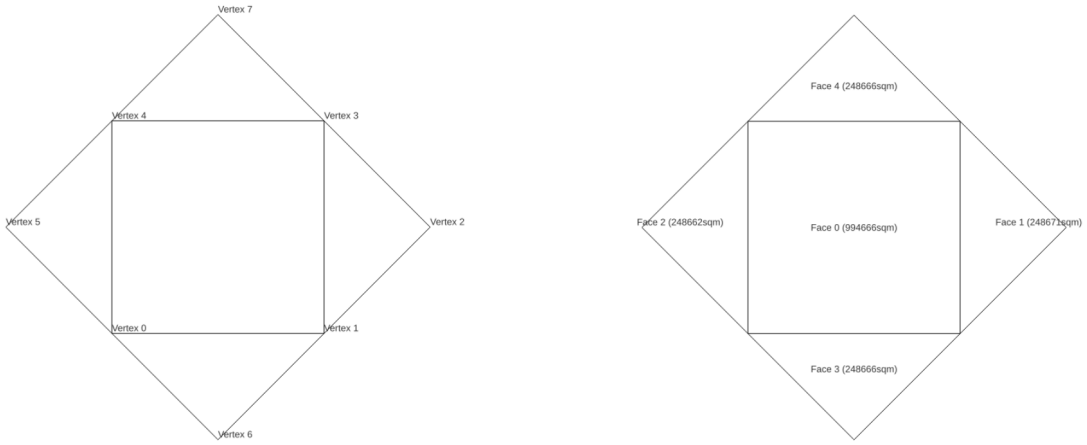


Fig. 14.11: Mesh Labeling methods

Once the labeling method has been selected, open Expression dialog and choose a *mesh function*. You can combine meshes functions with other functions from the *list* to modify the label.

Below are displayed options to customize the labels, under various tabs:

- *Text*
- *Formatting*
- *Buffer*
- *Mask*
- *Background*
- *Shadow*
- *Callouts*
- *Placement*
- *Rendering*

Description of how to set each property is exposed at 设置标注.

14.3.5 3D View Properties

Mesh layers can be used as *terrain in a 3D map view* based on their vertices Z values. From the *3D View* properties tab, it's also possible to render the mesh layer's dataset in the same 3D view. Therefore, the vertical component of the vertices can be set equal to dataset values (for example, level of water surface) and the texture of the mesh can be set to render other dataset values with color shading (for example velocity).

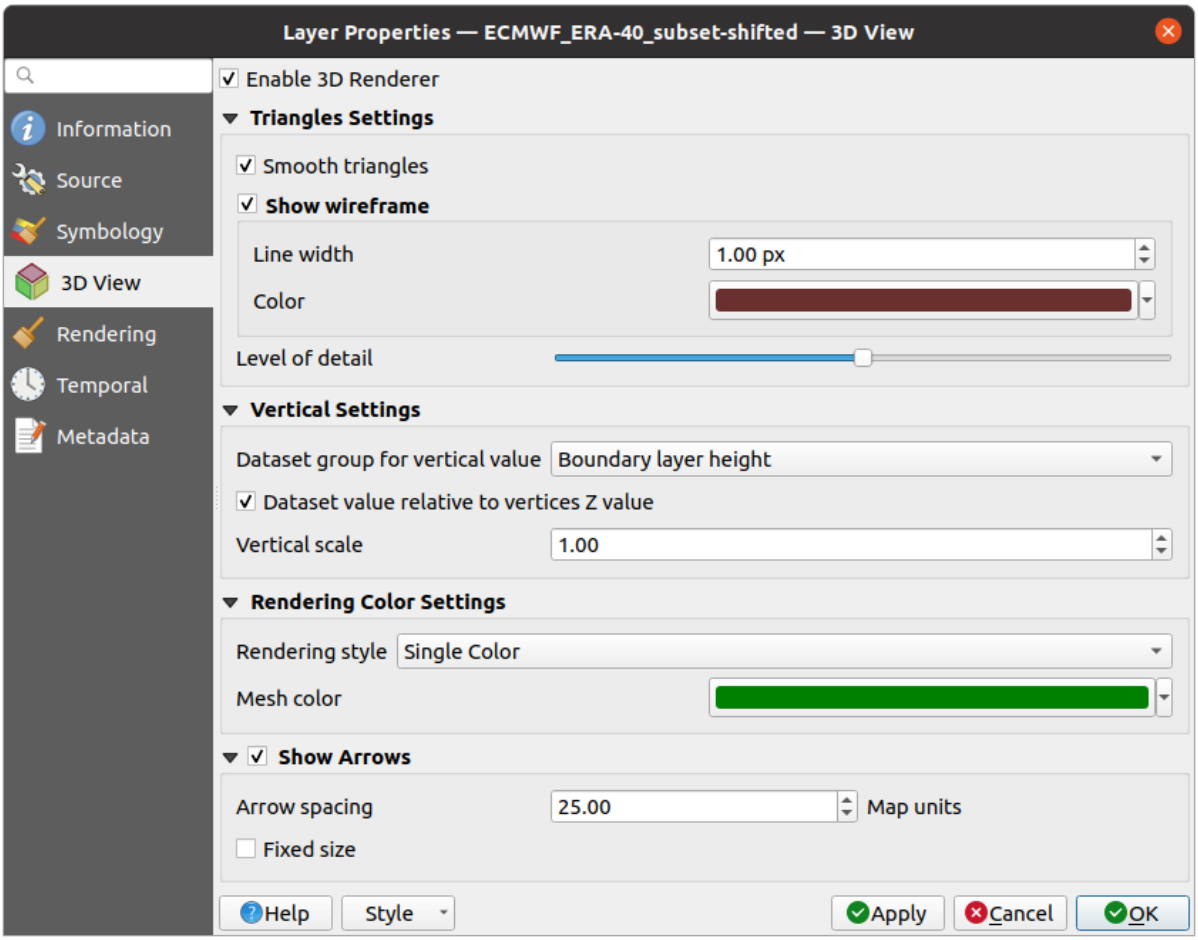


Fig. 14.12: Mesh dataset 3D properties

Check *Enable 3D Renderer* and you can edit following properties:

- Under *Triangle settings*
  - *Smooth triangles*: Angles between consecutive triangles are smoothed for a better 3D rendering
  - *Show wireframe* whose you can set the *Line width* and *Color*
  - *Level of detail*: Controls how *simplified* the mesh layer to render should be. On the far right, it is the base mesh, and the more you go left, the more the layer is simplified and is rendered with less details. This option is only available if the *Simplify mesh* option under the *Rendering* tab is activated.
- *Vertical settings* to control behavior of the vertical component of vertices of rendered triangles:
  - *Dataset group for vertical value*: the dataset group that will be used for the vertical component of the mesh
  - *Dataset value relative to vertices Z value*: whether to consider the dataset values as absolute Z coordinate or relative to the vertices native Z value
  - *Vertical scale*: the scale factor to apply to the dataset Z values
- *Rendering color settings* with a *Rendering style* that can be based on the color ramp shader set in *Contours Symbology* (*2D contour color ramp shader*) or as a *Single color* with an associated *Mesh color*
- *Show arrows*: displays arrows on mesh layer dataset 3D entity, based on the same vector dataset group used in the *vector 2D rendering*. They are displayed using the 2D color setting. It's also possible to define the *Arrow spacing* and, if it's of a *Fixed size* or scaled on magnitude. This spacing setting defines also the max size of arrows because arrows can't overlap.

14.3.6 Rendering Properties

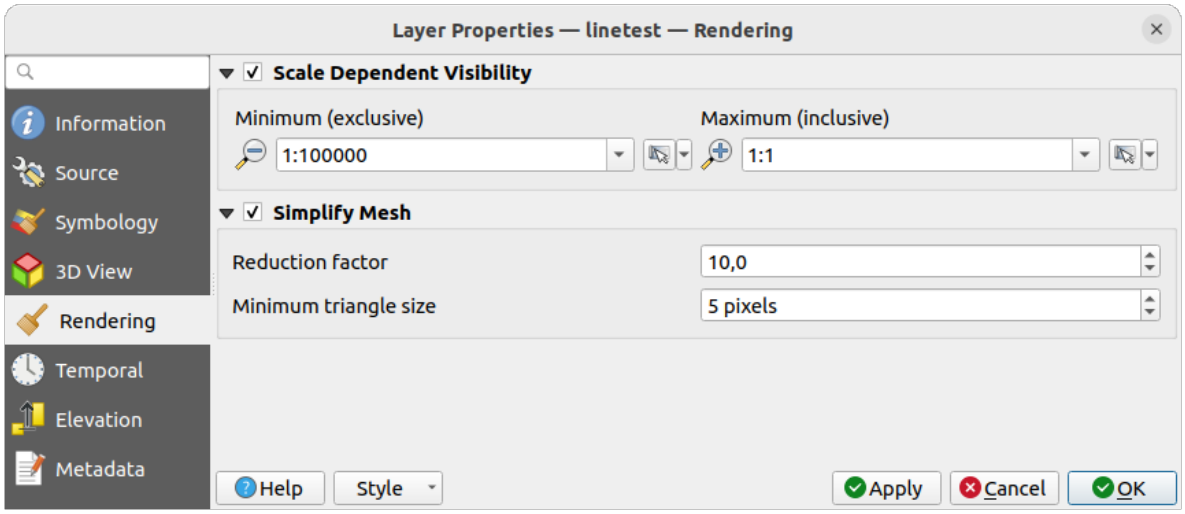


Fig. 14.13: Mesh rendering properties

Under the *Scale dependent visibility* group box, you can set the *Maximum (inclusive)* and *Minimum (exclusive)* scale, defining a range of scale in which mesh elements will be visible. Out of this range, they are hidden. The
Set to current canvas scale button helps you use the current map canvas scale as boundary of the range visibility. See 可见比例选择器 for more information.

> **Note**
>
> You can also activate scale dependent visibility on a layer from within the *Layers* panel: right-click on the layer and in the contextual menu, select *Set Layer Scale Visibility*.

As mesh layers can have millions of faces, their rendering can sometimes be very slow, especially when all the faces are displayed in the view whereas they are too small to be viewed. To speed up the rendering, you can simplify the mesh layer, resulting in one or more meshes representing different *levels of detail* and select at which level of detail you would like QGIS to render the mesh layer. Note that the simplify mesh contains only triangular faces.

From the *Rendering* tab, check *Simplify mesh* and set:

- a *Reduction factor*: Controls generation of successive levels of simplified meshes. For example, if the base mesh has 5M faces, and the reduction factor is 10, the first simplified mesh will have approximately 500 000 faces, the second 50 000 faces, the third 5000,... If a higher reduction factor leads quickly to simpler meshes (i.e. with triangles of bigger size), it produces also fewer levels of detail.
- *Minimum triangle size*: the average size (in pixels) of the triangles that is permitted to display. If the average size of the mesh is lesser than this value, the rendering of a lower level of details mesh is triggered.

14.3.7 Temporal Properties

The *Temporal* tab provides options to control the rendering of the layer over time. It allows to dynamically display temporal values of the enabled dataset groups. Such a dynamic rendering requires the *temporal navigation* to be enabled over the map canvas.

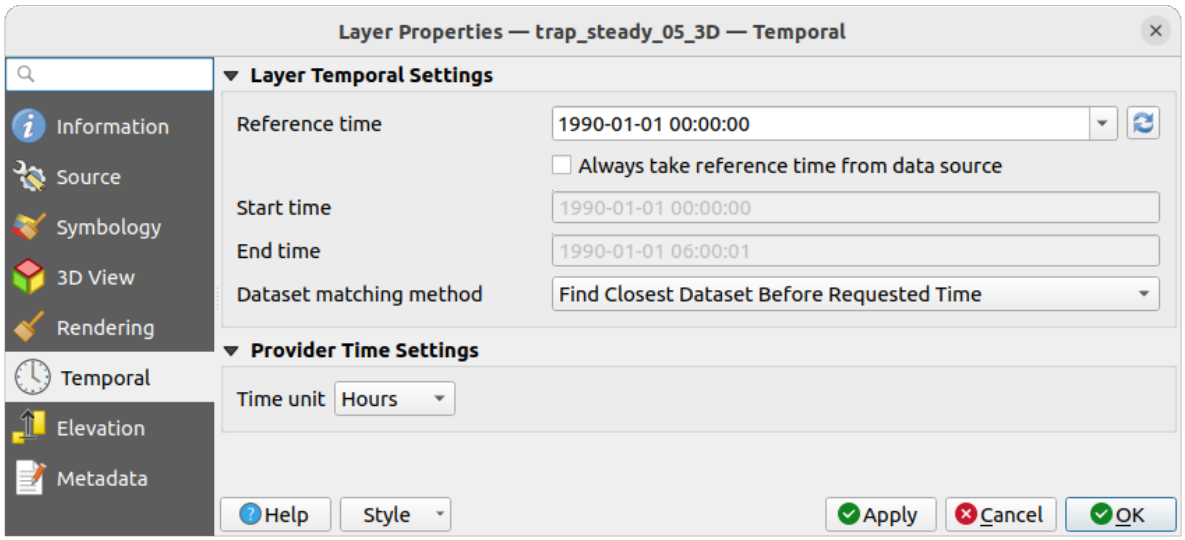


Fig. 14.14: Mesh Temporal properties

Layer temporal settings

- *Reference time* of the dataset group, as an absolute date time. By default, QGIS parses the source layer and returns the first valid reference time in the layer's dataset group. If unavailable, the value will be set by the project time range or fall back to the current date. The *Start time* and *End time* to consider are then calculated based on the internal timestamp step of the dataset.

  It is possible to set a custom *Reference time* (and then the time range), and revert the changes using the Reload from provider button. With *Always take reference time from data source* checked, you ensure that the time properties are updated from the file each time the layer is reloaded or the project reopened.

- *Dataset matching method*: determines the dataset to display at the given time. Options are *Find closest dataset before requested time* or *Find closest dataset from requested time (after or before)*.

Provider time settings

- *Time unit* extracted from the raw data, or user defined. This can be used to align the speed of the mesh layer with other layers in the project during map time navigation. Supported units are *Seconds*, *Minutes*, *Hours* and *Days*.

14.3.8 Elevation Properties

The *Elevation* tab provides options to control the layer elevation properties within a *3D map view* and *2D map view* and its appearance in the *profile tool charts*. Specifically, you can configure how heights from your dataset are interpreted:

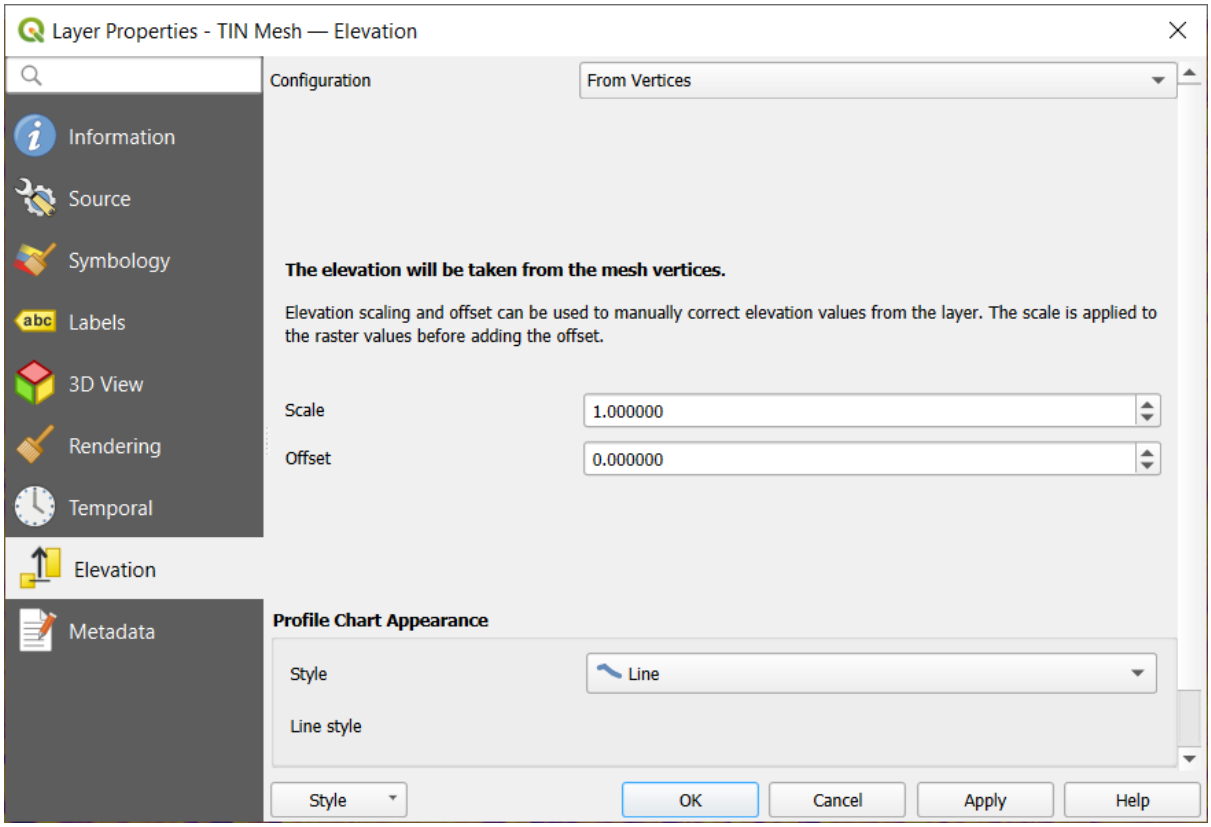


Fig. 14.15: Mesh Elevation properties

- *From vertices*: how the mesh layer vertices Z values should be interpreted as terrain elevation. You can apply a *Scale* factor and an *Offset*. This setting is available for *3D map view* and *profile tool charts*.
- *Fixed Elevation Range*: the mesh layer is linked to a fixed elevation range. This mode is applicable when a layer has either a single fixed elevation or a range (slice) of elevation values. If a range is specified, mesh values will be extruded over this range. You can set the *Lower* and *Upper* elevation range values for the layer, and specify whether the lower or upper *Limits* are inclusive or exclusive. When enabled, the layer will only be visible in *elevation filtered 2D maps* when the layer's range is included in the map's Z range.
- *Fixed Elevation Range Per Group*: each group in the mesh layer is associated with a fixed elevation range. This mode can be used when a layer has elevation data exposed through different dataset groups. This feature is exposed as a user-editable table for dataset groups with lower and upper values. You can either populate the lower and upper values manually or use an *Expression* to auto-fill all group values based on an expression. When enabled, the layer will be filtered in the *2D map view*, displaying only values within the map filter ranges. In *3D map view* or *profile tool charts*, the full extent of the layer will be displayed, ignoring the filtering.
- *Profile Chart Appearance*: controls the rendering of the mesh elements elevation in the profile chart. The profile *Style* can be set as:
  - a *Line* with a specific *Line style*
  - an elevation surface rendered using a fill symbol either above (*Fill above*) or below (*Fill below*) the elevation curve line. The surface symbology is represented using:
    - a *Fill style*
    - and a *Limit*: the maximum (respectively minimum) altitude determining how high the fill surface will be

14.3.9 元数据属性

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

14.4 Exploring mesh datasets

To identify mesh elements and their associated datasets, use the *Identify Features* tool from the *Attributes* toolbar. Click on a mesh element in the map canvas and the *Identify Results* panel opens, displaying information about the mesh layer and the identified element. The panel shows:

- Layer name
- Active dataset groups (scalar and/or vector) for the clicked element will be displayed first. On the other hand, inactive dataset groups are shown afterwards. For each group, the following derived information may also be shown, if applicable:
  - **Source** of the dataset (the file where it is stored).
  - **Time step** of the dataset value being displayed. For non-temporal dataset groups, this field remains empty.
  - **Vector x-component** and **Vector y-component** values for vector datasets.

Please note that when temporal navigation is disabled, the results only contain values from the *static dataset groups* defined.

- Geometry information shows the coordinates of the clicked point and may contain derived information in the following:
  - Centroid of the face.
  - Coordinates of a snapped vertex (if the click was snapped to a vertex).
  - Center of a snapped edge (if the click was snapped to an edge).

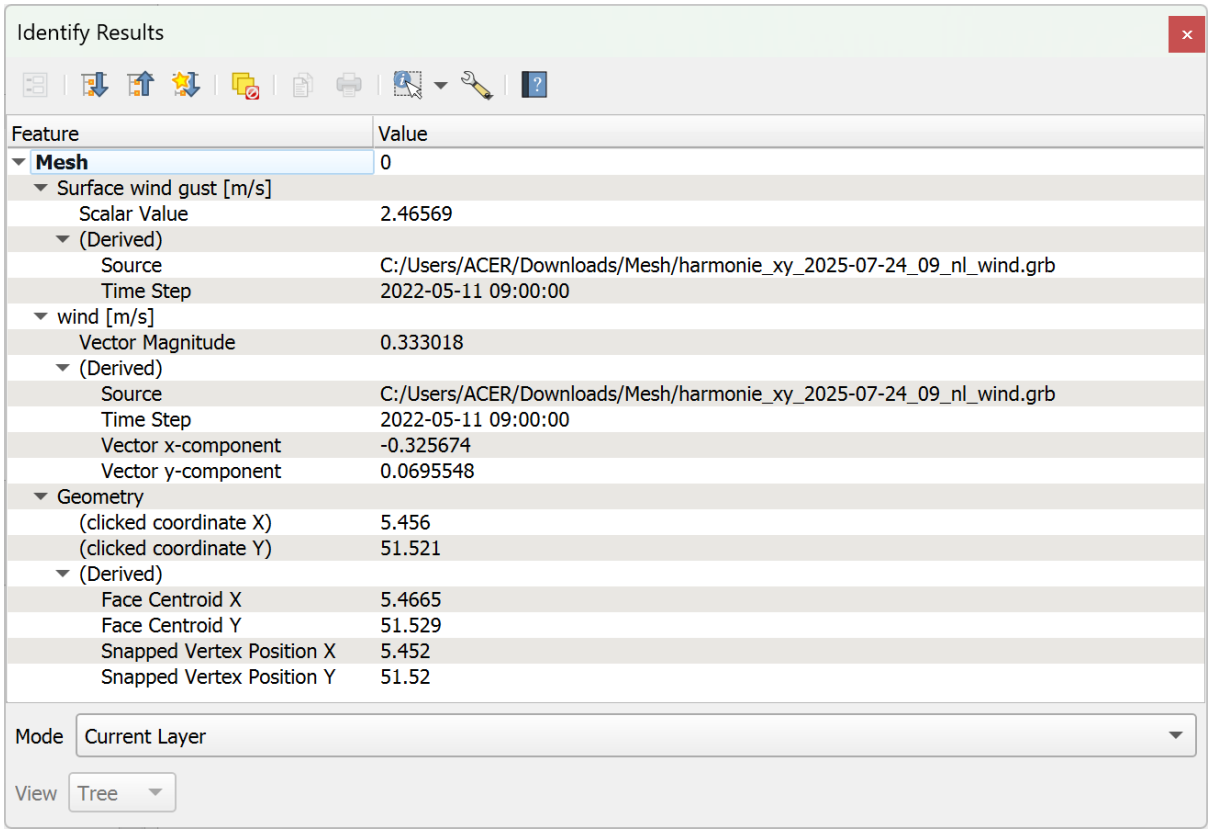


Fig. 14.16: Mesh Identify Results panel

14.5 Editing a mesh layer

QGIS allows to *create a mesh layer* from scratch or based on an existing one. You can create/modify the geometries of the new layer whom you can assign datasets afterwards. It's also possible to edit an existing mesh layer. Because the editing operation requires a frames-only layer, you will be asked to either remove any associated datasets first (make sure you have them available if they still are necessary) or create a copy (only geometries) of the layer.

Note

QGIS does not allow to digitize edges on mesh layers. Only vertices and faces are mesh elements that can be created. Also not all supported mesh formats can be edited in QGIS (see permissions).

14.5.1 Overview of the mesh digitizing tools

To interact with or edit a base mesh layer element, following tools are available.

Table 14.4: Tools for mesh digitizing

Label	目标	Location
Current Edits	Access to save, rollback or cancel changes in all or selected layers simultaneously	*Digitizing* toolbar
Toggle to Edit	Turn on/off the layer in edit mode	*Digitizing* toolbar
Save Edits	Save changes done to the layer	*Digitizing* toolbar
Undo	Undo the last change(s) - `Ctrl+Z`	*Digitizing* toolbar
Redo	Redo the last undone action(s) - `Ctrl+Shift+Z`	*Digitizing* toolbar
Enable Advanced Digitizing tools	Turn on/off the *Advanced Digitizing Panel*	*Advanced Digitizing* toolbar
Reindex Faces and Vertices	Recreate index and renumber the mesh elements for optimization	*Mesh* menu
Digitize Mesh Elements	Select/Create vertices and faces	*Mesh Digitizing* toolbar
Select Mesh Elements by Polygon	Select vertices and faces overlapped by a drawn polygon	*Mesh Digitizing* toolbar
Select Mesh Elements by Expressio	Select vertices and faces using an expression	*Mesh Digitizing* toolbar
Transform Vertices Coordinates	Modify coordinates of a selection of vertices	*Mesh Digitizing* toolbar
Force by Selected Geometries	Split faces and constrain Z value using a linear geometry	*Mesh Digitizing* toolbar

14.5.2 Exploring the Z value assignment logic

When a mesh layer is turned into edit mode, a *Vertex Z value* widget opens at the top right of the map canvas. By default, its value corresponds to the *Default Z value* set in *Settings* ► *Options* ► *Digitizing* tab. When there are selected vertices, the widget displays the average Z value of the selected vertices.

During editing, the *Vertex Z value* is assigned to new vertices. It is also possible to set a custom value: edit the widget, press `Enter` and you will override the default value and make use of this new value in the digitizing process. Click the icon in the widget to reset its value to the Options default value.

Rules of assignment

This is configured in the drop-down menu of the *Digitize mesh elements* tool as *New vertex Z value* option. Following methods are available:

Table 14.5: Z value assignment methods

Method	Description
Project terrain	Z value is always taken or calculated from the *project reference terrain*. • If it is **Flat terrain**, the value corresponds to the *Terrain height* set in the project properties. • If it is a **Mesh** layer, the value is interpolated from the surrounding vertices of the mesh. • If it is a **DEM (raster layer)**, the value is assigned from the raster pixel value. • If there is no terrain under the new vertex, the value falls back to the *Default Z value* set in *digitizing options*.
Z widget	Z value is taken from the *Vertex Z value* widget or the *Advanced Digitizing Panel* *Z* widget (if it is in Locked state).
Prefer mesh, then Z widget	Interpolates the Z value from the active mesh layer if the vertex is on an edge or face, otherwise falls back to the *Vertex Z value* or *Advanced Digitizing Panel* *Z* widget (if it is in Locked state).
Prefer mesh, then terrain	Interpolates the Z value from the active mesh layer if the vertex is on the edge or face, otherwise uses the project reference terrain value.

The following detailed logic describes the behavior of the **Prefer mesh, then Z widget** strategy.

Table 14.6: Matrix of Z value assignment to new vertex

Vertex creation	Vertices selected?	Source of assigned value	Assigned Z Value
”Free” vertex, not connected to any face or edge of a face	No	*Vertex Z value*	Default or user defined
		Advanced Digitizing Panel (if *z* widget is in Locked state)	*z* widget if in Locked state
	Yes	*Vertex Z value*	Average of the selected vertices or user-defined
Vertex on an edge	---	Mesh layer	Interpolated from the edge’s vertices
Vertex on a face	---	Mesh layer	Interpolated from the face’s vertices
Vertex snapped to a 2D vector feature	---	*Vertex Z value*	Default or user defined
Vertex snapped to a 3D vector vertex	---	Vector layer	Vertex
Vertex snapped to a 3D vector segment	---	Vector layer	Interpolated along the vector segment

Note

The *Vertex Z value* widget is deactivated if the *Advanced Digitizing Panel* is enabled and no mesh element is selected. The latter’s *z* widget then rules the Z value assignment.

Modifying Z value of existing vertices

To modify the Z value of vertices, the most straightforward way is:

1. Select one or many vertices. The *Vertex Z value* widget will display the average height of the selection.
2. Change the value in the widget.
3. Press `Enter`. The entered value is assigned to the vertices and becomes the default value of next vertices.

Another way to change the Z value of a vertex is to move and snap it on a vector layer feature with the Z value capability. If more than one vertex are selected, the Z value can't be changed in this way.

The *Transform mesh vertices* dialog also provides means to modify the Z value of a selection of vertices (along with their X or Y coordinates).

14.5.3 Selecting mesh elements

To select mesh elements, you can use the following tools:

- Digitize Mesh Elements to select different mesh elements, see more at *Using Digitize Mesh Elements*.
- Select Mesh Elements by Polygon to select different mesh elements by polygon, see more at *Using Select Mesh Elements by Polygon*.
- Select Mesh Elements by Expression to select different mesh elements by expression, you can choose to *Select by vertices* or *Select by faces*, see more at *Using Select Mesh Elements by Expression*.
- Select Isolated Vertices to select all vertices that are not part of any mesh face.
- Select All Vertices to select all vertices of the mesh layer.

Using *Digitize Mesh Elements*

Activate the Digitize Mesh Elements tool. Hover over an element and it gets highlighted, allowing you to select it.

- Click on a vertex, and it is selected.
- Click on the small square at the center of a face or an edge, and it gets selected. Connected vertices are also selected. Conversely, selecting all the vertices of an edge or a face also selects that element.
- Drag a rectangle to select overlapping elements (a selected face comes with all their vertices). Press `Alt` key if you want to select only completely contained elements.
- To add elements to a selection, press `Shift` while selecting them.
- To remove an element from the selection, press `Ctrl` and reselect it. A deselected face will also deselect all their vertices.

Using *Select Mesh Elements by Polygon*

Activate the Select Mesh Elements by Polygon tool and:

- Draw a polygon (left-click to add vertex, `Backspace` to undo last vertex, `Esc` to abort the polygon and right-click to validate it) over the mesh geometries. Any partially overlapping vertices and faces will get selected. Press `Alt` key while drawing if you want to select only completely contained elements.
- Right-click over the geometry of a vector layer's feature, select it in the list that pops up and any partially overlapping vertices and faces of the mesh layer will get selected. Use `Alt` while drawing to select only completely contained elements.

- To add elements to a selection, press `Shift` while selecting them.
- To remove an element from the selection, press `Ctrl` while drawing over the selection polygon.

Using *Select Mesh Elements by Expression*

Another tool for mesh elements selection is Select Mesh Elements by Expression. When pressed, the tool opens the mesh *expression selector dialog* from which you can:

1. Select the method of selection:
   - *Select by vertices*: applies the entered expression to vertices, and returns matching ones and their eventually associated edges/faces
   - *Select by faces*: applies the entered expression to faces, and returns matching ones and their associated edges/vertices
2. Write the expression of selection. Depending on the selected method, available functions in the *Meshes group* will be filtered accordingly.
3. Run the query by setting how the selection should behave and pressing:
   - *Select*: replaces any existing selection in the layer
   - *Add to current selection*
   - *Remove from current selection*

14.5.4 Modifying mesh elements

Adding vertices

To add vertices to a mesh layer:

1. Press the Digitize mesh elements button
2. A *Vertex Z value* widget is always available on the top right corner of the map canvas during digitizing. It allows setting or overriding the Z coordinate assigned to new vertices, depending on the selected *Z value assignment method*.
3. You can choose other method for assigning Z values to new vertices from the drop-down menu in the Digitize mesh elements tool.
4. Then double-click:
   - outside a face: adds a ”free vertex”, that is a vertex not linked to any face. This vertex is represented by a red dot when the layer is in editing mode.
   - on the edge of existing face(s): adds a vertex on the edge, splits the touching face(s) into triangles connected to the new vertex.
   - inside a face: splits the face into triangles whose edges connect the surrounding vertices to the new vertex.

     With the *Refine neighboring faces when adding vertices* option enabled in the Digitize mesh elements button drop-down menu, a check is applied to triangular faces that share at least one vertex with the face the new vertex is added to. If their edges do not satisfy Delaunay triangulation rules, they are flipped accordingly.

Adding faces

To add faces to a mesh layer:

1. Press the Digitize mesh elements button
2. A *Vertex Z value* widget appears on the top right corner of the map canvas. Set this value to the Z coordinate you would like to assign to the subsequent vertices.
3. Hover over a vertex and click the small triangle that appears next to it.
4. Move the cursor to the next vertex position; you can snap to existing vertex or left-click to add a new one.
5. Proceed as above to add as many vertices as you wish for the face. Press `Backspace` button to undo the last vertex.
6. While moving the mouse, a rubberband showing the shape of the face is displayed. If it is shown in green, then the expected face is valid and you can right-click to add it to the mesh. If it is red, the face is not valid (e.g. because it self-intersects, overlaps an existing face or vertex, creates a hole, ...) and can't be added. You'd need to undo some vertices and fix the geometry.
7. Press `Esc` to abort the face digitizing.
8. Right-click to validate the face.

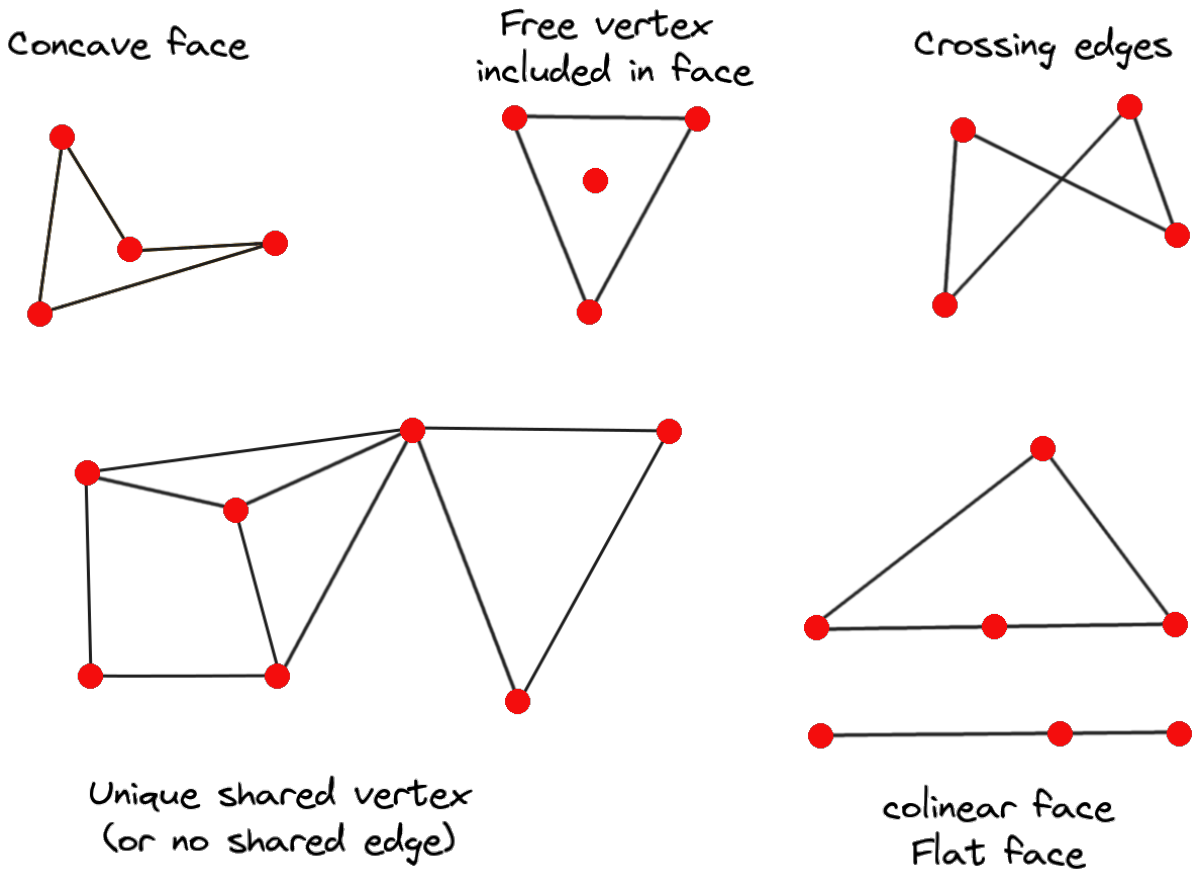


Fig. 14.17: Examples of invalid mesh

Removing mesh elements

1. *Select the target elements*
2. Enable the Digitize mesh elements tool
3. Right-click and select:
   - *Remove Selected Vertices and Fill Hole(s)* or press `Ctrl+Del`: removes vertices and linked faces and fills the hole(s) by triangulating from the neighbor vertices
   - *Remove Selected Vertices Without Filling Hole(s)* or press `Ctrl+Shift+Del`: removes vertices and linked faces and do not fill hole(s)
   - *Remove Selected Face(s)* or press `Shift+Del`: removes faces but keeps the vertices

These options are also accessible from the contextual menu when hovering over a single item without selecting.

Moving mesh elements

To move vertices and faces of a mesh layer:

1. *Select the target elements*
2. Enable the Digitize mesh elements tool
3. To start moving the element, click on a vertex or the centroid of a face/edge
4. Move the cursor to the target location (snapping to vector features is supported).
5. If the new location does not generate an *invalid mesh*, the moved elements appear in green. Click again to release them at this location. Faces whose vertices are all selected are translated, their neighbors are reshaped accordingly.

Transforming mesh vertices

The Transform Vertices Coordinates tool gives a more advanced way to move vertices, by editing their X, Y and/or Z coordinates thanks to expressions.

1. Select the vertices you want to edit the coordinates
2. Press Transform Vertices Coordinates. A dialog opens with a mention of the number of selected vertices. You can still add or remove vertices from the selection.
3. Depending on the properties you want to modify, you need to check the *X coordinate*, *Y coordinate* and/or *Z value*.
4. Then enter the target position in the box, either as a numeric value or an expression (using the Expression dialog)
5. With the Import Coordinates of the Selected Vertex pressed, the X, Y and Z boxes are automatically filled with its coordinates whenever a single vertex is selected. You can adjust Z value by using *Get the Z value from the project terrain* button and in that case the Z box is deactivated. A convenient and quick way to adjust vertices individually.
6. Press *Preview Transform* to simulate the vertices new location and preview the mesh with transformation.
   - If the preview is green, transformed mesh is valid and you can apply the transformation.
   - If the preview is red, the transformed mesh is invalid and you can not apply the transformation until it is corrected.
7. Press *Apply Transform* to modify the selected coordinates for the set of vertices.

Reshaping mesh geometry

The contextual menu

1. Enable the Digitize mesh elements
2. Select mesh item(s), or not
3. Hover over a mesh element, it gets highlighted.
4. Right-click and you can:
   - *remove the item(s)*
   - *Split Selected Face(s)* (*Split Current Face*): splits the face you are hovering over or each selected quad mesh faces into two triangles
   - *Delaunay Triangulation with Selected vertices*: builds triangular faces using selected free vertices.
   - *Refine Selected Face(s)* (*Refine Current Face*): splits the face into four faces, based on vertices added at the middle of each edge (a triangle results into triangles, a quad into quads). Also triangulates adjacent faces connected to the new vertices.

The edge markers

When the Digitize mesh elements is active and you hover over an edge, the edge is highlighted and it is possible to interact with it. Depending on the context, following markers may be available:

- a **square**, at the center of the edge: click on it to select extremity vertices.
- a **cross** if the two faces on either side can be merged: click on it to delete the edge and merge the faces.
- a **circle** if the edge is between two triangles: Click on it to flip the edge, i.e. connect it instead to the two other ”free” vertices of the faces

The *Force by Selected Geometries* tool

The Force by Selected Geometries tool provides advanced ways to apply break lines using lines geometry. A break line will force the mesh to have edges along the line. Note that the break line will not be considered persistent once the operation is done; resulting edges will not act as constraints anymore and can be modified like any other edge. This can be used for example to locally modify a mesh layer with accurate lines, as river banks or border of road embankments.

1. Enable the Force by Selected Geometries tool
2. Indicate the geometry to use as ”forcing line”; it can be:
   - picked from a line or polygon feature in the map canvas: right-click over the vector feature and select it from the list in the contextual menu.
   - a virtual line drawn over the mesh frame: left-click to add vertices, right-click for validation. Vertices Z value is set through the *Vertex Z value* widget or the *z* widget if the *Advanced Digitizing Panel* is on. If the line is snapped to a mesh vertex or a 3D vector feature’s vertex or segment, the new vertex takes the snapped element Z value.

Mesh faces that overlap the line geometry or the polygon’s boundary will be affected in a way that depends on options you can set from the Force by Selected Geometries tool drop-down menu:

- *Add new vertex on intersecting edges*: with this option, a new vertex is added each time the forcing line intersect an edge. This option leads to split along the line each encountered faces.

Without this option, encountered faces are removed and replaced by faces coming from a triangulation with only the existing vertices plus the vertices of the forcing lines (new vertices are also added on the boundary edge intersecting the forcing lines).

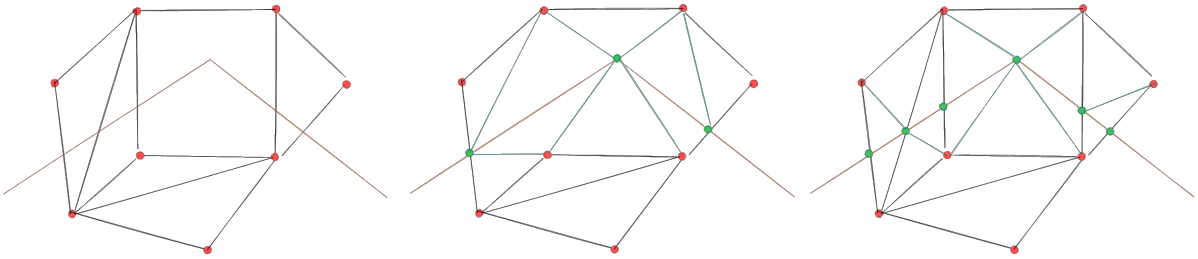


Fig. 14.18: Force Mesh using a line geometry - Results without (middle) and with (right) new vertex on edges intersection

- *Interpolate Z value from*: set how the new vertices Z value is calculated. It can be from:
  - the *Mesh* itself: the new vertices Z value is interpolated from vertices of the face they fall within
  - or the *Forcing line*: if the line is defined by a 3D vector feature or a drawn line then the new vertices Z value is derived from its geometry. In case of 2D line feature, the new vertices Z value is the *Vertex Z value*.
- *Tolerance*: when an existing mesh vertex is closer to the line than the tolerance value, do not create new vertex on the line but use the existing vertex instead. The value can be set in *Meters at Scale* or in *Map Units* (more details at *Unit Selector*).

14.5.5 Reindexing meshes

During edit, and in order to allow quick undo/redo operations, QGIS keeps empty places for deleted elements, which may lead to growing memory use and inefficient mesh structuring. The *Mesh* ► *Reindex Faces and Vertices* tool is designed to remove these holes and renumber the indices of faces and vertices so that they are continuous and somewhat reasonably ordered. This optimizes relation between faces and vertices and increases the efficiency of calculation.

Note

The *Reindex Faces and Vertices* tool saves the layer and clear the undo/redo stacks, disabling any rollback.

14.6 Mesh Calculator

The *Mesh Calculator* tool from the top *Mesh* menu allows you to perform arithmetic and logical calculations on existing dataset groups to generate a new dataset group (see Fig. 14.19).

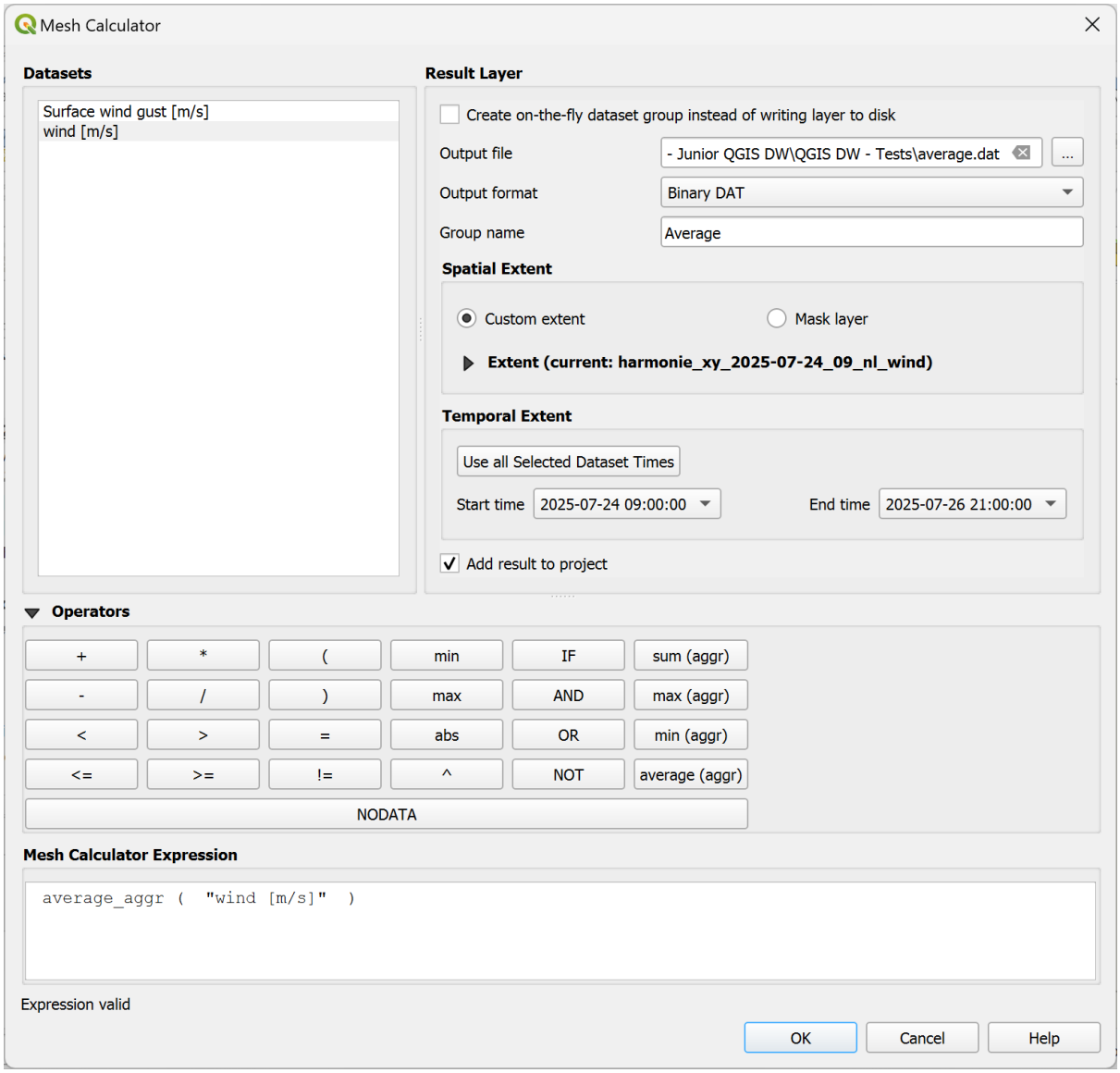


Fig. 14.19: Mesh Calculator

The *Datasets* list contains all dataset groups in the active mesh layer. To use a dataset group in an expression, double click its name in the list and it will be added to the *Mesh calculator expression* field. You can then use the operators to construct calculation expressions, or you can just type them into the box.

The *Result Layer* helps you configure properties of the output layer:

- *Create on-the-fly dataset group instead of writing layer to disk*:
  - If unchecked, the output is stored on disk as a new plain file. An *Output File* path and an *Output Format* are required.
  - If checked, a new dataset group will be added to the mesh layer. Values of the dataset group are not stored in memory but each dataset is calculated when needed with the formula entered in the mesh calculator. That virtual dataset group is saved with the project, and if needed, it can be removed or made persistent in file from the layer *Source* properties tab.

  In either case, you should provide a *Group Name* for the output dataset group.
- The *Spatial extent* to consider for calculation can be:
  - defined as a *Custom extent* using the *extent selector* widget.

  - defined by a polygon layer (*Mask layer*) of the project: the polygon features geometry are used to clip the mesh layer datasets

- The *Temporal extent* to take into account for datasets can be set with the *Start time* and *End time* options, selected from the existing dataset groups timesteps. They can also be filled using the *Use all selected dataset times* button to take the whole range.

- With the *Add result to project* checkbox, the result layer will automatically be added to the legend area and can be visualized. Checked by default.

The *Operators* section contains all available operators. To add an operator to the mesh calculator expression box, click the appropriate button. Mathematical calculations (`+`, `-`, `*`, ...) and statistical functions (`min`, `max`, `sum (aggr)`, `average (aggr)`, ...) are available. Conditional expressions (`=`, `!=`, `<`, `>=`, `IF`, `AND`, `NOT`, ...) return either 0 for false and 1 for true, and therefore can be used with other operators and functions. The `NODATA` value can also be used in the expressions.

The *Mesh Calculator Expression* widget shows and lets you edit the expression to execute.

CHAPTER 15

使用矢量瓦片

15.1 什么是矢量瓦片？

矢量瓦片是地理数据包，打包成预定义的大致正方形的”瓦片”，便于在 Web 传输。结合了预渲染的栅格地图瓦片和矢量地图瓦片。矢量瓦片服务器返回矢量地图数据，该数据已被裁剪到每个瓦片的边界，而不是预渲染的地图图像。裁剪的瓦片表示矢量瓦片服务的缩放级别，该缩放级别源自金字塔方式。与非瓦片的矢量地图相比，使用此结构可以减少数据传输。只需传输当前地图视图以及当前缩放级别的数据。此外，与瓦片栅格地图相比，数据传输也大大减少，因为矢量数据通常比渲染的位图小得多。矢量瓦片未分配任何样式信息，因此 QGIS 需要应用制图样式来显示数据。

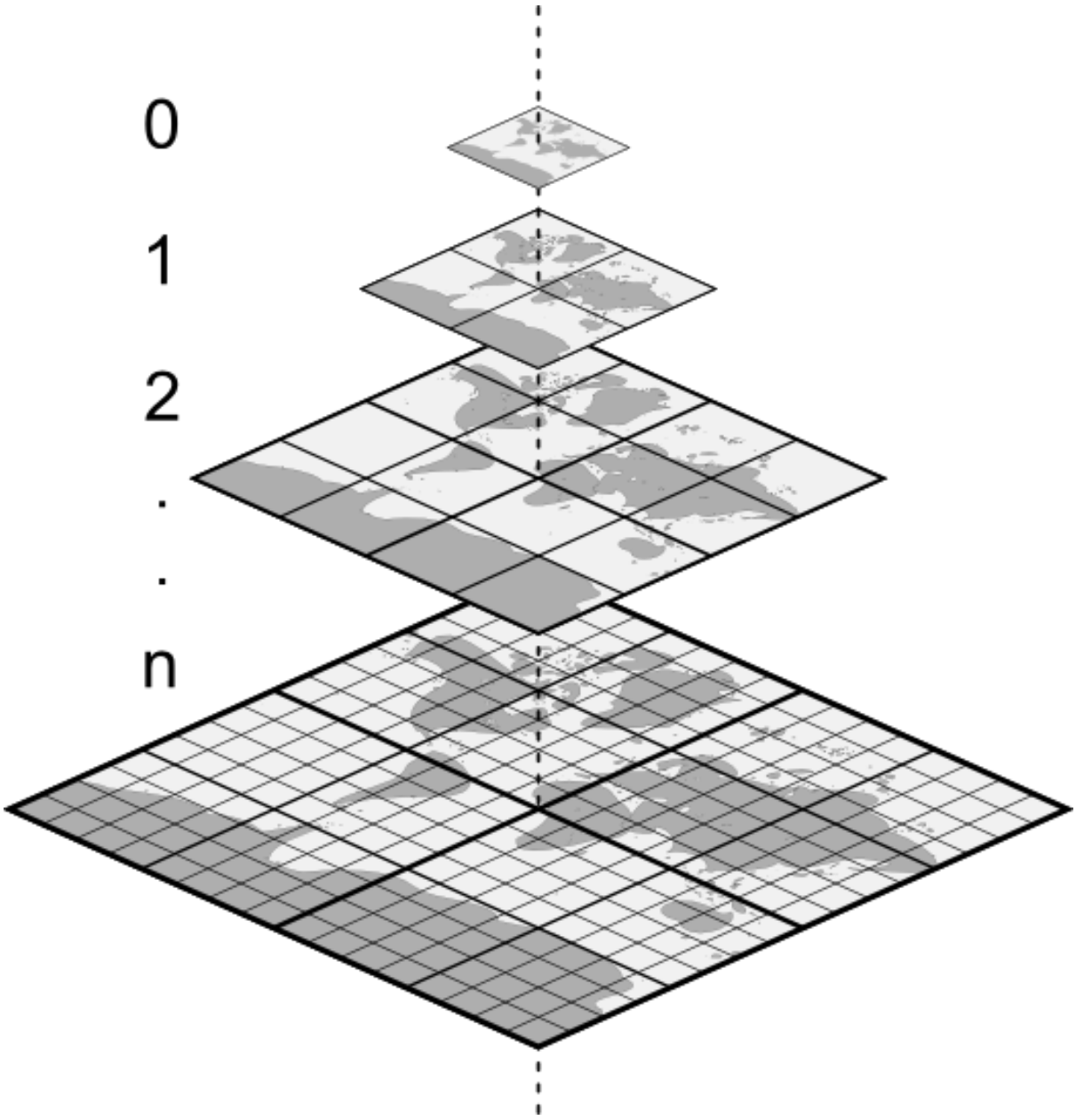


Fig. 15.1: 具有缩放级别的矢量瓦片的金字塔结构

15.2 支持的格式

通过以下方式支持矢量瓦片：

- 远程源 (HTTP/S) - 使用 XYZ 模板 - `type=xyz&url=http://example.com/{z}/{x}/{y}.pbf`
- 本地文件 - 使用 XYZ 模板 - 例如: `type=xyz&url=file:///path/to/tiles/{z}/{x}/{y}.pbf`
- 本地 MBTile 数据库 - 例如: `type=mbtiles&url=file:///path/to/file.mbtiles`

要将矢量瓦片数据集加载到 QGIS，请使用 *数据源管理器*对话框中的 矢量瓦片选项卡。更多信息，请参阅*使用矢量瓦片服务* 。

15.3 矢量瓦片数据集属性

矢量瓦片 图层属性对话框提供以下部分：

15.3.1 信息属性

信息选项卡是只读的，代表了一个有趣的地方，可以快速获取当前图层的摘要信息和元数据。提供的信息包括：

- 基于图层的提供程序：名称、URI、源类型和路径、缩放级别的数量
- 自定义属性，用于在活动项目中存储有关图层的附加信息。使用 PyQGIS 可以创建和管理更多的属性，特别是通过 setCustomProperty() 方法。
- the Coordinate Reference System: name, units, method, accuracy, reference (i.e., whether it is static or dynamic)
- 从已填充的元数据 中提取：访问、范围、链接、联系人、历史记录等

15.3.2 源属性

源选项卡显示所选矢量瓦片的基本信息，包括：

- 图层名显示在:guilabel:‘图层面板 ‘中；
- the *Coordinate Reference System*: Displays the layer's *Coordinate Reference System (CRS)*. You can change the layer's CRS, by selecting a recently used one in the drop-down list or clicking on the Select CRS button (see 坐标参照系选择器). Use this process only if the layer CRS is wrong or not specified.
- For file based vector tiles (`.vtpk` or `.mbtiles`), a *Layer source* group indicates the path to the source of the dataset and allows for replacing the loaded layer. Edit the path shown in the text box or press ... Browse to select another file on the disk.

[1] Also available in the *Layer styling panel*

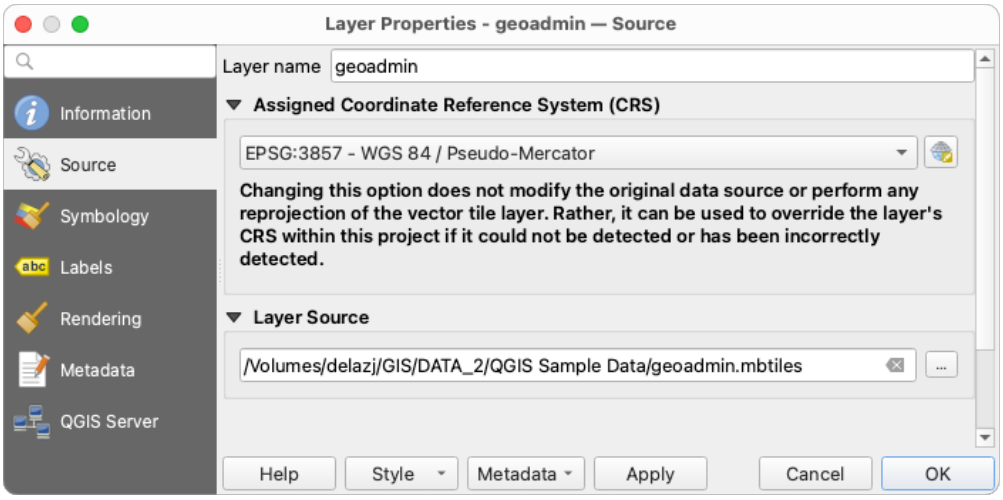


Fig. 15.2: Vector Tiles Properties - Source Dialog

15.3.3 Symbology and Labels Properties

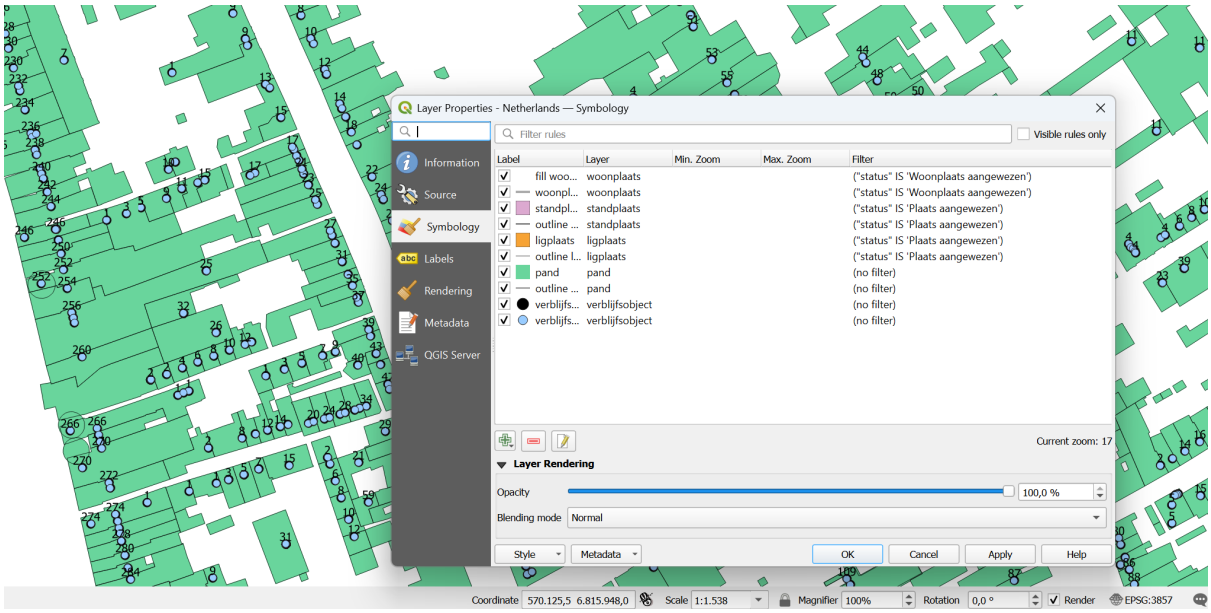


Fig. 15.3: 矢量瓦片图层符号化

Setting rules

As vector tiles consist of point, line and polygon geometries, the respective symbols are available. To apply a cartographic style (with symbology and/or labels), you can either:

- Use a *Style URL* when creating the *Vector Tiles Connection*. The symbology will be shown immediately in the *Symbology* tab after the layer is loaded in QGIS.
- Or build your own symbology and labeling in the corresponding tabs of the layer properties. By default, QGIS assigns an identical symbol to the features based on their geometry type.

In both cases, setting a style for a vector tile relies on a set of *rules* applied to the features, indicating:

- a vector *symbol* or *label*

- a *Label*, a title for comprehensive identification of the rule
- the name of a particular *Layer* the rule should apply to, if not applied to `(all layers)`
- a *Min. Zoom* and a *Max. Zoom*, for the range of display. Symbology and labeling can be dependent on the zoom level.
- a *Filter*, a QGIS expression to identify the features to apply the style to

Each rule is added pressing the Add rule button and selecting the type of symbols (*Marker*, *Line*, *Fill*) corresponding to the features geometry type. You can as well Remove selected rules or Edit current rule.

At the bottom the *Current Zoom* is shown. Check the *Visible rules only* option at the top of the dialog to filter the list of rules to only those that are visible at the current zoom level. This makes it easier to work with complex vector styling and to locate troublesome rules. The *Filter rules* text box also helps you easily find a rule, by searching the *Label*, *Layer* and *Filter* fields.

In Fig. 15.3 we set up style for the OpenStreetMap `landuse` layer. For better visibility most of the rules are deselected.

Layer rendering

From the *Symbology* tab, you can also set some options that invariably act on all features of the layer:

- *Opacity*: You can make the underlying layer in the map canvas visible with this tool. Use the slider to adapt the visibility of your vector layer to your needs. You can also make a precise definition of the percentage of visibility in the menu beside the slider.
- *Blending mode*: You can achieve special rendering effects with these tools that you may previously only know from graphics programs. The pixels of your overlaying and underlaying layers are mixed through the settings described in *Blending Modes*.

Styles

Available at the bottom of most of the tabs, the *Styles* ► menu provides shortcuts to save, load, create, switch styles to apply to the vector tiles. Vector tiles can have their style saved from QGIS as *QML* files and they can be imported as:

- *QML* files (*QML - QGIS 样式文件格式*)
- *MapBox GL Json* style configuration files

More details at 保存和共享图层属性.

15.3.4 Rendering Properties

Under *Scale dependent visibility*, you can set the *Maximum (inclusive)* and *Minimum (exclusive)* scales, defining a range of scales in which features will be visible. Out of this range, they are hidden. The Set to current canvas scale button helps you use the current map canvas scale as boundary of the range visibility. See 可见比例选择器 for more information.

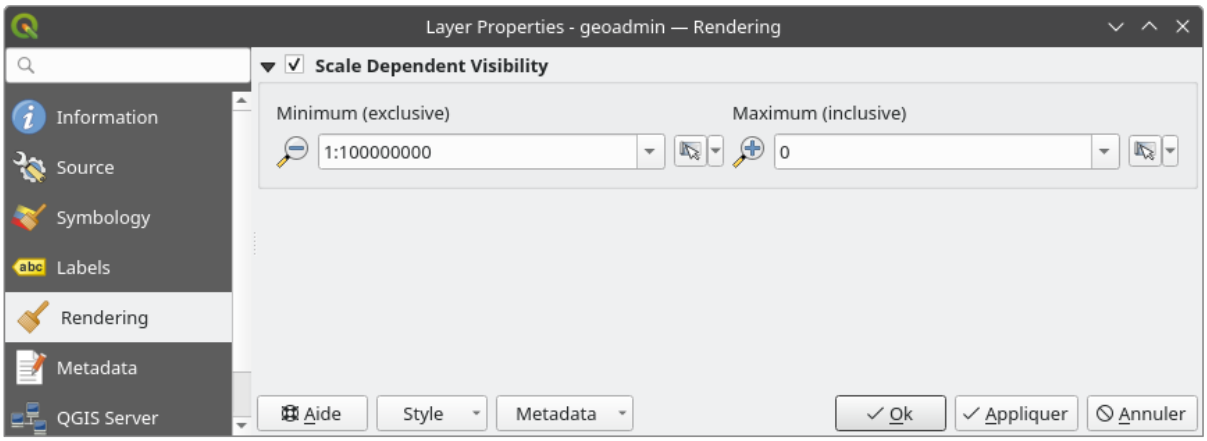


Fig. 15.4: Vector Tiles Properties - Rendering Dialog

15.3.5 元数据属性

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

15.4 QGIS Server Properties

The *QGIS Server* tab helps you configure settings of the data when published by QGIS Server. The configuration concerns:

- *Description*: provides information to describe the data, such as *Short name*, *Title*, *Abstract*, a *Keyword list*, and a *Data URL* whose *Format* can be in `text/html`, `text/plain` or `application/pdf`.
- *Attribution*: a *Title* and *URL* to identify who provides the data
- *Metadata URL*: a list of *URL* for the metadata that can be of `FGDC` or `TC211` *Type*, and in `text/plain` or `text/xml` *Format*
- *Legend URL*: a *URL* for the legend, in either `image/png` or `image/jpeg` *Format*

CHAPTER 16

Working with Point Clouds

16.1 Introduction to Point Clouds

What is A Point Cloud?

A point cloud is a three-dimensional image of a space made up of many individual of data points (up to billions, even trillions). Each of the points has an x, y and z coordinate. Depending on the capture method, point clouds usually also have additional attributes that came from the capture, such as color values or intensity. These attributes can be used, for instance, to display point clouds in different colors. In QGIS, a point cloud can be used to generate a three-dimensional image of the landscape (or of another space).

Supported Formats

QGIS supports the data formats Entwine Point Tile (EPT) and LAS/LAZ. To work with point clouds, QGIS always saves the data in EPT. EPT is a storage format that consists of several files stored in a common folder. To allow quick access to the data, EPT uses indexing. For more information on the EPT format, see entwine homepage

If the data is in LAS or LAZ format, QGIS will convert it to EPT when it is loaded for the first time. Depending on the size of the file, this may take some time. In this process, a subfolder is created in the folder in which the LAS/LAZ file is located according to the scheme `ept_` + `name_LAS/LAZ_file`. If such a subfolder already exists, QGIS loads the EPT immediately (which leads to a reduced loading time).

Worth Knowing

In QGIS it is not (yet) possible to edit point clouds. If you want to manipulate your point cloud, you can use CloudCompare, an open source point cloud processing tool. Also the Point Data Abstraction Library (PDAL - similar to GDAL) offers you options to edit point clouds (PDAL is command line only).

Due to the large number of data points, it is not possible to display an attribute table of point clouds in QGIS. However, the *Identify tool* supports point clouds, so you can display all attributes, even of a single data point.

If you want to create a new layer, in the same or another supported format, from an existing point cloud layer, see *Creating new layers from an existing layer*.

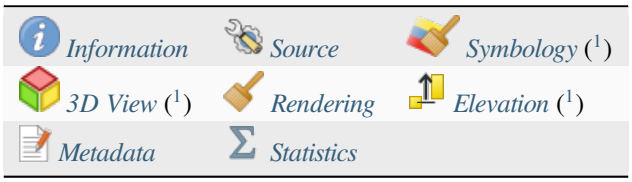
16.2 Point Clouds Properties

The *Layer Properties* dialog for a point cloud layer offers general settings for the layer and its rendering. It also provides information about the layer.

要访问 图层属性对话框，请执行以下操作：

- In the *Layers* panel, double-click the layer or right-click and select *Properties...* from the context menu;
- 选择图层后，转到 图层 ► 图层属性... 菜单。

The point cloud *Layer Properties* dialog provides the following sections:



Note

Most of the properties of a point cloud layer can be saved to or loaded from a `.qml` file using the *Style* menu at the bottom of the properties dialog. More details at 保存和共享图层属性

16.2.1 信息属性

The *Information* tab is read-only and represents an interesting place to quickly grab summarized information and metadata on the current layer. Provided information are:

- General such as name in the project, source path, last save time and size, the used provider
- custom properties, used to store in the active project additional information about the layer. Default custom properties may include *layer notes*. More properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- Based on the provider of the layer: extent and number of points
- The Coordinate Reference System: name, units, method, accuracy, reference (i.e. whether it's static or dynamic)
- Metadata delivered by the provider: creation date, version, data format, scale X/Y/Z, ...
- Picked from the *Metadata* tab (where they can be edited): access, extents, links, contacts, history...

[1] Also available in the *Layer styling panel*

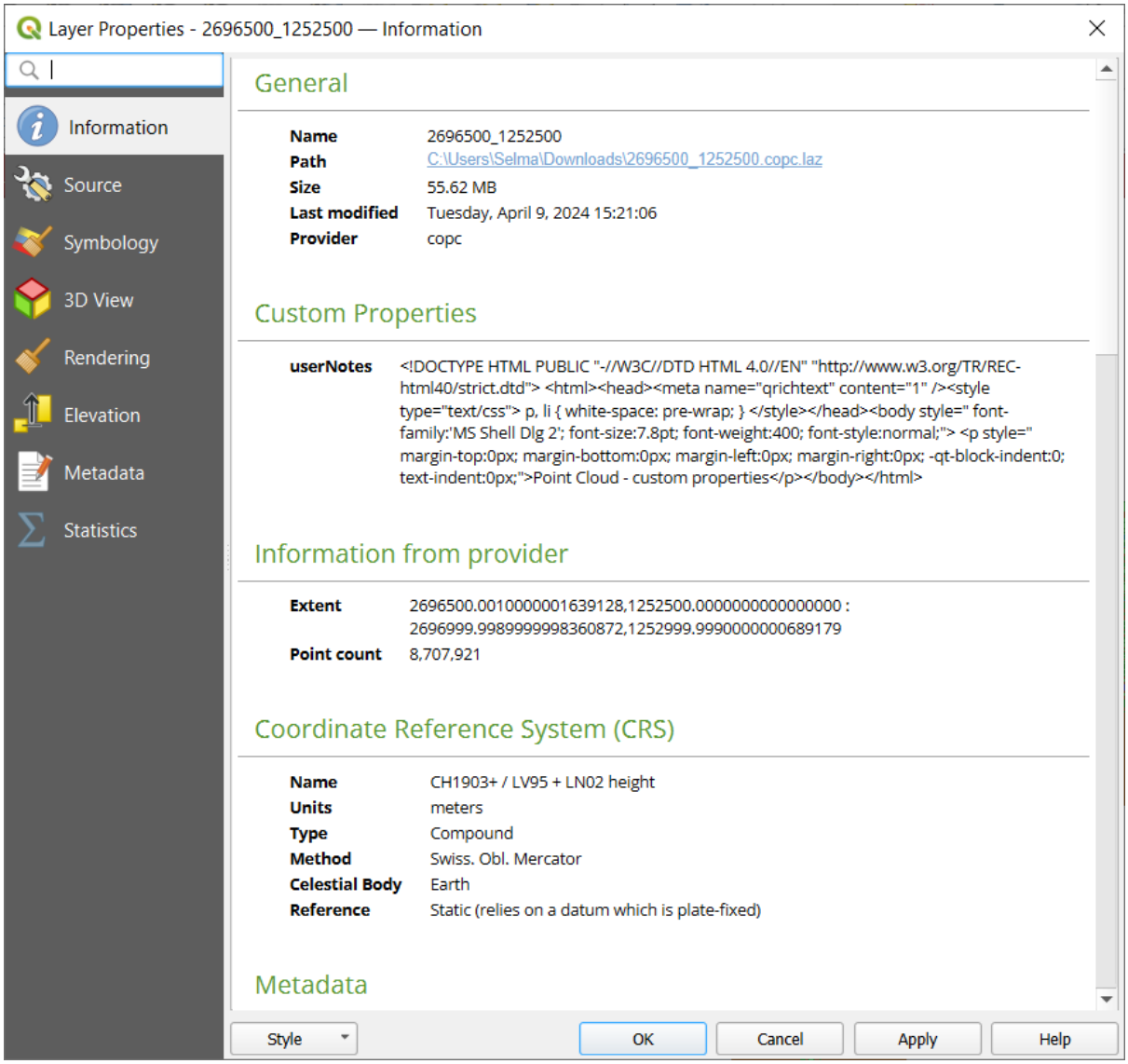


Fig. 16.1: Point cloud information tab

16.2.2 源属性

In the *Source* tab you can see and edit basic information about the point cloud layer:

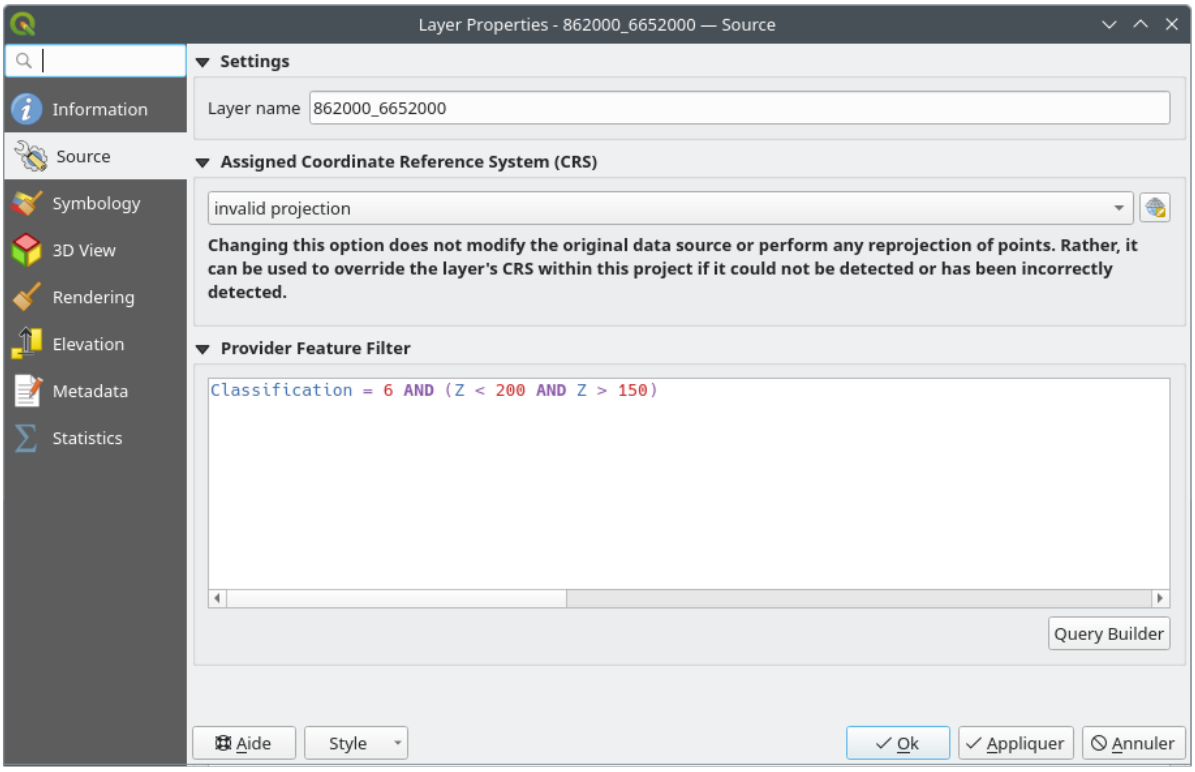


Fig. 16.2: Point cloud source tab

- *Settings*: Set a Layer name different from the layer filename that will be used to identify the layer in the project (in the Layers Panel, with expressions, in print layout legend,...)
- *Assigned Coordinate Reference System (CRS)*: Here you can change the layer's assigned *Coordinate Reference System*, selecting a recently used one in the drop-down list or clicking on set Projection Select CRS button (see 坐标参照系选择器). Use this process only if the CRS applied to the layer is a wrong one or if none was applied.
- *Provider Feature Filter*: allows to restrict data to use in the current project to a subset, using functions and expression at the PDAL data provider level. Press the *Query Builder* button at the bottom to start setting the filter.

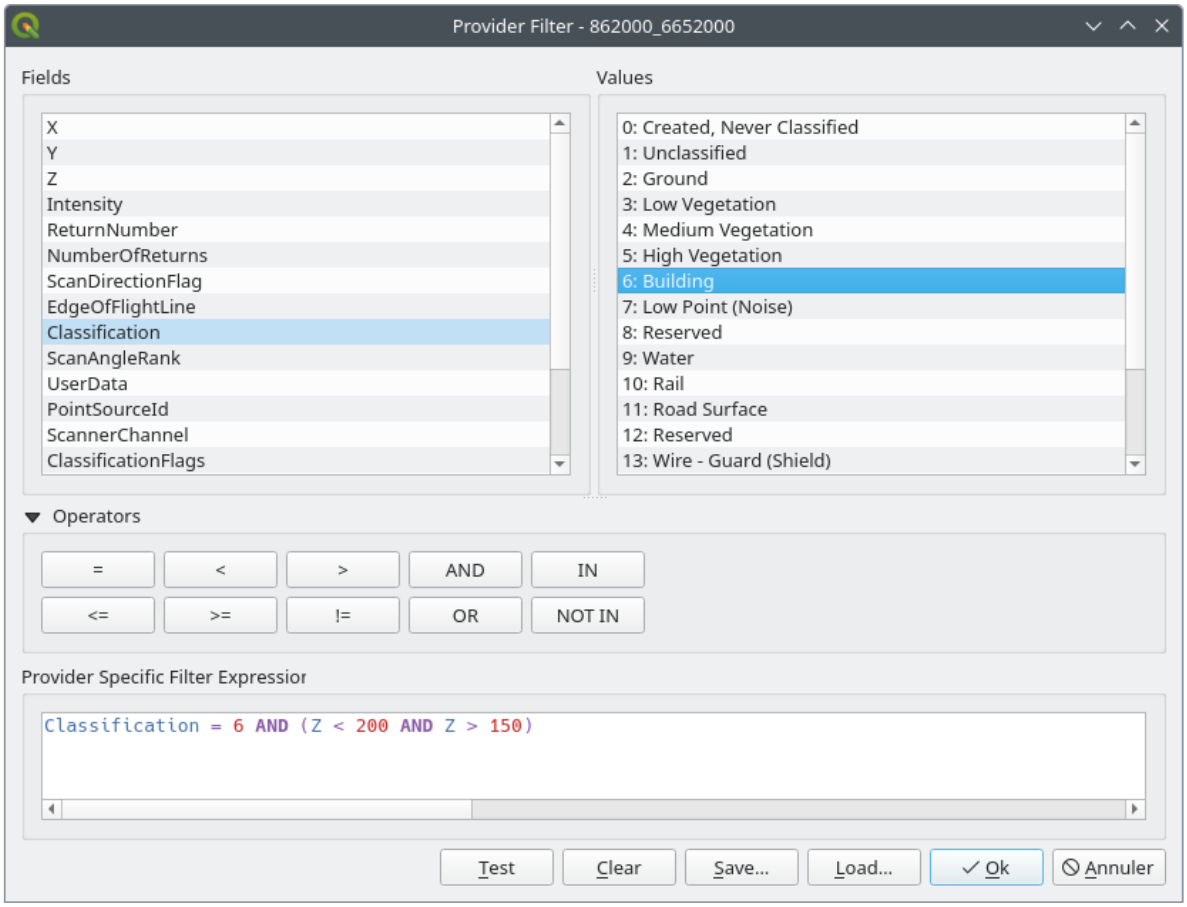


Fig. 16.3: Filtering point cloud layer to buildings at specific altitude

In the bottom part of the dialog, you can type a *Provider specific filter expression*. Such an expression can be built thanks to:

- *Fields*: the list contains all the attributes of the point cloud layer. To add an attribute to the expression, double-click its name or just type it into the text box.
- *Values*: frame lists values or statistics of the currently selected attribute, as a `key:value` pair. To add a value to the expression field, double click its name in the list: depending on the type of attribute, either the key or the value will be added to the expression. Or type the value in the expression text box.
- *Operators*: this toolbar contains all usable operators. To add an operator to the expression field, click the appropriate button. Relational operators (= , > , ...) and logical operators (`AND`, `OR`, ...) are available.

The *Test* button helps you check the syntax of your query. Use the *Clear* button to wipe the query and revert the layer to its original state (i.e., fully load all the points in the layer). It is possible to *Save...* the query as a `.QQF` file, or *Load...* the query from a file into the dialog.

When a filter is applied, QGIS treats the resulting subset as if it were the entire layer. For example if you applied the *filter above* for filtering buildings, you can not e.g. display, query, save or edit points that are of vegetation classification because they are not part of the subset.

Tip

Filtered layers are indicated in the Layers Panel

In the *Layers* panel, a filtered layer is listed with a Filter icon next to it indicating the query used when the mouse hovers over the icon. Double-click the icon to open the *Query Builder* dialog for edit. This can

also be achieved through the *Layer* ► *Filter...* menu.

16.2.3 Symbology Properties

In the *Symbology* tab the settings for the rendering of the point cloud are made. In the upper part, the settings of the different feature renderers can be found. In the lower part, there are sections with which general settings for the entire layer can be made and which apply over feature renderers.

Feature Rendering types

There are different options for rendering point clouds that can be selected using the drop-down menu at the top of the *Symbology* tab (see Fig. 16.4):

- *Extent Only*: Only a bounding box of the extent of the data is displayed; convenient for overviewing the data extent. As usual, the *Symbol widget* helps you configure any properties (color, stroke, opacity, sub-layers, ...) you'd like for the box.
- *Attribute by Ramp*: The data is drawn over a color gradient. See *Attribute by Ramp Renderer*
- *RGB*: Draw the data using red, green and blue color values. See *RGB Renderer*
- *Classification*: The data is drawn using different colors for different classes. See *Classification Renderer*

When a point cloud is loaded, QGIS follows a logic to select the best renderer:

- if the dataset contains color information (red, green, blue attributes), the RGB renderer will be used
- else if the dataset contains a `Classification` attribute, the classified renderer will be used
- else it will fall back to rendering based on Z attribute

If you do not know the attributes of the point cloud, the *Statistics tab* provides a good overview of which attributes are contained in the point cloud and in which ranges the values are located.

For each renderer, you can improve the data display adjusting the *point symbol size* or enabling *surface triangulation*.

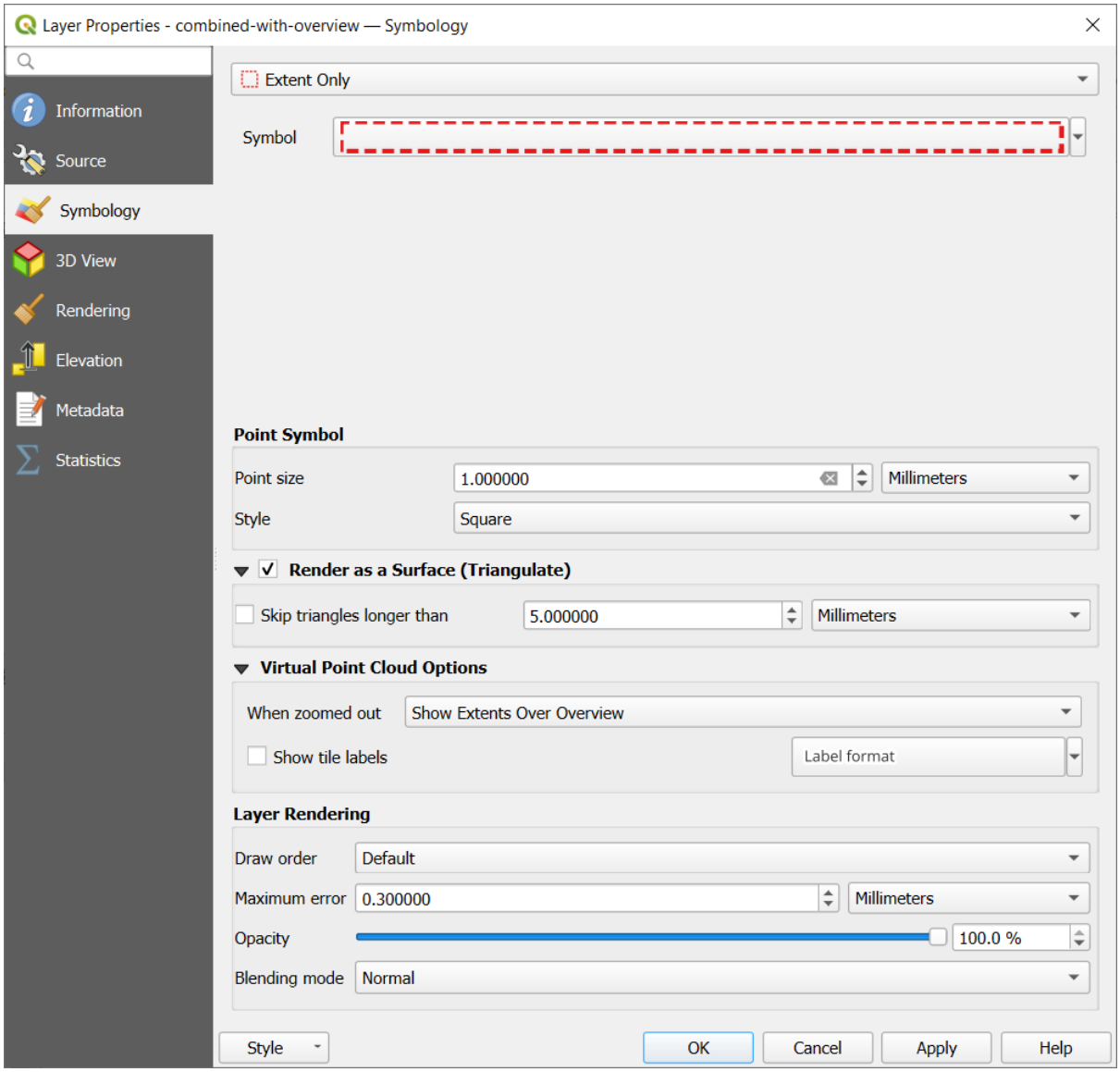


Fig. 16.4: Point cloud symbology tab

Attribute by Ramp Renderer

With *Attribute by Ramp*, the data can be displayed by numerical values over a color gradient. Such numerical values can be, for example, an existing intensity attribute or the Z-value. Depending on a minimum and a maximum value, the other values are spread to the color gradient via interpolation. The distinct values and their assignment to a certain color are called ”color map” and are shown in the table. There are various setting options, which are described below the figure.

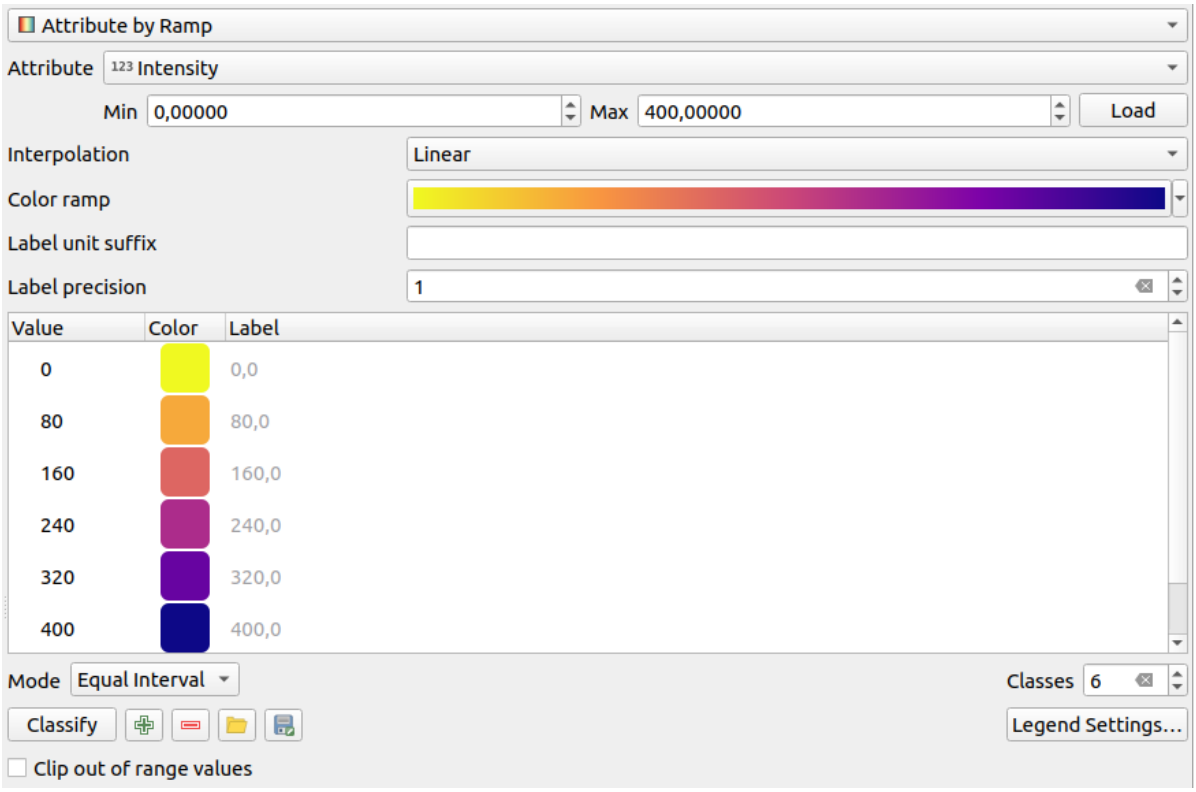


Fig. 16.5: Point cloud symbology tab: Attribute by Ramp

- *Min* and *Max* define the range that is applied to the color ramp: the *Min* value represents the left, the *Max* value the right end of the color ramp, the values in between are interpolated. By default QGIS detects the minimum and the maximum from the selected attribute but they can be modified. Once you have changed the values, you can restore the defaults by clicking on the *Load* button.
- The *Interpolation* entry defines how values are assigned their color:
  - *Discrete* (a <= symbol appears in the header of the *Value* column): The color is taken from the closest color map entry with equal or higher value
  - *Linear* The color is linearly interpolated from the color map entries above and below the pixel value, meaning that to each dataset value corresponds a unique color
  - *Exact* (a = symbol appears in the header of the *Value* column): Only pixels with value equal to a color map entry are applied a color; others are not rendered.
- The *Color ramp* widget helps you select the color ramp to assign to the dataset. As usual with *this widget*, you can create a new one and edit or save the currently selected one.
- The *Label unit suffix* adds a label after the value in the legend, and the *Label precision* controls the number of decimals to display.

The classification *Mode* helps you define how values are distributed across the classes:

- *Continuous*: Classes number and color are fetched from the color ramp stops; limits values are set following stops distribution in the color ramp (you can find more information on stops in 设置颜色渐变).
- *Equal interval*: The number of classes is set by the *Classes* field at the end of the line; limits values are defined so that the classes all have the same magnitude.

The classes are determined automatically and shown in the color map table. But you can also edit these classes manually:

- Double clicking in a *Value* in the table lets you modify the class value

- Double clicking in the *Color* column opens the 颜色选择器 widget, where you can select a color to apply for that value
- Double clicking in the *Label* column to modify the label of the class
- Right-clicking over selected rows in the color table shows a contextual menu to *Change Color...* and *Change Opacity...* for the selection

Below the table there are the options to restore the default classes with *Classify* or to manually Add values or Delete selected values from the table.

Since a customized color map can be very complex, there is also the option to Load an existing color map or to Save it for use in other layers (as a `txt` file).

If you have selected *Linear* for *Interpolation*, you can also configure:

- *Clip out of range values* By default, the linear method assigns the first class (respectively the last class) color to values in the dataset that are lower than the set *Min* (respectively greater than the set *Max*) value. Check this setting if you do not want to render those values.
- *Legend settings*, for display in the *Layers* panel and in the *layout legend*. Customization works the same way as with a raster layer (find more details at *Customize raster legend*).

RGB Renderer

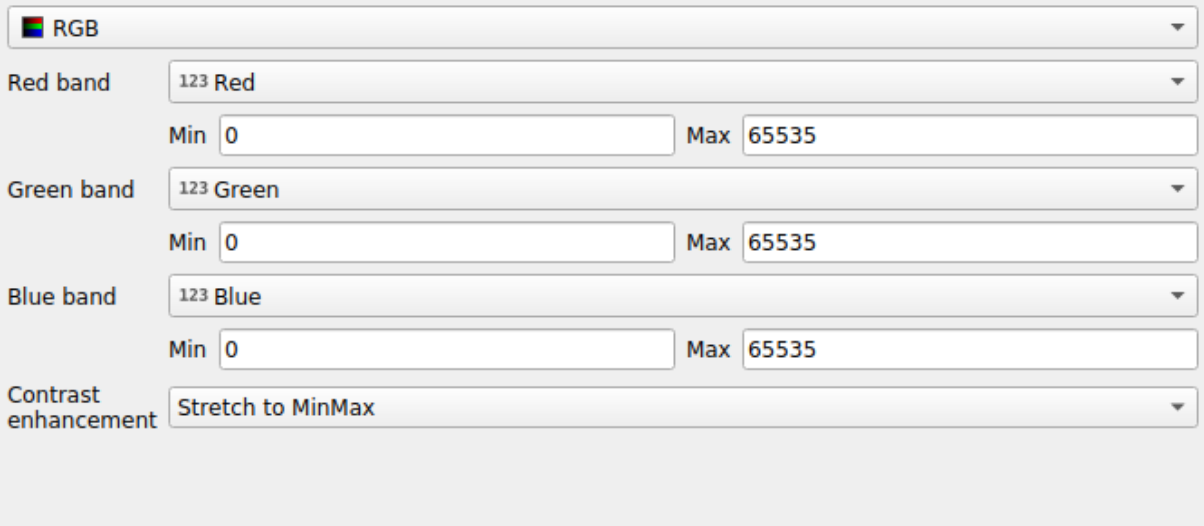


Fig. 16.6: The point cloud RGB renderer

With the *RGB* renderer, three selected attributes from the point cloud will be used as the red, green and blue component. If the attributes are named accordingly, QGIS selects them automatically and fetches *Min* and *Max* values for each band and scales the coloring accordingly. However, it is also possible to modify the values manually.

A *Contrast enhancement* method can be applied to the values: *No Enhancement*, *Stretch to MinMax*, *Stretch and Clip to MinMax* and *Clip to MinMax*

Note

The *Contrast enhancement* tool is still under development. If you have problems with it, you should use the default setting *Stretch to MinMax*.

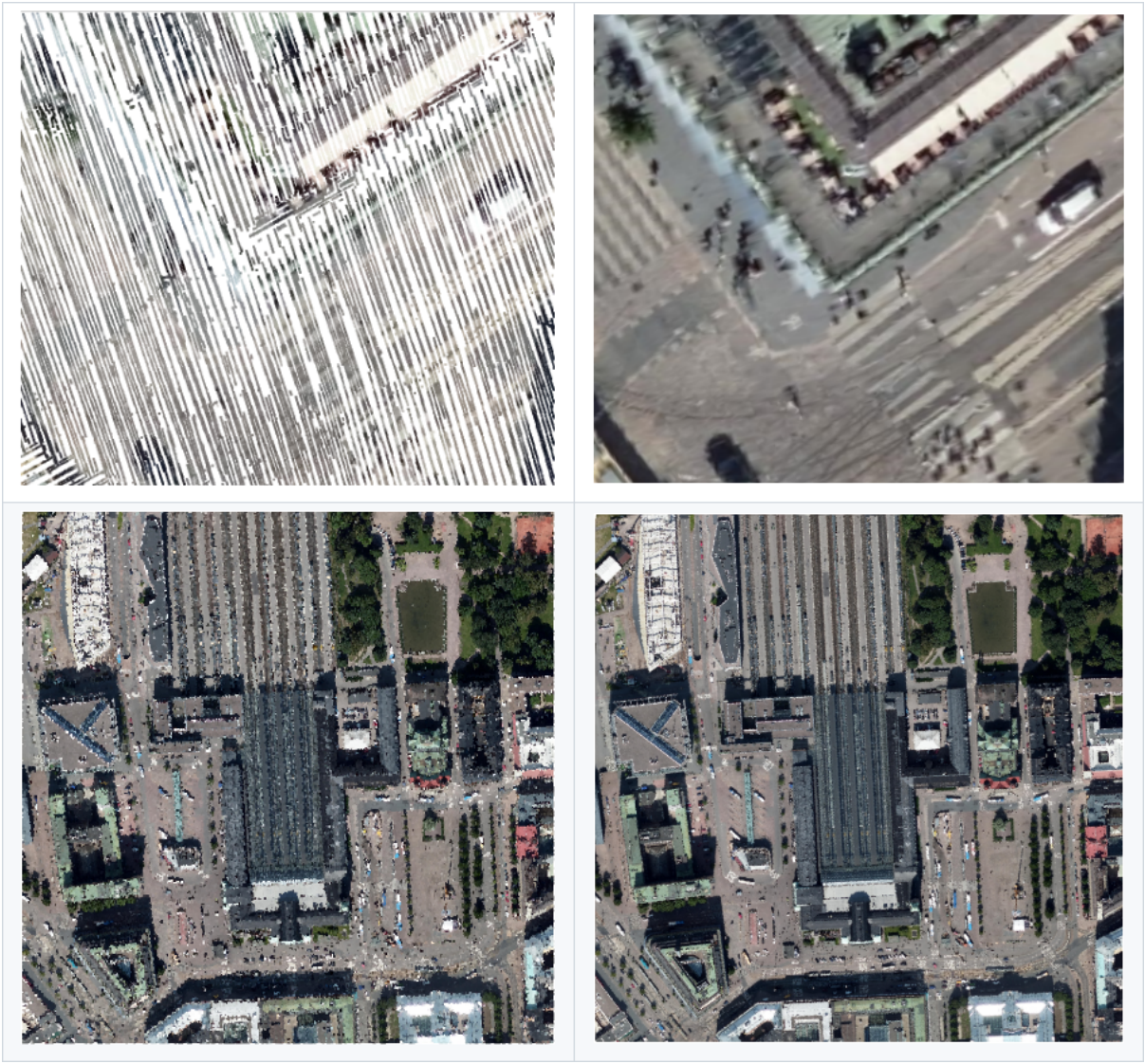


Fig. 16.7: Example of RGB renderer (left) combined with surface triangulation option (right)

Classification Renderer

In the *Classification* rendering, the point cloud is shown differentiated by color on the basis of an attribute. Any type of attribute can be used (numeric, string, ...). Point cloud data often includes a field called `Classification`. This usually contains data determined automatically by post-processing, e.g. about vegetation. With *Attribute* you can select the field from the attribute table that will be used for the classification. By default, QGIS uses the definitions of the LAS specification (see table 'ASPRS Standard Point Classes' in the PDF on ASPRS home page). However, the data may deviate from this schema; in case of doubt, you have to ask the person or institution from which you received the data for the definitions.

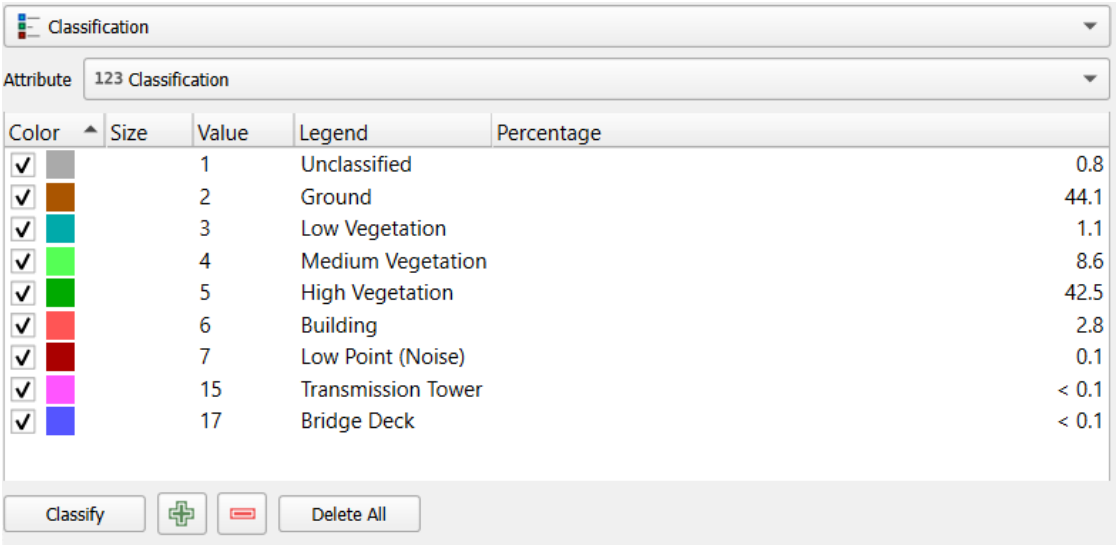


Fig. 16.8: The point cloud classification renderer

In the table all used values are displayed with the corresponding color and legend. At the beginning of each row there is a check box; if it is unchecked, this value is no longer shown on the map. With double click in the table, you can modify:

- the *Color*: the 颜色选择器 widget opens
- the *Size*: assigning a size of `0` to a category will revert it to use the default *point size* set for the layer
- the *Value*
- the *Legend*
- the *Percentage* column will show you the representation of the category within the layer.

You can also right-click one or more rows to open a context menu with the options to change the color, opacity or point size.

Below the table there are buttons with which you can change the default classes generated by QGIS:

- With the *Classify* button the data can be classified automatically: all values that occur in the attributes and are not yet present in the table are added
- With Add and Delete, values can be added or removed manually
- *Delete All* removes all values from the table

Hint

In the *Layers* panel, you can right-click over a class leaf entry of a layer to quickly configure visibility of the corresponding features.

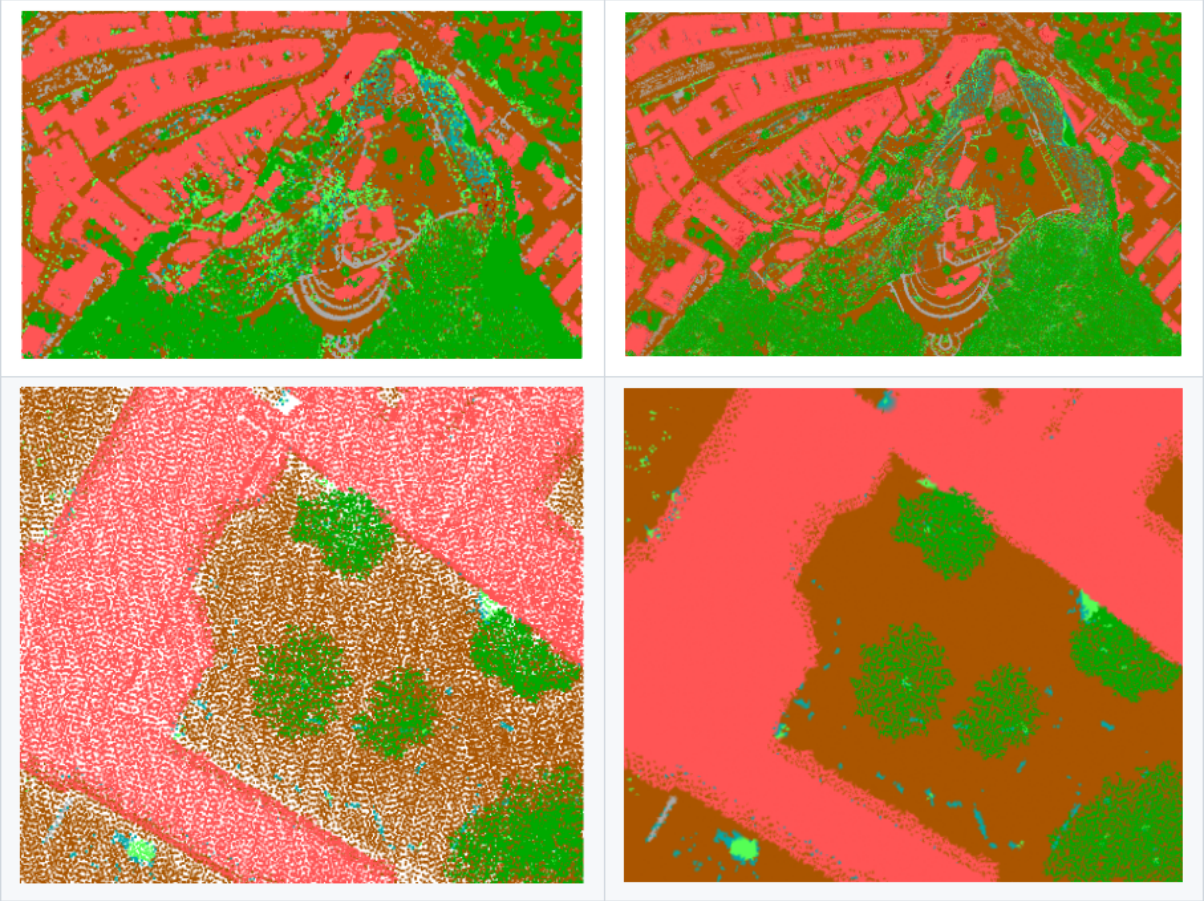


Fig. 16.9: Example of classification renderer (left) combined with surface triangulation option (right)

Point Symbol

Under *Point Symbol*, the size and the unit (e.g. millimeters, pixels, inches) with which each data point is displayed can be set. Either *Circle* or *Square* can be selected as the style for the points.

Render as a surface (Triangulate)

Check *Render as surface (Triangulate)* to enable the triangulation of the point cloud layer in the 2D view. This option allows rendering triangles instead of points. Each point keeps its color for interpolation in the triangle. You can control the horizontal length of computed triangles:

By checking the *Skip triangles longer than* option and setting up the threshold value, you can control the maximum lenght of a side of the triangles to consider in the horizontal plan. This can be particularly useful if you want to identify actual holes in the data.

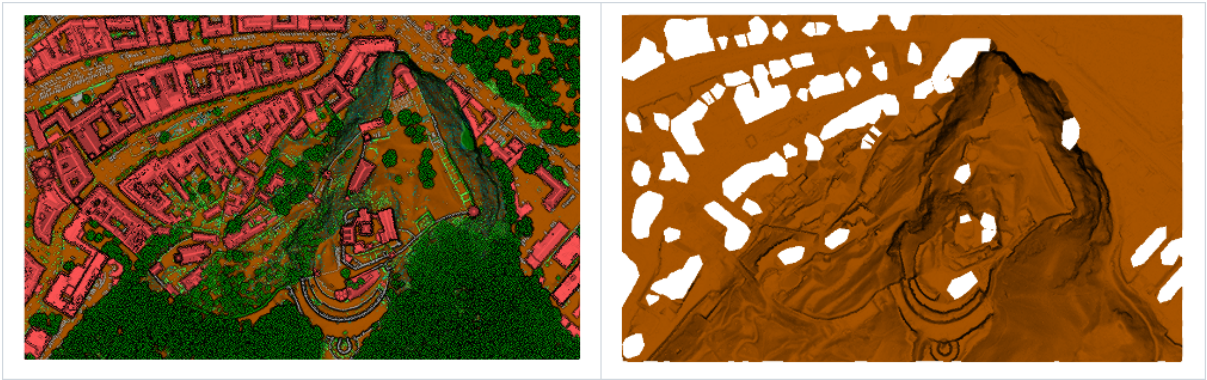


Fig. 16.10: Rendering data as a surface with map shading (left) and with map shading, filtering large triangles (right)

Virtual Point Cloud Options

The *Virtual Point Cloud Options* are available only when the layer is a *virtual point cloud (VPC)*. QGIS renders the overview of the VPC when zoomed out, if an overview is present. This provides a seamless transition from the overview to the detailed display of individual point clouds as you zoom in.

You can control how the VPC is displayed when zoomed out using the available options:

- *Show Extents Only*: Only the extents of the underlying point clouds are displayed.
- *Show Overview Only*: Only the overview is displayed (if available).
- *Show Extents Over Overview*: The extents are displayed on top of the overview.

You can also choose to *Show tile labels* to display the tile names and set the label format.

Layer Rendering

In the *Layer Rendering* section you have the following options to modify the rendering of the layer:

- *Draw order*: allows to control whether point clouds rendering order on 2d map canvas should rely on their Z value. It is possible to render :
  - with the *Default* order in which the points are stored in the layer,
  - from *Bottom to top* (points with larger Z values cover lower points giving the looks of a true ortho photo),
  - or from *Top to bottom* where the scene appears as viewed from below.
- *Maximum error*: Point clouds usually contains more points than are needed for the display. By this option you set how dense or sparse the display of the point cloud will be (this can also be understood as 'maximum allowed gap between points'). If you set a large number (e.g. 5 mm), there will be visible gaps between points. Low value (e.g. 0.1 mm) could force rendering of unnecessary amount of points, making rendering slower (different units can be selected).
- *Opacity*: You can make the underlying layer in the map canvas visible with this tool. Use the slider to adapt the visibility of your layer to your needs. You can also make a precise definition of the percentage of visibility in the menu beside the slider.
- *Blending mode*: You can achieve special rendering effects with this tool. The pixels of your overlaying and underlying layers are mixed through the settings described in *Blending Modes*.
- *Eye dome lighting*: this applies shading effects to the map canvas for a better depth rendering. Rendering quality depends on the *draw order* property; the *Default* draw order may give sub-optimal results. Following parameters can be controlled:
  - *Strength*: increases the contrast, allowing for better depth perception

  - *Distance*: represents the distance of the used pixels off the center pixel and has the effect of making edges thicker.

16.2.4 3D View Properties

In the *3D View* tab you can make the settings for the rendering of the point cloud in 3D maps.

3D Rendering modes

Following options can be selected from the drop down menu at the top of the tab:

- *No Rendering*: Data are not displayed
- *Follow 2D Symbology*: Syncs features rendering in 3D with *symbology assigned in 2D*
- *Single Color*: All points are displayed in the same *color* regardless of attributes
- *Attribute by Ramp*: Interpolates a given attribute over a color ramp and assigns to features their matching color. See *Attribute by Ramp Renderer*.
- *RGB*: Use different attributes of the features to set the Red, Green and Blue color components to assign to them. See *RGB Renderer*.
- *Classification*: differentiates points by color on the basis of an attribute. See *Classification Renderer*.

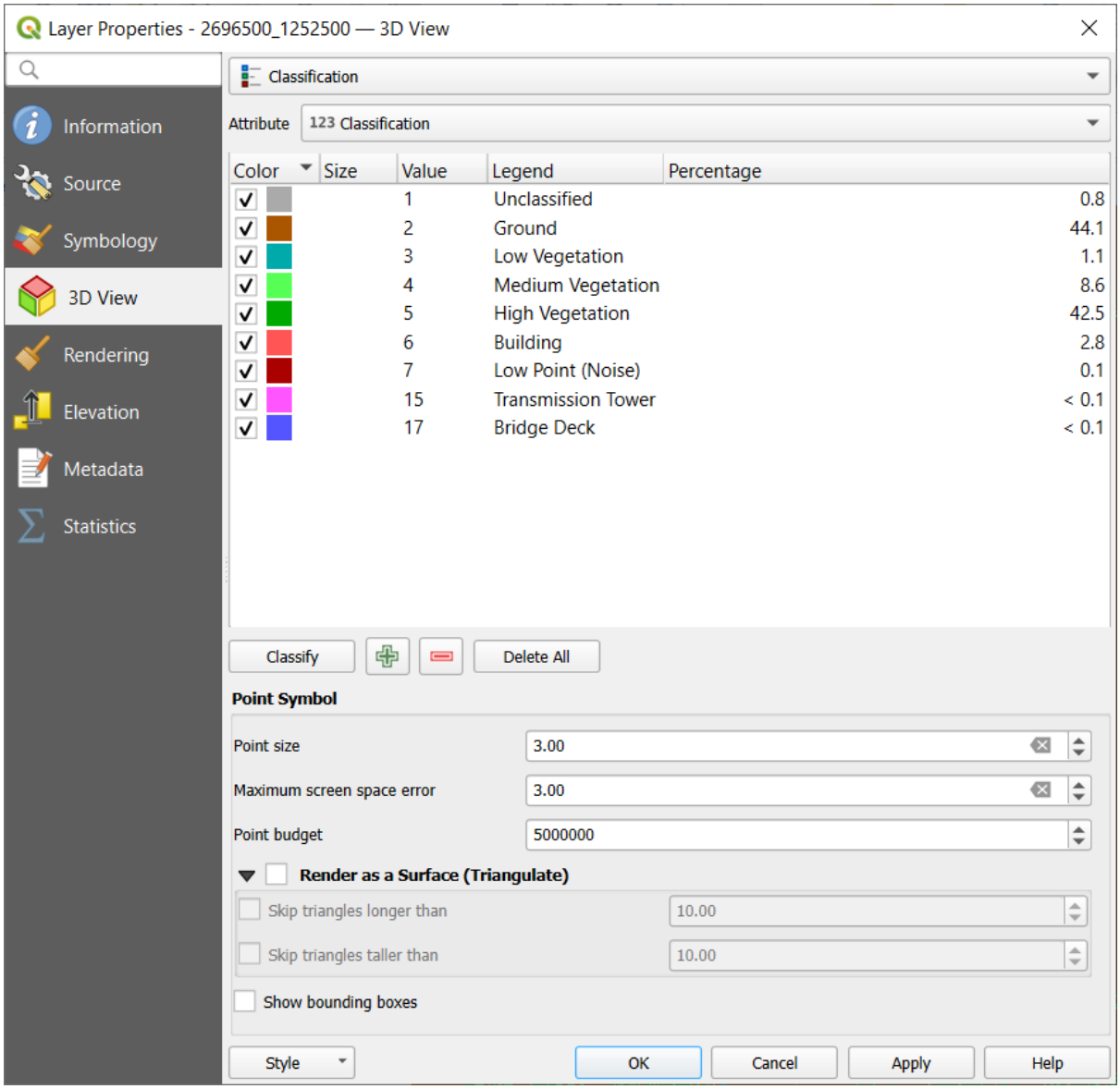


Fig. 16.11: The point cloud 3D view tab with the classification renderer

3D Point Symbol

In the lower part of the *3D View* tab you can find the *Point Symbol* section. Here you can make general settings for the entire layer which are the same for all renderers. There are the following options:

- *Point size*: The size (in pixels) with which each data point is displayed can be set
- *Maximum screen space error*: By this option you set how dense or sparse the display of the point cloud will be (in pixels). If you set a large number (e.g. 10), there will be visible gaps between points; low value (e.g. 0) could force rendering of unnecessary amount of points, making rendering slower (you can find more details at *Symbology Maximum error*).
- *Point budget*: To avoid long rendering, you can set the maximum number of points that will be rendered
- Check *Render as surface (Triangulate)* to render the point cloud layer in the 3D view with a solid surface obtained by triangulation. You can control dimensions of the computed triangles:
  - *Skip triangles longer than* a threshold value: sets in the horizontal plan, the maximum length of a side

of the triangles to consider

  - *Skip triangles taller than* a threshold value: sets in the vertical plan, the maximum height of a side of the triangles to consider

- *Show bounding boxes*: Especially useful for debugging, shows bounding boxes of nodes in hierarchy

16.2.5 Rendering Properties

Under the *Scale dependent visibility* group box, you can set the *Maximum (inclusive)* and *Minimum (exclusive)* scale, defining a range of scale in which features will be visible. Out of this range, they are hidden. The Set to current canvas scale button helps you use the current map canvas scale as boundary of the range visibility. See 可见比例选择器 for more information.

Note

You can also activate scale dependent visibility on a layer from within the *Layers* panel: right-click on the layer and in the contextual menu, select *Set Layer Scale Visibility*.

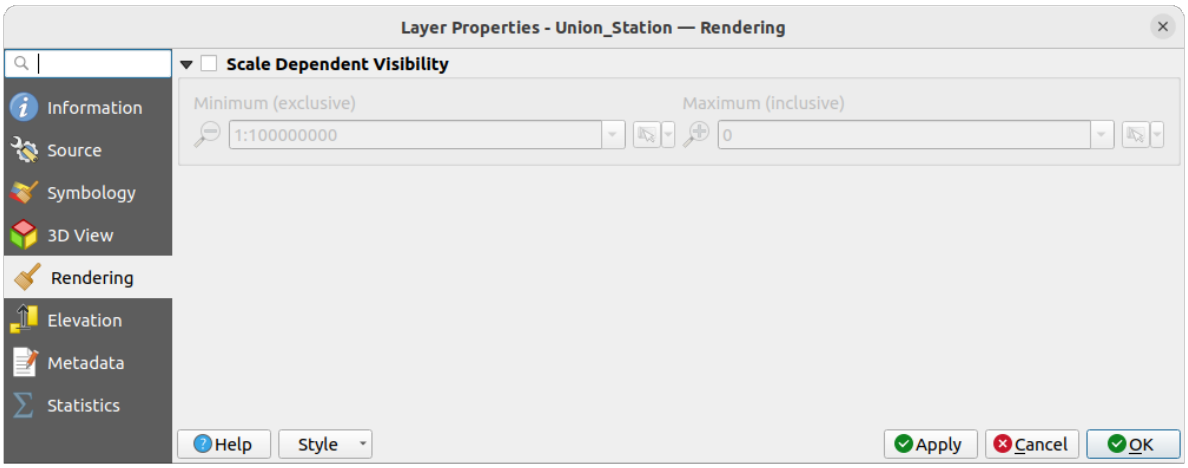


Fig. 16.12: The point cloud rendering tab

16.2.6 Elevation Properties

In the *Elevation* tab, you can set corrections for the Z-values of the data. This may be necessary to adjust the elevation of the data in 3D maps and its appearance in the *profile tool charts*. There are following options:

- *Vertical Reference System*: If the CRS of your point cloud layer is a compound one (including a Z dimension), then the vertical CRS used for the layer will be automatically derived from the vertical component of the layer's CRS. In this case, you cannot manually set a different vertical CRS, and the option to change it will be disabled. If your point cloud layer uses a horizontal (2D) CRS (though uncommon), you can manually select a specific vertical CRS by clicking on the Select CRS. Vertical reference systems are supported for point cloud layers in:
  - *Elevation profiles*
  - *Identify Tool Results*
  - *3D map views*
- Under *Elevation* group:

  - You can set a *Scale*: If `10` is entered here, a point that has a value Z = `5` is displayed at a height of `50`.
  - An *offset* to the z-level can be entered. This is useful to match different data sources in its height to each other. By default, the lowest z-value contained in the data is used as this value. This value can also be restored with the Refresh button at the end of the line.
- Under *Profile Chart Accuracy*, the *Maximum error* helps you control how dense or sparse the points will be rendered in the elevation profile. Larger values result in a faster generation with less points included.
- Under *Profile Chart Appearance*, you can control the point display:
  - *Point size*: the size to render the points with, in supported units (millimeters, map units, pixels, ...)
  - *Style*: whether to render the points as *Circle* or *Square*
  - Apply a single *Color* to all the points visible in the profile view
  - Check *Respect layer's coloring* to instead show the points with the color assigned via their *2D symbology*
  - *Apply opacity by distance from curve effect*, reducing the opacity of points which are further from the profile curve

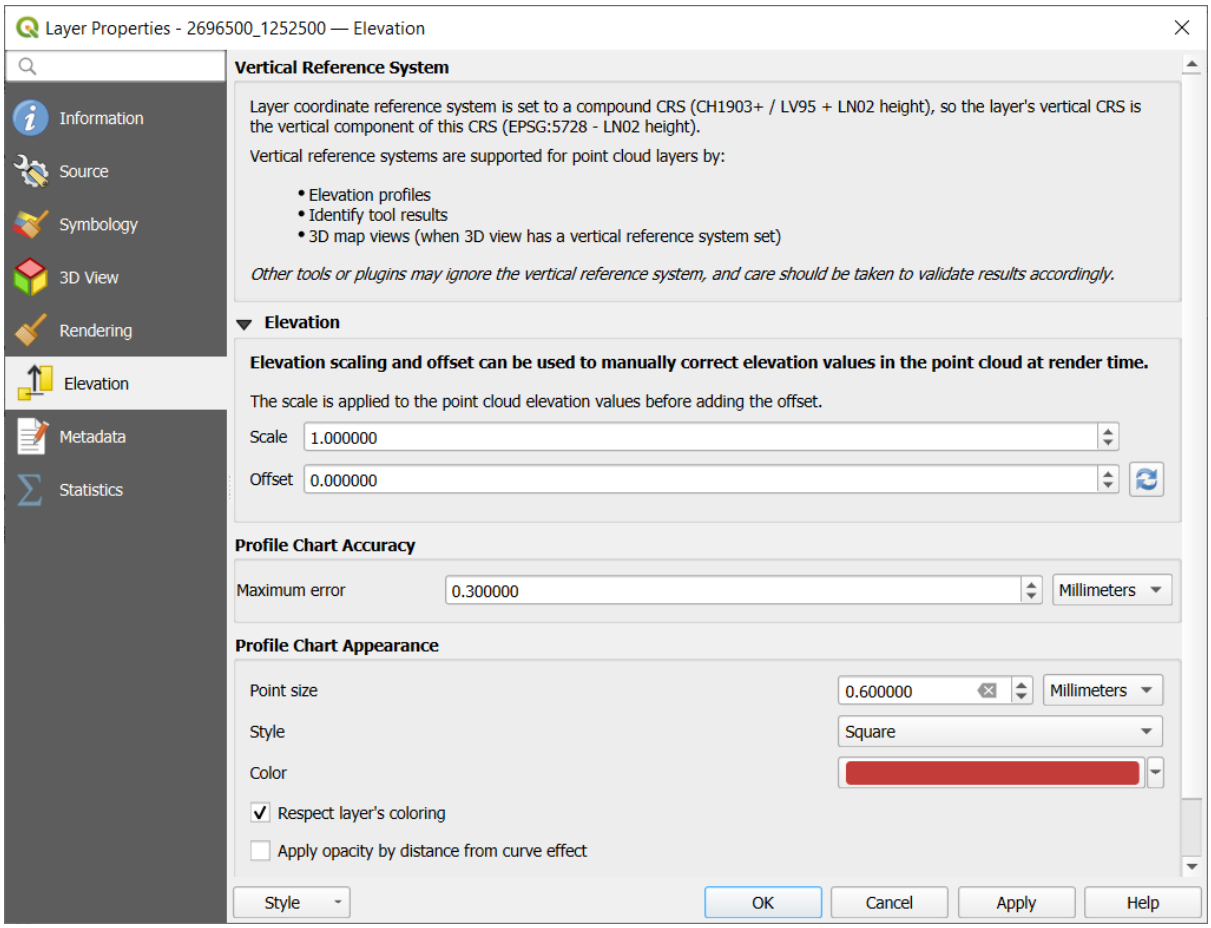


Fig. 16.13: The point cloud elevation tab

16.2.7 元数据属性

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

16.2.8 Statistics Properties

In the *Statistics* tab you can get an overview of the attributes of your point cloud and their distribution.

At the top you will find the section *Attribute Statistics*. Here all attributes contained in the point cloud are listed, as well as some of their statistical values: *Minimum*, *Maximum*, *Mean*, *Standard Deviation*

If there is an attribute *Classification*, then there is another table in the lower section. Here all values contained in the attribute are listed, as well as their absolute *Count* and relative *%* abundance.

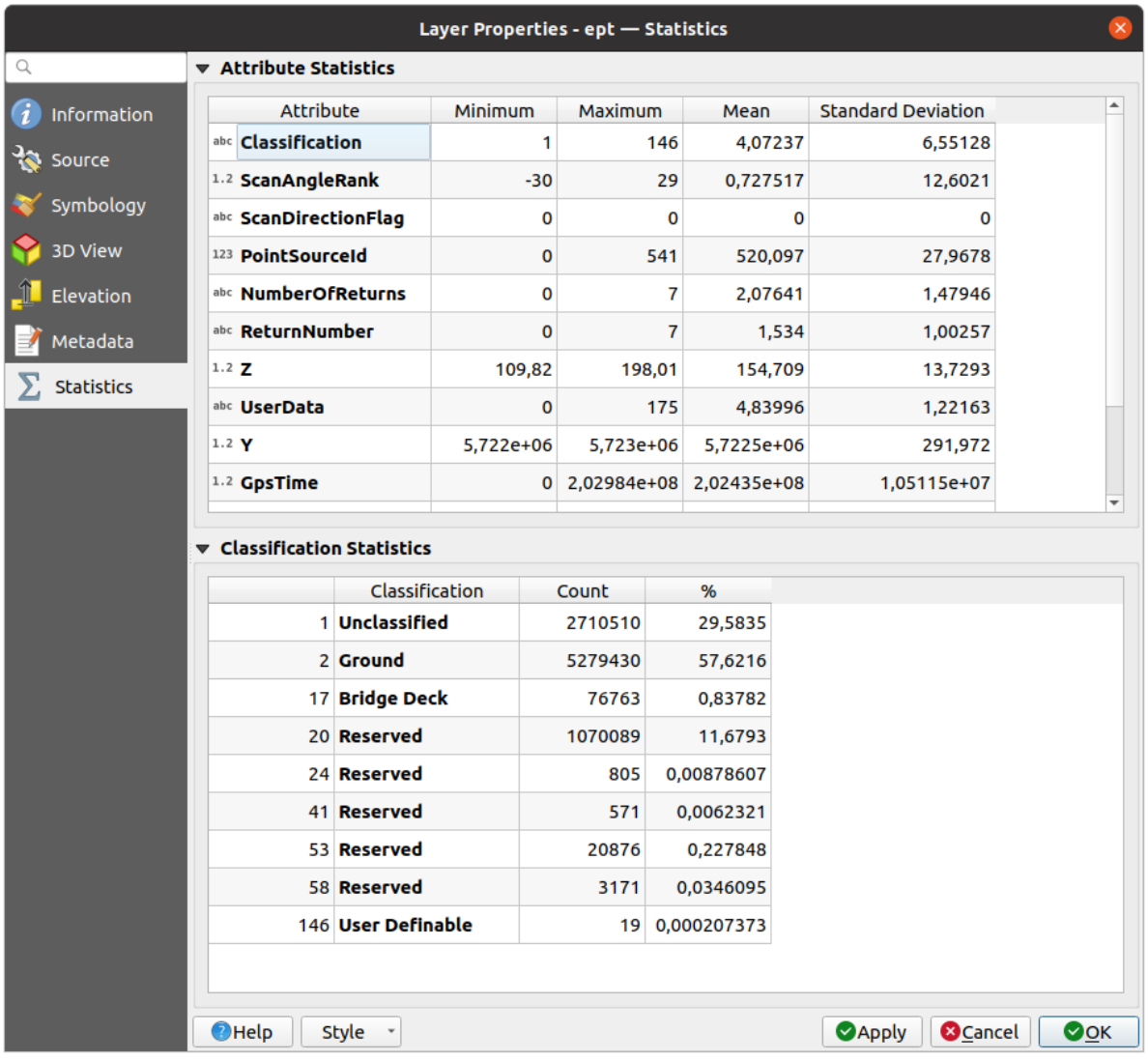


Fig. 16.14: The point cloud statistics tab

16.3 Virtual point cloud

Lidar surveys of larger areas are often multi-terabyte datasets with many billions of points. Having such large datasets represented as a single point cloud file is not practical due to the difficulties of storage, transfer, display and analysis. Point cloud data are therefore typically stored and distributed split into square tiles (e.g. `1km x 1km`), each tile having a more manageable file size (e.g. ~200 MB when compressed).

Tiling of data solves the problems with size of data, but it introduces issues when processing or viewing an area of interest that does not fit entirely into a single tile. Users need to develop workflows that take into account multiple tiles and special care needs to be taken to deal with data near edges of tiles to avoid unwanted artefacts in outputs. Similarly, when viewing point cloud data, it becomes cumbersome to load many individual files and apply the same symbology.

Here is an example of several point cloud tiles loaded in QGIS. Each tile is styled based on min/max Z values of the tile, creating visible artefacts on tile edges. The styling has to be adjusted for each layer separately:

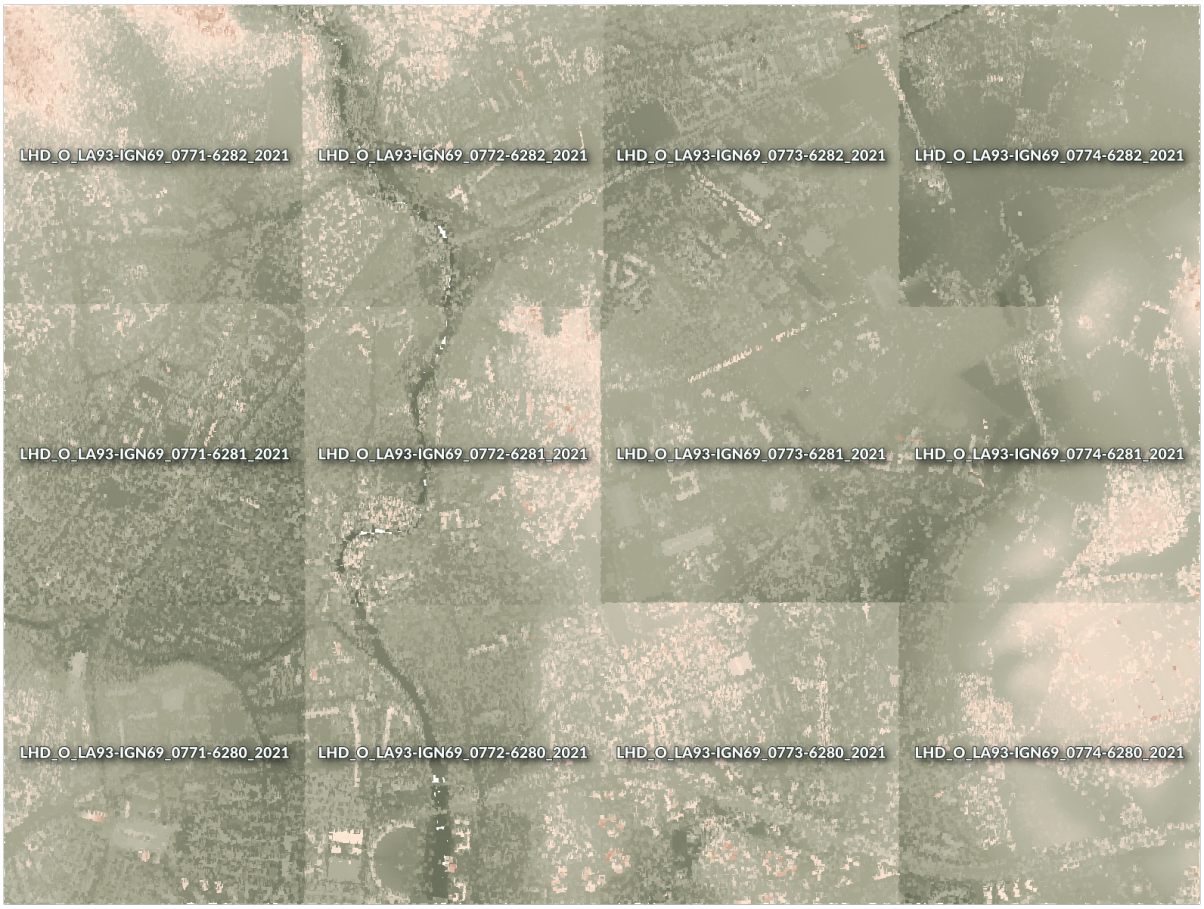


Fig. 16.15: Individual point cloud tiles loaded, with artefacts on edges

In the GIS world, many users are familiar with the concept of virtual rasters. A virtual raster is a file that simply references other raster files with actual data. In this way, GIS software then treats the whole dataset comprising many files as a single raster layer, making the display and analysis of all the rasters listed in the virtual file much easier.

Borrowing the concept of virtual rasters from GDAL, **virtual point cloud (VPC)** is a file format that references other point cloud files. Software supporting virtual point clouds handles the whole tiled dataset as a single data source.

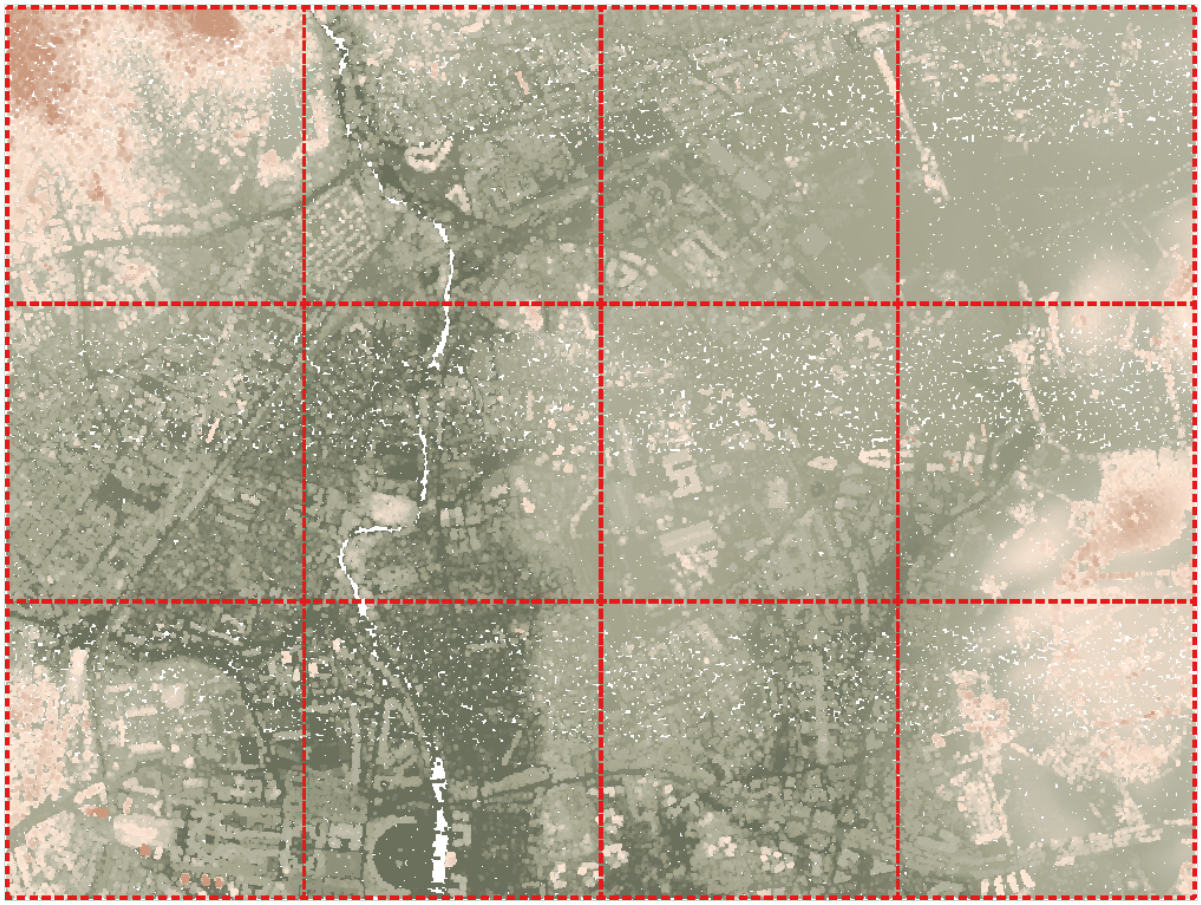


Fig. 16.16: The virtual point cloud

Displaying and manipulating virtual point cloud is much more fluent and easy.

At the core, a virtual point cloud file is a simple JSON file with `.vpc` extension, containing references to actual data files (e.g. `.LAS`, `.LAZ` or `.COPC` files) and additional metadata extracted from the files. Even though it is possible to write VPC files by hand, it is strongly recommended to create them using an automated tool, such as:

- The Processing *Build virtual point cloud (VPC)* algorithm
- The `build_vpc` command of PDAL wrench tool

For more details, please refer to the VPC specification that also contains best practices and optional extensions (such as overviews).

CHAPTER 17

Working with 3D Tiles

17.1 What are 3D Tiles?

3D tiles are specification for streaming and rendering large-scale 3D geospatial datasets. They use a hierarchical structure to efficiently manage and display 3D content, optimising performance by dynamically loading appropriate levels of detail. This technology is widely used in urban planning, architecture, simulation, gaming, and virtual reality, providing a standardised and interoperable solution for visualising complex geographical data.

Fig. 17.1: Example of 3D tiles

See *Using 3D tiled scene services* for instructions on how to add these data sources to QGIS.

17.1.1 State of 3D Tiles Support

Currently, QGIS supports two formats of 3D tiles:

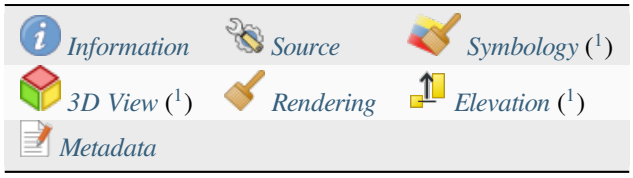
- "Cesium" 3D Tiles are used primarily for complex 3D models of buildings or whole cities. Such datasets can be provided by cloud-based platforms such as Cesium Ion or Google (*Photorealistic 3D Tiles*).
- Quantized Mesh tiles are used for terrain elevation data.

Supported *3D Tiles* features:

- QGIS currently has partial support for 3D Tiles 1.0 and partial support for 3D Tiles 1.1.
- For 3D Tiles 1.0, the tile format Batched 3D Model (`b3dm`) is supported. The tile formats Instanced 3D Model (`i3dm`), Point Cloud (`pnts`) and Composite (`cmpt`) are not supported yet.
- For 3D Tiles 1.1, the `3DTILES_content_gltf` extension is supported, so tile contents may also be encoded as glTF 2.0 (text or binary). The `EXT_mesh_gpu_instancing` extension is not supported yet.
- Currently only explicit tiling is supported. Implicit tiling is not supported yet
- Currently only tilesets using EPSG:4978 are supported.

17.2 3D Tiles Properties

The 3D tiles *Layer Properties* dialog provides the following sections:



17.2.1 信息属性

Information 选项卡为只读状态，这是引人关注的位置，可以快速获取当前图层的摘要信息和元数据。提供的信息包括：

- based on the provider of the layer: name, URL, source type and path, number of zoom levels
- 自定义属性，用于在活动项目中存储有关层的附加信息。使用 PyQGIS 可以创建和管理更多的属性，特别是通过 setCustomProperty() 方法。
- 坐标参照系：名称、单位、方法、精度、参照 (即静态还是动态)
- picked from the *filled metadata*: access, extents, links, contacts, history...

17.2.2 源属性

The *Source* tab displays basic information about the selected 3D tile, including:

- 图层名显示在:guilabel:'图层面板 '中；
- the *Coordinate Reference System*: Displays the layer's *Coordinate Reference System (CRS)*. You can change the layer's CRS, by selecting a recently used one in the drop-down list or clicking on the Select CRS button (see 坐标参照系选择器). Use this process only if the layer CRS is wrong or not specified.

17.2.3 Symbology Properties

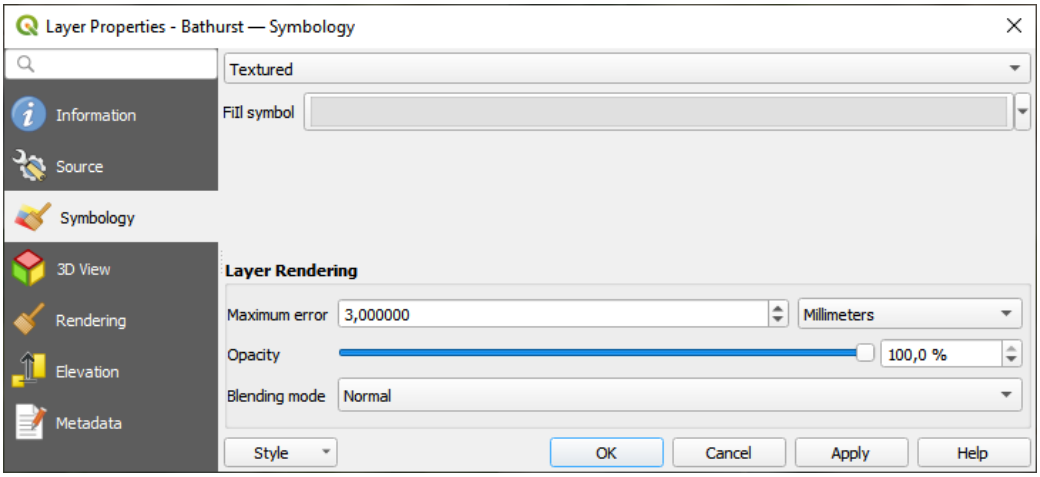


Fig. 17.2: 3D Tile Layer Symbology

[1] Also available in the *Layer styling panel*

By default, the layer is styled using texture, but you can change it to see the wireframe mesh behind the scene by choosing *Wireframe* in the drop-down menu. You can also, change the mesh fill and line symbols similar to the vector polygons. Checking *Use texture colors* will render each mesh element with the average value of the full texture. This is a good option to try when dealing with a large dataset and want to get a quick overview of the data.

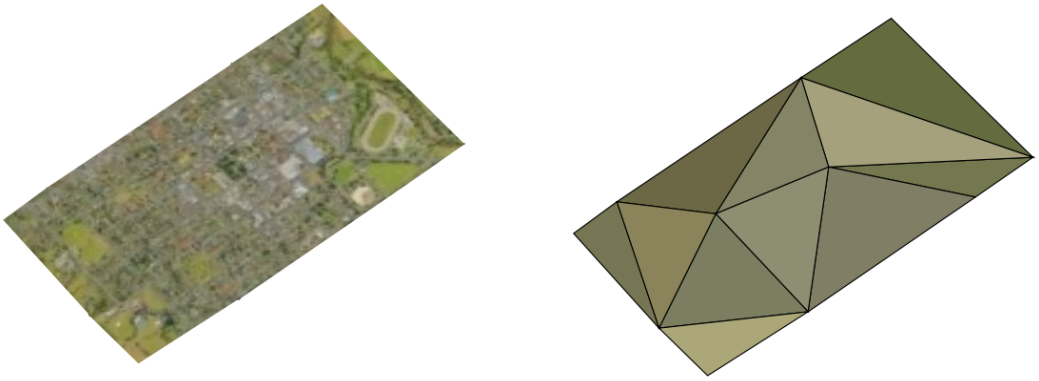


Fig. 17.3: 3D Tiles - textured and wireframe

From the *Symbology* tab, you can also set some options that invariably act on all features of the layer:

- *Maximum error*: This parameter determines the level of detail displayed in the 3D model. Similar to point clouds, 3D tiles often contain more information than necessary for visual representation. By adjusting this setting, you control the balance between display density and rendering speed. A larger value (e.g., 5 mm) may introduce noticeable gaps between elements, while a smaller value (e.g., 0.1 mm) could lead to the rendering of an excessive number of details, potentially slowing down the rendering process. Different units can be selected to tailor the setting to your specific needs.
- *Opacity*: Adjusts the visibility of the underlying layer on the map canvas using this tool. Use slider to tailor the visibility of your scene layer according to your preferences. Alternatively, specify the exact percentage of visibility through the menu next to the slider.
- *Blending mode*: You can achieve special rendering effects with these tools that you may previously only know from graphics programs. The pixels of your overlaying and underlaying layers are mixed through the settings described in *Blending Modes*.

17.2.4 3D View Properties

- *Maximum screen space error*: Determines the threshold for swapping terrain tiles with more detailed ones (and vice versa) - i.e. how soon the 3D view will use higher quality tiles. Lower numbers mean more details in the scene at the expenses of increased rendering complexity.
- *Show bounding boxes*: Shows 3D bounding boxes of the terrain tiles (useful for troubleshooting terrain issues).

To view the data you can open *New 3D map view*.

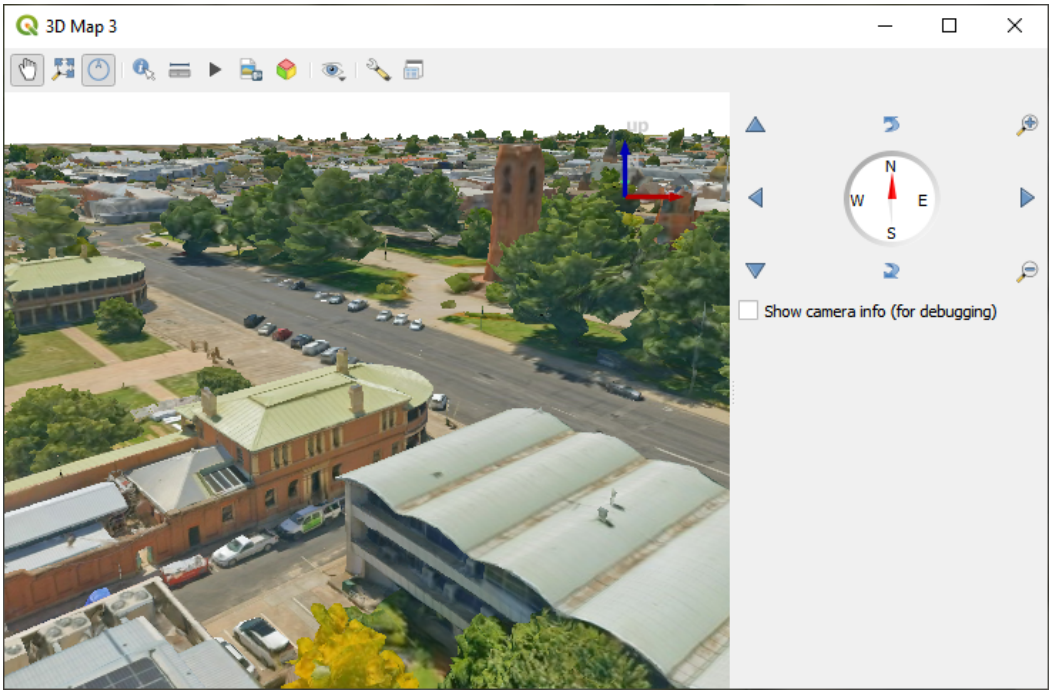


Fig. 17.4: 3D 地图视图

17.2.5 Rendering Properties

Under *Scale dependent visibility*, you can set the *Maximum (inclusive)* and *Minimum (exclusive)* scales, defining a range of scales in which features will be visible. Out of this range, they are hidden. The Set to current canvas scale button helps you use the current map canvas scale as boundary of the range visibility. See 可见比例选择器 for more information.

17.2.6 Elevation Properties

The *Elevation* tab provides options to control the layer elevation properties within a *3D map view*. Specifically, you can set:

- *Elevation Surface*: how the 3D layer vertices Z values should be interpreted as terrain elevation. You can apply a *Scale* factor and an *Offset*.

17.2.7 元数据属性

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

CHAPTER 18

Laying out the maps

With Print Layouts and Reports you can create maps and atlases, and print them or save them as image, PDF or SVG files.

18.1 Overview of the Print Layout

The print layout provides growing layout and printing capabilities. It allows you to add elements such as the QGIS 2D or 3D map canvas, text labels, images, legends, scale bars, basic shapes, arrows, attribute and simple tables, elevation profiles and HTML frames... You can size, group, align, position and rotate each element and adjust their properties to create your layout.

The layout can be printed or exported to image formats, PostScript, PDF or to SVG. You can save the layout as a template and load it again in another session. Finally, generating several maps based on a template can be done through the atlas generator.

18.1.1 Sample Session for beginners

Before you start to work with the print layout, you need to load some raster or vector layers in the QGIS map canvas and adapt their properties to suit your own convenience. After everything is rendered and symbolized to your liking, click the New Print Layout icon in the *Project* toolbar or choose *Project* ► *New Print Layout*. You will be prompted to choose a title for the new layout.

To demonstrate how to create a map please follow the next instructions.

1. On the left side, select the Add map toolbar button and draw a rectangle on the canvas holding down the left mouse button. Inside the drawn rectangle the QGIS map view to the canvas.

2. Select the Add scalebar toolbar button and click with the left mouse button on the print layout canvas. A scalebar will be added to the canvas.

3. Select the Add legend toolbar button and draw a rectangle on the canvas holding down the left mouse button. Inside the drawn rectangle the legend will be drawn.

4. Select the Select/Move item icon to select the map on the canvas and move it a bit.

5. While the map item is still selected you can also change the size of the map item. Click while holding down the left mouse button, in a white little rectangle in one of the corners of the map item and drag it to a new location to change its size.
6. Click the *Item Properties* panel on the left down side and find the setting for the orientation. Change the value of the setting *Map orientation* to '15.00°'. You should see the orientation of the map item change.
7. Now, you can print or export your print layout to image formats, PDF or to SVG with the export tools in *Layout* menu.
8. Finally, you can save your print layout within the project file with the Save Project button.

You can add multiple elements to the print layout. It is also possible to have more than one map view or legend or scale bar in the print layout canvas, on one or several pages. Each element has its own properties and, in the case of the map, its own extent. If you want to remove any elements from the layout canvas, you can do that with the `Delete` or the `Backspace` key.

18.1.2 The Layout Manager

The *Layout Manager* is the main window to manage print layouts in the project. It gives you an overview of existing print layouts and reports in the project and offers tools to:

- search for a layout;
- add new print layout or new report from scratch, template or duplicating an existing one;
- rename or delete any of them;
- open them in the project.

To open the layout manager dialog:

- from the main QGIS dialog, select *Project* ► *Layout Manager...* menu or click on the Layout Manager button in the *Project Toolbar*;
- from a print layout or report dialog, select *Layout* ► *Layout Manager...* menu or click on the Layout Manager button in the *Layout Toolbar*.

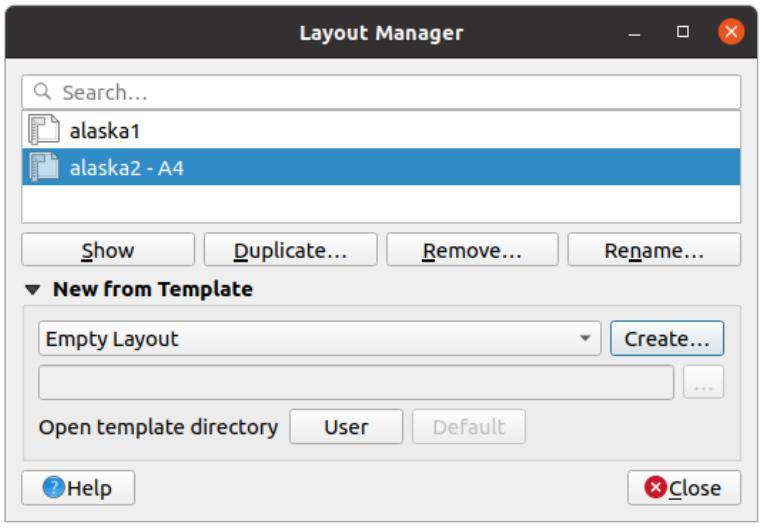


Fig. 18.1: The Print Layout Manager

The layout manager lists in its upper part all the available print layouts or reports in the project with tools to:

- show the selection: you can select multiple reports and/or print layout(s) and open them in one-click. Double-click a name also opens it;

- duplicate the selected print layout or report (available only if one item is selected): it creates a new dialog using the selected one as template. You'll be prompted to choose a new title for the new layout;
- rename the report or layout (available only if one item is selected): you'll be prompted to choose a new title for the layout;
- remove the layout: the selected print layout(s) will be deleted from the project.

In the lower part, it's possible to create new print layouts or reports from scratch or a template. By default, QGIS will look for templates in the *user profile* folder and the application template directories (accessible with the two buttons at the bottom of the frame) but also in any folder declared as *Path(s) to search for extra print templates* in *Settings* ► *Options* ► *Layouts*. Found templates are listed in the combobox. Select an item and press the *Create* button to generate a new report or print layout.

You can also use layout templates from a custom folder; in that case, select *specific* in the templates drop-down list, browse to the template and press *Create*.

> **Tip**
>
> **Creating template-based print layouts from Browser panel**
>
> Drag-and-drop a print layout template `.qpt` file from any file browser onto the map canvas or double-click it in the *Browser panel* generates a new print layout from the template.

18.1.3 Menus, tools and panels of the print layout

Opening the print layout provides you with a blank canvas that represents the paper surface when using the print option. Initially you find buttons on the left beside the canvas to add print layout items: 2D or 3D map canvases, text labels, images, legends, scale bars, basic shapes, arrows, attribute or simple tables, HTML frames, elevation profiles,... In this toolbar you also find buttons to navigate, zoom in on an area and pan the view on the layout as well as buttons to select any layout item and to move the extents of the map items.

Fig. 18.2 shows the initial view of the print layout before any elements are added.

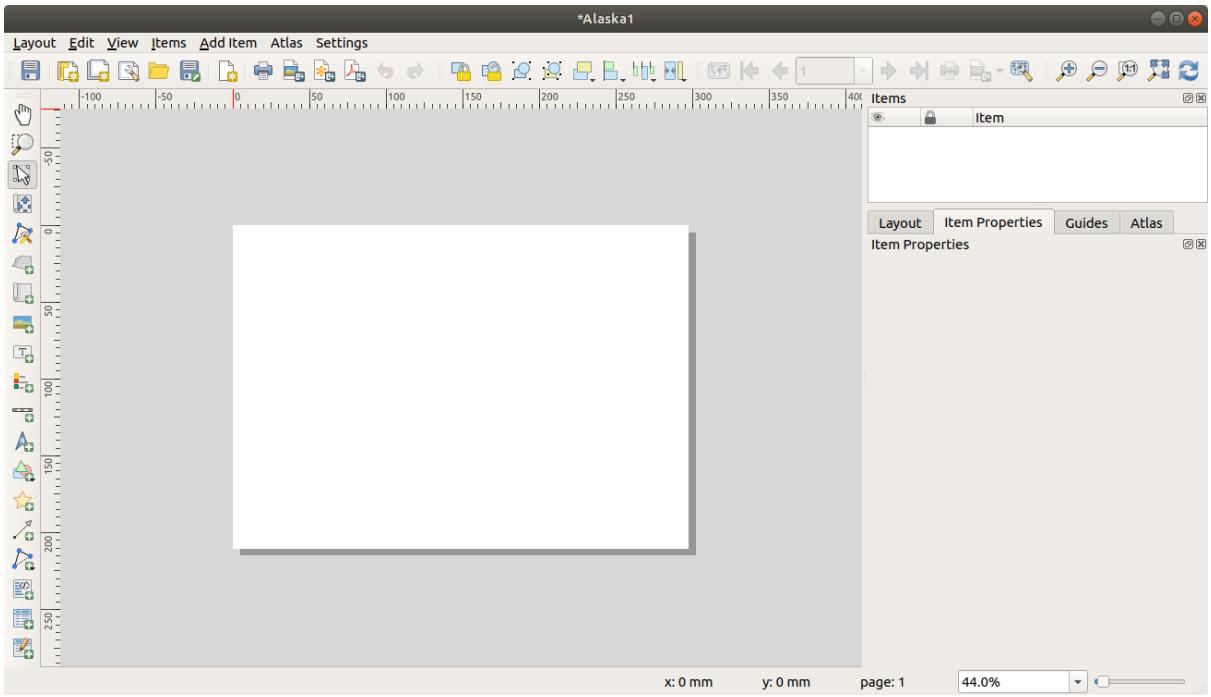


Fig. 18.2: 打印布局

On the right beside the canvas you find two set of panels. The upper one holds the panels *Items* and *Undo History* and the lower holds the panels *Layout*, *Item properties* and *Atlas generation*.

- The *Items* panel provides a list of all the print layout items added to the canvas and ways to globally interact with them (see *The Items Panel* for more information).
- The *Undo History* panel displays the history of all changes applied to the layout. With a mouse click, it is possible to undo and redo layout steps back and forth to a certain status.
- The *Layout* panel allows you to set general parameters to apply to the layout when exporting or working within (see *The Layout Panel* for more details);
- The *Item Properties* panel displays the properties for the selected item. Click the Select/Move item icon to select an item (e.g., legend, scale bar or label) on the canvas. Then click the *Item Properties* panel and customize the settings for the selected item (see 布局项 for detailed information on each item settings).
- The *Atlas* panel allows you to enable the generation of an atlas for the current layout and gives access to its parameters (see *Generate an Atlas* for detailed information on atlas generation usage).

In the bottom part of the print layout window, you can find a status bar with mouse position, current page number, a combo box to set the zoom level, the number of selected items if applicable and, in the case of atlas generation, the number of features.

In the upper part of the print layout window, you can find menus and other toolbars. All print layout tools are available in menus and as icons in a toolbar.

The toolbars and the panels can be switched off and on using the right mouse button over any toolbar, or through *View* ► *Toolbars* ► or *View* ► *Panels* ►.

Menus and Tools

Layout menu

The *Layout* provides action to manage the layout:

- Save the project file directly from the print layout window.
- Create a new and blank print layout with *New Layout...*.
- *Duplicate Layout...* : Create a new print layout by duplicating the current one.
- Remove the current layout with *Delete Layout...*.
- Open the *Layout Manager...*.
- *Layouts* ► : Open an existing print layout.

Once the layout is designed, with *Save as Template* and *Add Items from Template* icons, you can save the current state of a print layout session as a `.qpt` template file and load its items again in another session/print layout.

In the *Layout* menu, there are also powerful ways to share geographical information produced with QGIS that can be included in reports or published. These tools are *Export as Image...*, *Export as PDF...*, *Export as SVG...* and *Print...*.

Below is a list of all the available tools in this menu with some convenient information.

Table 18.1: ”Layout” Menu Tools

工具	快捷键	工具栏	参考
保存工程	Ctrl+S	布局	*QGIS 工程简介*
新建布局	Ctrl+N	布局	*The Layout Manager*
创建布局副本		布局	*The Layout Manager*
删除布局			
布局管理器...		布局	*The Layout Manager*
布局 ►			
布局属性...			*The Layout Panel*
重命名布局...			
Page Properties...			*Working with the page properties*
添加页面...		布局	*Working with the page properties*
由模板添加项		布局	*Creating a layout item*
另存为模板...		布局	*The Layout Manager*
导出为图像...		布局	*Export as Image*
导出为 *SVG* 文件...		布局	*Export as SVG*
导出为 *PDF* 文件...		布局	*Export as PDF*
Printer Page Setup...	Ctrl+Shift+P		
打印...	Ctrl+P	布局	*Creating an Output*
关闭	Ctrl+Q		

Edit menu

The *Edit* menu offers tools to manipulate print layout items. It includes common actions like selection tools, Copy/Cut/Paste and undo/redo (see *The Undo History Panel: Revert and Restore actions*) functionality for the items in the layout.

When using the Paste action, the elements will be pasted according to the current mouse position. Using the *Edit* ► *Paste in Place* action or pressing Ctrl+Shift+V will paste the items into the current page, at the same position they were in their initial page. It ensures to copy/paste items at the same place, from page to page.

Below is a list of all the available tools in this menu with some convenient information.

Table 18.2: ”Edit” Menu Tools

工具	快捷键	工具栏	参考
撤销 (上次修改)	Ctrl+Z	布局	*The Undo History Panel: Revert and Restore actions*
重做 (上次恢复的修改)	Ctrl+Y	布局	*The Undo History Panel: Revert and Restore actions*
删除	Del		
剪切	Ctrl+X		
复制	Ctrl+C		
粘贴	Ctrl+V		
就地粘贴	Ctrl+Shift+V		

continues on next page

Table 18.2 – continued from previous page

工具	快捷键	工具栏	参考
全选	Ctrl+A		
取消全选	Ctrl+Shift+A		
反选			
向下选择下一个项	Ctrl+Alt+[		
向上选择下一个项	Ctrl+Alt+]		
平移布局	P	工具箱	
缩放	Z	工具箱	
选择/移动项	V	工具箱	*Interacting with layout items*
移动内容	C	工具箱	*The Map Item*
编辑节点项		工具箱	基于结点的形状项

View menu

The *View* menu gives access to navigation tools and helps to configure general behavior of the print layout. Beside the common zoom tools, you have means to:

- Refresh view (if you find the view in an inconsistent state);
- enable a *grid* you could snap items to when moving or creating them. Grids setting is done in *Settings* ► *Layout Options...* or in the *Layout Panel*;
- enable *guides* you could snap items to when moving or creating them. Guides are red lines that you can create by clicking in the ruler (above or at the left side of the layout) and drag and drop to the desired location;
- *Smart Guides*: uses other layout items as guides to dynamically snap to as you move or reshape an item;
- *Clear Guides* to remove all current guides;
- *Show Bounding box* around the items to better identify your selection;
- *Show Rules* around the layout;
- *Show Pages* or set up pages to transparent. Often layout is used to create non-print layouts, e.g. for inclusion in presentations or other documents, and it's desirable to export the composition using a totally transparent background. It's sometimes referred to as "infinite canvas" in other editing packages.

In the print layout, you can change the zoom level using the mouse wheel or the slider and combo box in the status bar. If you need to switch to pan mode while working in the layout area, you can hold the Spacebar or the mouse wheel. With Ctrl+Spacebar, you can temporarily switch to Zoom In mode, and with Ctrl+Alt+Spacebar, to Zoom Out mode.

Panels and toolbars can be enabled from the *View* ► menu. To maximise the space available to interact with a composition you can check the *View* ► *Toggle Panel Visibility* option or press Ctrl+Tab; all panels are hidden and only previously visible panels are restored when unchecked.

It's also possible to switch to a full screen mode to have more space to interact with by pressing F11 or using *View* ► *Toggle Full Screen*.

Table 18.3: "View" Menu Tools

工具	快捷键	工具栏	参考
刷新	F5	导航	

continues on next page

Table 18.3 – continued from previous page

工具	快捷键	工具栏	参考
预览 ►			
放大	Ctrl++	导航	
缩小	Ctrl+-	导航	
缩放到 *100%*	Ctrl+1	导航	
全图显示	Ctrl+0	导航	
缩放到宽度			
显示网格	Ctrl+'		*Guides and Grid*
捕捉到网格	Ctrl+Shift+'		*Guides and Grid*
显示参考线	Ctrl+;		*Guides and Grid*
捕捉到参考线	Ctrl+Shift+;		*Guides and Grid*
智能参考线	Ctrl+Alt+;		
管理参考线...			*The Guides Panel*
清除参考线			*The Guides Panel*
显示标尺	Ctrl+R		
显示边界框	Ctrl+Shift+B		
显示页面			
工具栏 ►			面板和工具栏
面板 ►			面板和工具栏
切换全屏模式	F11		视图
切换面板可见性	Ctrl+Tab		视图

Items menu

The *Items* helps you configure items' position in the layout and the relations between them (see *Interacting with layout items*).

Table 18.4: "Items" Menu Tools

工具	快捷键	工具栏	参考
Group	Ctrl+G	*Actions*	*Grouping items*
Ungroup	Ctrl+Shift+G	*Actions*	*Grouping items*
Raise	Ctrl+]	*Actions*	*Alignment and distribution*
Lower	Ctrl+[	*Actions*	*Alignment and distribution*
Bring to Front	Ctrl+Shift+]	*Actions*	*Alignment and distribution*
Send to Back	Ctrl+Shift+[	*Actions*	*Alignment and distribution*
Lock Selected Items	Ctrl+L	*Actions*	*Locking items*
Unlock All	Ctrl+Shift+L	*Actions*	*Locking items*
Align Items ►		*Actions*	*Alignment and distribution*
Distribute Items ►		*Actions*	*Moving and resizing items*
Resize ►		*Actions*	*Moving and resizing items*

Add Item menu

These are tools to create layout items. Each of them is deeply described in 布局项 chapter.

Table 18.5: ”Add Item” Menu Tools

工具	工具栏	参考
Add Map	工具箱	*The Map Item*
Add 3D Map	工具箱	*The 3D Map Item*
Add Picture	工具箱	*The Picture Item*
Add Label	工具箱	*The Label Item*
Add Dynamic Text ►		*The Label Item*
Add Legend	工具箱	图例项
Add Scale Bar	工具箱	*The Scale Bar Item*
Add North Arrow	工具箱	*The North Arrow Item*
Add Shape ►	工具箱	常规图形项
► *Add Rectangle*	工具箱	常规图形项
► *Add Ellipse*	工具箱	常规图形项
► *Add Triangle*	工具箱	常规图形项
Add Marker	工具箱	*The Marker Item*
Add Arrow	工具箱	*The Arrow Item*
Add Node Item ►	工具箱	基于结点的形状项
► *Add Polygon*	工具箱	基于结点的形状项
► *Add Polyline*	工具箱	基于结点的形状项
Add HTML	工具箱	*The HTML Frame Item*
Add Attribute Table	工具箱	*The attribute table item*
Add Fixed Table	工具箱	*The fixed table item*
Add Elevation Profile	工具箱	*The Elevation Profile Item*

Atlas menu

Table 18.6: ”Atlas” Menu Tools

工具	快捷键	工具栏	参考
Preview Atlas	`Ctrl+ALt+/`	*Atlas*	*Preview and generate an atlas*
First Feature	`Ctrl+<`	*Atlas*	*Preview and generate an atlas*
Previous Feature	`Ctrl+,`	*Atlas*	*Preview and generate an atlas*
Next Feature	`Ctrl+.`	*Atlas*	*Preview and generate an atlas*
Last feature	`Ctrl+>`	*Atlas*	*Preview and generate an atlas*
Print Atlas...		*Atlas*	*Preview and generate an atlas*

continues on next page

Table 18.6 – continued from previous page

工具	快捷键	工具栏	参考
Export Atlas as Images...		*Atlas*	*Preview and generate an atlas*
Export Atlas as SVG...		*Atlas*	*Preview and generate an atlas*
Export Atlas as PDF...		*Atlas*	*Preview and generate an atlas*
Atlas Settings		*Atlas*	*Generate an Atlas*

Settings Menu

The *Settings* ► *Layout Options...* menu is a shortcut to *Settings* ► *Options* ► *Layouts* menu of QGIS main canvas. Here, you can set some options that will be used as default on any new print layout:

- *Layout defaults* let you specify the default font to use;
- With *Grid appearance*, you can set the grid style and its color. There are three types of grid: **Dots**, **Solid** lines and **Crosses**;
- *Grid and guide defaults* defines spacing, offset and tolerance of the grid (see *Guides and Grid* for more details);
- *Layout Paths*: to manage list of custom paths to search print templates.

The *Settings* ► *Keyboard Shortcuts...* menu allows you to use the *shortcuts manager* in the print layout interface.

Contextual menus

Depending on where you right-click in the print layout dialog, you open a contextual menu with various features:

- Right-click on the menu bar or any toolbar and you get the list of layout panels and toolbars you can enable or disable in one-click.
- Right-click over a ruler and you can *Show Guides*, *Snap to Guides*, *Manage Guides...* opening the *Guides panel* or *Clear Guides*. It's also possible to hide the rulers.
- Right-click in the print layout canvas and:
  - You'll be able to *Undo* and *Redo* recent changes, or *Paste* any copied item (only available if no item is selected).
  - If you click over a page, you can additionally access the current *Page Properties* panel or *Remove Page*.
  - If you click on a selected item then you can cut or copy it as well as open the *Item Properties* panel.
  - If more than one item are selected, then you can either group them and/or ungroup if at least one group is already in the selection.
- Right-click inside a text box or spinbox widget of any layout panel provides edit options to manipulate its content.

The Layout Panel

In the *Layout* panel, you can define the global settings of your print layout.

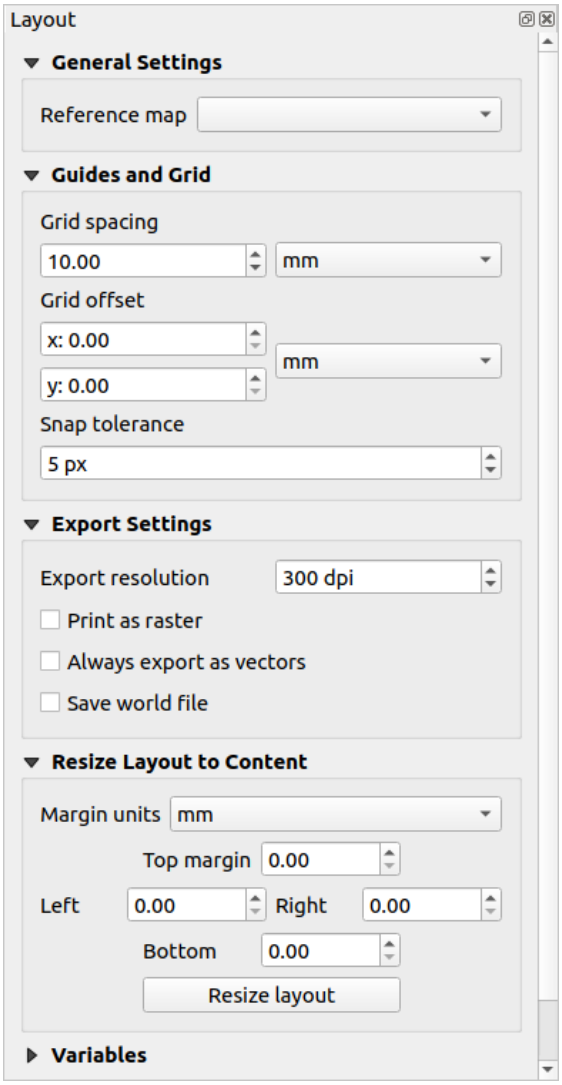


Fig. 18.3: Layout Settings in the Print Layout

General settings

In a print layout, you can use more than one map item. The *Reference map* represents the map item to use as the layout's master map. It's assigned as long as there's a map item in the layout. The layout will use this map in any of their properties and variables calculating units or scale. This includes exporting the print layout to georeferenced formats.

Moreover, new layout items such as scale bar, legend or north arrow have by default their settings (orientation, displayed layers, scale, ...) bound to the map item they are drawn over, and fall back to the reference map if there is no overlapping map.

Guides and Grid

You can put some reference marks on your paper sheet to help you accurately place some items. These marks can be:

- simple horizontal or vertical lines (called **Guides**) put at the position you want (see *The Guides Panel* for guides creation).
- or regular **Grid**: a network of horizontal and vertical lines superimposed over the layout.

Settings like *Grid spacing* or *Grid offset* can be adjusted in this group as well as the *Snap tolerance* to use for items. The tolerance is the maximum distance below which the mouse cursor is snapped to a grid or a guide, while moving, resizing or creating an item.

Whether grid or guides should be shown is set in *View* menu. There, you can also decide if they might be used to snap layout items. When both a grid line and a guide line are within tolerance of a point, guides will always take precedence - since they have been manually set (hence, assumption that they have been explicitly placed at highly desirable snapping locations, and should be selected over the general grid).

> **Note**
>
> In the *Settings* ► *Layout Options* menu, you can also set the grid and guides parameters exposed above. However, these options will only apply as defaults to new print layouts.

Export settings

You can define a resolution to use for all exported maps in *Export resolution*. This setting can then be overridden each time you export a map.

Because of some advanced rendering options (*blending mode*, *effects*...), a layout item may need rasterization in order to be exported correctly. QGIS will individually rasterize it without forcing every other item to also be rasterized. This allows printing or saving as PostScript or PDF to keep items as much as possible as vectors, e.g. a map item with layer opacity won't force labels, scale bars, etc to be rasterized too. You can however:

- force all the items to be rasterized checking the *Print as raster* box;
- or use the opposite option, i.e. *Always export as vectors*, to force the export to keep items as vectors when exported to a compatible format. Note that in some cases, this could cause the output to look different to layout.

Where the format makes it possible (e.g., `.TIF`, `.PDF`) exporting a print layout results by default in a georeferenced file (based on the *Reference map* item in the *General settings* group). For other formats, georeferenced output requires you to generate a world file by checking *Save world file*. The world file is created beside the exported map(s), has the name of the page output with the reference map item and contains information to georeference it easily.

Resize layout to content

Using the *Resize page* tool in this group, you create a unique page composition whose extent covers the current contents of the print layout (with some optional *margins* around the cropped bounds).

Note that this behavior is different from the *crop to content* option in that all the items are placed on a real and unique page in replacement of all the existing pages.

Variables

The *Variables* lists all the variables available at the layout's level (which includes all global and project's variables).

It also allows the user to manage layout-level variables. Click the button to add a new custom layout-level variable. Likewise, select a custom layout-level variable from the list and click the button to remove it.

More information on variables usage in the *General Tools* section.

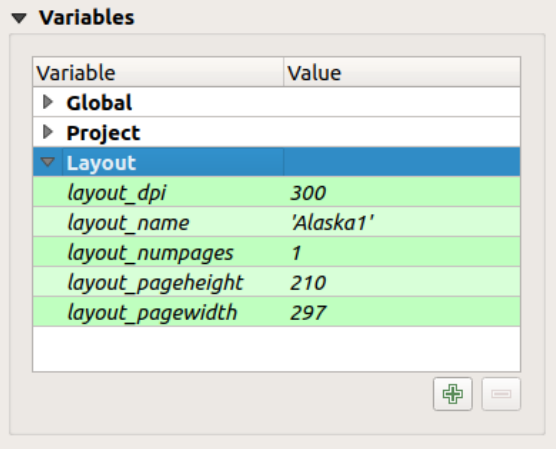


Fig. 18.4: Variables Editor in the Print Layout

Working with the page properties

A layout can be composed of several pages. For instance, a first page can show a map canvas, and a second page can show the attribute table associated with a layer, while a third one shows an HTML frame linking to your organization website. Or you can add many types of items on each page.

Adding a new page

Futhermore, a layout can be made using different size and/or orientation of pages. To add a page, select the *Add Pages...* tool from the *Layout* menu or *Layout Toolbar*. The *Insert Pages* dialog opens and you are asked to fill:

- the number of pages to insert;
- the position of the page(s): before or after a given page or at the end of the print layout;
- The *Page size*: it could be of a preset format page (`A4`, `B0`, `Legal`, `Letter`, `ANSI A`, `Arch A` and their derivatives as well as a resolution type, such as `1920x1080` or `1024x768`) with associated *Orientation* (Portrait or Landscape).

  The page size can also be of a `custom` format; In that case, you'd need to enter its *Width* and *Height* (with locked size ratio if needed) and select the unit to use among `mm`, `cm`, `px`, `pt`, `in`, `ft`... Conversion of entered values is automatically applied when switching from one unit to another.

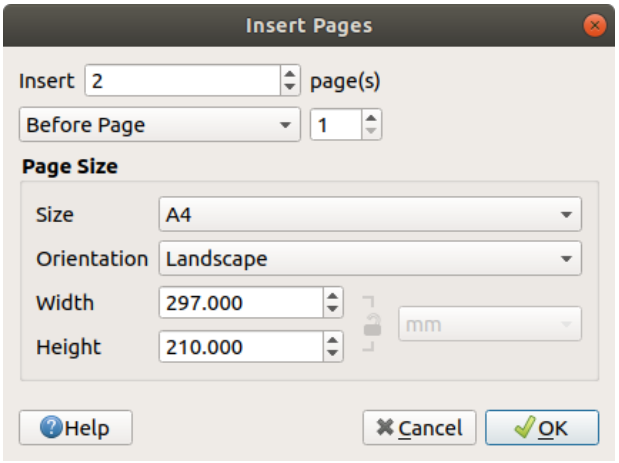


Fig. 18.5: Creating a new page in the Print Layout

Updating page properties

Any page can be later customized through the Page *Item Properties* panel. To access a page's properties, either:

- Left-click on an empty section of the page
- Right-click on a page and select *Page Properties...*.
- Go to *Layout* ► *Page properties...* while hovering over the page

The *Item Properties* panel opens with settings such as:

- for layouts with several pages, it displays the active page number and the count of available pages
- the *Page size* frame described above. You can modify each property using the data defined override options (see *Explore Data-defined override buttons with atlas* for a use case);
- the *Background* of the current page using the *color* or *symbol* you want.
- the *Apply to all Pages* button updates the current page properties to the other existing pages in the layout;
- the *Exclude page from exports* to control whether the current page with its content should be included in the *layout output*;

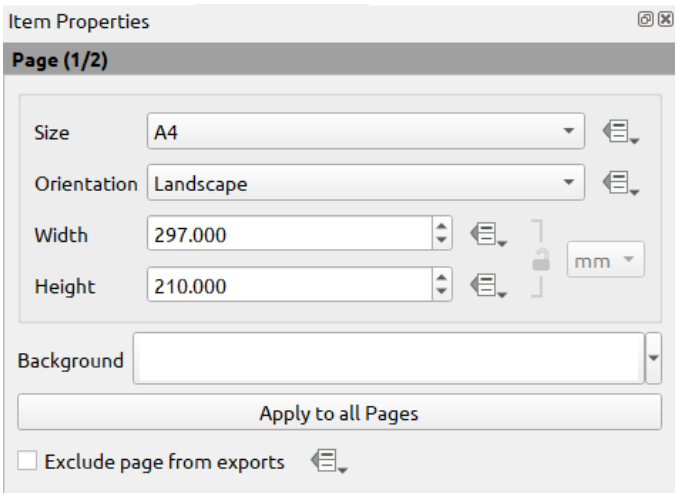


Fig. 18.6: Page properties dialog

The Guides Panel

Guides are vertical or horizontal line references you can place on a layout page to assist you on items placement, when creating, moving or resizing them. To be active, guides require the *View* ► *Show Guides* and *View* ► *Snap to Guides* options to be checked. To create a guide, there are two different methods:

- if the *View* ► *Show Rulers* option is set, drag out a ruler and release the mouse button within the page area, at the desired position.
- for more precision, use the *Guides* panel from the *View* ► *Toolbox* ► or by selecting *Manage guides for page...* from the page's contextual menu.

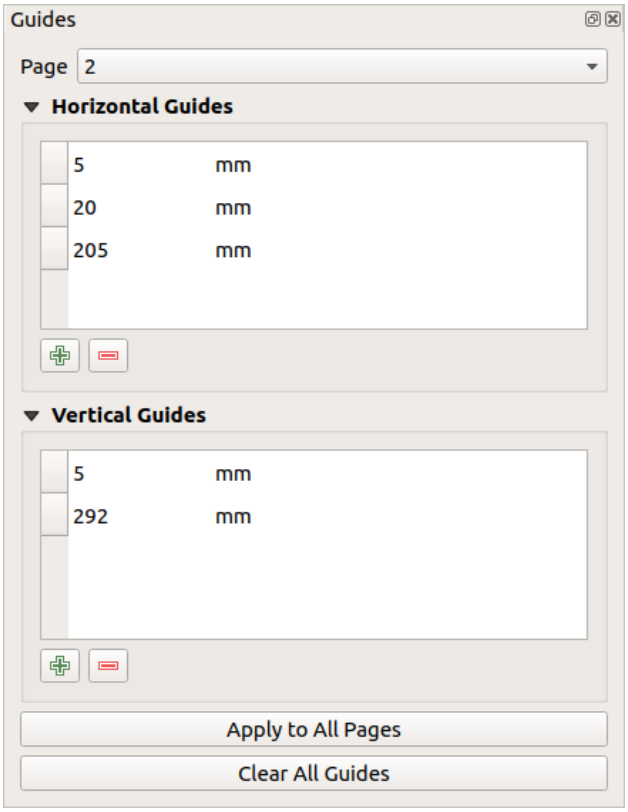


Fig. 18.7: The Guides panel

The *Guides* panel allows creation of snap lines at specific locations:

1. Select the *Page* you'd like to add the guides to
2. Click the Add new guide button and enter the coordinates of the horizontal or vertical line. The origin is at the top left corner. Different units are available for this.

   The panel also allows adjusting the position of existing guides to exact coordinates: double-click and replace the value.
3. The *Guides* panel lists only the items for the current page. It allows creation or removal of guides only in the current page. However, you can use the *Apply to All Pages* button to replicate the guide configuration of the current page to the other pages in the layout.
4. To delete a guide, select it and press the Remove selected guide button. Use *Clear All Guides* to remove all the guides in the current page.

> **Tip**
>
> **Snapping to existing layout items**
>
> Other than guides and grids, you can use existing items as snapping references when moving, resizing or creating new items; these are called **smart guides** and require *View* ► *Smart Guides* option to be checked. Anytime the mouse pointer is close to an item's bound, a snapping cross appears.

The Items Panel

The *Items* panel offers some options to manage selection and visibility of items. All the items added to the print layout canvas (including *items group*) are shown in a list and selecting an item makes the corresponding row selected in the list as well as selecting a row does select the corresponding item in the print layout canvas. This is thus a handy way to select an item placed behind another one. Note that a selected row is shown as bold. Multiple items selection is also possible holding `Shift` for contiguous items, and `Ctrl` key for non-contiguous ones.

For any available item, you can:

- set it visible or not;
- lock or unlock its position;
- sort its Z position. You can move up and down each item in the list with a click and drag. The upper item in the list will be brought to the foreground in the print layout canvas. By default, a newly created item is placed in the foreground.
- change the item ID by double-clicking the text;
- right-click an item and select whether to copy or delete it or open its *properties panel*.

Once you have found the correct position for an item, you can lock it by ticking the box in column. Locked items are **not** selectable on the canvas. Locked items can be unlocked by selecting the item in the *Items* panel and unchecking the tickbox or you can use the icons on the toolbar.

The Undo History Panel: Revert and Restore actions

During the layout process, it is possible to revert and restore changes. This can be done with the revert and restore tools available in the *Edit* menu, the *Layout* toolbar or the contextual menu any time you right-click in the print layout area:

- Revert last change
- Restore last change

This can also be done by mouse click within the *Undo history* panel (see Fig. 18.8). The History panel lists the last actions done within the print layout. Select the point you want to revert to and once you do a new action all the actions done after the selected one will be removed.

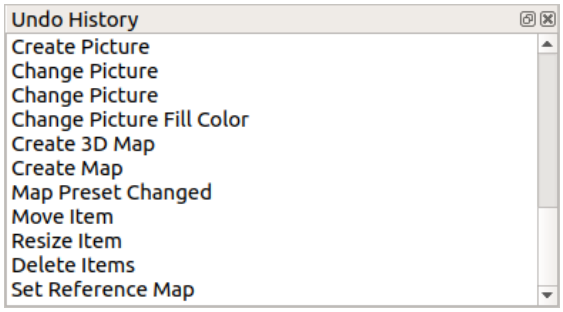


Fig. 18.8: Undo History in the Print Layout

18.2 布局项

18.2.1 Layout Items Common Options

QGIS provides a large set of items to layout a map. They can be of map, legend, scale bar, picture, table, north arrow, image type... They however share some common options and behavior that are exposed below.

Creating a layout item

Items can be created using different tools, either from scratch or based on existing items.

To create a layout item from scratch:

1. Select the corresponding tool either from the *Add Item* menu or the *Toolbox* bar.
2. Then:
   - Click on the page and fill the size and placement information requested in the *New Item Properties* dialog that pops up (for details, see *Position and Size*);

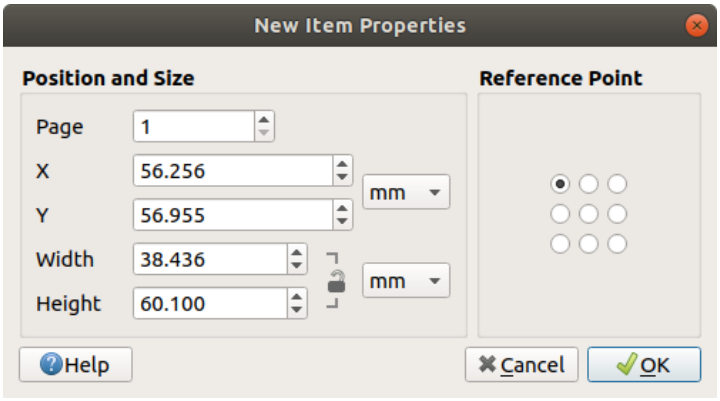


Fig. 18.9: New Item properties dialog

   - Or click-and-drag to define the initial size and placement of the item. You can rely on *grids and guides* snapping for a better position.

> **Note**
>
> Because they can have particular shapes, drawing node or arrow items does not work with one-click nor click-and-drag methods; you need to click and place each node of the item. See 基于结点的形状项 for more details.

You can also:

1. Select an existing item with the Select/Move item button from the *Toolbox* toolbar
2. Use the contextual menu or the *Edit* menu tools to copy/cut the item and paste it at the mouse position as a new item.

   You can also use the *Paste in Place* (`Ctrl+Shift+V`) command to duplicate an item from one page to another and place it in the new page at the same coordinates as the original.

Moreover, you can create items using a print layout template (for details, see *The Layout Manager*) through the *Layout* ► *Add Items from Template...* command.

> **Tip**
>
> **Add layout items using the file browser**
>
> From your file browser or using the *Browser* panel, drag-and-drop a print layout template (`.qpt` file) onto a print layout dialog and QGIS automatically adds all items from that template to the layout.

Interacting with layout items

Each item inside the print layout can be moved and resized to create a perfect layout. For both operations the first step is to activate the Select/Move item tool and click on the item.

You can select multiple items with the Select/Move item button: click and drag over the items or hold the `Shift` button and click on each of the items you want. To deselect an item, click on it holding the `Shift` button.

Each time there's a selection, count of selected items is displayed on the status bar. Inside the *Edit* menu, you can find actions to select all the items, clear all selections, invert the current selection and more...

Moving and resizing items

Unless *View* ► *Show Bounding Boxes* option is unchecked, a selected item will show squares on its boundaries; moving one of them with the mouse will resize the item in the corresponding direction. While resizing, holding `Shift` will maintain the aspect ratio. Holding `Alt` will resize from the item center.

To move a layout item, select it with the mouse and move while holding the left button. If you need to constrain the movements to the horizontal or vertical axis, hold the `Shift` button on the keyboard while moving the mouse. You can also move a selected item using the `Arrow keys` on the keyboard; if the movement is too slow, you can speed it up by holding `Shift`. If you need better precision, use the *Position and size* properties, or grid/guides snapping as explained above for item's creation.

Resizing or moving several items at once is made the same way as for a single item. QGIS however provides some advanced tools to automatically resize a selection of items following different rules:

- each item height matches the tallest or the shortest selected item;
- each item width matches the widest or the narrowest selected item;
- resizes items to squares: each item is enlarged to shape a square.

Likewise, there are *tools* available to organize multiple items position by distributing equidistantly:

- edges (left, right, top or bottom) of items;
- centers of items horizontally or vertically;
- spacing between items horizontally or vertically.

Grouping items

Grouping items allows you to manipulate a set of items like a single one: you can easily resize, move, delete, copy the items as a whole.

To create a group of items, select more than one and press the . *Group* button on the *Items* menu or the *Actions* toolbar, or from the right-click menu. A row named `Group` is added to the *Items* panel and can be locked or hidden like any other *Items panel's object*. Grouped items are **not individually** selectable on the canvas; use the Items panel for direct selection and access the item's properties panel.

Locking items

Once you have found the correct position for an item, you can lock it by using the *Lock selected items* button in the *Items* menu or the *Actions* toolbar, or ticking the box next to the item in the *Items* panel. Locked items are **not** selectable on the canvas.

Locked items can be unlocked by selecting the item in the *Items* panel and unchecking the tickbox or you can use the icons on the toolbar.

Alignment and distribution

Tools for raising or lowering the Z position of items in the layout are inside the

pull-down menu. Choose an element on the print layout canvas and select the matching functionality to raise or lower the selected element over the other elements. This order is shown in the *Items* panel. You can also raise or lower objects in the *Items* panel by clicking and dragging an object's label in this list.

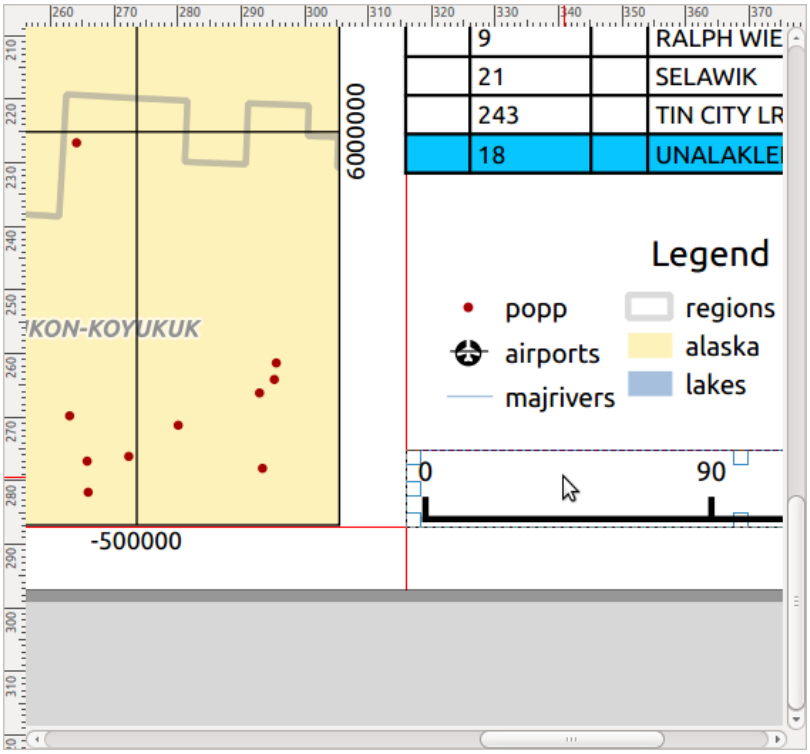


Fig. 18.10: Alignment helper lines in the print layout

There are several alignment options available within the Align selected items pull-down menu (see Fig. 18.10). To use an alignment function, you first select the elements and then click on one of the alignment icons:

- *Align Left* or *Align Right*;
- *Align Top* or *Align Bottom*;
- *Align Center* horizontally or *Align Center Vertical*.

All selected elements will then be aligned to their common bounding box. When moving items on the layout canvas, alignment helper lines appear when borders, centers or corners are aligned.

Another way to improve layout items placement is by adjusting the spacing between them, over the layout page. This can be done by selecting the items and press the *Distribute Left Edges* drop-down menu to:

- *Distribute Left Edges* or *Distribute Right Edges* of items equidistantly
- *Distribute Top edges* or *Distribute Bottom Edges* of items equidistantly
- *Distribute Horizontal Centers* or *Distribute Vertical Centers* of items equidistantly
- Add equal space between items: *Distribute Horizontal Spacing Equally* or *Distribute Vertical Spacing Equally*

Items Common Properties

Layout items have a set of common properties you will find at the bottom of the *Item Properties* panel: Position and size, Rotation, Frame, Background, Item ID, Variables and Rendering (see Fig. 18.11).

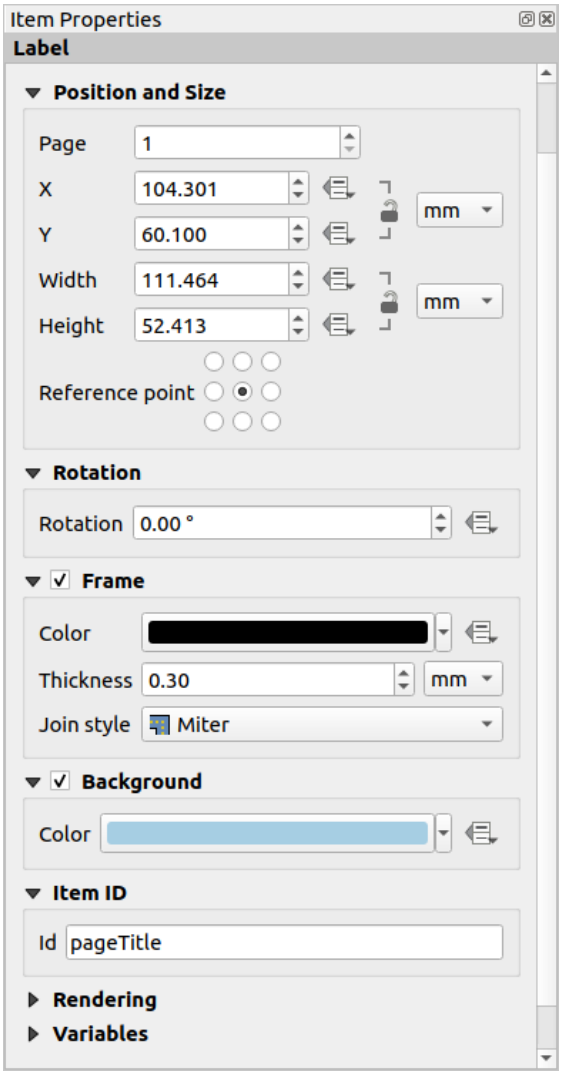


Fig. 18.11: Common Item Properties groups

Note

The Data defined override icon next to most of the options means that you can associate that property with a layer, features attributes, geometry or with any other layout item's property, using *expressions* or *variables*. For more information see *Data defined override setup*.

- The *Position and size* group lets you define the size and position of the frame which contains the item (see *Position and Size* for more information).
- The *Rotation* sets the rotation of the element (in degrees).
- The *Frame* shows or hides the frame around the item. Use the *Color*, *Thickness* and *Join style* widgets to adjust those properties.

- In the *Background* group you can create or pick a background *Color*. Transparency can be adjusted through altering the alpha field settings.
- Use the *Item ID* to create a relationship to other print layout items. This is used with QGIS server and other potential web clients. You can set an ID on an item (for example, a map or a label), and then the web client can send data to set a property (e.g., label text) for that specific item. The GetProjectSettings command will list the items and IDs which are available in a layout.
- *Rendering* mode helps you set whether and how the item can be displayed: you can, for instance, apply *blending mode*, adjust the opacity of the item or *Exclude item from exports*.

Position and Size

Extending the features of the *New Item Properties* dialog with data-defined capabilities, this group allows you to place the items accurately.

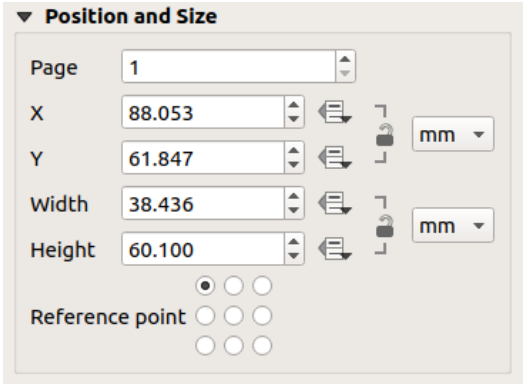


Fig. 18.12: Position and size

- the actual number of the page to place the item on;
- the reference point of the item;
- the *X* and *Y* coordinates of the *Reference point* of the item on the chosen page. The ratio between these values can be locked by clicking on the button. Changes made to a value using the widget or the Select/Move item tool will be reflected in both of them;
- the *Width* and *Height* of the item bounding box. As for coordinates, the ratio between width and height can be locked.

Rendering mode

QGIS allows advanced rendering for layout items just like vector and raster layers.

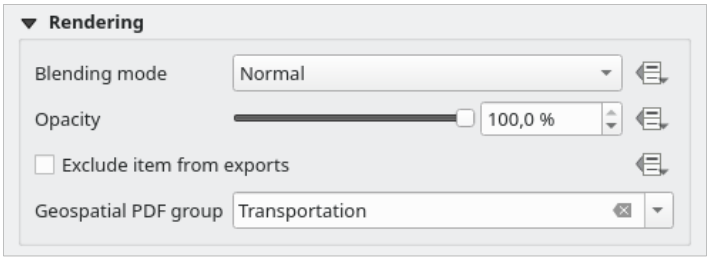


Fig. 18.13: Rendering mode

- *Blending mode*: With this tool you can achieve effects which would otherwise only be achieved using graphic rendering software. The pixels of your overlaying and underlaying items can be mixed according to the mode set (see *Blending Modes* for description of each effect).
- *Opacity*: You can make the underlying item in the layout visible with this tool. Use the slider to adapt the visibility of your item to your needs. You can also make a precise definition of the percentage of visibility in the menu beside the slider.
- *Exclude item from exports*: Combined with data-defined expressions, you can decide to make an item invisible in some exports. After activating this checkbox, the item will not be included in export to PDF, print etc..
- *Geospatial PDF group*: When set, a matching layer tree group will be created in the exported Geospatial PDF and the item will only be visible when this group is checked. This allows content to be selectively displayed as a group by viewers of the Geospatial PDF. E.g., it can allow extra layout content such as descriptive labels or legends to only be shown when layers from the group are visible, making the file export much more flexible.

Variables

The *Variables* lists all the variables available at the layout item's level (which includes all global, project and layout's variables). Layout map items alhave an additional *Map settings* section for variables that provide easy access to values like the map's scale, extent, and so on.

In *Variables*, it's also possible to manage layout item level variables. Click the button to add a new custom variable. Likewise, select any custom item-level variable from the list and click the button to remove it.

More information on variables usage in the 将值存储在变量中 section.

18.2.2 The Map Item

The map item is the main frame that displays the map you've designed in the map canvas. Use the *Add Map* tool following *items creation instructions* to add a new map item that you can later manipulate the same way as exposed in *Interacting with layout items*.

By default, a new map item shows the current status of the *map canvas* with its extent and visible layers. You can customize it thanks to the *Item Properties* panel. Other than the *items common properties*, this feature has the following functionalities:

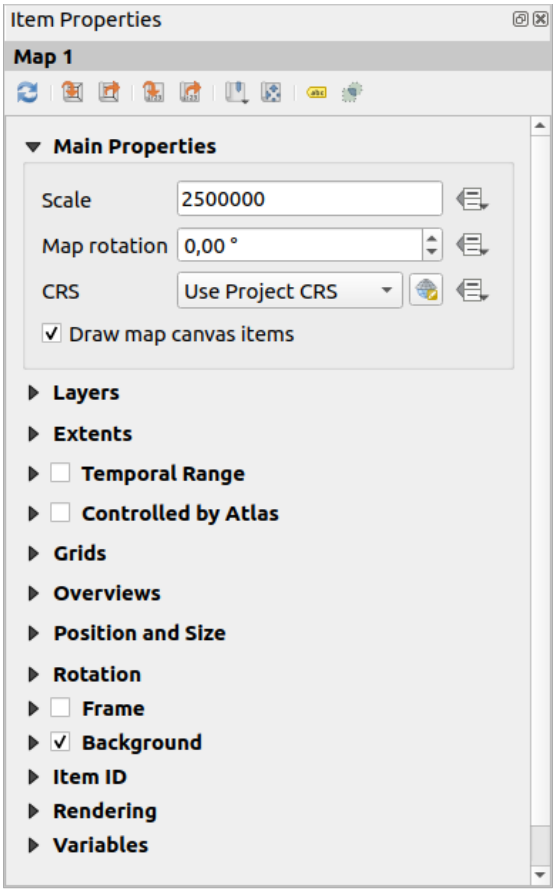


Fig. 18.14: Map Item Properties Panel

The Toolbar

The Map *Item Properties* panel embeds a toolbar with the following functionalities:

- Update map preview
- Set map canvas to match main canvas extent
- View current map extent in main canvas
- Set map scale to match main canvas scale
- Set main canvas to match current map scale
- Bookmarks: set the map item extent to match an existing spatial bookmark
- Interactively edit map extent: pan and zoom interactively within the map item
- Labeling settings: control feature label behaviour (placement, visibility...) in the layout map item extent:
  - set a *Margin from map edges*, a data definable distance from the map item's limits inside which no label should be displayed
  - *Allow truncated labels on edges of map*: controls whether labels which fall partially outside of the map item allowed extent should be rendered. If checked, these labels will be shown (when there's no way to place them fully within the visible area). If unchecked then partially visible labels will be skipped.

  - *Label blocking items*: allows other layout items (such as scalebars, north arrows, inset maps, etc) to be marked as a blockers for the map labels in the **active** map item. This prevents any map labels from being placed under those items - causing the labeling engine to either try alternative placement for these labels or discard them altogether.

    If a *Margin from map edges* is set, the map labels are not placed closer than the specified distance from the checked layout items.

  - *Show unplaced labels*: can be used to determine whether labels are missing from the layout map (e.g. due to conflicts with other map labels or due to insufficient space to place the label) by highlighting them in a *predefined color*.

- Clipping settings: allows to clip the map item to the atlas feature and to shape and polygon items:

  - *Clip to atlas feature*: you can determine that the layout map item will be clipped automatically to the current *atlas feature*.

    There are different clipping modes available:

    * *Clip During Render Only*: applies a painter based clip, so that portions of vector features which sit outside the atlas feature become invisible
    * *Clip Feature Before Render*: applies the clip before rendering features, so borders of features which fall partially outside the atlas feature will still be visible on the boundary of the atlas feature
    * *Render Intersecting Features Unchanged*: renders all features which intersect the current atlas feature, but without clipping their their geometry.

    You can *Force labels inside atlas feature*. If you don't want to *Clip all layers* to the atlas feature you can use the *Clip selected layers* option.

  - *Clip to item*: it is possible to change the shape of the map item by using a *shape* or *polygon* item from the print layout. When you enable this option the map will be automatically clipped to the selected shape in the combobox. Again, the above mentioned clipping modes are available and labels can be forced to display only inside the clipping shape.

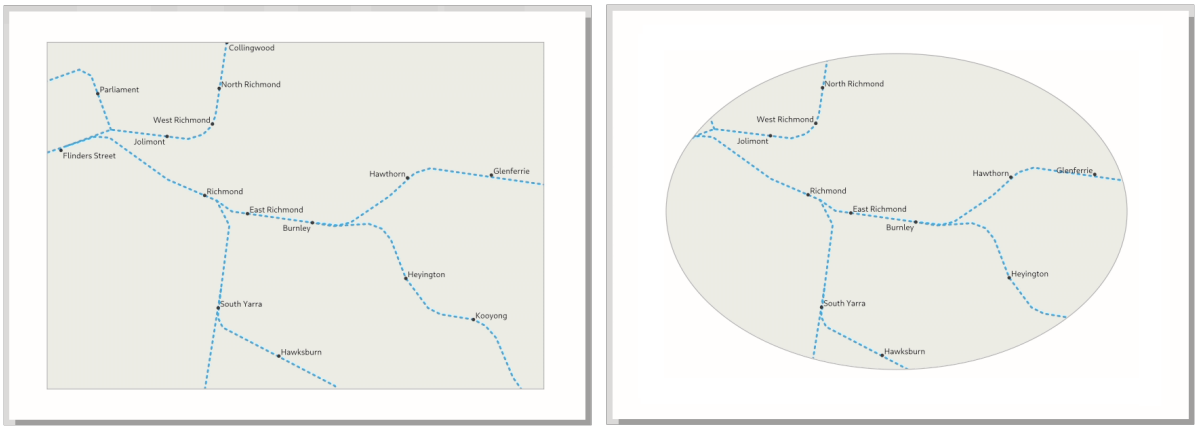


Fig. 18.15: Clipping a layout map item to shapes

主要属性

In the *Main properties* group (see Fig. 18.14) of the map *Item Properties* panel, available options are:

- The *Update Preview* button to refresh the map item rendering if the view in map canvas has been modified. Note that most of the time, the map item refresh is automatically triggered by the changes;
- The *Scale* to manually set the map item scale;
- The *Map rotation* allows you to rotate the map item content clockwise in degrees. The rotation of the map canvas can be imitated here;
- The *CRS* allows you to display the map item content in any *CRS*. It defaults to `Use project CRS`;
- *Draw map canvas items* lets you show in the print layout *annotations* that are placed on the main map canvas.

Layers

By default, map item appearance is synced with the map canvas rendering meaning that toggling visibility of the layers or modifying their style in the *Layers Panel* is automatically applied to the map item. Because, like any other item, you may want to add multiple map items to a print layout, there's a need to break this synchronization in order to allow showing different areas, layer combinations, at different scales... The *Layers* properties group (see Fig. 18.16) helps you do that.

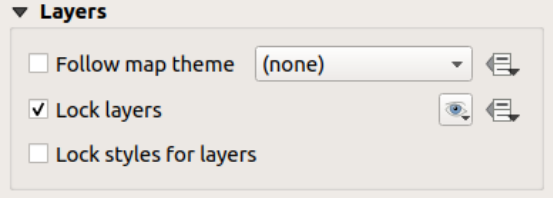


Fig. 18.16: Map Layers group

If you want to keep the map item consistent with an existing *map theme*, check *Follow map theme* and select the desired theme in the drop-down list. Any changes applied to the theme in QGIS' main window (using the replace theme function) will automatically affect the map item. If a map theme is selected, the *Lock styles for layers* option is disabled because *Follow map theme* also updates the style (symbology, labels, diagrams) of the layers.

To lock the layers shown in a map item to the current map canvas visibility, check *Lock layers*. When this option is enabled, any changes on the layers' visibility in QGIS' main window will not affect the layout's map item. Nevertheless, style and labels of locked layers are still refreshed according to QGIS' main window. You can prevent this by using *Lock styles for layers*.

Instead of using the current map canvas, you can also lock the layers of the map item to those of an existing map theme: select a map theme from the Set layer list from a map theme drop-down button, and the *Lock layers* is activated. The set of visible layers in the map theme is from now on used for the map item until you select another map theme or uncheck the *Lock layers* option. You then may need to refresh the view using the Refresh view button of the *Navigation* toolbar or the *Update Preview* button seen above.

Note that, unlike the *Follow map theme* option, if the *Lock layers* option is enabled and set to a map theme, the layers in the map item will not be refreshed even if the map theme is updated (using the replace theme function) in QGIS' main window.

Locked layers in the map item can also be *data-defined*, using the icon beside the option. When used, this overrides the selection set in the drop-down list. You need to pass a list of layers separated by `|` character. The following example locks the map item to use only layers `layer 1` and `layer 2`:

```
concat ('layer 1', '|', 'layer 2')
```

Extents

The *Extents* group of the map item properties panel provides the following functionalities (see Fig. 18.17):

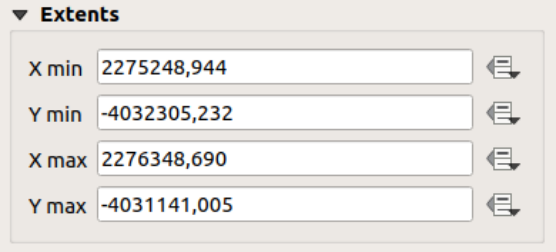


Fig. 18.17: Map Extents group

The **Extents** area displays `X` and `Y` coordinates of the area shown in the map item. Each of these values can be manually replaced, modifying the map canvas area displayed and/or map item size. The extent can also be modified using tools at the top of the map item panel such as:

- Set map canvas to match main canvas extent
- Set map scale to match main canvas scale

You can also alter a map item extent using the Move item content tool: click-and-drag within the map item to modify its current view, keeping the same scale. With the tool enabled, use the mouse wheel to zoom in or out, modifying the scale of the shown map. Combine the movement with `Ctrl` key pressed to have a smaller zoom.

Elevation range

The *Elevation range* setting in the map item properties allows you to filter the content of certain layers based on a specific elevation range. While all layers remain visible, those that support elevation filtering (currently point clouds and raster DEMs) will have their data filtered. Only the parts of these layers that fall within the elevation range set by the *Lower* and *Upper* values will be displayed.

The *Elevation range* can be data-defined. This means you can set different elevation ranges for different features in the *atlas* or *report*.

Temporal range

The *Temporal range* group of the map item properties panel provides the options to control layers rendering in the map item based on a temporal range. Only layers whose temporal properties overlap with the time range set by the *Start* and *End* dates are displayed in the map item.

The associated data-defined widgets help make the time range dynamic, and allow outputting temporal *atlases*, i.e. automated maps with fixed spatial extent and whose contents vary based on time. For example, using as coverage layer a csv file with a start and end pair of fields and a number of rows representing date ranges, enable both the temporal range and control by atlas in the map item properties and hit atlas export.

Controlled by atlas

The *Controlled by atlas* group properties is available only if an *atlas* is active in the print layout. Check this option if you want the map item being ruled by the atlas; when iterating over the coverage layer, the map item extent is panned/zoomed to the atlas feature following:

- *Margin around features*: zooms to the feature at the best scale, keeping around each a margin representing a percentage of the map item width or height. The margin can be the same for all features or *set variable*, e.g., depending on map scale;
- *Predefined scale (best fit)*: zooms to the feature at the project *predefined scale* where the atlas feature best fits;
- *Fixed scale*: atlas features are panned from one to another, keeping the same scale of the map item. Ideal when working with features of same size (e.g., a grid) or willing to highlight size differences among atlas features.

Grids

With grids, you can add, over your map, information relative to its extent or coordinates, either in the map item projection or a different one. The *Grids* group provides the possibility to add several grids to a map item.

- With the and buttons you can add or remove a selected grid;
- With the and buttons you can move up and down a grid in the list, hence move it on top or bottom of another one, over the map item.

Double-click the added grid to rename it.

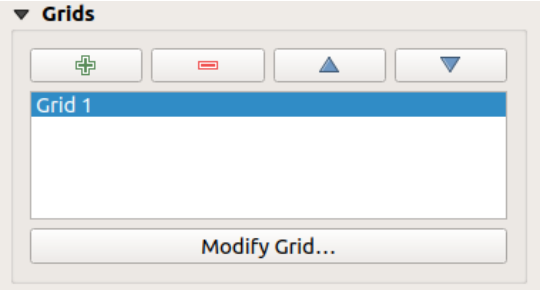


Fig. 18.18: Map Grids Dialog

To modify a grid, select it and press the *Modify Grid...* button to open the *Map Grid Properties* panel and access its configuration options.

Grid Appearance

In the *Map Grid Properties* panel, check *Grid enabled* to show the grid on the map item.

As grid type, you can specify to use a:

- **Solid**: shows a line across the grid frame. The *Line style* can be customized using *color* and *symbol* selector widget;
- **Cross**: displays segment at the grid lines intersection for which you can set the *Line style* and the *Cross width*;
- **Markers**: only displays customizable markers symbol at grid lines intersection;
- or **Frame and annotations only**.

Other than the grid type, you can define:

- the *CRS* of the grid: by default, it will follow the map item CRS. Press Select CRS button to set it to a different CRS.
- the *Interval* type to use for the grid references:
  - **Map Units**: you set a distance within the map (in the unit of the grid CRS) between consecutive grid references in the *X* and *Y* directions. The number of grid ticks will vary depending on the map scale.
  - choosing **Fit Segment Width** will dynamically select the grid interval based on the map extent to a "pretty" interval. That optimal interval is calculated within a range of distances whose *Minimum* and *Maximum* values can be customized.
  - With **Millimeters** or **Centimeters**, you set a distance on the paper between consecutive grid references in the *X* and *Y* directions. The number of grid ticks will be the same whatever the map scale.
- the *Offset* from the map item edges, in the *X* and/or the *Y* direction
- and the *Blend mode* of the grid (see *Blending Modes*) when compatible.

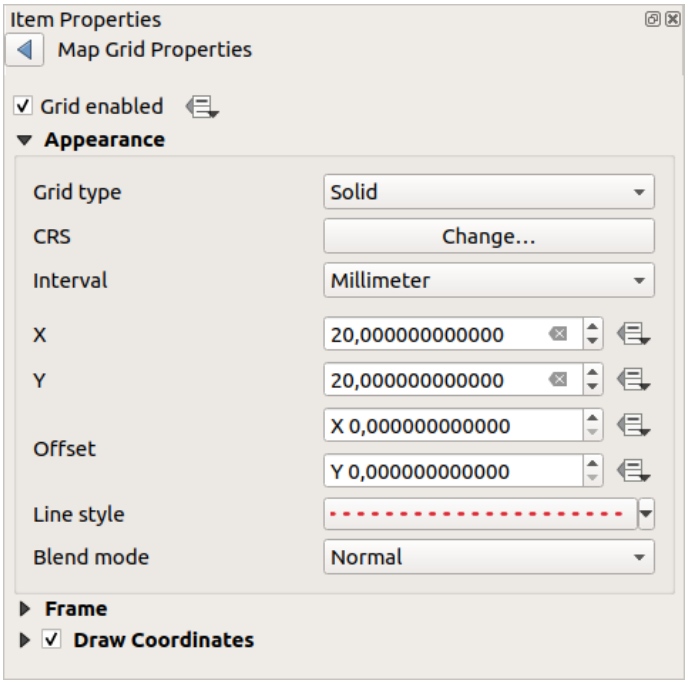


Fig. 18.19: Grid Appearance Dialog

Grid Frame

There are different options to style the frame that holds the map. The following options are available: `No Frame`, `Zebra`, `Zebra (nautical)`, `Interior ticks`, `Exterior ticks`, `Interior and Exterior ticks`, `Line border` and `Line border (nautical)`.

When compatible, it's possible to set the *Frame size*, a *Frame margin*, the *Frame line thickness* with associated color and the *Frame fill colors*.

Using `Latitude/Y only` and `Longitude/X only` values in the divisions section you can prevent a mix of latitude/Y and longitude/X coordinates showing on each side when working with rotated maps or reprojected grids. Also you can choose to set visible or not each side of the grid frame.

When the map item extent is rotated (from the *Main properties* group) or the grid has a different CRS applied, grid lines may not be orthogonal to the map item sides. This can result in bad looking of the grid when styled with interior

and/or exterior ticks. Checking *Follow grid rotation* will align the ticks with grid lines. Moreover, you can adjust some more properties:

- *Ticks alignment*: The interior and/or exterior ticks will be parallel to their corresponding grid line. Their alignment can be:
  - **Orthogonal**: ticks on the same side end at one line, parallel to the side. This can result e.g. in some ticks getting longer when with a low angle to the frame.
  - **Fixed length**: all ticks have the same length, so they may not align
- *Skip below angle*: prevents displaying ticks for grid lines intersecting the frame border below a specified threshold
- *Margin from map corner*: prevents displaying ticks too close to the map corners, because they could overlap and/or be out of bounds.

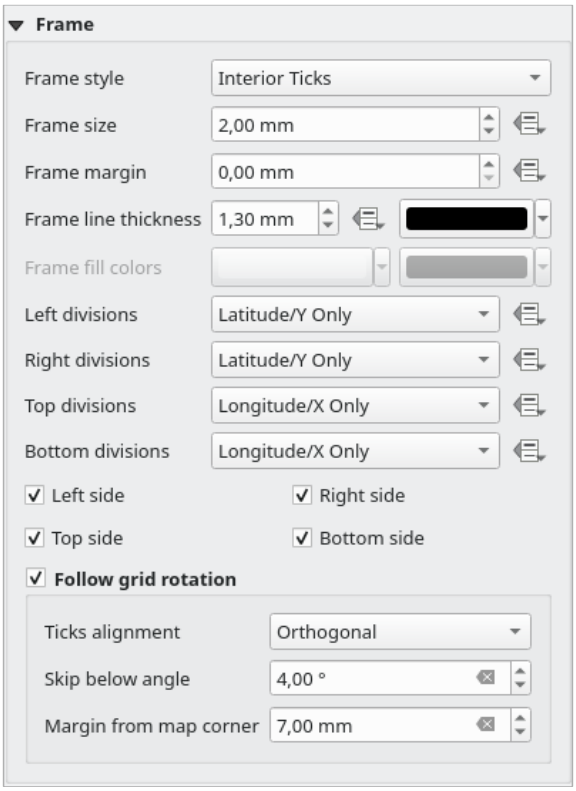


Fig. 18.20: Grid Frame Dialog

Coordinates

The *Draw coordinates* checkbox allows you to add coordinates to the map frame. Displayed values relate to the chosen *grid interval* unit. You can choose the annotation numeric format, the options range from decimal to degrees, minute and seconds, with or without suffix, aligned or not and a custom format using the expression dialog.

For each of the *Left*, *Right*, *Top* and *Bottom* sides of the grid frame, you can indicate:

- whether to render the coordinates: **Show all**, **Show latitude/Y only**, **Show longitude/X only**, **Disabled**. Showing only Latitude/Y or Longitude/X values in the divisions helps prevent a mix of latitude/Y and longitude/X coordinates showing on each side when working with rotated maps or reprojected grids.
- the relative position of the text to the grid frame: **Outside frame** or **Inside frame**
- the placement and orientation of the annotation:

- **Horizontal**
- **Vertical ascending**, **Vertical descending**
- **Boundary direction**
- **Above tick**, **On tick**, **Under tick** when a tick-based frame is used

You can also define the *Font properties* (font, size, color, buffer,...) the *Distance to the map frame* and the *Coordinate precision* (number of decimals) for the drawn annotations.

Follow grid rotation: available when the map extent is rotated or the grid is reprojected, it helps you adjust the annotations placement. Depending on the selected placement mode, the annotations are also rotated:

- *Annotations alignment*: it can be **Orthogonal** or of **Fixed length**
- *Skip below angle*: prevents displaying annotations for grid lines intersecting the frame border below a specified threshold
- *Margin from map corner*: prevents displaying annotations too close to the map corners, because they could overlap and/or be out of bounds.

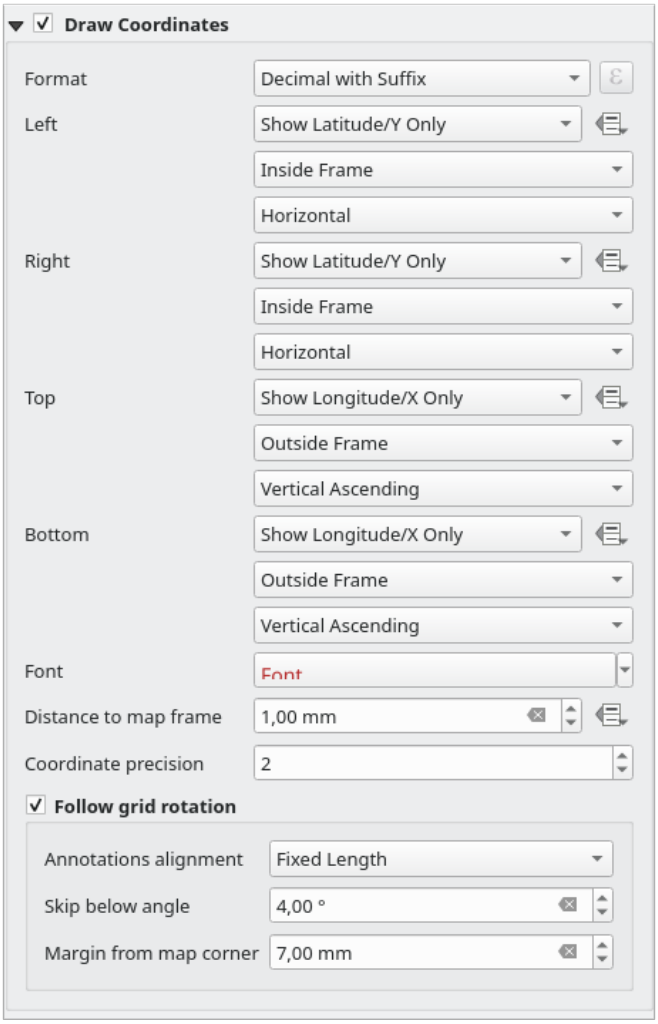


Fig. 18.21: Grid Draw Coordinates dialog

Overviews

Sometimes you may have more than one map in the print layout and would like to locate the study area of one map item on another one. This could be for example to help map readers identify the area in relation with its larger geographic context shown in the second map.

The *Overviews* group of the map panel helps you create the link between two different maps extent and provides the following functionalities:

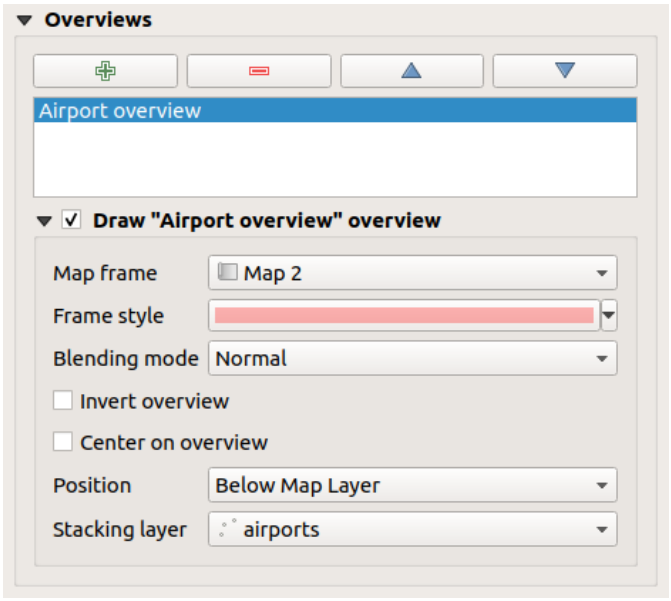


Fig. 18.22: Map Overviews group

To create an overview, select the map item on which you want to show the other map item's extent and expand the *Overviews* option in the *Item Properties* panel. Then press the button to add an overview.

Initially this overview is named 'Overview 1' (see Fig. 18.22). You can:

- Rename it with a double-click
- With the and buttons, add or remove overviews
- With the and buttons, move an overview up and down in the list, placing it above or below other overviews in the map item (when they are at the same *stack position*).

Then select the overview item in the list and check the *Draw "<name_overview>" overview* to enable the drawing of the overview on the selected map frame. You can customize it with:

- The *Map frame* selects the map item whose extents will be shown on the present map item.
- The *Frame Style* uses the *symbol properties* to render the overview frame.
- The *Blending mode* allows you to set different transparency blend modes.
- The *Invert overview* creates a mask around the extents when activated: the referenced map extents are shown clearly, whereas the rest of the map item is blended with the frame fill color (if a fill color is used).
- The *Center on overview* pans the map item content so that the overview frame is displayed at the center of the map. You can only use one overview item to center, when you have several overviews.
- The *Position* controls exactly where in the map item's layer stack the overview will be placed, e.g. allowing an overview extent to be drawn below some feature layers such as roads whilst drawing it above other background layers. Available options are:

- *Below map*
- *Below map layer* and *Above map layer*: place the overview frame below and above the geometries of a layer, respectively. The layer is selected in the *Stacking layer* option.
- *Below map labels*: given that labels are always rendered above all the feature geometries in a map item, places the overview frame above all the geometries and below any label.
- *Above map labels*: places the overview frame above all the geometries and labels in the map item.

18.2.3 The 3D Map Item

The 3D Map item is used to display a *3D map view*. Use the *Add 3D Map* button, and follow *items creation instructions* to add a new 3D Map item that you can later manipulate the same way as demonstrated in *Interacting with layout items*.

By default, a new 3D Map item is empty. You can set the properties of the 3D view and customize it in the *Item Properties* panel. In addition to the *common properties*, this feature has the following functionalities (Fig. 18.23):

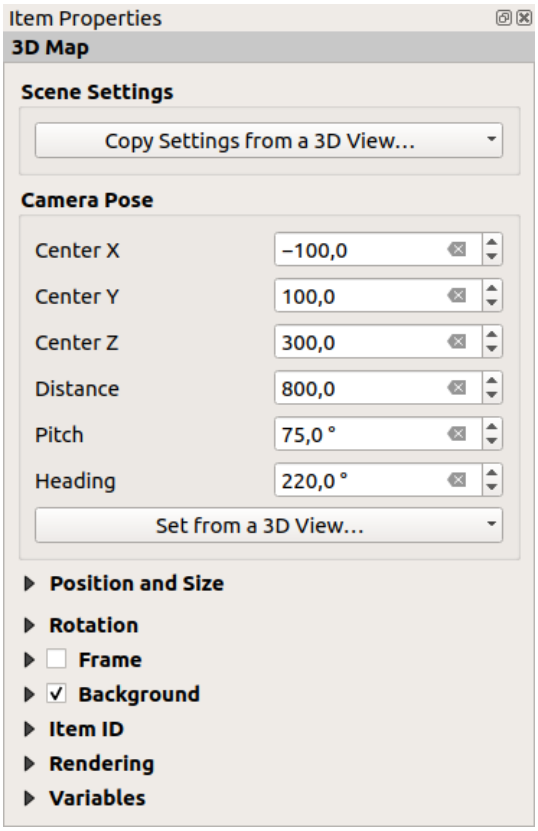


Fig. 18.23: 3D Map Item Properties

Scene settings

Press *Copy Settings from a 3D View...* to choose the 3D map view to display.

The 3D map view is rendered with its current configuration (layers, terrain, lights, camera position and angle...).

Camera pose

- *Center X* sets the X coordinate of the point the camera is pointing at
- *Center Y* sets the Y coordinate of the point the camera is pointing at
- *Center Z* sets the Z coordinate of the point the camera is pointing at
- *Distance* sets the distance from the camera center to the point the camera is pointing at
- *Pitch* sets the rotation of the camera around the X-axis (vertical rotation). Values from 0 to 360 (degrees). 0°: terrain seen straight from above; 90°: horizontal (from the side); 180°: straight from below; 270°: horizontal, upside down; 360°: straight from above.
- *Heading* sets the rotation of the camera around the Y-axis (horizontal rotation - 0 to 360 degrees). 0°/360°: north; 90°: west; 180°: south; 270°: east.

The *Set from a 3D View...* pull-down menu lets you populate the items with the parameters of a 3D View.

18.2.4 The Label Item

The *Label* item is a tool that helps decorate your map with texts that would help to understand it; it can be the title, author, data sources or any other information... You can add a label with the *Add Label* tool following *items creation instructions* and manipulate it the same way as exposed in *Interacting with layout items*.

By default, the label item provides a default text that you can customize using its *Item Properties* panel. Other than the *items common properties*, this feature has the following functionalities (see Fig. 18.24):

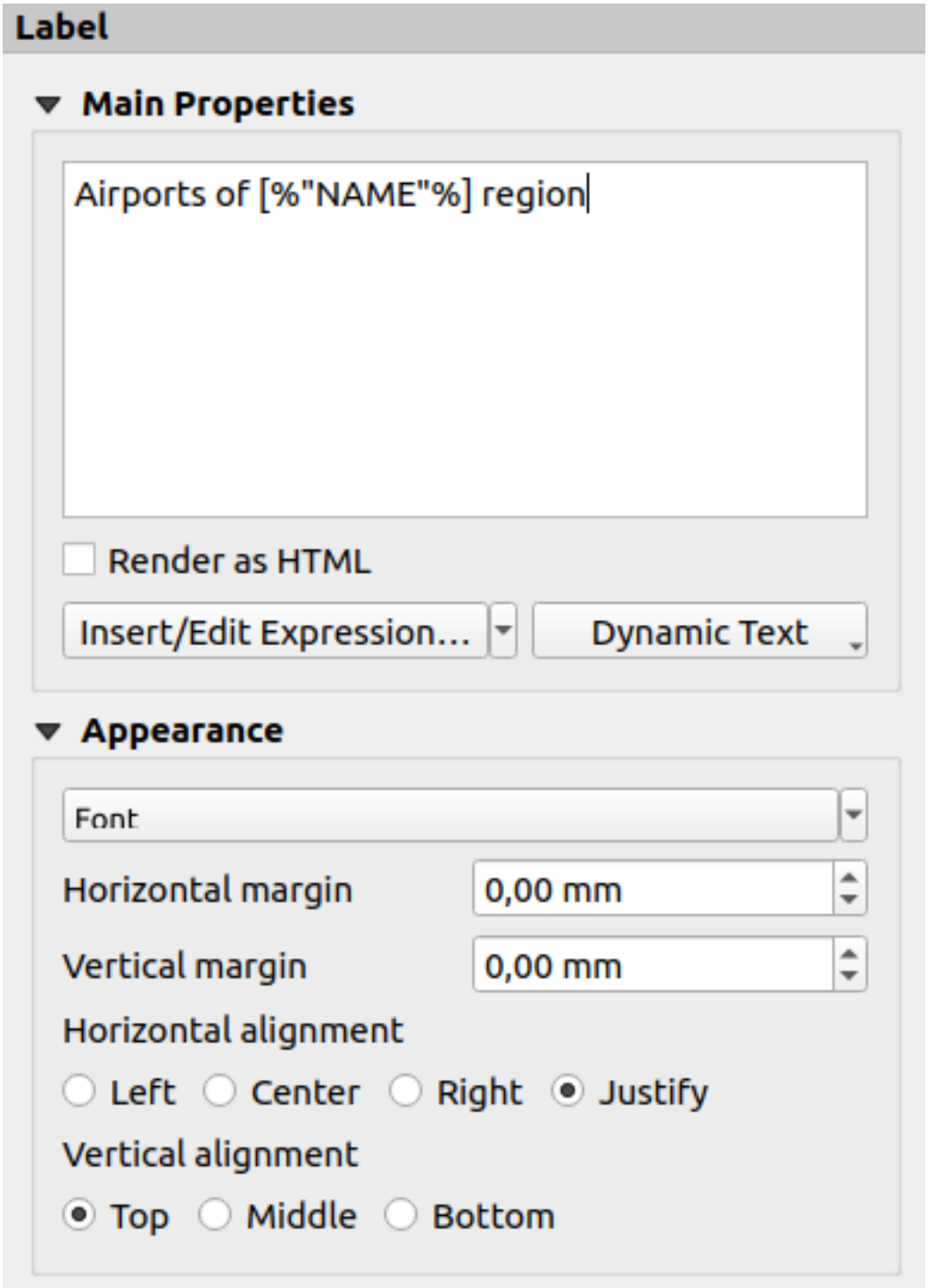


Fig. 18.24: Label Item Properties Panel

主要属性

The *Main properties* group is the place to provide the text of the label. The text can be static, dynamic with *expression* functions and variables, and/or formatted with HTML. Dynamic parts of a label need to be surrounded by `[%` and `%]` in order to be interpreted and evaluated as such.

- To use expressions in labels, you can click on *Insert/Edit Expression...* button, write your formula as usual and when the dialog is applied, QGIS automatically adds the surrounding characters.

  > **Hint**
  >
  > When you click the *Insert/Edit Expression...* button with no selection made in the textbox, the new expression will be appended to the existing text. However, if you want to modify an existing expression, the behavior changes based on your selection: If some text is selected and the selection is within an expression (between '[%' and '%]'), or if no text is selected but the cursor is inside an expression, the whole expression will be selected. In all other cases, the selection will remain as is.

  Because maps are usually filled with some common textual information (date, author, title, page number, ...), QGIS provides a direct access to the corresponding expressions or variables: press the *Dynamic text* button to select and insert them into your label.

  > **Tip**
  >
  > The top menu *Add Item* ► *Add Dynamic Text* ► can be used to create a new label item filled with the selected predefined expression.

  It's possible to turn a dynamic label into static: press the drop-down arrow next to the *Insert/Edit Expression...* button and select *Convert to Static*. Any dynamic parts of the label's contents will be evaluated and replaced with their current values. You can then manually tweak the resulting text when needed.

- Labels can be interpreted as HTML code: check *Render as HTML*. You can now insert HTML tags or styles, URL, a clickable image that links to a web page, or something more complex...

The following code combines HTML rendering with expressions, for an advanced labeling and will output Fig. 18.25:

```
<html>
 <head>
   <style>
      /* Define some custom styles, with attribute-based size */
      name {color:red; font-size: [% ID %]px; font-family: Verdana; text-shadow:␣
↪grey 1px 0 10px;}
      use {color:blue;}
   </style>
 </head>

 <body>
   <!-- Information to display -->
   <u>Feature Information</u>
   <ul style="list-style-type:disc">
     <li>Feature Id: [% ID %]</li>
     <li>Airport: <name>[% NAME %]</name></li>
     <li>Main use: <use>[% USE %]</use></li>
   </ul>
   Last check: [% concat( format_date( "control_date", 'yyyy-MM-dd'), ' by <b><i>',
↪ @user_full_name, '</i></b>' ) %]

   <!-- Insert an image -->
   <p align=center><img src="path/to/logos/qgis-logo-made-with-color.svg" alt=
↪"QGIS icon" style="width:80px;height:50px;"</p>
```

(continues on next page)

(continued from previous page)

```
 </body>
</html>
```

Fig. 18.25: Leveraging a label with HTML styling

Appearance

- Define font and style of the text by clicking on the *Font* button. In the *Label Font* menu you can use some of the options for *Formatting the label text*.
- You can specify different horizontal and vertical margins in mm. This is the margin from the edge of the layout item. The label can be positioned outside the bounds of the label e.g. to align label items with other items. In this case you have to use negative values for the margin.
- Using the text alignment is another way to position your label. It can be:
  - *Left*, *Center*, *Right* or *Justify* for *Horizontal alignment*
  - and *Top*, *Vertical Center*, *Bottom* for *Vertical alignment*.

Tip

Resize the layout label item to fit its contents

By default, the size of a layout label item does not depend on the contained text. It may be too large with useless blank space or too small, hiding part of the text. To automatically resize a label so it fits the text inside it perfectly, double-click one of the square handles on the label. The label will adjust its size to match the text, either expanding or shrinking as needed, while the opposite handle of the label remains fixed in place. This allows you to quickly ensure the label fits the text without manually resizing it. So e.g., double-clicking the left-middle handle causes the left side of the label to move.

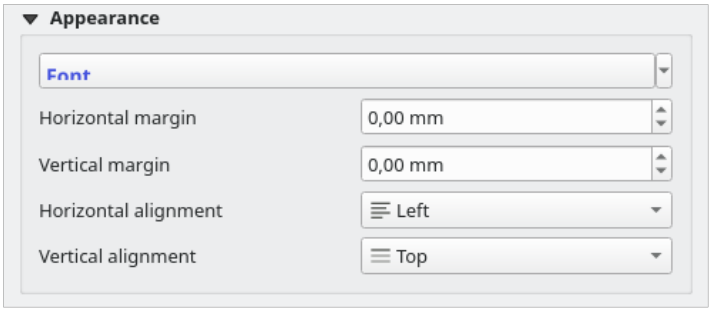


Fig. 18.26: Label Item Appearance Properties

Exploring expressions in a label item

Below some examples of expressions you can use to populate the label item with interesting information - remember that the code, or at least the calculated part, should be surrounded by `[%` and `%]` in the *Main properties* frame:

- Display a title with the current atlas feature value in "field1":

```
'This is the map for ' || "field1"
```

  or, written in the *Main properties* section:

```
This is the map for [% "field1" %]
```

- Add a pagination for processed atlas features (eg, `Page 1/10`):

```
concat( 'Page ', @atlas_featurenumber, '/', @atlas_totalfeatures )
```

- Return the name of the airports of the current atlas region feature, based on their common attributes:

```
aggregate( layer := 'airports',
           aggregate := 'concatenate',
           expression := "NAME",
           filter := fk_regionId = attribute( @atlas_feature, 'ID' ),
           concatenator := ', '
         )
```

  Or, if an *attributes relation* is set:

```
relation_aggregate( relation := 'airports_in_region_relation',
                    aggregate := 'concatenate',
                    expression := "NAME",
                    concatenator := ', '
                  )
```

- Return the name of the airports contained in the current atlas region feature, based on their spatial relationship:

```
aggregate( layer := 'airports',
           aggregate := 'concatenate',
           expression := "NAME",
           filter := contains( geometry( @parent ), $geometry ),
           concatenator := ', '
         )
```

  OR:

```
array_to_string( array:= overlay_contains( layer := 'airports',
                                           expression := "NAME" ),
                 delimiter:= ', '
               )
```

- Return the lower X coordinate of the `Map 1` item's extent:

```
x_min( map_get( item_variables( 'Map 1' ), 'map_extent' ) )
```

- Retrieve the name of the layers in the current layout `Map 1` item, and formats in one name by line:

```
array_to_string(
 array_foreach(
  map_get( item_variables( 'Map 1' ), 'map_layers' ), -- retrieve the layers␣
↪list
  layer_property( @element, 'name' ) -- retrieve each layer name
 ),
```

(continues on next page)

(continued from previous page)

```
 '\n' -- converts the list to string separated by breaklines
)
```

- Display the list of layers with their license strings (usage rights) in a layout `Map 1` item. You need to fill the layers' *Access metadata* properties first.

```
array_to_string( map_credits( 'Map 1', true ) )
```

18.2.5 图例项

图例项是一个方框或表格，用于说明地图中所用符号的含义。图例随后会绑定到一个地图项上。您可以使用 添加图例工具，按照*items creation instructions* 的说明添加图例项，并以*Interacting with layout items* 中所述的方式对其进行操作。

默认情况下，图例项会显示所有可用图层，并可通过其 项属性面板进行细化。除了*items common properties* 中所述的通用属性外，该功能还具有以下特性（见 Fig. 18.27）：

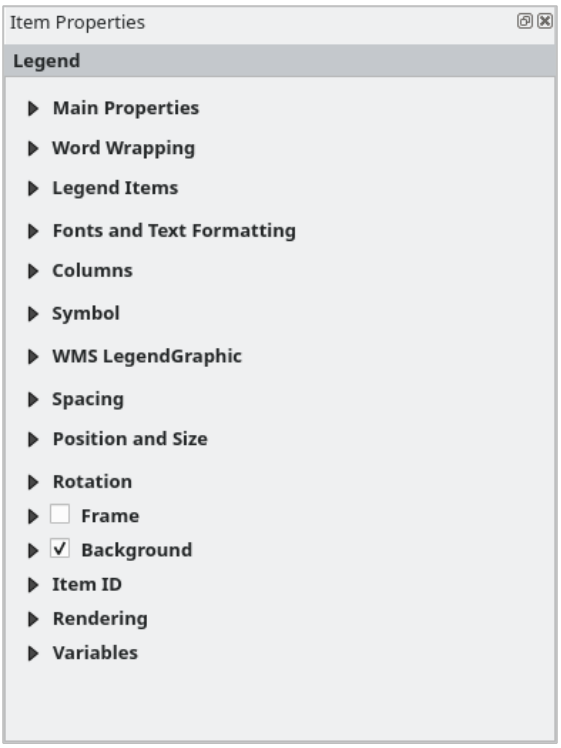


Fig. 18.27: 图例的项属性面板

主要属性

图例 项属性面板的 主要属性组提供了以下功能 (看图:numref:*figure_layout_legend_ppt*):

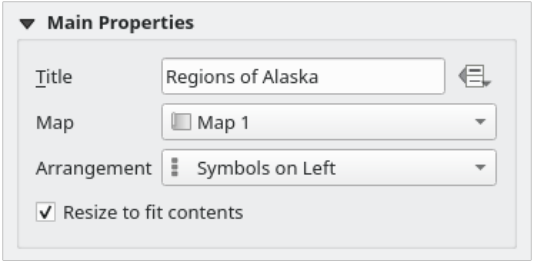


Fig. 18.28: 图例的主要属性组

在主要属性，你可以

- 更改图例的 标题。可以使用*data-defined override* 设置使其动态，例如在生成地图集时很有用；
- 选择哪个 地图做当前图例参考的项目。默认情况下，会选择绘制图例项所在的地图。如果没有，则回退到*reference map*。

> **Note**
>
> 链接的地图项的*Variables* 例如 (@map_id, @map_scale, @map_extent...) 也可以从图例的数据定义属性中访问。

- 设置图例中的符号和文本位置：排列可以是 符号居左或者 符号居右。默认值取决于所使用的语言环境（是否从右向左）。
- 使用 调整至内容大小来控制图例是否应该自动调整大小以适合其内容。如果取消勾选，则图例将永远不会调整大小，而只是坚持用户设置的大小。任何不适合尺寸的内容都会被裁剪掉。

Word wrapping

The *Word wrapping* group of the legend *Item Properties* panel provides the following functionalities:

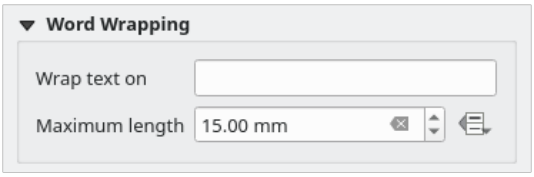


Fig. 18.29: Legend's word wrapping group

- 在给定字符上换行图例的文本：每当该字符出现时，它就被换行符替换；
- Allow legend text to be automatically wrapped after a set line length (in millimeters), preventing very wide auto-generated legends. The *Maximum length* can be set as a static or data-defined value, convenient for dynamic layouts which adjust legend size based on for example page orientation or displayed features, or atlases where you may want to tweak the legend appearance on different pages.

  Examples:

  – Control legend width based on layout page orientation

```
IF( @layout_pageheight >= @layout_pagewidth, 35, 80 )
```

- Set the legend column width when a specific layer is displayed in the linked map

```
CASE WHEN
  array_contains(
    map_get( item_variables('legend_1'), 'map_layer_ids' ),
    'a_specific_layer_id' )
THEN 60
END
```

图例项

图例 项属性面板的 图例项组提供了以下功能 (看图 Fig. 18.30):

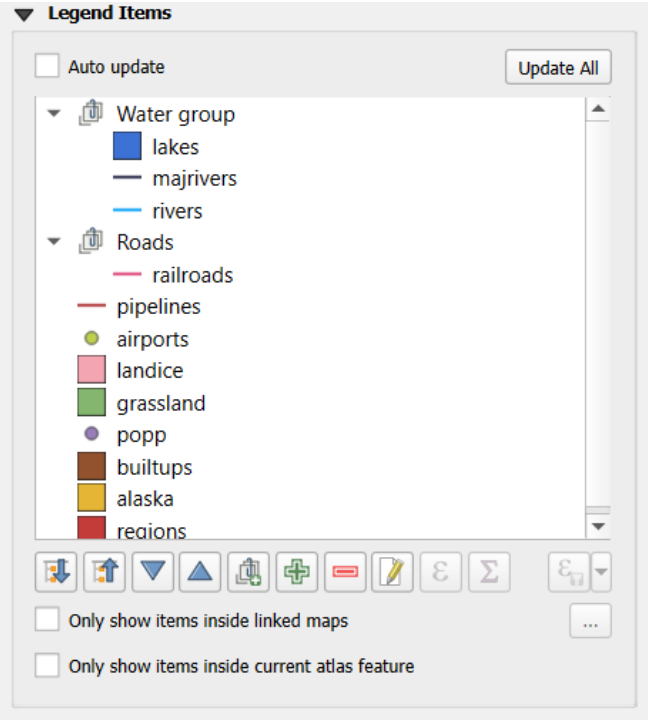


Fig. 18.30: 图例项组

- 如果选中 自动更新，图例将自动更新。当 自动更新被选中时，这将给你更多的控制图例项目。图例项目列表下面的所有图标将被激活。
- 图例项窗口列出了所有图例项目，并允许您更改项目顺序、组层、删除和恢复列表中的项目、编辑图层名称和符号或添加过滤器。
  - 使用 全部展开 和 lcollapseTreel 全部折叠 按钮分别展开或折叠图例树中的所有组和子组。确保 自动更新未选中以使用这些按钮。
  - 可以使用 向上和 向下按钮或使用“拖放”功能更改项目顺序。不能更改 WMS 图例图形的顺序。
  - 使用 按钮添加图例组。
  - 使用 按钮添加层，使用 按钮删除组，层或符号类。

- 按钮用于编辑图层、组名或标题。首先，您需要选择图例项。双击该项还会打开文本框以重命名它。
- 按钮使用表达式来定制所选图层的每个符号标签 (参见数据定义图例标签)
- 按钮为每一类向量层添加一个特征计数。
- 过滤表达式 可以帮助你过滤一个层的哪些图例项将被显示，即使用一个有不同图例项的层（例如，来自基于规则的或分类的符号），你可以指定一个布尔表达式从图例树中删除，样式没有满足条件的特征。请注意，这些特性仍然保留并显示在布局地图项中。

虽然图例项的默认行为是模仿 图层面板树，显示相同的组，层和符号类，右键单击任何项目为您提供隐藏层的名称或将其提升为组或子组的选项。如果你对一个图层做了一些改变，你可以通过在图例条目的上下文菜单中选择 重置为默认来恢复它们。

在 QGIS 主窗口更改符号后，您可以单击 全部更新来适应打印布局的图例元素的更改。

- 使用 仅显示当前地图内容关联的图例项，只有在链接地图中可见的图例项目将在图例中列出。如果你有多个地图，你可以点击 ... 并从布局中选择其他地图。当 自动更新激活时，此工具仍然可用。
- 在生成具有多边形特征的地图集时，您可以过滤掉位于当前地图集要素之外的图例项目。要做到这一点，勾选 只显示当前地图集要素内部的项选项。

数据定义图例标签

允许您为给定图层的每个符号标签添加*expressions* 新增变量 (`@symbol_label` , `@symbol_id` 和 “@symbol_count“) 可以帮助您与图例条目进行交互。

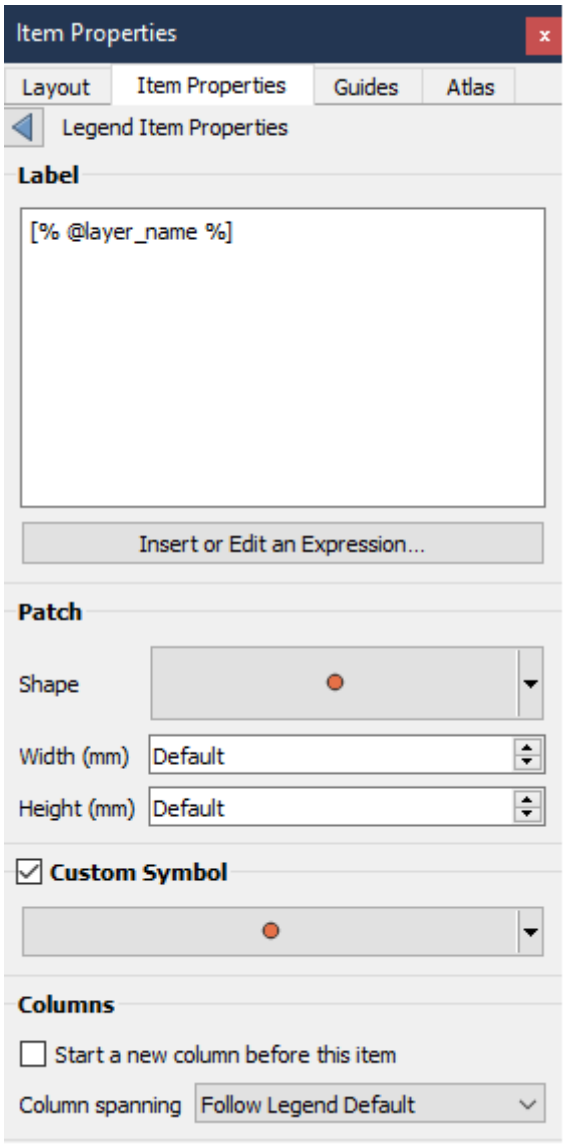
例如，对于一个按 `type` 字段进行分类的 `regions` 图层，您可以在图例中每个类别后附加其要素数量和总面积，例如：`Borough (3) - 850ha`。

1. 在图例树中选择图层条目
2. 按 按钮，打开 表达式构建器对话框
3. 输入如下表达式 (假设没有编辑过符号标签):

```
format( '%1 (%2) - %3ha',
        @symbol_label,
        @symbol_count,
        round( aggregate(@layer, 'sum', $area, filter:= "type"=@symbol_label)/
↪10000 )
      )
```

4. 按 *OK* 按钮

自定义图例项



图例项目也可以在 图例项属性 '中单独定制。但是这些定制只能在 |checkbox| :guilabel:'自动更新禁用时使用.

双击一个项或按 编辑所选项的属性 允许进一步定制。

标注

对于所有项目类型，它允许通过使用 插入或编辑一个表达式... 输入或插入表达式来修改标签文本。表达式也可以通过使用 [% expression %] 表示法直接添加到条目标签的任何位置。

列

图例项属性还允许您通过强制列分割发生在特定项或层的所有符号之后来控制列分割行为。在这个控件中，还可以允许或阻止层及其子层的自动分割。

图斑

对于带有符号的项，图例项属性允许您指定符号占用的最大高度和宽度。

对于矢量符号，您可以为符号指定自定义形状。形状通常由表达式定义，以表示简单平面中的几何形状，但这些符号也可以保存在样式管理器中，以后再导入。每个几何图形类型的默认符号也可以通过样式管

理器来控制。

自定义符号

也可以为矢量符号指定自定义符号。这在需要微调某个特定符号的渲染效果、在图例中对其进行增强，或使用一个与其真实符号预览无关的符号时非常有用。该自定义符号将覆盖图例中的符号，但仍会考虑所指定的符号 图斑设置。

字体和文本格式

图例 项目属性面板的 字体与文本格式组提供以下功能：

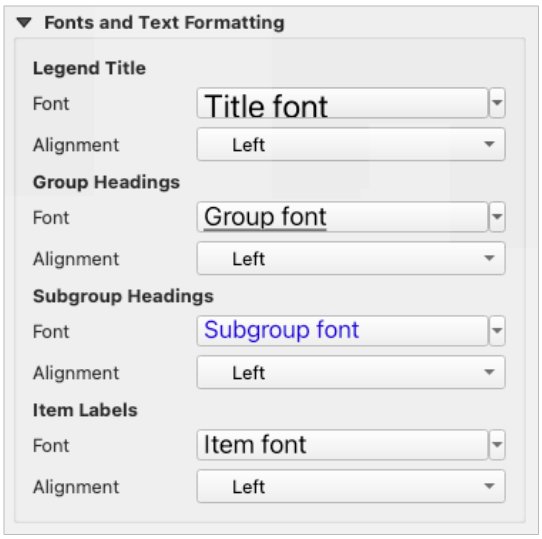


Fig. 18.31: 图例字体属性

- 您可以使用*font selector* 控件，分别更改图例项中的图例标题、组、子组和项（要素）的字体，并享有*text formatting* 的全部功能（包括字间距控制、混合 HTML 格式、颜色、混合模式、背景、文字缓冲、阴影等）。
- 对于这些内容，你可以设置文本:guilabel: 对齐：它可以是 左对齐（默认为从左到右的区域），水平居中或 右对齐（默认为从右到左的区域）。

列

在“图例项属性”面板的 列组下，图例项可以排列在几个列上：

- 设置 计数 1,00 字段中的列数。这个值可以是动态的，例如：图集特征、图例内容、帧大小……
- 列宽相等设置图例列的调整方式。
- 分割图层选项允许在列之间划分分类或分级图层图例。

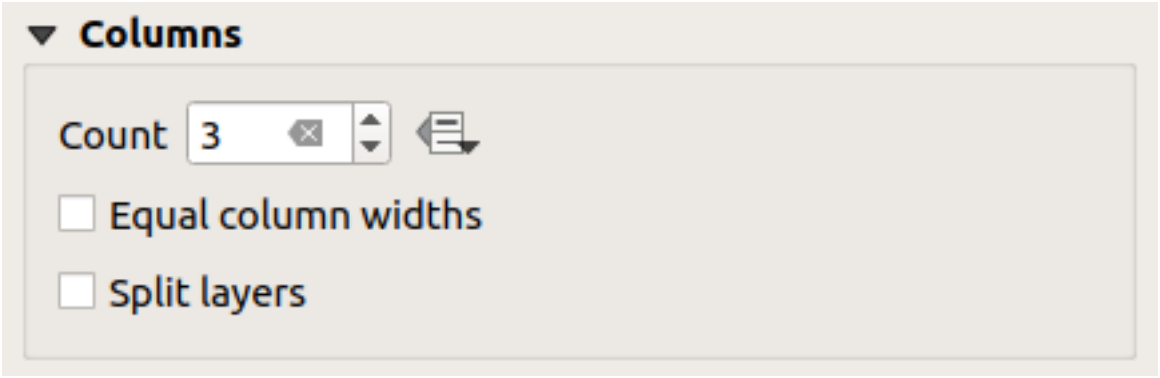


Fig. 18.32: 图例列设置

符号

图例 项属性面板的 符号组配置图例标签旁边显示的符号的大小。您可以:

- 设置 符号宽度和 符号高度
- 设置标记的' 最小符号大小和 最大符号大小: 0.00mm 表示没有设置值。
- 为栅格符号绘制描边: 这将为栅格图层的波段符号添加一个轮廓; 你可以设置 描边颜色和 厚度。

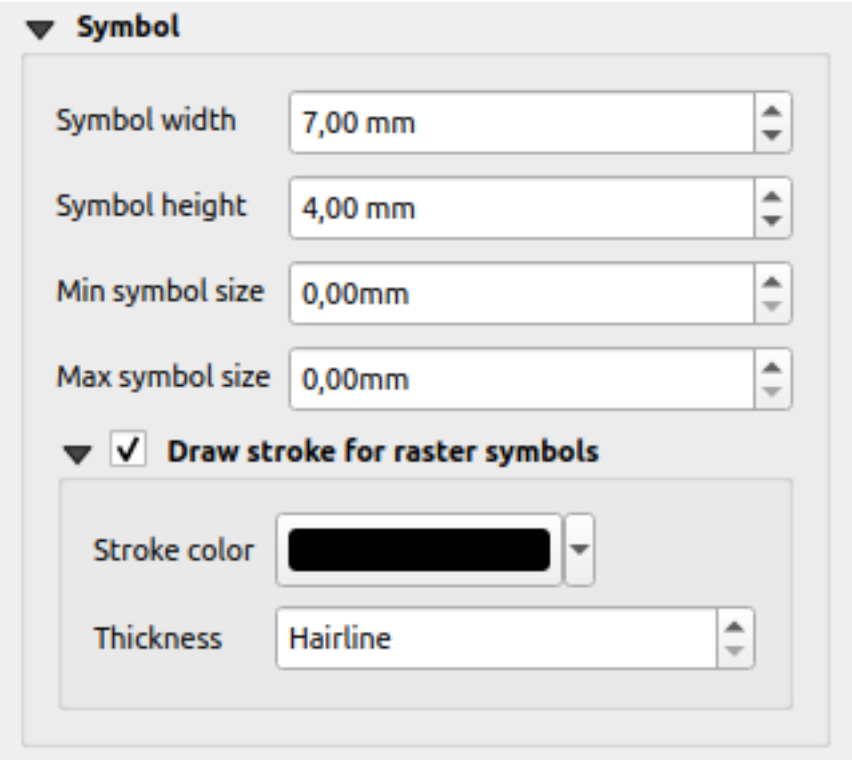


Fig. 18.33: 图例的符号配置

WMS 图例图形

图例的项属性面板中的 *WMS* 图例图形‘(*WMS LegendGraphic*) 部分提供以下功能（参见:numref:‘figure_layout_legend_wms’）：

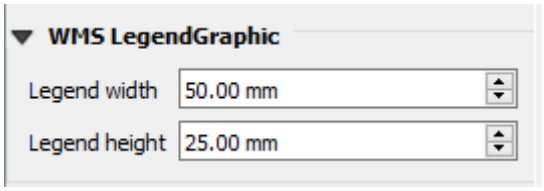
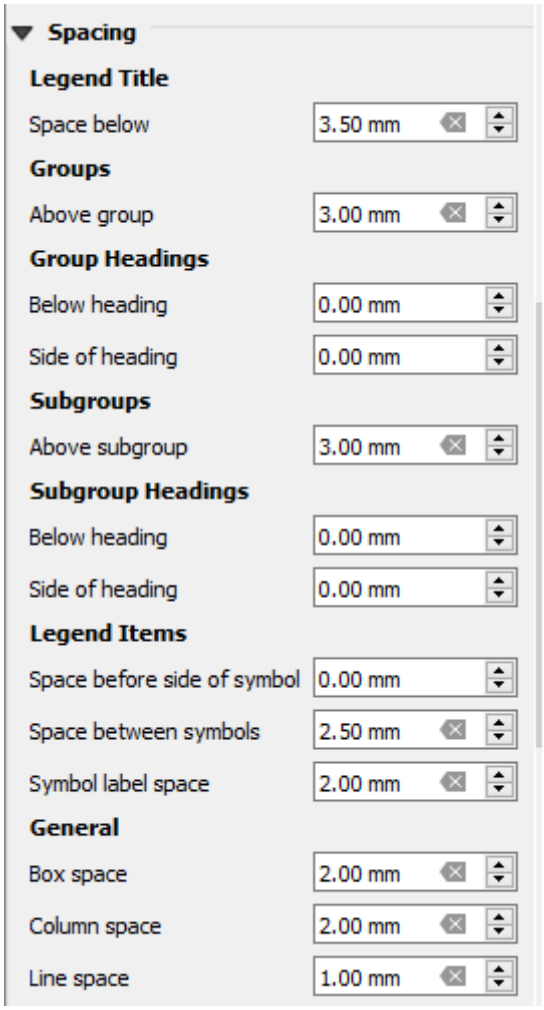


Fig. 18.34: WMS 图例图形

当您添加了 WMS 图层并插入图例项时，系统会向 WMS 服务器发送请求以获取 WMS 图例。该图例仅在 WMS 服务器支持 GetLegendGraphic 功能时才会显示，且图例内容将以栅格图像形式提供。

WMS 图例图形用于调整 WMS 图例栅格图像的 图例宽度和 图例高度。

间距



The *Spacing* section allows you to customize the spacing within the legend. Spacing can greatly help denote the groupement of items in the legend and their relation.

标题、组、子组、符号、标签、框、列和行周围和之前的 间距可以通过这个对话框自定义。

18.2.6 The Scale Bar Item

Scale bars provide a visual indication of the size of features, and distance between features, on the map item. A scale bar item requires a map item. Use the *Add Scale Bar* tool following *items creation instructions* to add a new scale bar item that you can later manipulate the same way as exposed in *Interacting with layout items*.

By default, a new scale bar item shows the scale of the map item over which it is drawn. If there is no map item below, the *reference map* is used. You can customize it in the *Item Properties* panel. Other than the *items common properties*, this feature has the following functionalities (see Fig. 18.35):

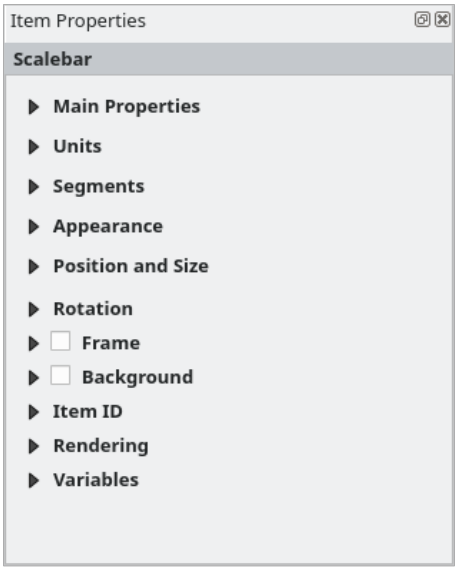


Fig. 18.35: Scale Bar Item Properties Panel

主要属性

The *Main properties* group of the scale bar *Item Properties* panel provides the following functionalities (see Fig. 18.36):

Fig. 18.36: Scale Bar Main properties group

1. First, choose the map the scale bar will be attached to
2. Then, choose the style of the scale bar. Available styles are:
   - **Single box** and **Double box** styles, which contain one or two lines of boxes alternating colors;
   - **Middle**, **Up** or **Down** line ticks;
   - **Stepped line** style that draws a stepped line representation of a scalebar
   - **Hollow** style that draws a single box with alternating color for the segments, with horizontal lines through alternating segments
   - **Numeric**, where the scale ratio is printed (e.g., `1:50000`).

3. Set properties as appropriate

Units

The *Units* group of the scale bar *Item Properties* panel provides the functionalities to set the units of display and some text formatting (see Fig. 18.37):

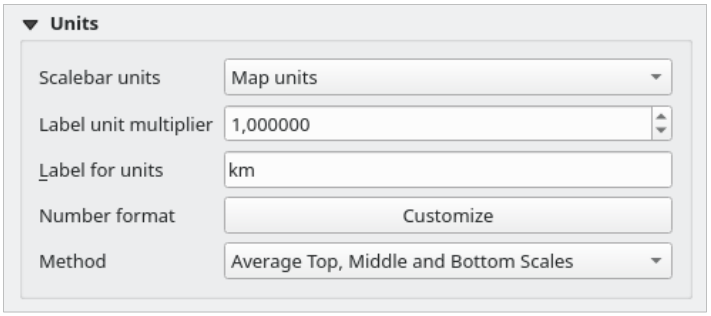


Fig. 18.37: Scale Bar Units group

- Select the units you want to use with *Scalebar units*. There are many possible choices: **Map Units** (the default one), **Meters**, **Feet**, **Miles** or **Nautical Miles**... and some derivatives. Units conversion is handled automatically.
- The *Label unit multiplier* specifies how many scale bar units per labeled unit. Eg, if your scale bar units are set to ”meters”, a multiplier of 1000 will result in the scale bar labels in ”kilometers”.
- The *Label for units* field defines the text used to describe the units of the scale bar, eg `m` or `km`. This should be matched to reflect the multiplier above.
- Press *Customize* next to *Number format* to have control over all the formatting properties for the numbers in the scale bar, including thousand separators, decimal places, scientific notation, etc. (see *Number Formatting* for more details). Very useful in the case of making maps for audiences outside of the current QGIS locale, or when you would like to vary the style from the locale defaults (e.g. adding thousands separators when the locale default is to hide them).
- Select the *Method* for scale bar calculation: Depending on the map CRS and extent, the distance reported in the scale bar may not be the same measured over the map. This option, which defaults to the *project's scale calculation method*, helps you specify the one you would like to use for this particular scale bar.

Segments

The *Segments* group of the scale bar *Item Properties* panel provides the functionalities to configure the number and size of segments and subdivisions (see Fig. 18.38):

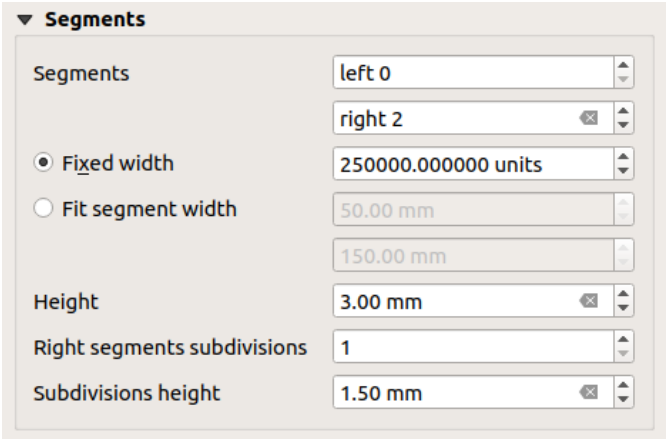


Fig. 18.38: Scale Bar Segments group

- You can define the number of *Segments* that will be drawn at the left and right sides of the `0` of the scale bar:
  - number of subdivisions of a unique segment on the *Left* side
  - number of segments on the *Right* side
- You can set the width of a segment or a range for the total length of the scale bar:
  - Set how long a segment will be in your scale bar units (*Fixed width*)
  - Or limit the total scale bar size in `mm` with *Fit segment width* option. In the latter case, each time the map scale changes, the scale bar is resized (and its label updated) to fit between the upper and lower range set.
- *Height* is used to define the height of the bar.
- *Right segment subdivisions* is used to define the number of sections the right-side segments of the scale bar can have (for *Line Ticks Down*, *Line Ticks Middle* and *Line Ticks Up* scale bar styles) .
- *Subdivision height* is used to define the height of the subdivision segment.

Appearance

The *Appearance* group of the scale bar *Item Properties* panel provides the following functionalities:

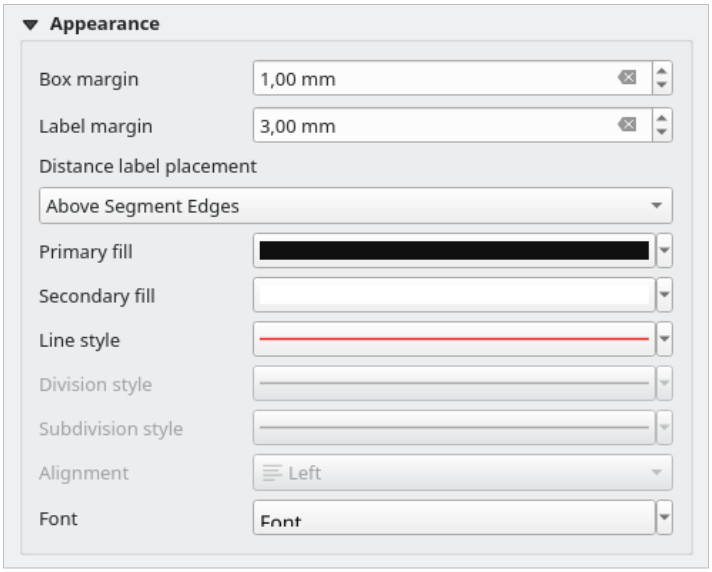


Fig. 18.39: Scale Bar Appearance group

You can define how the scale bar will be displayed in its frame.

- *Box margin* : space between text and frame borders
- *Label margin* : space between text and scale bar drawing
- *Distance label placement* defines the position of the text relative to the scale bar segments: *Above Segment Edges*, *Above Segment Centers*, *Below Segment Edges* or *Below Segment Centers*
- *Primary fill* and *Secondary fill* of the scale bar drawing using *fill symbols properties* (color, opacity, patterns, effects...) --- for *Single Box*, *Double Box* and *Hollow* styles
- *Line style* of the scale bar drawing using *line symbols properties* (color, stroke, join, cap style, patterns, effects...) --- for all but *Numeric* style
- *Division style* and *Subdivision style* respectively for division and subdivision segments in *Line Ticks Up*, *Line Ticks Middle* and *Line Ticks Down* scale bar styles using *line symbols properties* (color, stroke, join, cap style, patterns, effects...)
- *Alignment* puts text on the left, center or right side of the frame (only for *Numeric* scale bar style)
- *Font* to set the *properties* (size, font, color, letter spacing, shadow, background...) of the scale bar label.

Since most of the display properties of the scale bar rely on symbols whose properties can be data-defined, it's possible to render data-defined scale bars.

Example: The following code applied to the bold property of the scale labels will display numbers in bold when they are a multiple of 500:

```
-- returns True (or 1) if the value displayed on the bar
-- is a multiple of 500

@scale_value % 500 = 0
```

18.2.7 The Table Items

You can use table items to decorate and explain your map:

- *Attribute table*: automatically extracts a subset of the attributes of a layer, based on predefined rules
- *Fixed table*: allows for creation of tables with contents manually entered (i.e. spreadsheet style), so that you can create completely custom tables.

The attribute table item

Any layer in the project can have its attributes shown in the print layout. Use the *Add Attribute Table* tool following *items creation instructions* to add a new table item that you can later manipulate the same way as exposed in *Interacting with layout items*.

By default, a new attribute table item loads first rows of the first (alphabetically sorted) layer, with all the fields. You can however customize the table thanks to its *Item Properties* panel. Other than the *items common properties*, this feature has the following functionalities (see Fig. 18.40):

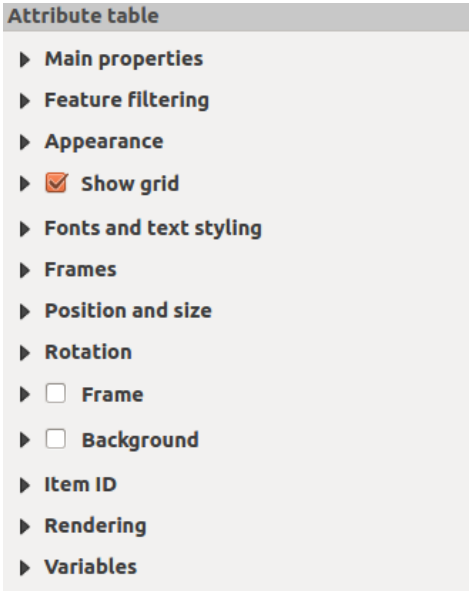


Fig. 18.40: Attribute table Item Properties Panel

主要属性

The *Main properties* group of the attribute table provides the following functionalities (see Fig. 18.41):

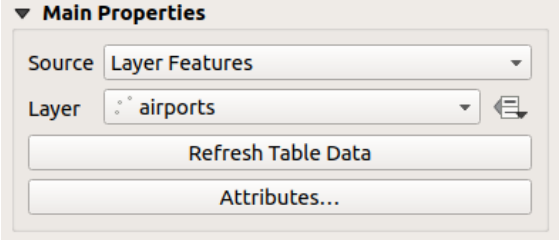


Fig. 18.41: Attribute table Main properties Group

- For *Source* you can by default only select **Layer features** allowing you to select a *Layer* from the vector layers loaded in the project.

The Data-defined override button near the layer list allows you to dynamically change the layer which is used to populate the table, e.g. you could fill the attribute table with different layer attributes per atlas page. Note that the table structure used (Fig. 18.44) is the one of the layer shown in the *Layer* drop-down list and it is left intact, meaning that setting a data defined table to a layer with different field(s) will result in empty column(s) in the table.

In case you activate the *Generate an atlas* option in the *Atlas* panel (see *Generate an Atlas*), there are two additional *Source* possible:

- **Current atlas feature** (see Fig. 18.42): you won't see any option to choose the layer, and the table item will only show a row with the attributes from the current feature of the atlas coverage layer.
- and **Relation children** (see Fig. 18.43): an option with the relation names will show up. This feature can only be used if you have defined a *relation* using your atlas coverage layer as parent, and the table will show the children rows of the atlas coverage layer's current feature.

- The button *Refresh Table Data* can be used to refresh the table when the actual contents of the table has changed.

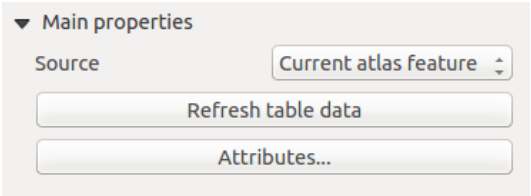


Fig. 18.42: Attribute table Main properties for 'Current atlas feature'

Fig. 18.43: Attribute table Main properties for 'Relation children'

- The button *Attributes...* starts the *Select Attributes* dialog, (see Fig. 18.44) that can be used to change the visible contents of the table. The upper part of the window shows the list of the attributes to display and the lower part helps you sort the data.

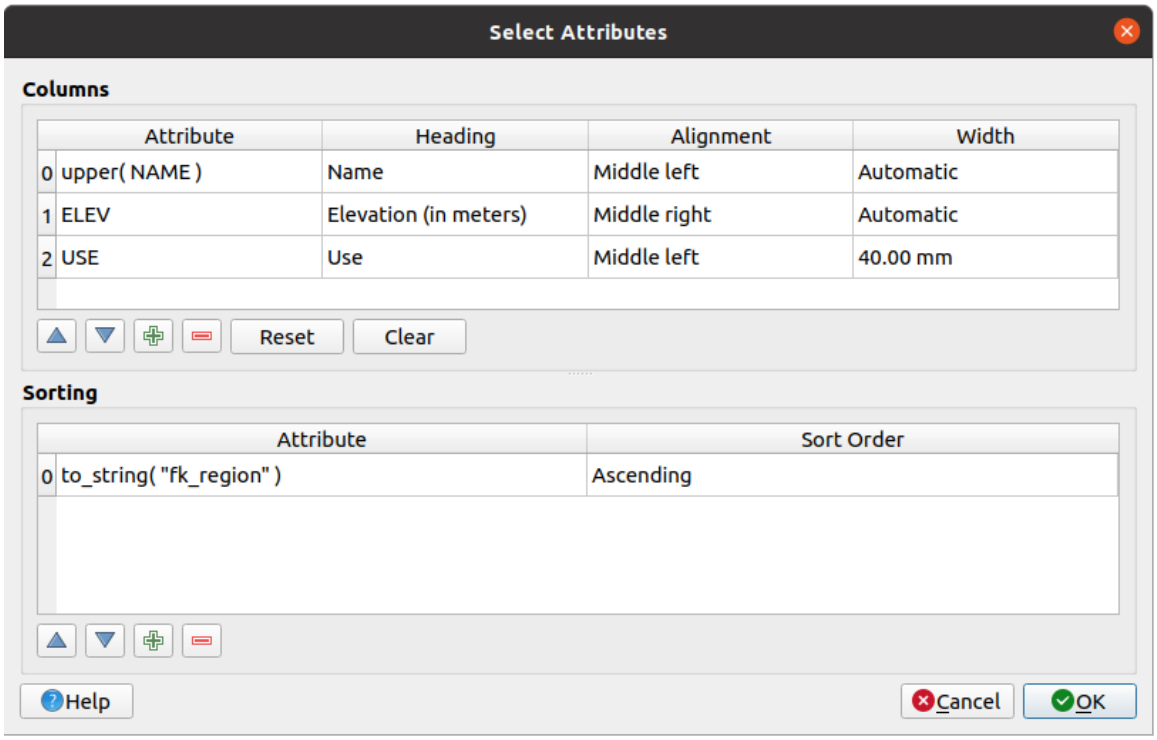


Fig. 18.44: Attribute table Select attributes Dialog

In the *Columns* section you can:

- Move attributes up or down the list by selecting the rows and then using the and buttons to shift the rows. Multiple rows can be selected and moved at any one time.
- Add an attribute with the button. This will add an empty row at the bottom of the table where you can select a field to be the attribute value or create an attribute via a regular expression.
- Remove an attribute with the button. Multiple rows can be selected and removed at any one time.
- Reset the attribute table back to its default state with the *Reset* button.
- Clear the table using the *Clear* button. This is useful when you have a large table but only want to show a small number of attributes. Instead of manually removing each row, it may be quicker to clear the table and add the rows needed.
- Cell headings can be altered by adding the custom text in the *Heading* column.
- Cell alignment can be managed with the *Alignment* column which will dictate the texts position within the table cell.
- Cell width can be manually managed by adding custom values to the *width* column.

In the *Sorting* section you can:

- Add an attribute to sort the table with: press the button and a new empty row is added. Insert a field or an expression in the *Attribute* column and set the *Sort order* to **Ascending** or **Descending**.
- Select a row in the list and use the and buttons to change the sort priority on attribute level. Selecting a cell in the *Sort Order* column helps you change the sorting order of the attribute field.
- Use the button to remove an attribute from the sorting list.

Feature filtering

The *Feature filtering* group of the attribute table provides the following functionalities (see Fig. 18.45):

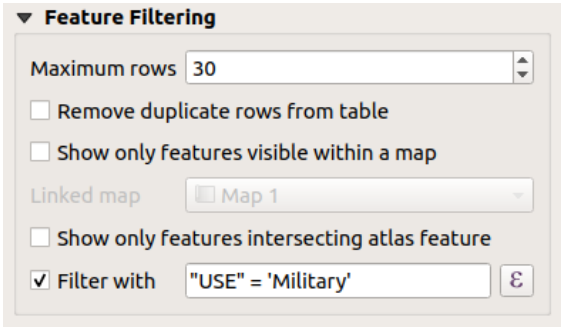


Fig. 18.45: Attribute table Feature filtering Group

您可以：

- Define the *Maximum rows* to be displayed.
- Activate *Remove duplicate rows from table* to show unique records only.
- Activate *Show only visible features within a map* and select the corresponding *Linked map* whose visible features attributes will be displayed.
- Activate *Show only features intersecting Atlas feature* is only available when *Generate an atlas* is activated. When activated it will show a table with only the features which intersect the current atlas feature.
- Activate *Filter with* and provide a filter by typing in the input line or insert a regular expression using the given expression button. A few examples of filtering statements you can use when you have loaded the airports layer from the Sample dataset:
  - `ELEV > 500`
  - `NAME = 'ANIAK'`
  - `NAME NOT LIKE 'AN%'`
  - `regexp_match( attribute( $currentfeature, 'USE' ) , '[i]')`

  The last regular expression will include only the airports that have a letter 'i' in the attribute field 'USE'.

Appearance

The *Appearance* group of the attribute table provides the following functionalities (see Fig. 18.46):

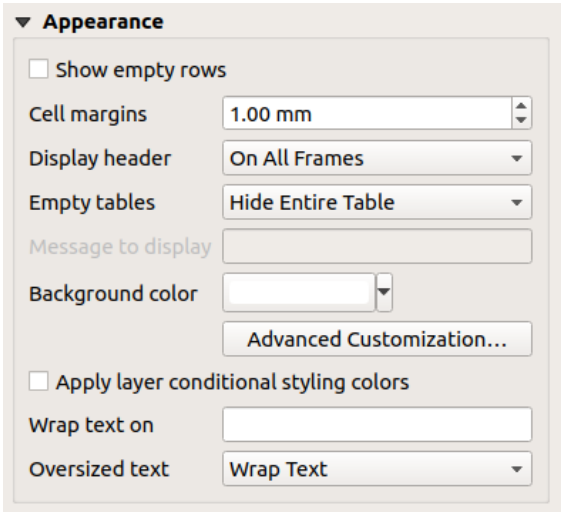


Fig. 18.46: Attribute table appearance Group

- Click *Show empty rows* to fill the attribute table with empty cells. This option can also be used to provide additional empty cells when you have a result to show!
- With *Cell margins* you can define the margin around text in each cell of the table.
- With *Display header* you can select from a list one of 'On first frame', 'On all frames' default option, or 'No header'.
- The option *Empty table* controls what will be displayed when the result selection is empty.
  - **Draw headers only**, will only draw the header except if you have chosen 'No header' for *Display header*.
  - **Hide entire table**, will only draw the background of the table. You can activate *Don't draw background if frame is empty* in *Frames* to completely hide the table.
  - **Show set message**, will draw the header and adds a cell spanning all columns and display a message like 'No result' that can be provided in the option *Message to display*
- The option *Message to display* is only activated when you have selected **Show set message** for *Empty table*. The message provided will be shown in the table in the first row, when the result is an empty table.
- With *Background color* you can set the background color of the table using the *color selector* widget. The *Advanced customization* option helps you define different background colors for each cell (see Fig. 18.47)

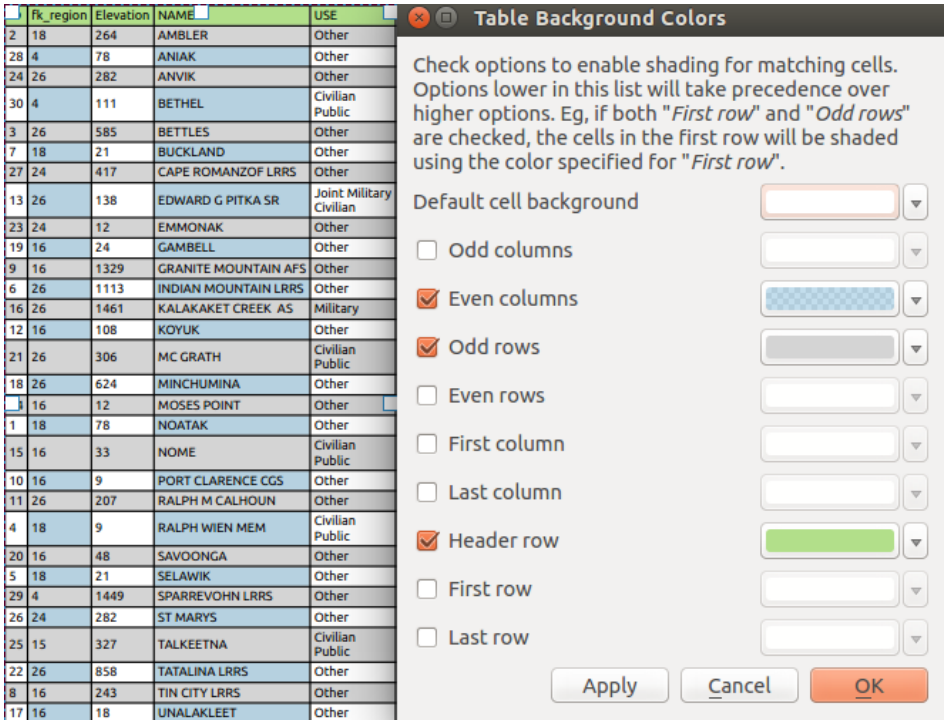


Fig. 18.47: Attribute table Advanced Background Dialog

- *Apply layer conditional styling colors*: the *conditional table formatting* present in the layer is applied inside the layout attribute table (background color, font family and properties such as bold, italic, strikeout, underline, color, ...). Conditional formatting rules take precedence over other layout table formatting settings, e.g. they will override other cell background color settings such as alternating row colors.
- With the *Wrap text on* option, you can define a character on which the cell content will be wraped each time it is met
- With *Oversized text* you define the behavior when the width set for a column is smaller than its content's length. It can be **Wrap text** or **Truncate text**.

> **Note**
>
> More properties of the attribute table item are described in the *Tables common functionalities* section.

The fixed table item

Additional information about the map can be inserted manually into a table by choosing *Add Fixed Table* and by following *items creation instructions* to add a new table item that you can later manipulate the same way as exposed in *Interacting with layout items*.

By default, an empty table with two minimized columns and rows appears in the map layout. You have to customize the table in the *Item Properties* panel. Other than the *items common properties*, this feature has the following functionalities:

主要属性

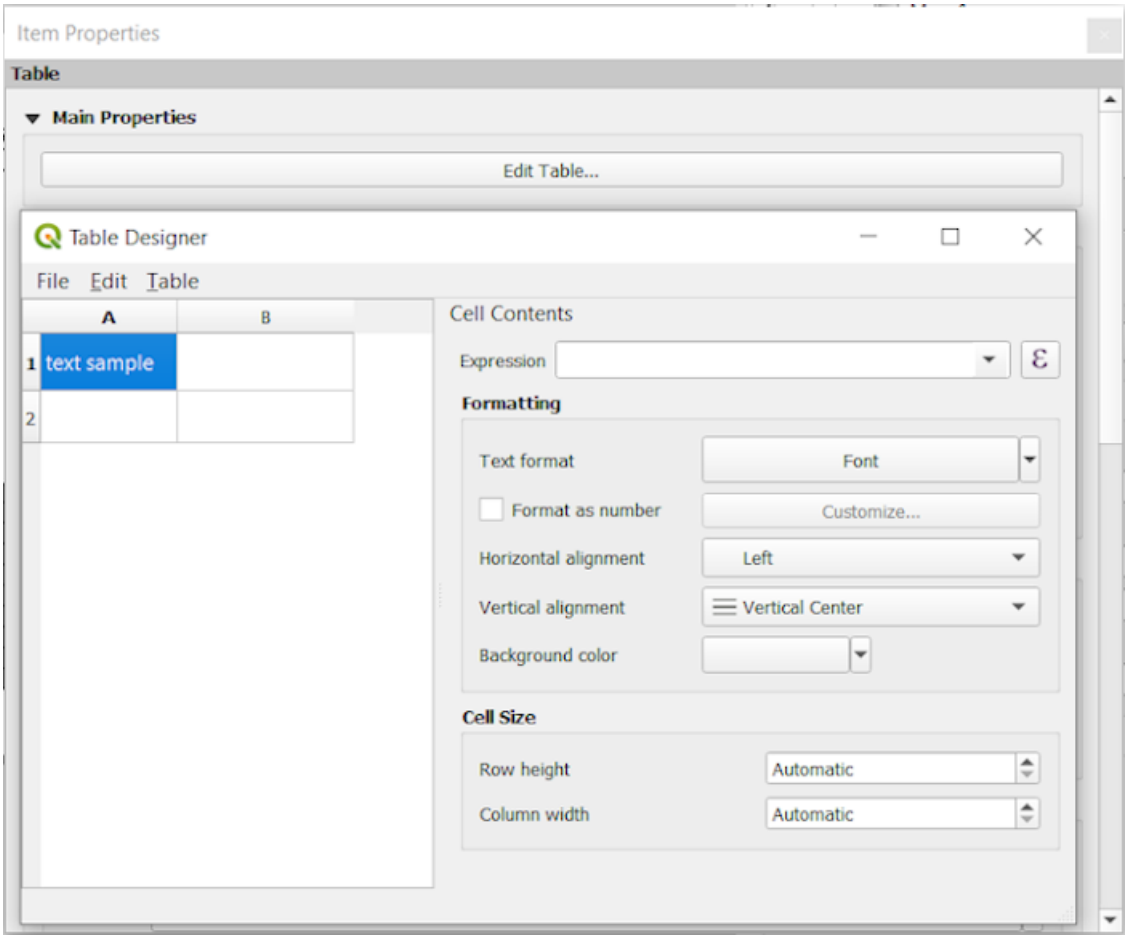


Fig. 18.48: Fixed table Item Properties Panel with Table designer

In *Main properties*, press *Edit table...* to open the *Table designer* dialog where you can build your table (double-clicking the table item will also open the editor dialog):

- On the left hand of the dialog, a table frame displays the active table item with its content and settings. Click or double-click on a cell and write a text to modify its contents. Press `Ctrl+Enter` to add linebreak (for a multiline text). The logic for selecting multiple cells in the table is the same as in the vector layer *attribute table*. Grab and move a row or column separator to resize the row above or the column before. A double-click on the separator will resize the row above or the column before to fit their contents. A right-click over column or row headers shows a contextual menu to insert columns or rows around the selection or to delete them.
- Through the menus on top of the dialog, it is possible to:
  - *File* ► *Import Content From Clipboard*: it overrides contents of the table.
  - *File* ► *Close editor*: it closes the *Table Designer* dialog.
  - From the *Edit* menu, you can work with selection functionalities for rows and columns:
    - *Select all* cells in the table
    - *Select columns* or *Select Rows*: when cells are selected in the table, you can extend the selection respectively to their column(s) or row(s)
    - *Clear Cells*: deletes the content of the selected cells
  - The *Table* menu is the place where you design the structure of the table. You can:

    - *Insert rows* ► above or below the selection
    - *Insert columns* ► before or after the selection
    - *Delete Rows* or *Delete Columns* of selected cells
    - *Merge selected cells*: select multiple cells in a rectangular shape and you can merge them and concatenate their contents into the top left cell (whose styling is also applied to the merged cell, unless there is a last row/column cell in the selection).
    - *Split selected cells*: select a merged cell and you can split it back to its individual cells. The current text is kept in the top left cell, and the other cells are filled with their contents before they get merged. They are also reapplied their original styling.
    - *Include Header Row* whose styling can only be controlled from the *Fonts and text styling* widget
- Select the cells and, on the right of the table frame, you can provide the *Cell Contents*:
  - Either manually enter the contents of each cell, or use an *Expression* to automatically populate it.
  - Under the *Formatting* group, set specific options that apply to the selection and take precedence over the *global table content styling*:
    - by choosing the *Text format*
    - by *Format as number* (several formats are available)
    - by defining the *Horizontal alignment* and the *Vertical alignment*
    - by choosing a *Background color*
  - Define the *Cell Size* with *Row height* and *Column width*.

Appearance

The *Appearance* group of the fixed table provides the following functionalities:

- Click *Show empty rows* to fill the attribute table with empty cells.
- With *Cell margins* you can define the margin around text in each cell of the table.
- With *Display header* you can select from a list one of 'On first frame', 'On all frames' default option, or 'No header'.
- With *Background color* you can set the background color of the table using the *color selector* widget. The *Advanced customization* option helps you define different background colors for each cell.
- With *Oversized text* you define the behavior when the width set for a column is smaller than its content's length. It can be **Wrap text** or **Truncate text**.

> **Note**
>
> More properties of the fixed table item are described in the *Tables common functionalities* section.

Tables common functionalities

Show grid

The *Show grid* group of the table items provides the following functionalities (see Fig. 18.49):

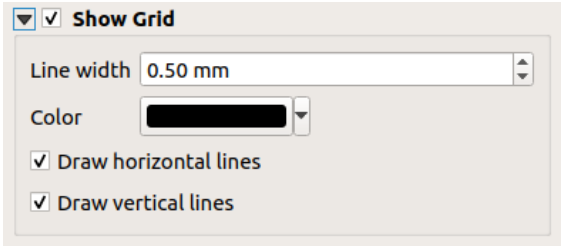


Fig. 18.49: Attribute table Show grid Group

- Activate *Show grid* when you want to display the grid, the outlines of the table cells. You can also select to either *Draw horizontal lines* or *Draw vertical lines* or both.
- With *Line width* you can set the thickness of the lines used in the grid.
- The *Color* of the grid can be set using the color selection widget.

Fonts and text styling

The *Fonts and text styling* group of the table items provides the following functionalities (see Fig. 18.50):

Fig. 18.50: Attribute table Fonts and text styling Group

- You can define *Font* properties for *Table heading* and *Table contents*, using the advanced *text settings* widget (with buffer, shadow, paint effects, transparence, background, coloring, ...). Note that these changes do not affect the cells that have custom font assigned, either from the *Appearance* section or the *Table Designer* dialog. Only cells with the default rendering are overwritten.
- For *Table heading* you can additionally set the *Alignment* to `Follow column alignment` or override this setting by choosing `Left`, `Center` or `Right`. The column alignment is set using the *Select Attributes* dialog (see Fig. 18.44).

Frames

The *Frames* group of the table item properties provides the following functionalities (see Fig. 18.51):

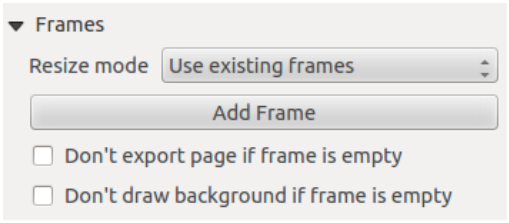


Fig. 18.51: Attribute table Frames Group

- With *Resize mode* you can select how to render the attribute table contents:
  - `Use existing frames` displays the result in the first frame and added frames only.
  - `Extend to next page` will create as many frames (and corresponding pages) as necessary to display the full selection of attribute table. Each frame can be moved around on the layout. If you resize a frame, the resulting table will be divided up between the other frames. The last frame will be trimmed to fit the table.
  - `Repeat until finished` will also create as many frames as the *Extend to next page* option, except all frames will have the same size.
- Use the *Add Frame* button to add another frame with the same size as selected frame. The result of the table that will not fit in the first frame will continue in the next frame when you use the Resize mode `Use existing frames`.
- Activate *Don't export page if frame is empty* prevents the page to be exported when the table frame has no contents. This means all other layout items, maps, scalebars, legends etc. will not be visible in the result.
- Activate *Don't draw background if frame is empty* prevents the background to be drawn when the table frame has no contents.

18.2.8 The Marker, Picture and North Arrow Items

- *The Picture Item*
  - 主要属性
  - *Size and placement*
  - *Image rotation*
- *The North Arrow Item*
- *The Marker Item*

Along with the map or legend items in the print layout, you may want to decorate your realization with images or annotations. QGIS provides different tools to achieve this:

- the *picture item*: decorates the layout with an image raster or SVG file (e.g. logos, pictures, north arrows, ...)
- the *north arrow item*: a picture item predefined with a north arrow image
- the *marker item*: decorates the layout with QGIS vector *symbols*. It can be used to place markers over a map item or for creation of advanced custom legends.

The Picture Item

You can add a picture by dragging it from your file manager onto the canvas, pasting it directly into the layout by using `Ctrl+V` or *Edit* ► *Paste* and by using the Add Picture, following *items creation instructions*. Then you can manipulate it, as explained in *Interacting with layout items*.

When using Add Picture, the picture item will be a blank frame that you can customize using its *Item Properties* panel. Other than the *items common properties*, this feature has the following functionalities:

主要属性

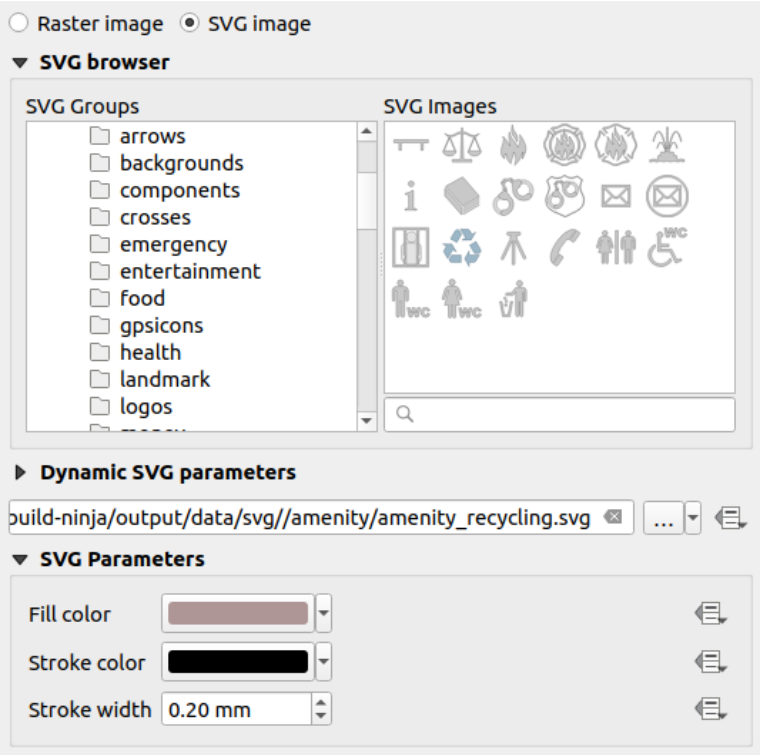


Fig. 18.52: Picture Item Properties panel

The picture item supports two types of images:

- *Raster Image*: a file selector widget can be used to fetch the data. Use the ... Browse button to select a file on your computer or enter the path directly in the text field. You can even provide a remote URL that points to a picture. The associated image can also be *embedded* in the layout.

  Use the data defined override button to set the image source from a feature attribute or using a regular expression.

- *SVG Image*: using by default the SVG libraries provided in *Settings* ► *Options* ► *System* ► *SVG Paths*. You can however use any other file, and the file selection follows the same rules as for the raster image. The SVG parameters can as well be set dynamic.

  The QGIS provided (default) `.SVG` files are customizable, meaning that you can easily apply other *Fill color*, *Stroke color* (including opacity) and *Stroke width* than the original, using their corresponding feature in the *SVG Parameters* group. These properties can also be *data-defined*.

  If you add an `.SVG` file that does not enable these properties, you may need to add the following tags to the file in order to add support e.g. for transparency:

  – *fill-opacity="param(fill-opacity)"*

– *stroke-opacity="param(outline-opacity)"*

More details at 参数化 *SVG*.

Note

Drag-and-drop an image file (raster or SVG) into the layout page will create a layout picture item with corresponding settings.

Size and placement

Fig. 18.53: Layout pictures size and placement properties

With the *Resize mode* option, you can set how the image is displayed when the frame is resized:

- `Zoom`: enlarges/reduces the image to the frame while maintaining the aspect ratio of picture
- `Stretch`: stretches the image to fit inside the frame
- `Clip`: use this mode for raster images only, it sets the size of the image to the original image size without scaling, and the frame is used to clip the image. So only the part of the image that is inside the frame will be visible.
- `Zoom and resize frame`: enlarges the image to fit the frame, and then resizes frame to fit the resulting image dimensions
- `Resize frame to image size`: sets the size of the frame to match the original size of the image (no scaling)

Depending on the selected *Resize mode*, the *Placement* and *Image rotation* options may be disabled. *Placement* lets you select the position of the image inside its frame (top/middle/bottom and left/center/right).

Image rotation

Images can be rotated with the *Image rotation* field. Activating the *Sync with map* checkbox synchronizes the rotation of the image with the rotation applied to the selected map item. This is a convenient feature to make any picture behave as a north arrow. The *North alignment* can be:

- **Grid north**: the direction of a grid line which is parallel to the central meridian of the national/local grid
- **True north**: direction of a meridian of longitude.

You can also apply a declination *Offset* to the picture rotation.

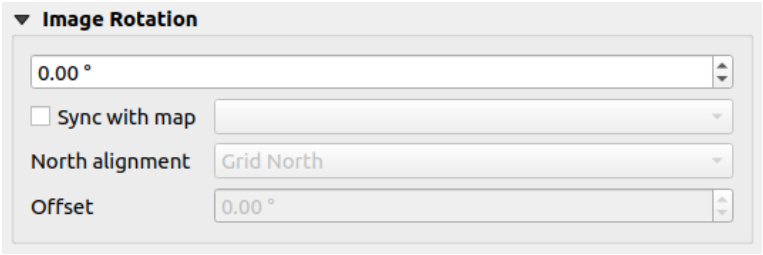


Fig. 18.54: Layout pictures image rotation properties

The North Arrow Item

You can add a north arrow with the Add North Arrow button, following *items creation instructions* and manipulate it the same way as exposed in *Interacting with layout items*.

Since north arrows are images, the *North Arrow* item has the same properties as the *picture item*. The main differences are:

- A default north arrow is used when adding the item, instead of a blank frame
- The north arrow item is synced with a map item by default: the *Sync with map* property is the map over which the north arrow item is drawn. If none, it falls back to the *reference map*.

> **Note**
>
> Many of the north arrows do not have an 'N' added in the north arrow. This is done on purpose, since there are languages that do not use an 'N' for North.

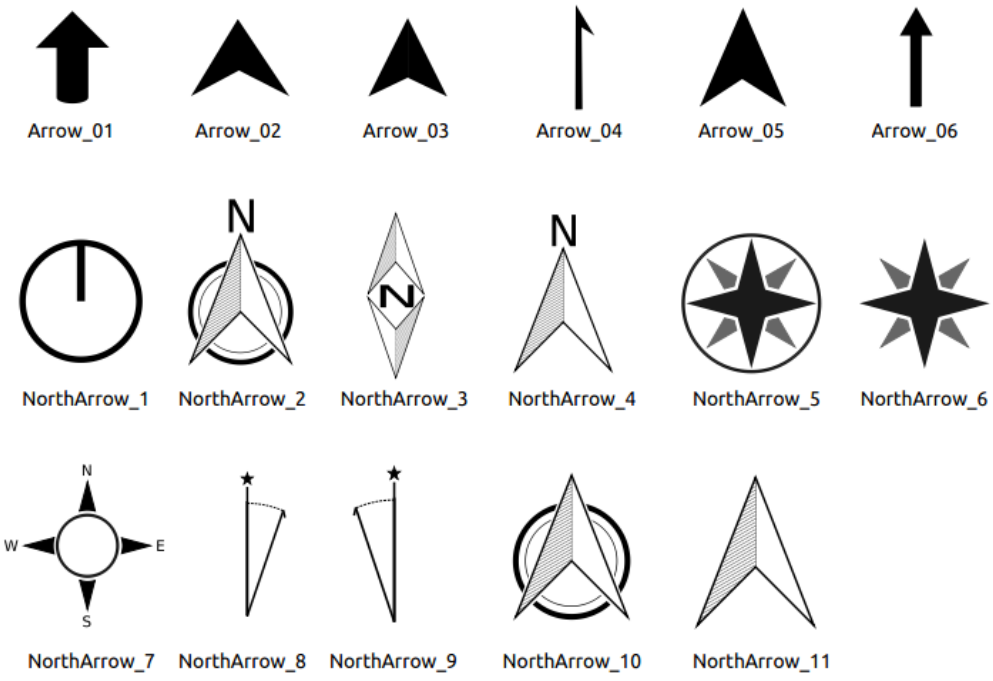


Fig. 18.55: North arrows available for selection in provided SVG library

The Marker Item

To add a marker item, select the Add Marker button, and click on the page. A default point marker symbol is added. Then you can manipulate it, as explained in *Interacting with layout items*. But note that unlike most of the other items, you resize the item given that its size is controlled by the embedded symbols properties.

The marker item can be customized from the *Item Properties* panel. Other than the *items common properties*, you can also:

- modify the *Symbol*, relying on all the symbol *widget capabilities*
- sync the marker item rotation with the map's (see *Image rotation*), acting as a north arrow. The map rotation is added to any existing marker symbol level rotation (so .e.g if you have to rotate the triangle marker 90° to get it pointing straight up, it will still work nicely in north arrow mode!)

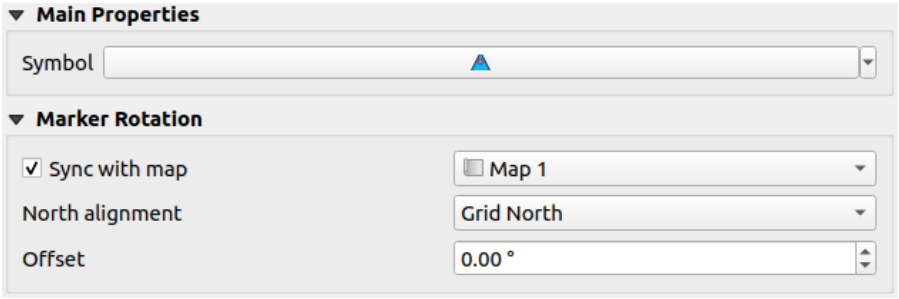


Fig. 18.56: The marker item custom properties

18.2.9 The Elevation Profile Item

The Elevation Profile item is used to display an *elevation profile view* in a layout. Use the *Add Elevation Profile* button, and follow *items creation instructions* to add a new elevation profile item that you can later manipulate the same way as demonstrated in *Interacting with layout items*.

A new elevation profile item has default settings that make it render an empty chart. You can customize its properties in the *Item Properties* panel. In addition to the *common properties*, this feature has the following functionalities:

The Elevation profile *Item Properties* panel embeds a top toolbar with the following functionalities:

- Update elevation profile to refresh the item rendering
- Copy from elevation profile: a drop-down menu to select an elevation profile view from. The view settings are applied to the layout elevation profile item, and can be later modified.

Layers

Under the *Layers* group, check in the tree view the layers you would like to render in the profile item. Remember to properly configure the *Elevation* properties of the selected layers.

Profile curve

- *Controlled by atlas*: the *profile curve* will be taken from the current atlas feature and the elevation profile view updated as you walk through the atlas features. This is currently supported for an active layout atlas or report, using a line geometry type coverage layer.
- The *Tolerance* distance, which can be data-defined, helps you control how far a feature of the visible layers should be from the profile curve in order to display in the layout elevation view. Only point features are currently returned.

Show Subsections Indicator

It is possible to highlight the position of the vertices of the curve on the Elevation Profile. Select the *Show Subsections Indicator* to display vertical lines at the distances of the vertices. Use the *Lines* drop-down list to set the symbol, color, opacity and width of the Subsections Indicator lines.

Chart ranges

A layout elevation profile item does not necessarily display the full extent of the elevation profile view it is based on. You can limit the area to render, providing:

- on the X axis, the *Minimum distance* and *Maximum distance* from the profile curve starting point
- On the Y axis, the *Minimum elevation* and *Maximum elevation*

Distance and elevation axes

The *Distance axis* and *Elevation axis* groups give options to tweak the grid over the elevation profile item, respectively on the X and Y axes:

- the distance display *Unit*, allowing to override the map canvas unit
- the graduation on the axis with both a *Major interval* and *Minor interval*
- the line symbols to apply to the corresponding *Major grid lines* and *Minor grid lines*
- how regular the graduation items should be labeled (*Label interval*) as well as their *Label format* and *Label font*
- the *Distance labels*: configures whether the distance unit symbol should be placed next to *Every value*, *First value*, *Last value*, *First and last values*, or skipped (*None*)

Chart area

Under *Chart area*, you can configure the rendering of the area in which the elevation profile plot is actually displayed:

- a *Background* fill symbol
- a *Border* line symbol
- the margins from the elevation profile item border

18.2.10 The HTML Frame Item

It is possible to add a frame that displays the contents of a website or even create and style your own HTML page and display it! You can add a picture with the *Add HTML* following *items creation instructions* and manipulate it the same way as exposed in *Interacting with layout items*. Note that the HTML scale is controlled by the layout export resolution at the time the HTML frame is created.

The HTML item can be customized using its *Item Properties* panel. Other than the *items common properties*, this feature has the following functionalities (see Fig. 18.57):

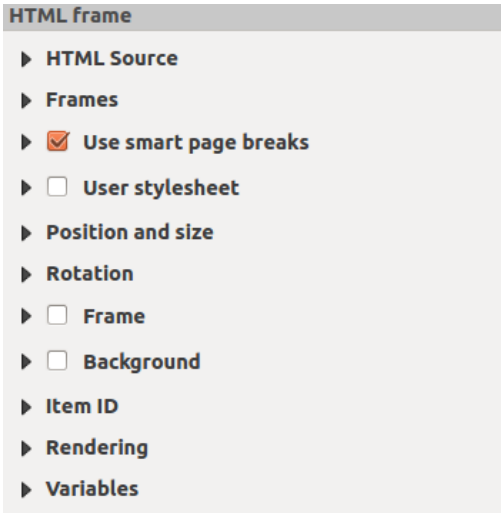


Fig. 18.57: HTML Frame, the Item Properties Panel

HTML Source

The *HTML Source* group of the HTML frame *Item Properties* panel provides the following functionalities (see Fig. 18.58):

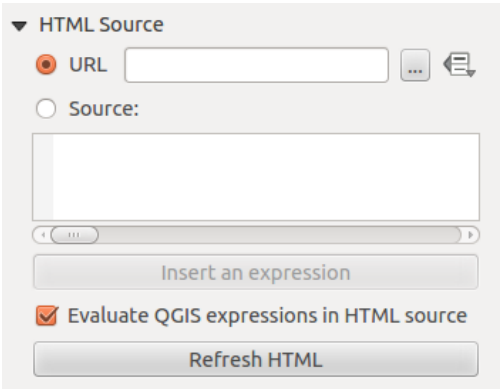


Fig. 18.58: HTML frame, the HTML Source properties

- In *URL* you can enter the URL of a webpage you copied from your Internet browser or select an HTML file using the ... Browse button. There is also the option to use the Data-defined override button, to provide a URL from the contents of an attribute field of a table or using a regular expression.
- In *Source* you can enter text in the textbox with some HTML tags or provide a full HTML page.
- The *Insert or Edit an Expression...* button can be used to add an expression like `[%Year($now)%]` in the Source textbox to display the current year. This button is only activated when radiobutton *Source* is selected.

After inserting the expression click somewhere in the textbox before refreshing the HTML frame, otherwise you will lose the expression.

- Activate *Evaluate QGIS expressions in HTML code* to see the result of the expression you have included, otherwise you will see the expression instead.
- Use the *Refresh HTML* button to refresh the HTML frame(s) and see the result of changes.

Frames

The *Frames* group of the HTML frame *Item Properties* panel provides the following functionalities (see Fig. 18.59):

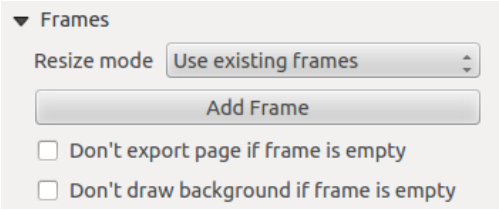


Fig. 18.59: HTML frame, the Frames properties

- With *Resize mode* you can select how to render the HTML contents:
  - `Use existing frames` displays the result in the first frame and added frames only.
  - `Extend to next page` will create as many frames (and corresponding pages) as necessary to render the height of the web page. Each frame can be moved around on the layout. If you resize a frame, the webpage will be divided up between the other frames. The last frame will be trimmed to fit the web page.
  - `Repeat on every page` will repeat the upper left of the web page on every page in frames of the same size.
  - `Repeat until finished` will also create as many frames as the `Extend to next page` option, except all frames will have the same size.
- Use the *Add Frame* button to add another frame with the same size as selected frame. If the HTML page does not fit in the first frame it will continue in the next frame when you use *Resize mode* or *Use existing frames*.
- Activate *Don't export page if frame is empty* prevents the page from being exported when the frame has no HTML contents. This means all other layout items, maps, scale bars, legends etc. will not be visible in the result.
- Activate *Don't draw background if frame is empty* prevents the HTML frame being drawn if the frame is empty.

Use smart page breaks and User style sheet

The *Use smart page breaks* dialog and *User style sheet* dialog of the HTML frame *Item Properties* panel provides the following functionalities (see Fig. 18.60):

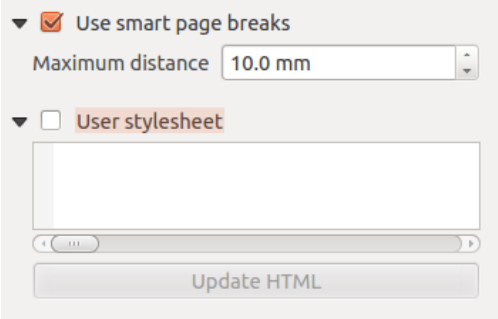


Fig. 18.60: HTML frame, Use smart page breaks and User style sheet properties

- Activate *Use smart page breaks* to prevent the html frame contents from breaking mid-way a line of text so it continues nice and smooth in the next frame.
- Set the *Maximum distance* allowed when calculating where to place page breaks in the html. This distance is the maximum amount of empty space allowed at the bottom of a frame after calculating the optimum break location. Setting a larger value will result in better choice of page break location, but more wasted space at the bottom of frames. This is only used when *Use smart page breaks* is activated.
- Activate *User style sheet* to apply HTML styles that often is provided in cascading style sheets. An example of style code is provided below to set the color of `<h1>` header tag to green and set the font and font size of text included in paragraph tags `<p>`.

```
h1 {color: #00ff00;
}
p {font-family: "Times New Roman", Times, serif;
   font-size: 20px;
}
```

- Use the *Update HTML* button to see the result of the style sheet settings.

18.2.11 图形项

QGIS 在打印布局中提供了一系列工具绘制常规或复杂图形

> **Note**
>
> 与其他打印布局项不同，你不能自定义框架风格，也不能自定义形状边界的背景颜色（默认设定为透明色）

常规图形项

布局:guilabel: 图形项是一个有助于布置你地图的工具, 包含三角形、矩形和椭圆形等常规图形... 你可以使用 laddBasicShapel:sup: 添加图形工具来添加一个常规图形，点击该工具集可以访问特定的工具，例如 添加矩形,laddBasicCirclel 添加椭圆 和 添加三角形. 选择了合适的工具之后, 你可以根据接下来的:ref:‘项目创建指令 <create_layout_item>‘绘制图形. 和其他布局项一样, 常规形状的操作方法和:ref:‘interact_layout_item‘中提示的操作一样.

> **Note**
>
> 在绘制基本图形时要按住:kbd:‘Shift‘键，使用单击 + 拖动的方式，可以帮助你创建一个完美的正方形，圆或者三角形.

默认的图形项可以使用它的:guilabel: ‘项目属性’ 面板来定制。除了:ref:’ 项目基本属性 <item_common_properties>‘, 该特性还有以下功能 (参见:numref: ’ figure_layout_basic_shape ’):

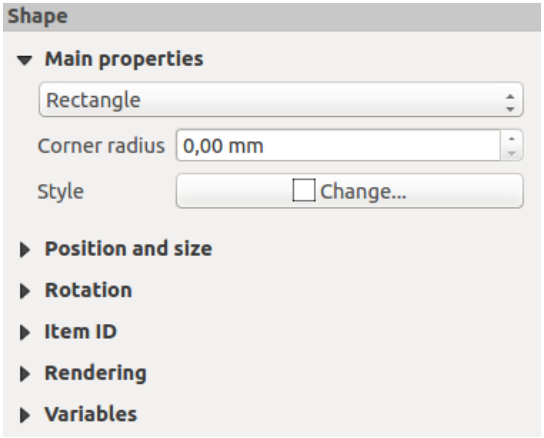


Fig. 18.61: 图形项属性面板

主要特性组中，并在图框中允许你转换形状项的类型 (**椭圆**, **矩形** or **三角形**).

你可以使用高级选择器控件:ref: 符号 <*symbol-selector*> and 颜色 设置形状的样式

对于矩形图形, 你可以使用不同的单位来更改:guilabel: 圆角半径设置圆角.

基于结点的形状项

尽管 laddBasicShapel 添加形状工具提供了创建简单和预定义几何项的方式, :guilabel:‘添加结点项 ‘工具帮助你创建一个自定义或更高级的几何项. 对于折线或多边形, 你可以画任意数量的线或者边, 并且项目的顶点可以使用 :guilabel:‘编辑结点项 ‘独立且直接地操作. 项目本身也可以在:ref:‘interact_layout_item‘中操作.

添加一个基于结点的形状

1. 单击 添加结点项 图标
2. 选择 添加多边形 或 添加折线 工具
3. Perform consecutive left clicks to add nodes of your item. If you hold down the `Shift` key while drawing a segment, it is constrained to follow an orientation multiple of 45°.
4. 完成绘制后，右键单击，结束图形绘制。

你可以在:guilabel: 项目特性面板自定义形状的外观.

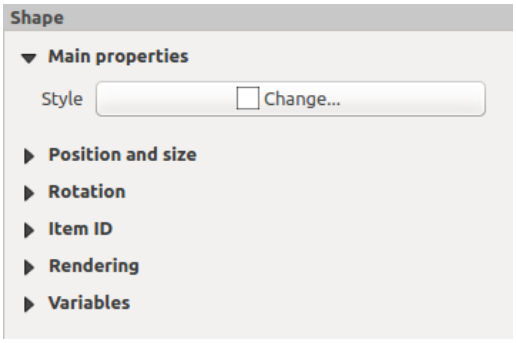


Fig. 18.62: 多边形结点形状项目特性面板

在 主要特性 '中, 你可以使用高级选择器控件:ref:'符号 <*symbol-selector*> 和:ref: 颜色 <*color-selector*> 设置样式...

对于折线结点项, 你同样可以使用:guilabel:'线路标记 '参数化后添加

- 使用选项开始/结束标记:
  - *无*: 绘制一个简单折线
  - *箭头*: 添加一个自定义箭头指向的常规三角形.
  - *SVG* 标记: 使用文件:file:*SVG* 文件作为项目目标.
- 自定义箭头指向
  - *Arrow stroke color*: sets the stroke color of the arrow head.
  - *Arrow fill color*: sets the fill color of the arrow head.
  - *Arrow stroke width*: sets the stroke width of the arrow head.
  - *Arrow head width*: sets the size of the arrow head.

SVG images are automatically rotated with the line. Stroke and fill colors of QGIS predefined SVG images can be changed using the corresponding options. Custom SVG may require some tags following this *instruction*.

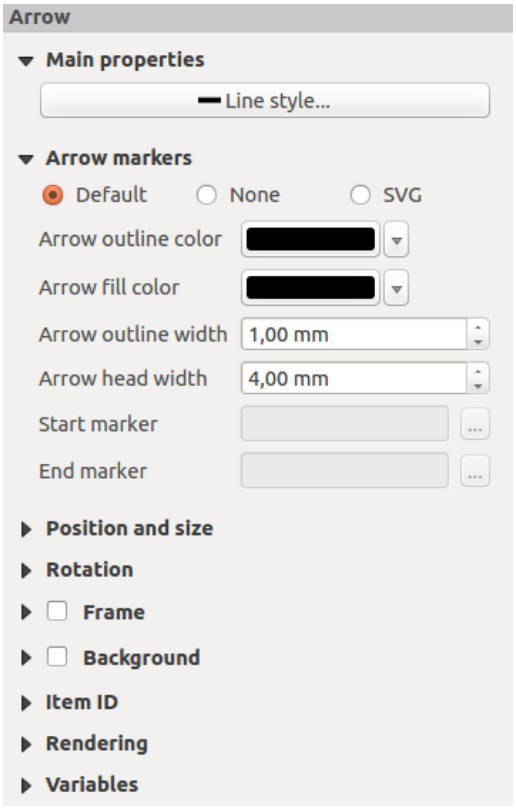


Fig. 18.63: Polyline Node Shape Item Properties Panel

The Arrow Item

The Add Arrow tool is a shortcut to create an arrow-enabled polyline by default and thus has the same properties and behavior as a *polyline node item*.

Actually, the arrow item can be used to add a simple arrow, for example, to show the relation between two different print layout items. However, to create a north arrow, the *image item* should be considered first as it gives access to a set of north arrows in `.SVG` format that you can sync with a map item so that it rotates automatically with it.

Editing a node item geometry

A specific tool is provided to edit node-based shapes through Edit Nodes Item. Within this mode, you can select a node by clicking on it (a marker is displayed on the selected node). A selected node can be moved either by dragging it or by using the arrow keys. Moreover, in this mode, you are able to add nodes to an existing shape: double-click on a segment and a node is added at the place you click. Finally, you can remove the currently selected node by hitting the `Del` key.

18.3 Creating an Output

Fig. 18.64 shows an example print layout including all the types of layout items described in the previous section.

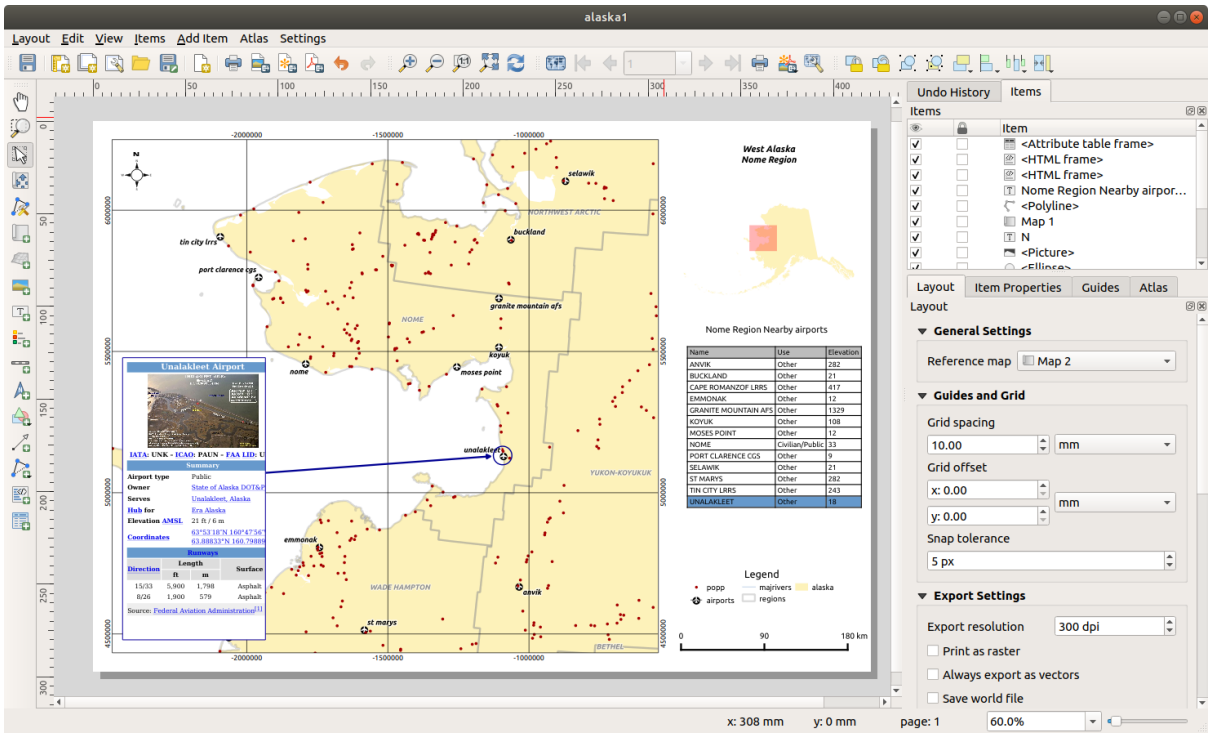


Fig. 18.64: Print Layout with map view, legend, image, scale bar, coordinates, text and HTML frame added

From the *Layout* menu or toolbar, you can output the print layout to different file formats, and it is possible to modify the resolution (print quality) and paper size:

- The Print icon allows you to print the layout to a connected printer or a PostScript file, depending on the installed printer drivers.
- The Export as image icon exports the print layout image formats such as `PNG`, `BMP`, `TIF`, `JPG`, and many others...
- The Export as SVG icon saves the print layout as an `SVG` (Scalable Vector Graphic).
- The Export as PDF icon saves the defined print layout directly as a `PDF` (Portable Document Format) file.

18.3.1 Export settings

Whenever you export a print layout, there are a selection of export settings QGIS needs to check in order to produce the most appropriate output. These configurations are:

- The *Export settings* of the *Layout* panel, such as *Export resolution*, *Print as raster* *Always export as vectors* or *Save world file*
- *Exclude page from exports* in the *page item properties* panel
- *Exclude item from exports* in the *item properties* panel

Moreover, a number of predefined checks are automatically applied to the layout. Currently these checks include testing that scalebars are correctly linked to map items, and that map overview items are also correctly linked to a map. If the checks fail, you are shown a nice warning advising you of the issue.

18.3.2 Export as Image

To export a layout as an image:

1. Click the Export as image icon
2. Select the image format, the folder and filename (e.g. `myill.png`) to use. If the layout contains more than one page, each page will be exported to a file with the given filename with the page number appended (e.g. `myill_2.png`).
3. In the next (*Image Export Options*) dialog:
   - You can override the print layout *Export resolution* and the exported page dimensions (as set in *Layout* panel).
   - For JPEG and JPG image formats, you can set the *JPEG Quality* of the output image.
   - Image rendering can also be improved with the *Enable antialiasing* option.
   - If you want to export your layout as a **georeferenced image** (e.g., to share with other projects), check the *Generate world file* option, and an *ESRI World File* with the same name as the exported image, but a different extension (`.tfw` for TIFF, `.pnw` for PNG, `jgw` for JPEG, ...) will be created when exporting. This option can also be checked by default in the *layout panel*.

     > **Note**
     >
     > For multi-page output, only the page that contains the *reference map* will get a world file (assuming that the *Generate world file* option is checked).

   - By checking *Crop to content* option, the image output by the layout will include the minimal area enclosing all the items (map, legend, scale bar, shapes, label, image...) of each page of the composition:
     - If the composition includes a single page, then the output is resized to include EVERYTHING on the composition. The page can then be reduced or extended to all items depending on their position (on, above, below, left or right of the page).
     - In case of a multi-page layout, each page will be resized to include items in its area (left and right sides for all pages, plus top for the first page and bottom for the last page). Each resized page is exported to a separate file.

     The *Crop to content* dialog also lets you add margins around the cropped bounds.
   - By checking *Open file after exporting* the exported file will automatically open in the default image viewer.

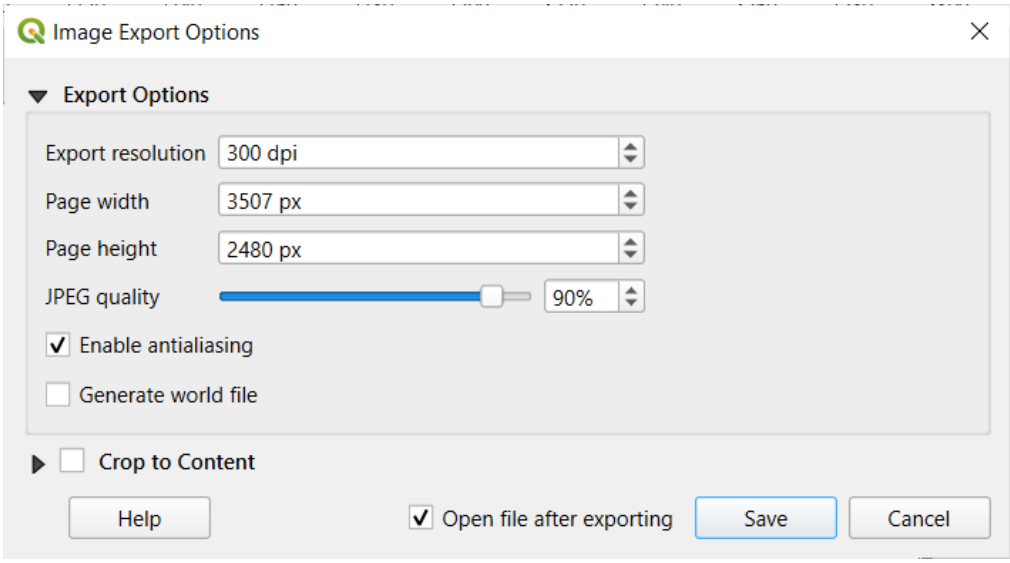


Fig. 18.65: Image Export Options

Tip

Use image formats that support transparency when items extend beyond the paper extent

Layout items may be placed outside the paper extent. When exporting with the *Crop to content* option, the resulting image may therefore extend beyond the paper extent. Since the background outside of the paper extent will be transparent, for image formats that do not support transparency (e.g. `BMP` and `JPG`) the transparent background will be rendered as full black, "corrupting" the image. Use transparency-compatible formats (e.g. `TIFF` and `PNG`) in such cases.

Note

When supported by the format (e.g. `PNG`) and the underlying Qt library, the exported image may include *project metadata* (author, title, date, description...)

18.3.3 Export as SVG

To export a layout as SVG:

1. Click the Export as SVG icon
2. Fill in the path and filename (used as a base name for all the files in case of multi-page composition, as for image export)
3. In the next *SVG Export Options* dialog, you can override the layout default *export settings* or configure new ones:
   - *Export map layers as SVG groups*: exported items are grouped within layers whose name matches the layer names from QGIS, making it much easier to understand the contents of the document.
   - *Always export as vectors*: some rendering options require items to be rasterized for a better rendering. Check this option to keep the objects as vectors with the risk that the appearance of the output file may not match the print layout preview (for more details, see *Export settings*).
   - 文档的 *RDF* 元数据，例如标题、作者、日期、描述...

- *Simplify geometries to reduce output file size*: this avoids exporting ALL geometry vertices, which can result in a ridiculously complex and large export file size that could fail to load in other applications. Geometries will be simplified while exporting the layout in order to remove any redundant vertices which are not discernably different at the export resolution (e.g. if the export resolution is `300 dpi`, vertices that are less than `1/600 inch` apart will be removed).
- Set the *Text export*: controls whether text labels are always or preferably exported as *text or outline objects*.
- Apply *Crop to content option*
- *Disable tiled raster layer exports*: When exporting files, QGIS uses a built-in raster layer tiled rendering that saves memory. Sometimes, this can cause visible "seams" in the rasters for generated files. Checking this option would fix that, at the cost of a higher memory usage during exports.
- By checking *Open file after exporting* the exported file will automatically open in the default SVG viewer.

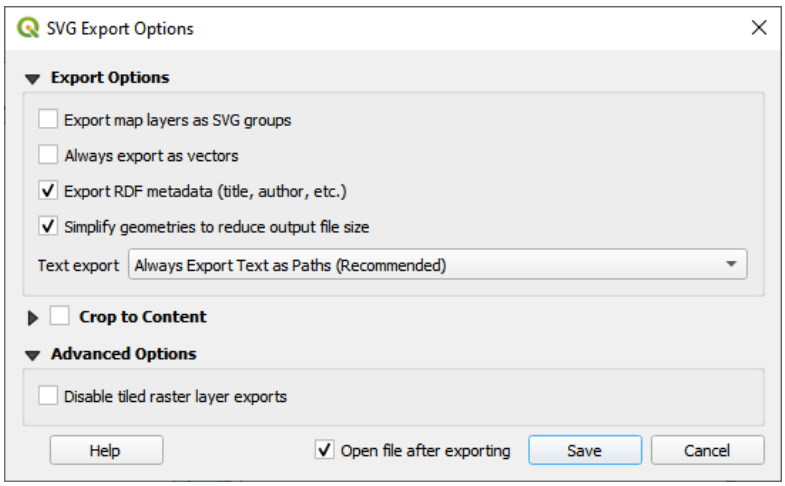


Fig. 18.66: SVG Export Options

Note

Currently, the SVG output is very basic. This is not a QGIS problem, but a problem with the underlying Qt library. This will hopefully be sorted out in future versions.

18.3.4 Export as PDF

To export a layout as PDF:

1. Click the Export as PDF icon
2. Fill in the path and filename: unlike for image and SVG export, all the pages in the layout are exported to a single PDF file.
3. In the next *PDF Export Options* dialog, you can override the layout default *export settings* or configure new ones:
   - *Always export as vectors*: some rendering options require items to be rasterized for a better rendering. Check this option to keep the objects as vectors with the risk that the appearance of the output file may not match the print layout preview (for more details, see *Export settings*).

> **Note**
>
> **Mind opacity settings for vector layers**
>
> When you set a layer-wide opacity (or opacity for any vector symbology) to less than 100%, it forces a rasterized rendering of those objects during PDF exports. Depending on the size of the layer, this may considerably increase the PDF file size.

- ☑ *Append georeference information*: available only if the *reference map*, from which the information is taken, is on the first page.
- 文档的 ☑ 导出 *RDF* 元数据，例如标题、作者、日期、描述...
- Set the *Text export*: controls whether text labels are always or preferably exported as *text or outline objects*.
- Control the PDF *Image compression* using:
  - *Lossy (JPEG)*, which is the default compression mode
  - or *Lossless*, which creates bigger files in most cases, but is much more suitable for printing outputs or for post-production in external applications (requires Qt 5.13 or later).
- ☐ *Create Geospatial PDF*: Generate a georeferenced PDF file.
- ☐ *Disable tiled raster layer exports*: When exporting files, QGIS uses tiled based rendering that saves memory. Sometimes, this can cause visible ”seams” in the rasters for generated files. Checking this option would fix that, at the cost of a higher memory usage during exports.
- ☑ *Simplify geometries to reduce output file size*: Geometries will be simplified while exporting the layout by removing vertices that are not discernably different at the export resolution (e.g. if the export resolution is `300 dpi`, vertices that are less than `1/600 inch` apart will be removed). This can reduce the size and complexity of the export file (very large files can fail to load in other applications).
- By checking ☑ *Open file after exporting* the exported file will automatically open in the default PDF viewer.

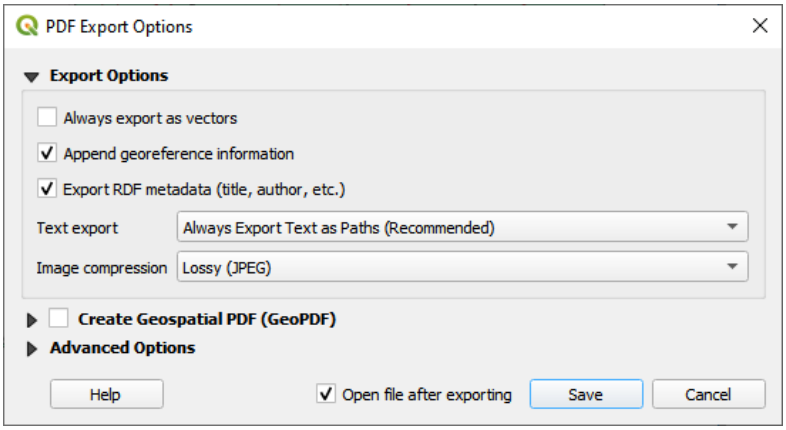


Fig. 18.67: PDF Export Options

> **Note**
>
> Geospatial PDF export is supported, and a number of Geospatial PDF specific options are available:
>
> - *Format* (Geospatial PDF format - there are some variations),
> - *Include multiple map themes* (specify map themes to include),

- *Include vector feature information* (choose the layers and group them into logical PDF groups).

Note

Exporting a print layout to formats that supports georeferencing (e.g. `PDF` and `TIFF`) creates a georeferenced output by default.

Note

When exporting maps or layouts to vector formats such as PDF or SVG, line pattern fills are rendered using a line-by-line vector-based approach instead of rasterized tiles. This change results in smaller and higher-quality output files, with individual pattern lines remaining fully editable in external vector graphics software.

18.3.5 Generate an Atlas

Atlas functions allow you to create map books in an automated way. Atlas uses the features of a table or vector layer (*Coverage layer*) to create an output for each feature (**atlas feature**) in the table / layer. The most common usage is to zoom a map item to the current atlas feature. Further use cases include:

- a map item showing, for another layer, only features that share the same attribute as the atlas feature or are within its geometry.
- a label or HTML item whose text is replaced as features are iterated over
- a table item showing attributes of associated *parent or children* features of the current atlas feature...

For each feature, the output is processed for all pages and items according to their exports settings.

Tip

Use variables for more flexibility

QGIS provides a large panel of functions and *variables*, including atlas related ones, that you can use to manipulate the layout items, but also the symbology of the layers, according to atlas status. Combining these features gives you a lot of flexibility and helps you easily produce advanced maps.

To enable the generation of an atlas and access atlas parameters, refer to the *Atlas* panel. This panel contains the following (see Fig. 18.68):

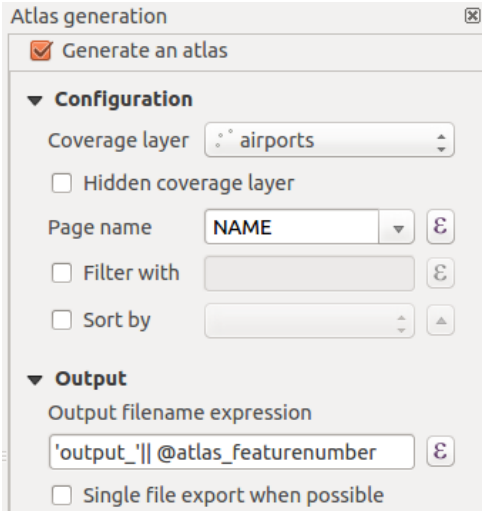


Fig. 18.68: Atlas Panel

- *Generate an atlas* enables or disables atlas generation.
- *Configuration*
  - A *Coverage layer* combo box that allows you to choose the table or vector layer containing the features to iterate over.
  - An optional *Hidden coverage layer* that, if checked, will hide the coverage layer (but not the other layers) during the generation.
  - An optional *Page name* combo box to specify the name for the feature page(s). You can select a field of the coverage layer or set an *expression*. If this option is empty, QGIS will use an internal ID, according to the filter and/or the sort order applied to the layer.
  - An optional *Filter with* text area that allows you to specify an expression for filtering features from the coverage layer. If the expression is not empty, only features that evaluate to `True` will be processed.
  - An optional *Sort by* that allows you to sort features of the coverage layer (and the output), using a field of the coverage layer or an expression. The sort order (either ascending or descending) is set by the two-state *Sort direction* button that displays an up or a down arrow.
- *Output* - this is where the output of the atlas can be configured:
  - An *Output filename expression* textbox that is used to generate a filename for each atlas feature. It is based on expressions. is meaningful only for rendering to multiple files.
  - A *Single file export when possible* that allows you to force the generation of a single file if this is possible with the chosen output format (`PDF`, for instance). If this field is checked, the value of the *Output filename expression* field is meaningless.
  - An *Image export format* drop-down list to select the output format when using the Export atlas as Images... button.

Control map by atlas

The most common usage of atlas is with the map item, zooming to the current atlas feature, as iteration goes over the coverage layer. This behavior is set in the *Controlled by atlas* group properties of the map item. See *Controlled by atlas* for different settings you can apply on the map item.

Customize labels with expression

In order to adapt labels to the feature the atlas iterates over, you can include expressions. Make sure that you place the expression part (including functions, fields or variables) between `[%` and `%]` (see *The Label Item* for more details).

For example, for a city layer with fields `CITY_NAME` and `ZIPCODE`, you could insert this:

```
The area of [% concat( upper(CITY_NAME), ',', ZIPCODE, ' is ',
format_number($area/1000000, 2) ) %] km2
```

or, another combination:

```
The area of [% upper(CITY_NAME)%],[%ZIPCODE%] is
[%format_number($area/1000000,2) %] km2
```

The information `[% concat( upper(CITY_NAME), ',', ZIPCODE, ' is ', format_number($area/1000000, 2) ) %]` is an expression used inside the label. Both expressions would result in the following type of label in the generated atlas:

```
The area of PARIS,75001 is 1.94 km2
```

Explore Data-defined override buttons with atlas

There are several places where you can use a Data defined override button to override the selected setting. This is particularly useful with atlas generation. See *Data defined override setup* for more details on this widget.

For the following examples the `Regions` layer of the QGIS sample dataset is used and selected as *Coverage layer* for the atlas generation. We assume that it is a single page layout containing a map item and a label item.

When the height (north-south) of a region extent is greater than its width (east-west), you should use *Portrait* instead of *Landscape* orientation to optimize the use of paper. With a Data Defined Override button you can dynamically set the paper orientation.

Right-click on the page and select *Page Properties* to open the panel. We want to set the orientation dynamically, using an expression depending on the region geometry, so press the button of field *Orientation*, select *Edit...* to open the *Expression string builder* dialog and enter the following expression:

```
CASE WHEN bounds_width(@atlas_geometry) > bounds_height(@atlas_geometry)
THEN 'Landscape' ELSE 'Portrait' END
```

Now if you *preview the atlas*, the paper orients itself automatically, but item placements may not be ideal. For each Region you need to reposition the location of the layout items as well. For the map item you can use the button of its *Width* property to set it dynamic using the following expression:

```
@layout_pagewidth - 20
```

Likewise, use the button of the *Height* property to provide the following expression to constrain map item size:

```
@layout_pageheight - 20
```

To ensure the map item is centered in the page, set its *Reference point* to the upper left radio button and enter `10` for its *X* and *Y* positions.

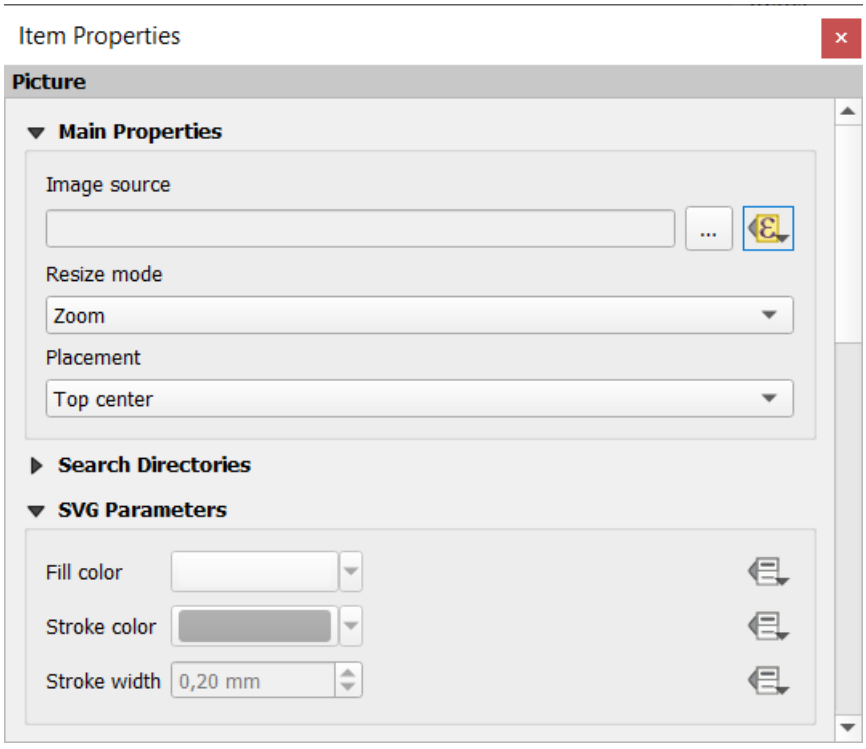
Let's add a title above the map in the center of the page. Select the label item and set the horizontal alignment to *Center*. Next move the label to the right position, choose the middle button for the *Reference point*, and provide the following expression for field *X*:

```
@layout_pagewidth / 2
```

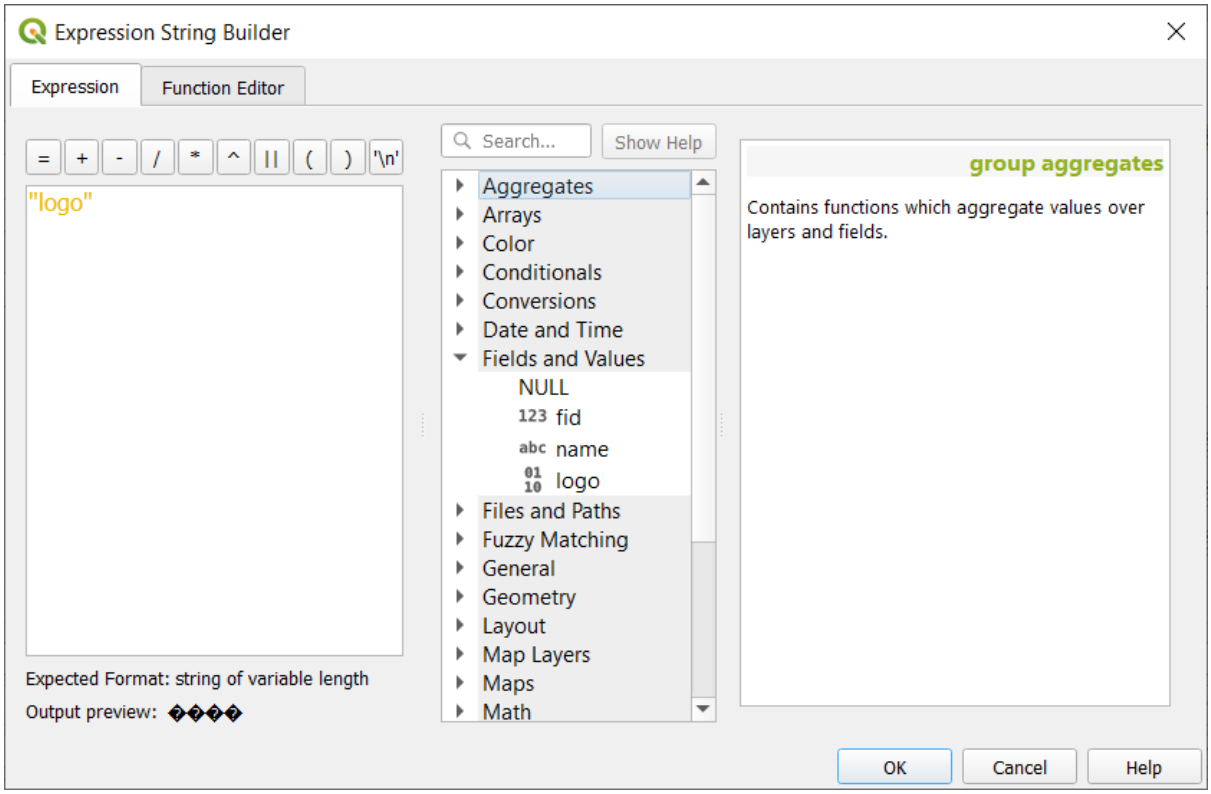
For all other layout items you can set the position in a similar way so they are correctly positioned both for portrait and landscape. You can also do more tweaks such as customizing the title with feature attributes (see *Customize labels with expression* example), changing images, resizing the number of legend columns number according to page orientation, ...

The information provided here is an update of the excellent blog (in English and Portuguese) on the Data Defined Override options Multiple_format_map_series_using_QGIS_2.6 .

Another example for using data-defined override buttons is the usage of a dynamic picture. For the following examples we use a geopackage layer containing a BLOB field called `logo` with the field type binary (see 新建地理包图层). For every feature there is defined a different picture so that the atlas can iterate over as described in *Preview and generate an atlas*. All you need to do is add a picture in the print layout and go to its *Item properties* in the atlas context. There you can find a data-defined override button in the *Image source* section of the *Main Properties*.



In the following window choose *Edit* so that the *Expression String Builder* opens. From the *Fields and values* section you can find the BLOB field that was defined in the geopackage layer. Double-click the field name `logo` and click *OK*.



The atlas iterates over the entries in the BLOB field provided that you choose the geopackage layer as *Coverage layer* (further instructions you can find in *Preview and generate an atlas*).

These are just two examples of how you can use some advanced settings with atlas.

Preview and generate an atlas

Fig. 18.69: Atlas Preview toolbar

Once the atlas settings have been configured, and layout items (map, table, image...) linked to it, you can create a preview of all the pages by choosing *Atlas* ► *Preview Atlas* or clicking the Preview Atlas icon. You can then use the arrows to navigate through all the features:

- First feature
- Previous feature
- Next feature
- Last feature

You can also use the combo box to select and preview a specific feature. The combo box shows atlas feature names according to the expression set in the atlas *Page name* option.

As for simple compositions, an atlas can be generated in different ways (see *Creating an Output* for more information - just use tools from the *Atlas* menu or toolbar instead of the *Layout* menu.

This means that you can directly print your compositions with *Atlas* ► *Print Atlas*. You can also create a PDF using *Atlas* ► *Export Atlas as PDF...*: You will be asked for a directory to save all the generated PDF files, except if the *Single file export when possible* has been selected. In that case, you'll be prompted to give a filename.

With *Atlas* ► *Export Atlas as Images...* or *Atlas* ► *Export Atlas as SVG...* tool, you're also prompted to select a folder. Each page of each atlas feature composition is exported to the image file format set in *Atlas* panel or to SVG.

Note

With multi-page output, an atlas behaves like a layout in that only the page that contains the *General settings* will get a world file (for each feature output).

Tip

Print a specific atlas feature

If you want to print or export the composition of only one feature of the atlas, simply start the preview, select the desired feature in the drop-down list and click on *Layout* ► *Print* (or *Export...* to any supported file format).

Use project defined relations for atlas creation

For users with HTML and Javascript knowledge it is possible to operate on GeoJSON objects and use project defined relations from the QGIS project. The difference between this approach and using expressions directly inserted into the HTML is that it gives you a full, unstructured GeoJSON feature to work with. This means that you can use existing Javascript libraries and functions that operate on GeoJSON feature representations.

The following code includes all related child features from the defined relation. Using the JavaScript `setFeature` function it allows you to make flexible HTML which represents relations in whatever format you like (lists, tables, etc). In the code sample, we create a dynamic bullet list of the related child features.

```
// Declare the two HTML div elements we will use for the parent feature id
// and information about the children
<div id="parent"></div>
<div id="my_children"></div>

<script type="text/javascript">
   function setFeature(feature)
   {
     // Show the parent feature's identifier (using its "ID" field)
     document.getElementById('parent').innerHTML = feature.properties.ID;
     //clear the existing relation contents
     document.getElementById('my_children').innerHTML = '';
     feature.properties.my_relation.forEach(function(child_feature) {
     // for each related child feature, create a list element
     // with the feature's name (using its "NAME" field)
       var node = document.createElement("li");
       node.appendChild(document.createTextNode(child_feature.NAME));
       document.getElementById('my_children').appendChild(node);
     });
   }
</script>
```

During atlas creation there will be an iteration over the coverage layer containing the parent features. On each page, you will see a bullet list of the related child features following the parent's identifier.

18.4 Creating a Report

This section will help you set up a report in QGIS.

18.4.1 What is it?

By definition, a GIS report is a document containing information organized in a narrative way, containing maps, text, graphics, tables, etc. A report can be prepared ad hoc, periodic, recurring, regular, or as required. Reports may refer to specific periods, events, occurrences, subjects or locations.

In QGIS, a *Report* is an extension of a *Layouts*.

Reports allow users to output their GIS projects in a simple, quick and structured way.

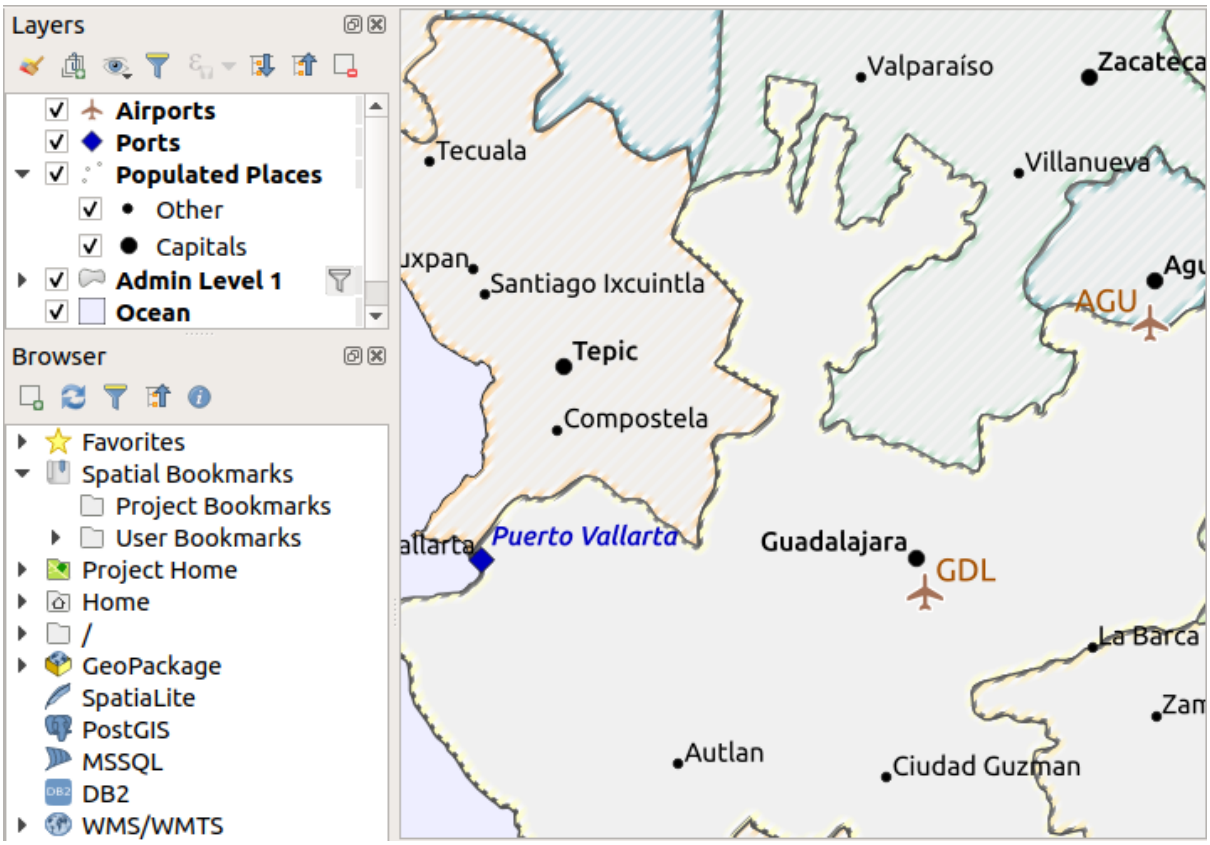
A report can be created with *Project* ► *New Report* or inside the *Project* ► *Layout Manager*.

> **Note**
>
> The maps in QGIS reports behave in the same way as maps in print layouts and atlases. We will concentrate on the specifics of QGIS reports. For details on map handling, see the sections on *print layouts* and *atlases*.

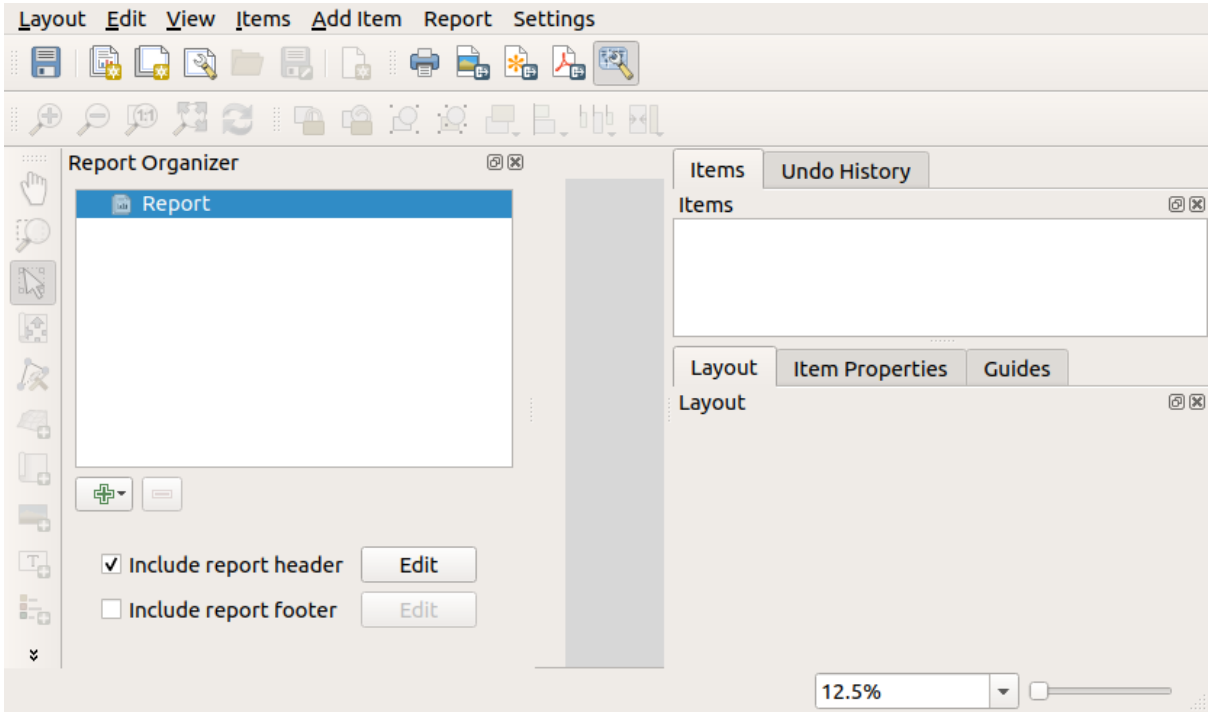
18.4.2 Get started

In the *Layout Manager* dialog a report can be created through *New from template* by selecting the dropdown option *Empty Report* and hitting the *Create...* button.

For this example, we use some administrative boundaries, populated places, ports and airports from the Natural Earth dataset (1:10M).

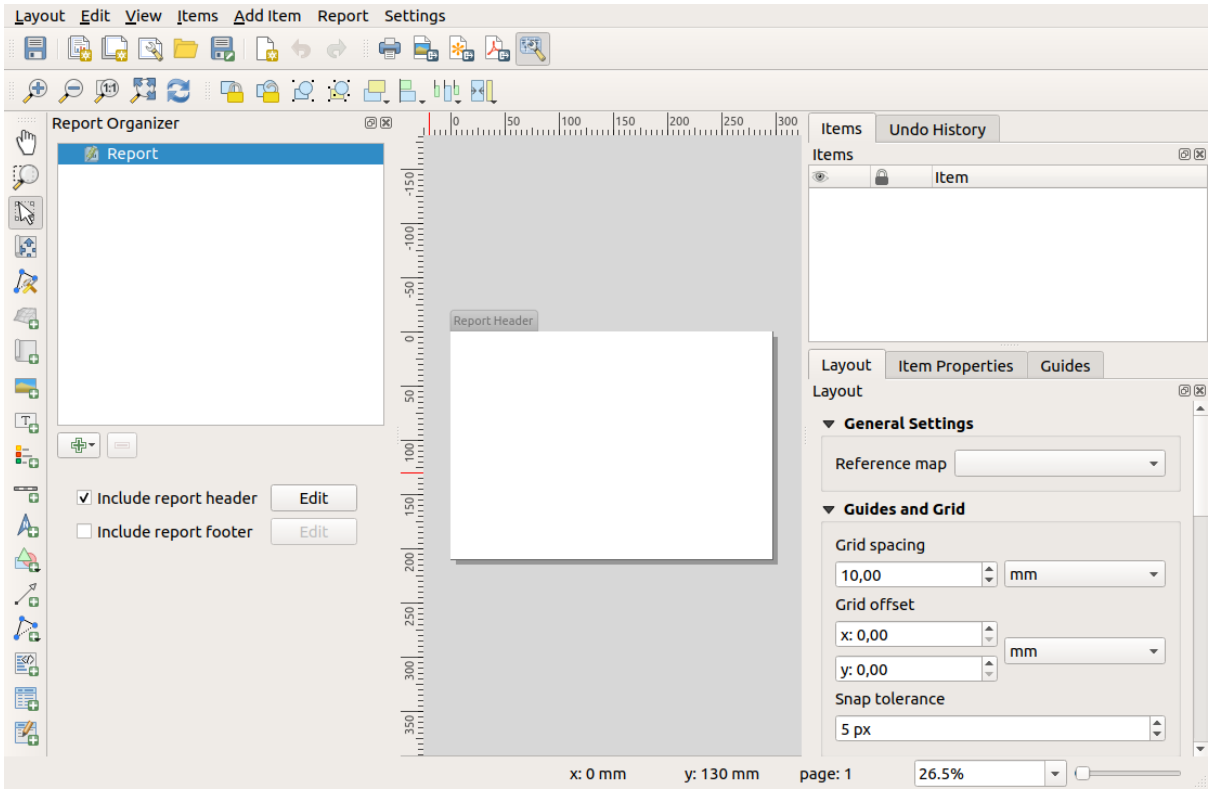


Using the *Project* ► *New Report* command, we create a blank report. Initially, there is not much to look at –the dialog which is displayed looks much like the print layout designer, except for the *Report Organizer* panel to the left:



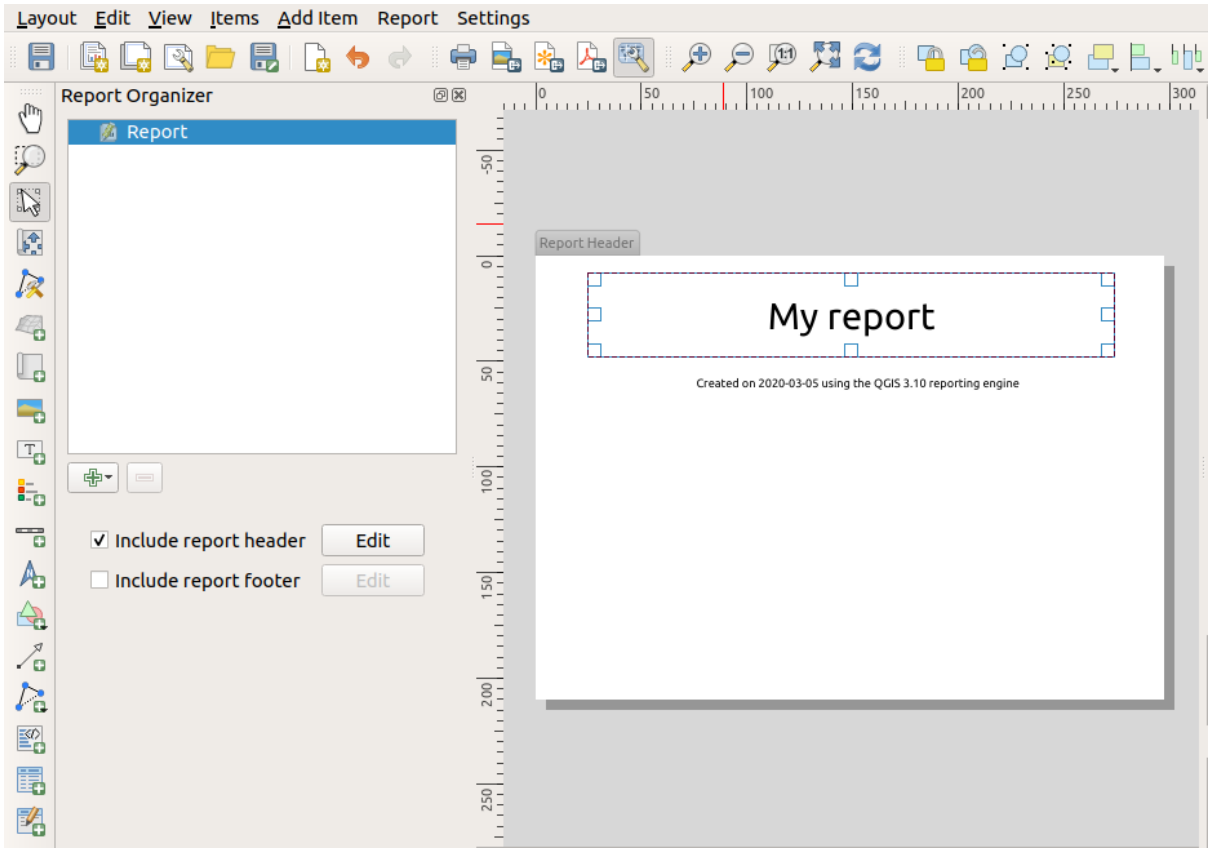
18.4.3 Layout Report Workspace

QGIS reports can consist of multiple, nested sections. In our new blank report we initially only have the main report section. The only options for this report section is *Include report header* and *Include report footer*. If we enable these options, a header will be included as the first page(s) (individual parts of reports can be multi-page if desired) in the report, and a footer will constitute the last page(s). Enable the header (*Include report header*), and hit the *Edit* button next to it:

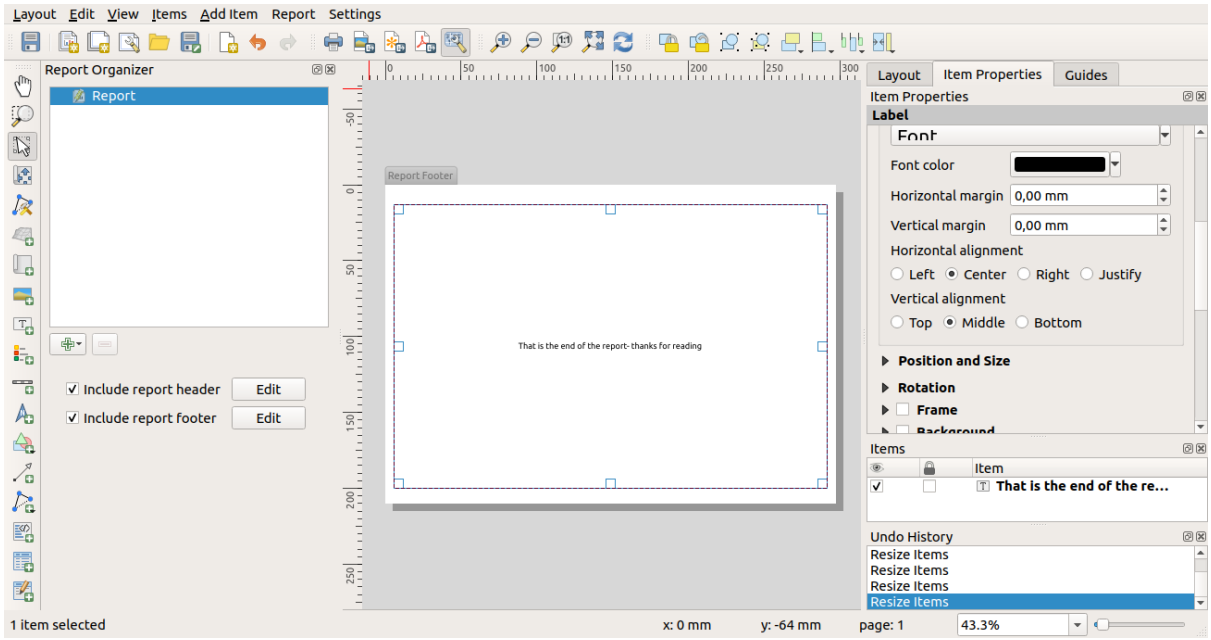


A few things happen as a result. Firstly, an edit pencil is shown next to *Report* in the *Report Organizer*, indicating that the report section is currently being edited in the designer. We also see a new page with a small *Report Header* title. The page has *landscape* orientation by default, but this (and other properties of the page) can be changed by right-clicking on the page and choosing *Page properties*. This will bring up the *Item properties* tab for the page, and page *Size*, *Width*, *Height*, and more can be specified.

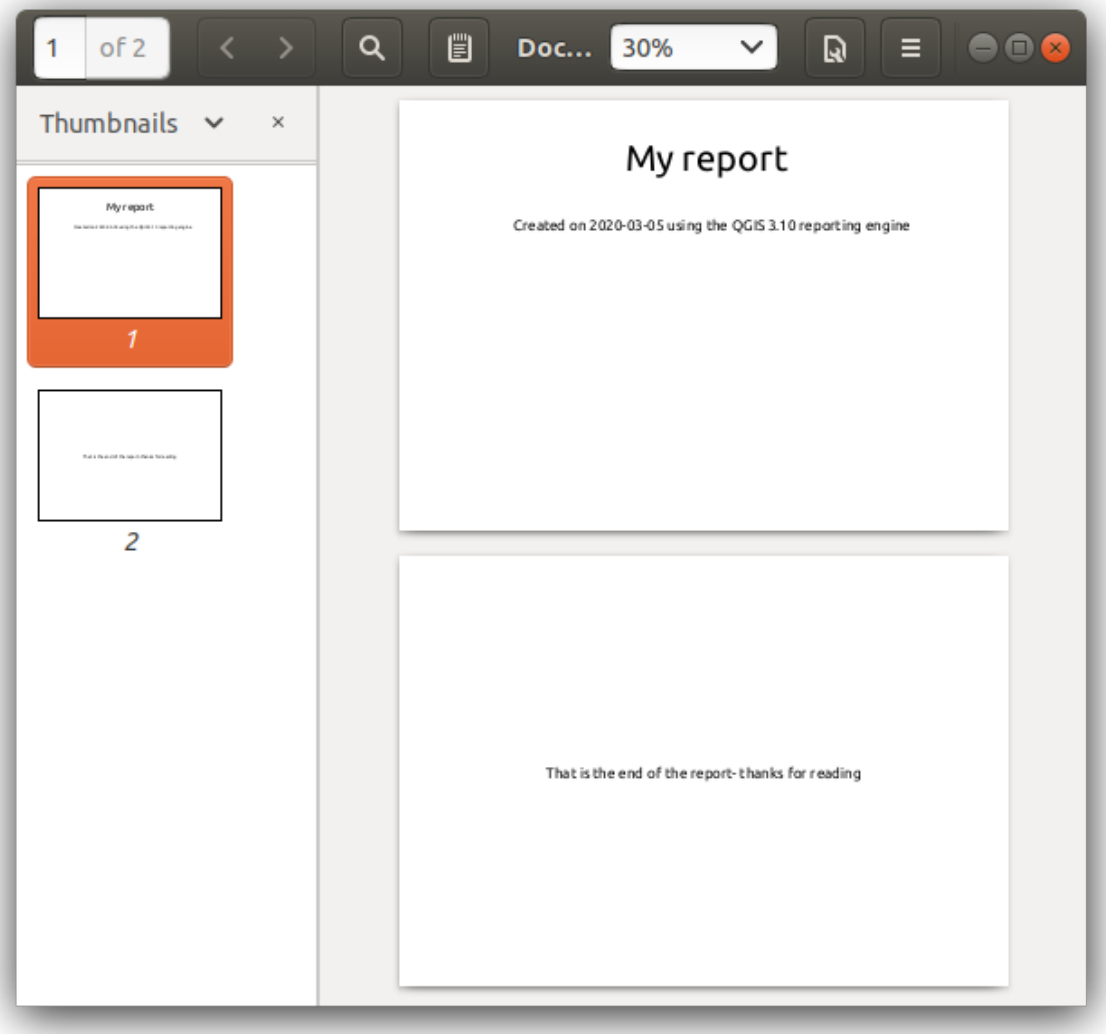
In QGIS reports, every component of the report is made up of individual layouts. They can be created and modified using the same tools as for standard print layouts –so you can use any desired combination of labels, pictures, maps, tables, etc. Let us add some items to our report header to demonstrate:



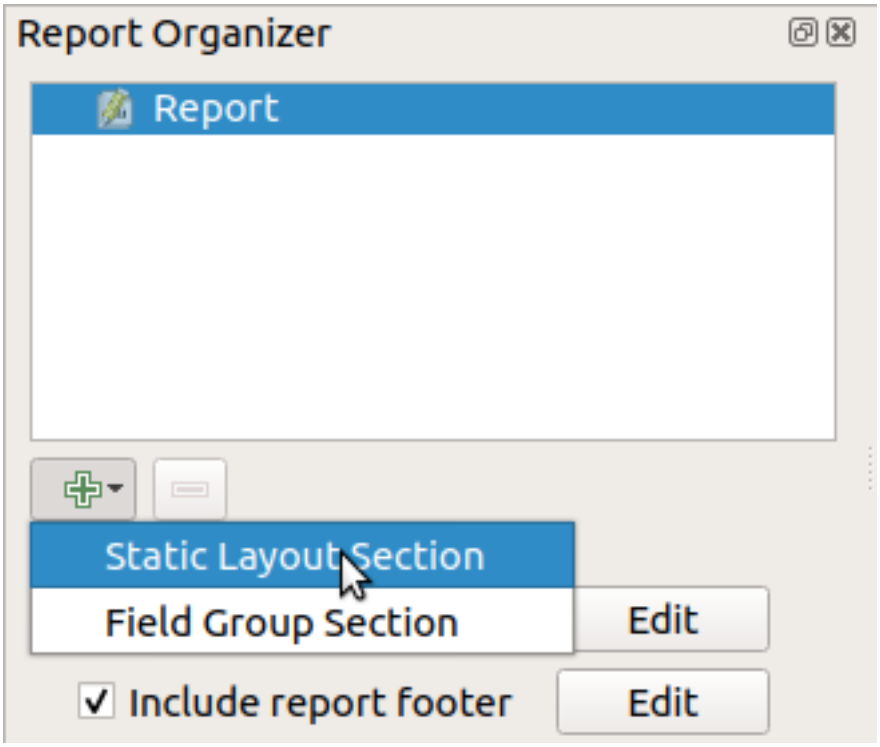
We will also create a simple footer for the report by checking the *Include report footer* option and hitting *Edit*.



Before proceeding further, let us export this report and see what we get. Exporting is done from the *Report* menu –in this case we select *Export Report as PDF...* to render the whole report to a PDF file. Here is the not-very-impressive result –a two page PDF consisting of our header and footer:



Let us make things more interesting. By hitting the Add Section button in the *Report Organizer*, we are given a choice of new sections to add to our report.

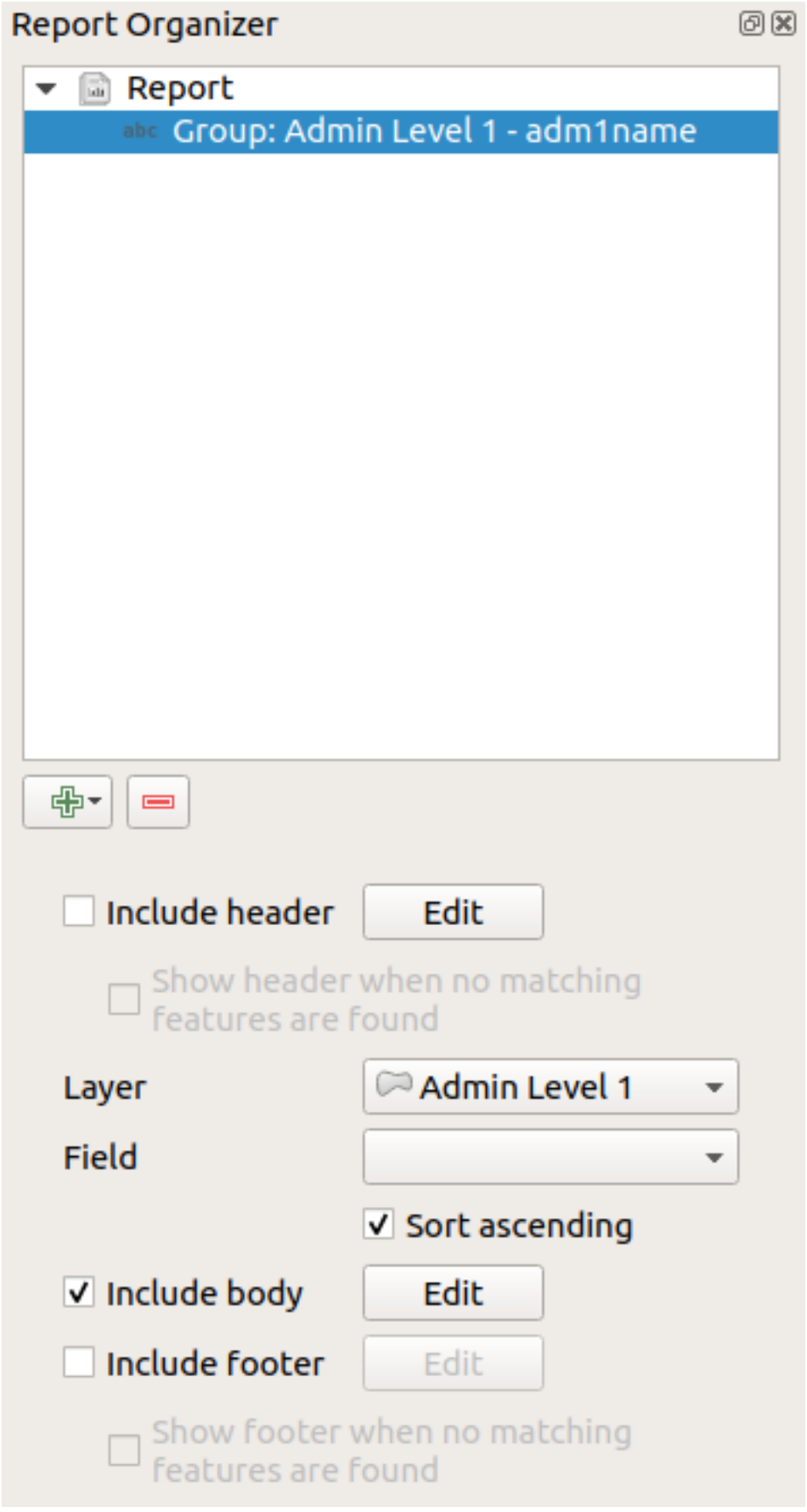


There are two options: *Static Layout Section* and *Field Group Section*.

The *Add Static Layout Section* is a single, static body layout. This can be used to embed static layouts mid-way through a report.

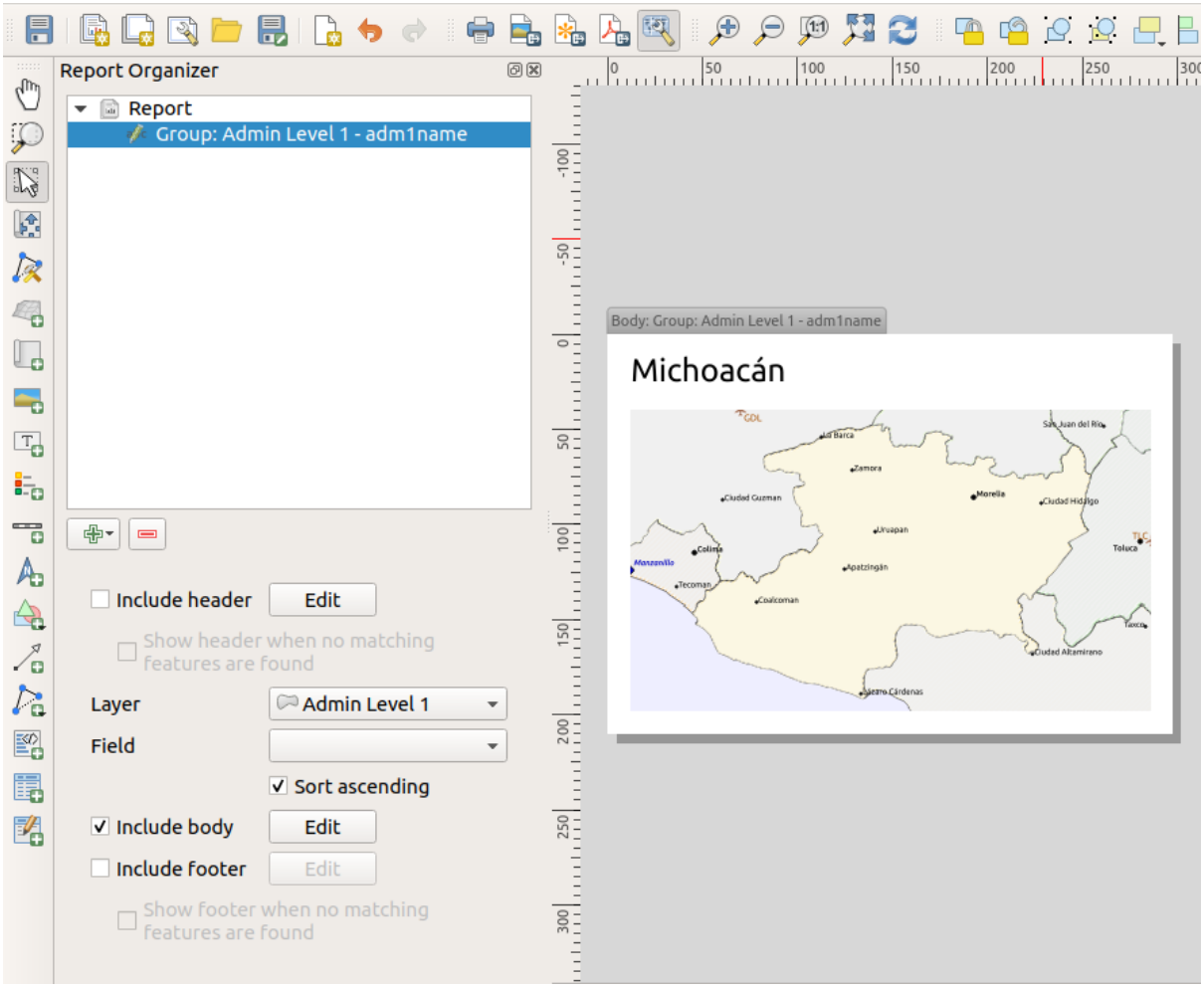
The *Field Group Section* repeats its body layout for every feature of a layer. The features are sorted by the selected grouping feature (with an option for ascending/descending sort). If a field group section has child sections (e.g. another field group section with a different field), then only features with unique values for the group feature are iterated over. This allows nested reports.

For now we will add a *Field Group Section* to our report. At its most basic level, you can think of a *Field Group Section* as the equivalent of a *print atlas*: you select a layer to iterate over, and the report will insert a section for each feature found. Selecting the new *Field Group Section* reveals a number of new related settings:



In this case we've setup our Field Group so that we iterate over all the states from the *Admin Level 1* layer, using

the values from the *adm1name* field. The same options to include header and footer are present, together with a new option to include a *body* for this section. We' ll do that, and edit the body:

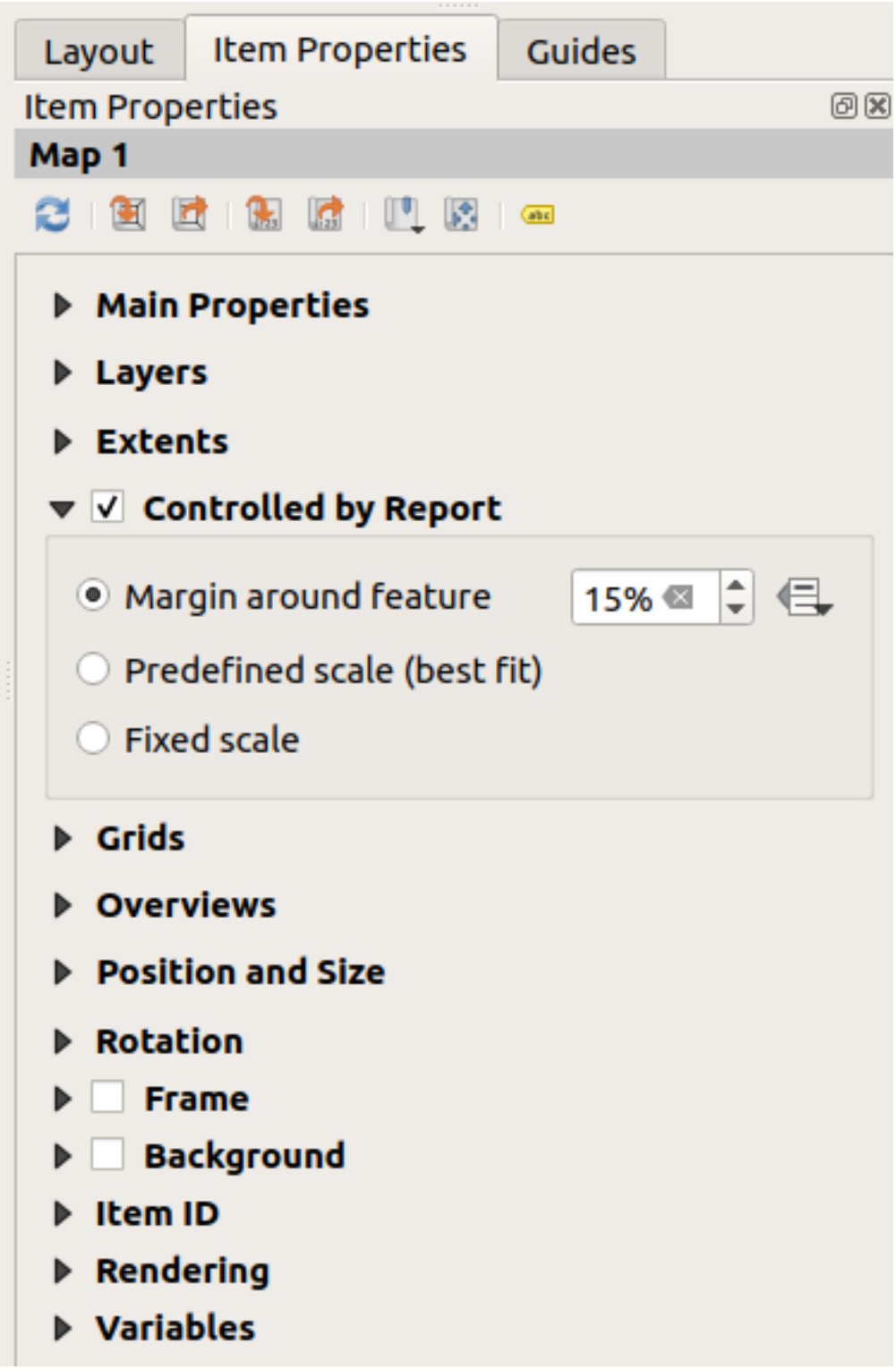


Our body now consists of a map and a label showing the name of the state. To include the name of the state, we selected *Add Item* ► *Add Label* and data defined the text under *Main Properties* with the help of *Insert or Edit an Expression...*.

The result was the following expression (*name* is the name of the attribute in the *Admin Level 1* layer that contains the name of the state):

```
[% "name" %]
```

The map is set to follow the current report feature (enabled by checking *Controlled by Report* –just like a map item in an atlas will follow the current atlas feature when *Controlled by Atlas* is checked):



If we went ahead and exported our report now, we'd get something like this:

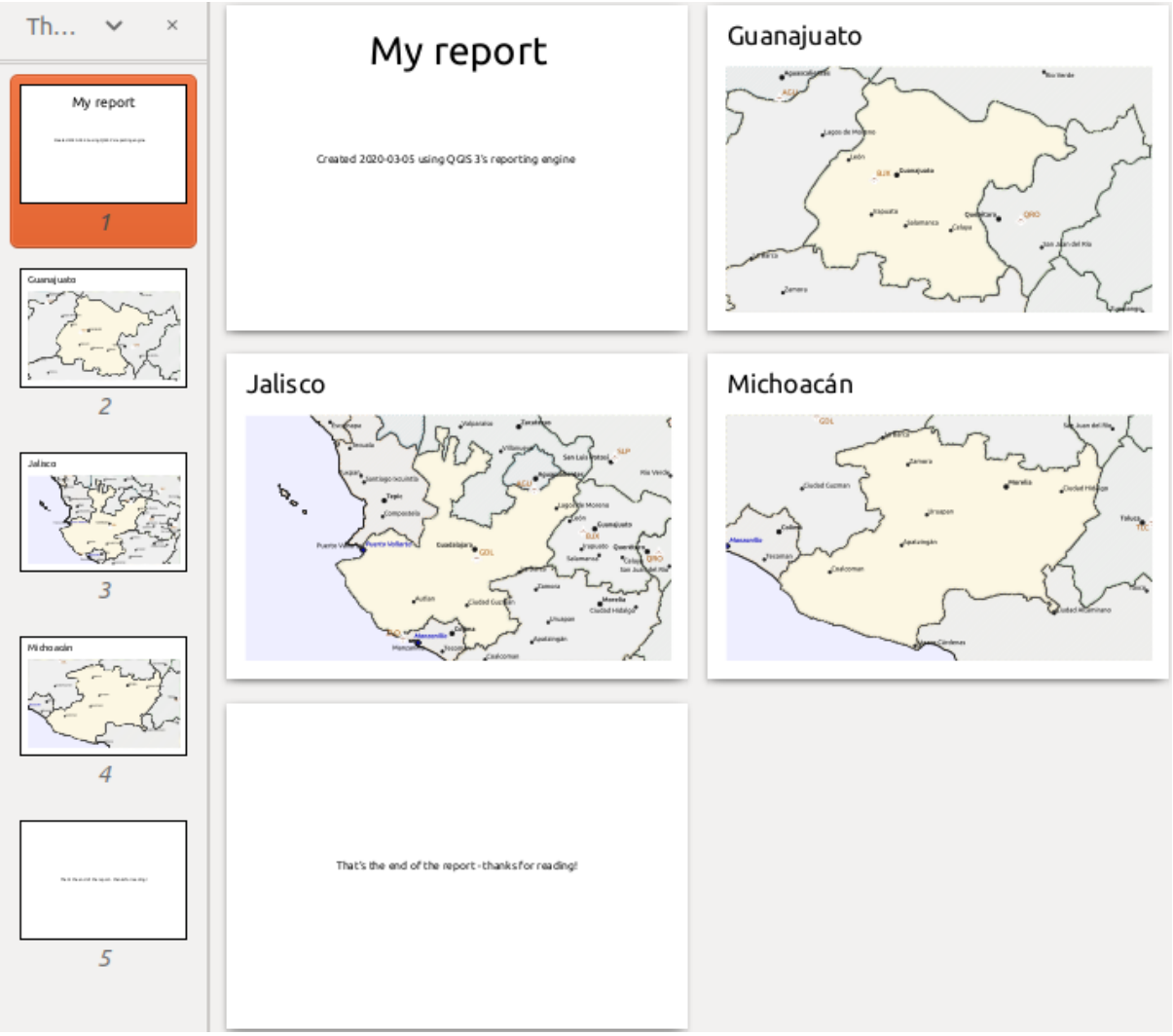
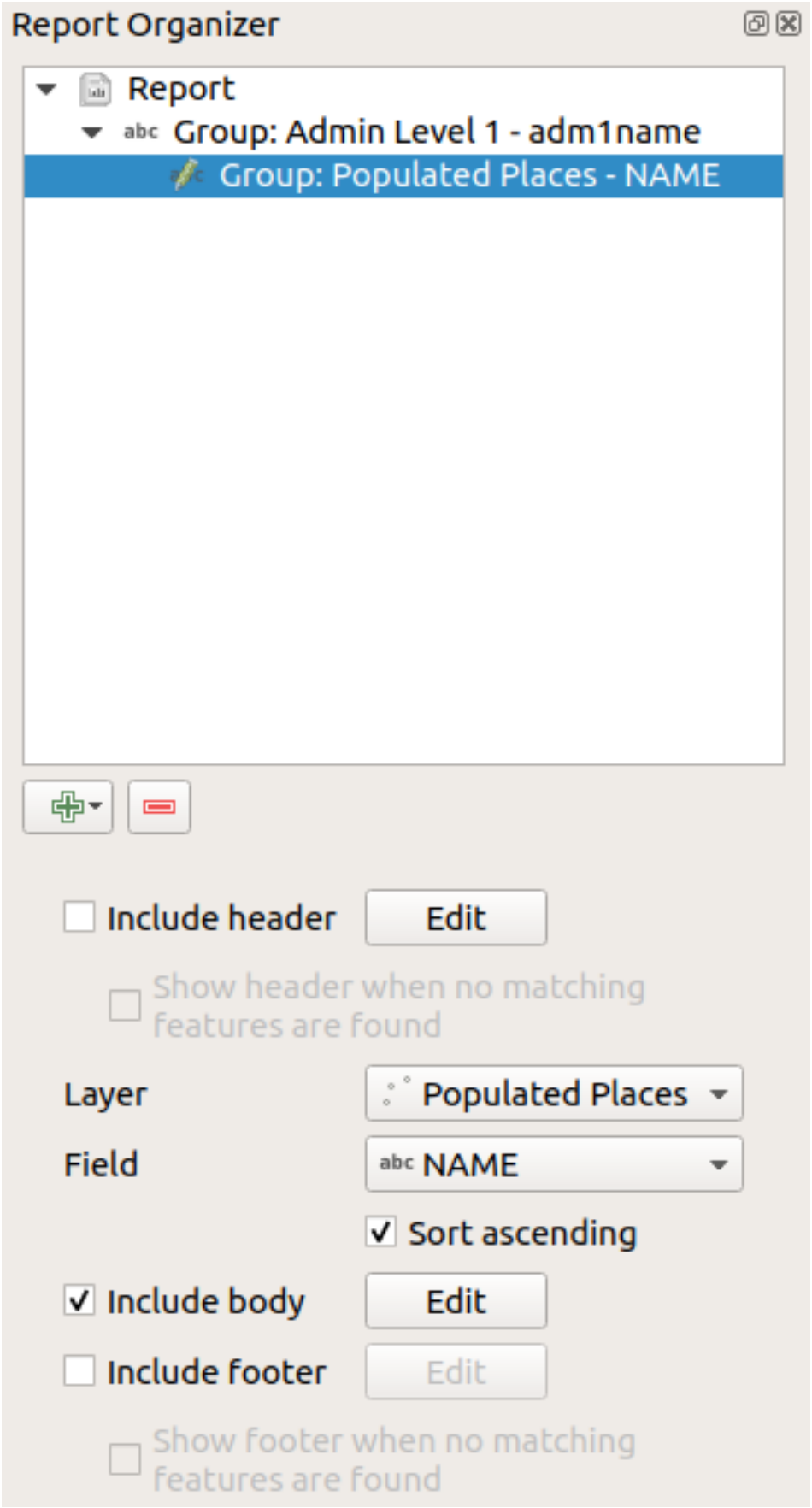


Fig. 18.70: The report header, a page for each state, and the report footer.

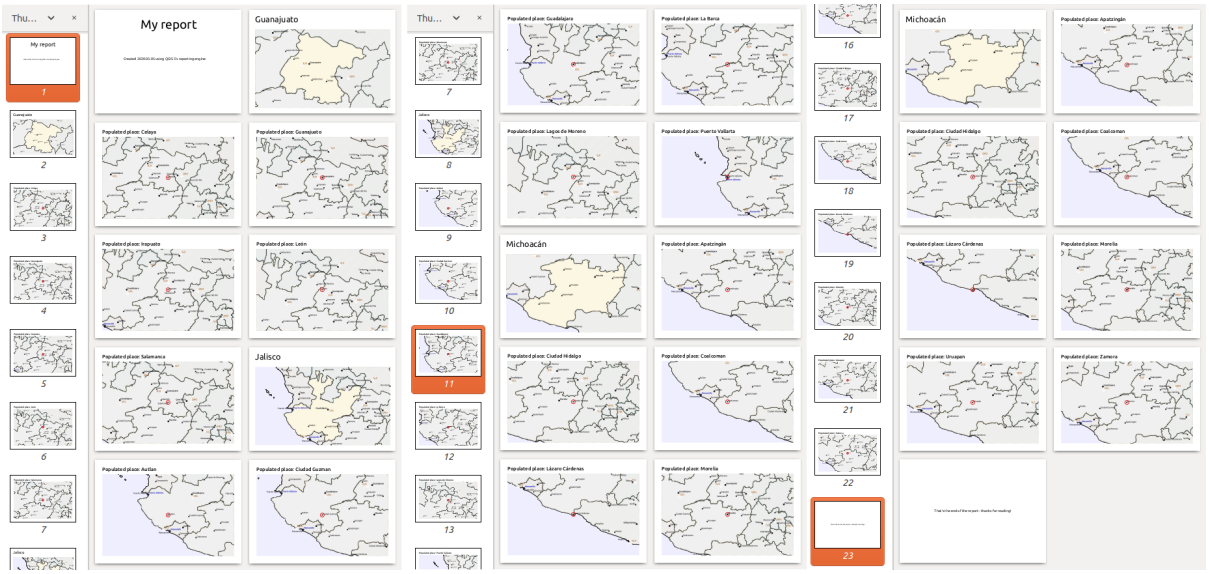
So more or less an atlas, but with a header and footer page.

Let us make things more interesting by adding a subsection to our state group. We do this by first selecting the *Admin Level 1* field group in the organizer, then hitting the Add Field button and adding a new *Field Group Section*:



When iterating over the features of a *Field Group Section*, the features will be filtered to match the defining field

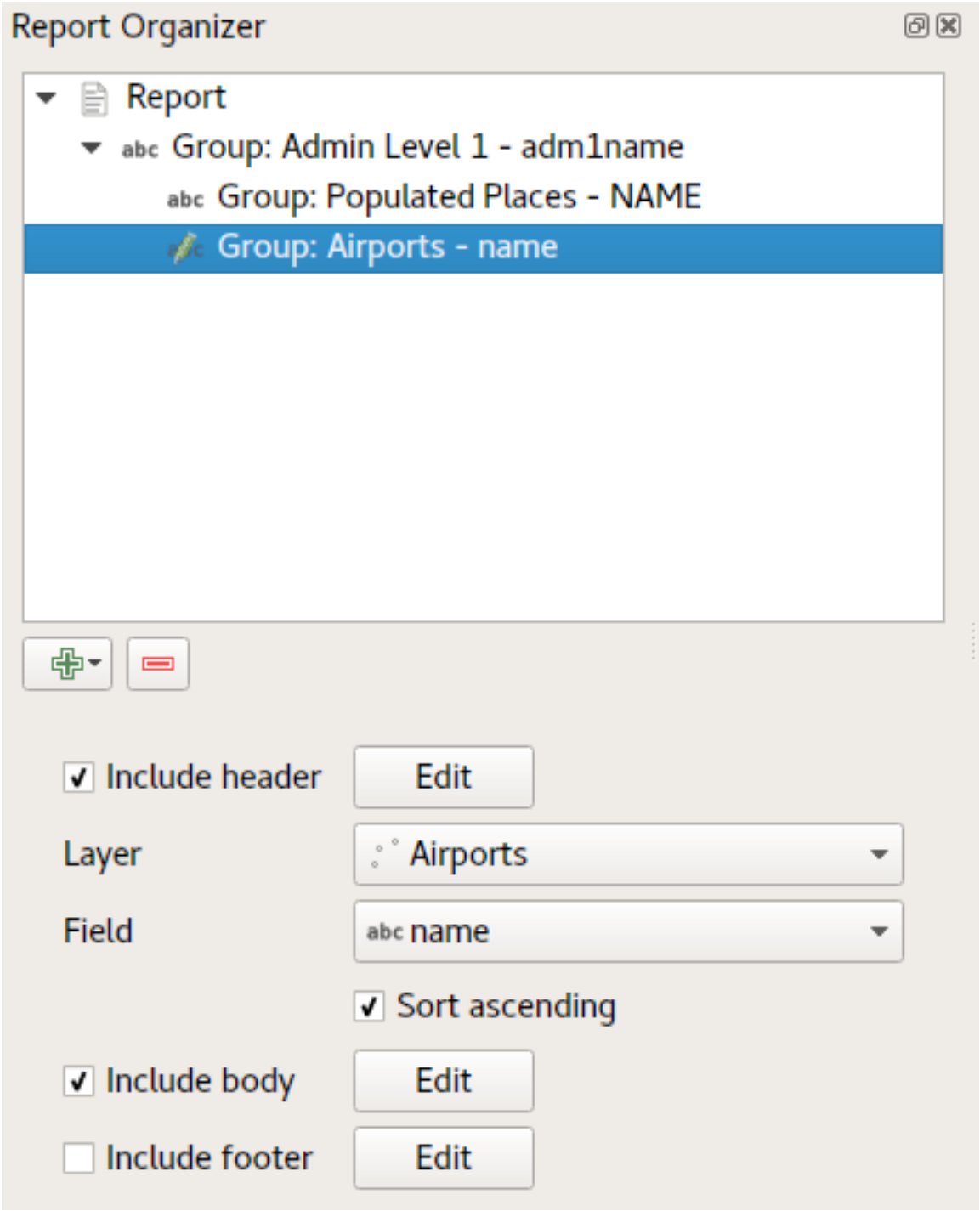
of its parent group (`adm1name` in this case). Here, the subsection we added will iterate over a *Populated Places* layer, including a body section for each place encountered. The magic here is that the *Populated Places* layer has an attribute with the same name as the defining field in the parent layer, *adm1name*, tagging each place with the state it is contained within (if you' re lucky your data will already be structured like this –if not, run the *Join Attributes by Location* Processing algorithm and create your own field). When we export this report, QGIS will grab the first state from the *Admin Level 1* layer, and then iterate over all the *Populated Places* with a matching *adm1name* value. Here' s what we get:



Here we created a basic body for the Populated Places group, including a map of the place and a table of some place attributes. So our report is now a report header, a page for the first state, followed by a page for every populated place within that state, then the rest of the states with their populated places, and finally the report footer. If we were to add a header for the Populated Places group, it would be included just before listing the populated places for each state, as shown in the illustration below.

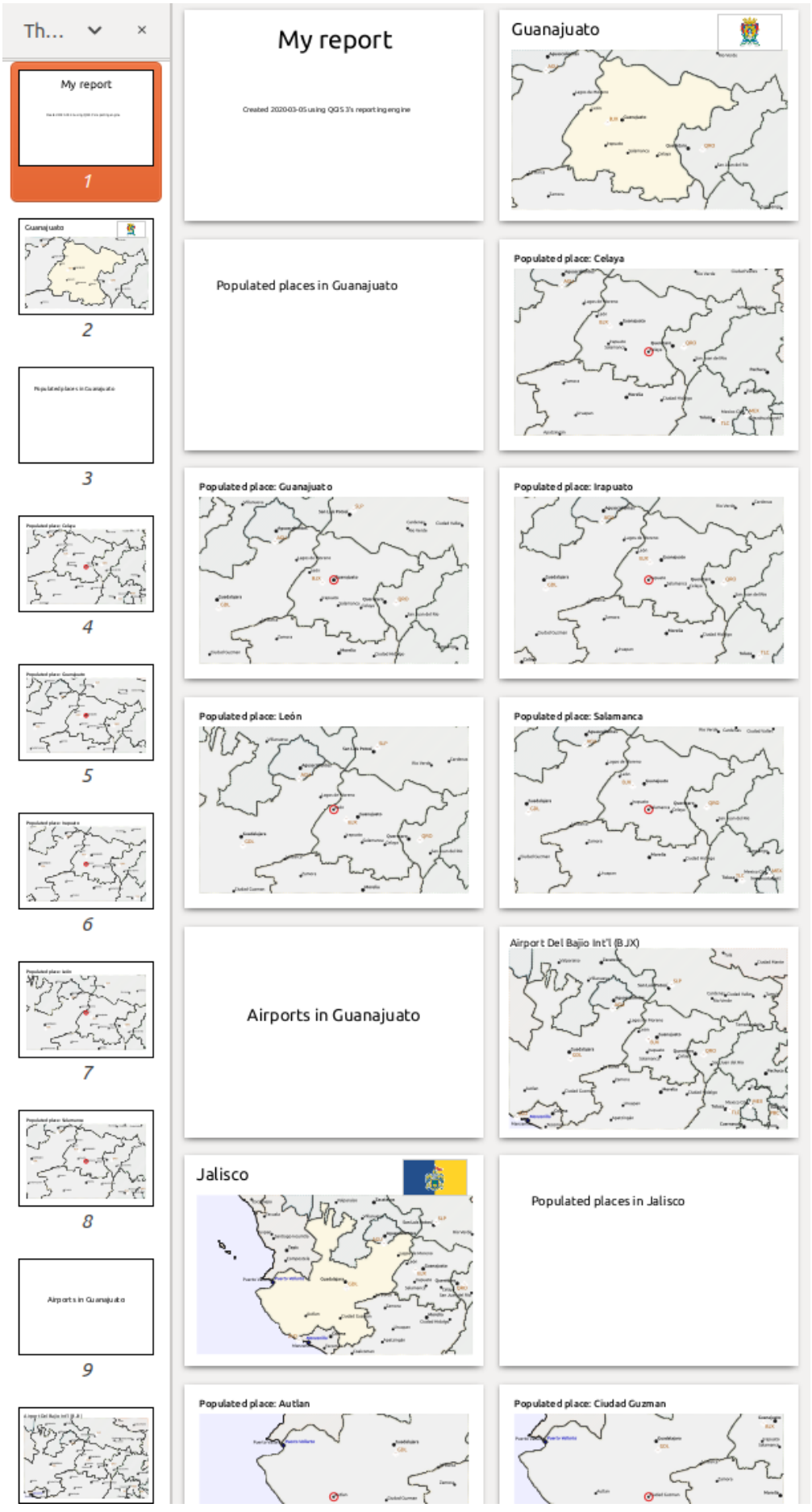
Similarly, a footer for the Populated Places group would be inserted after the final place for each state is included.

In addition to nested subsections, subsections in a report can also be included consecutively. If we add a second subsection to the *Admin Level 1 group* for *Airports*, then (if the *Airports* layer has an attribute *adm1name* that can link it to the parent group) our report will first list ALL the populated places for each state, followed by all the airports within that state, before proceeding to the next state.



The key point here is that our *Airports group* is a subsection of the *Admin Level 1 group* –not the *Populated Places group*.

In this case our report would be structured like this (note that state flags have also been included - the procedure for adding feature specific pictures in this way is described below):


Th...
My report
1
2
3
4
5
6
7
8
9
My report
Created 2020-03-05 using QGIS 3's reporting engine
Guanajuato
Populated places in Guanajuato
Populated place: Celaya
Populated place: Guanajuato
Populated place: Irapuato
Populated place: León
Populated place: Salamanca
Airports in Guanajuato
Airport Del Bajio Int'l (BJX)
Jalisco
Populated places in Jalisco
Populated place: Autlan
Populated place: Ciudad Guzman

Including pictures in a report

Pictures can be quite useful in reports, and QGIS allows pictures in both the static and dynamic parts of a report. Pictures are added in the same way as for standard print layouts, and for the static report parts (and static pictures in dynamic parts) there is not more to it.

But if you want illustrations that are tailored to the report features, your layer must have an attribute that can be used to define the picture to include.

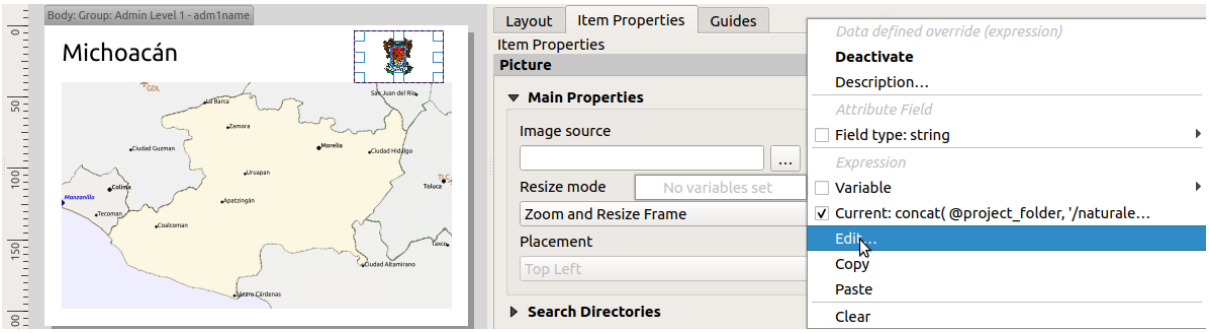
QGIS depends on absolute file names for images in reports.

For dynamic pictures, you first add a picture to the body part of the group, as usual. In the *Item properties* of the picture, you set the *Image Source* using the Data defined override button, and either select an attribute that contains the absolute path of the images or *Edit...* (to enter an expression that generates the absolute image path).

Below is an example expression that uses string concatenation to specify the absolute path to the pictures, using the directory where the project file is located (`@project_path`) and an attribute (`adm1name`) from which the file name is generated (in this case by transforming the string in the `adm1name` attribute to uppercase, and appending '_flag.png'):

```
concat(@project_folder, '/naturalearth/pictures/' ,
       upper("adm1name"), '_flag.png')
```

This means that the pictures are located in the `naturalearth/pictures` subdirectory of the project file directory.



Highlighting the current report feature in a map

In the above report, the report features are emphasized in the maps using highlighting (state) and circles (populated places). To emphasize the report features in the maps (apart from placing them at the centre of the maps), you must data define the style using a comparison between its `@id` and the `@atlas_featureid`, as for atlases.

For instance, if you would like to use a thicker line / border for the report feature than the other features you can data define the line width:

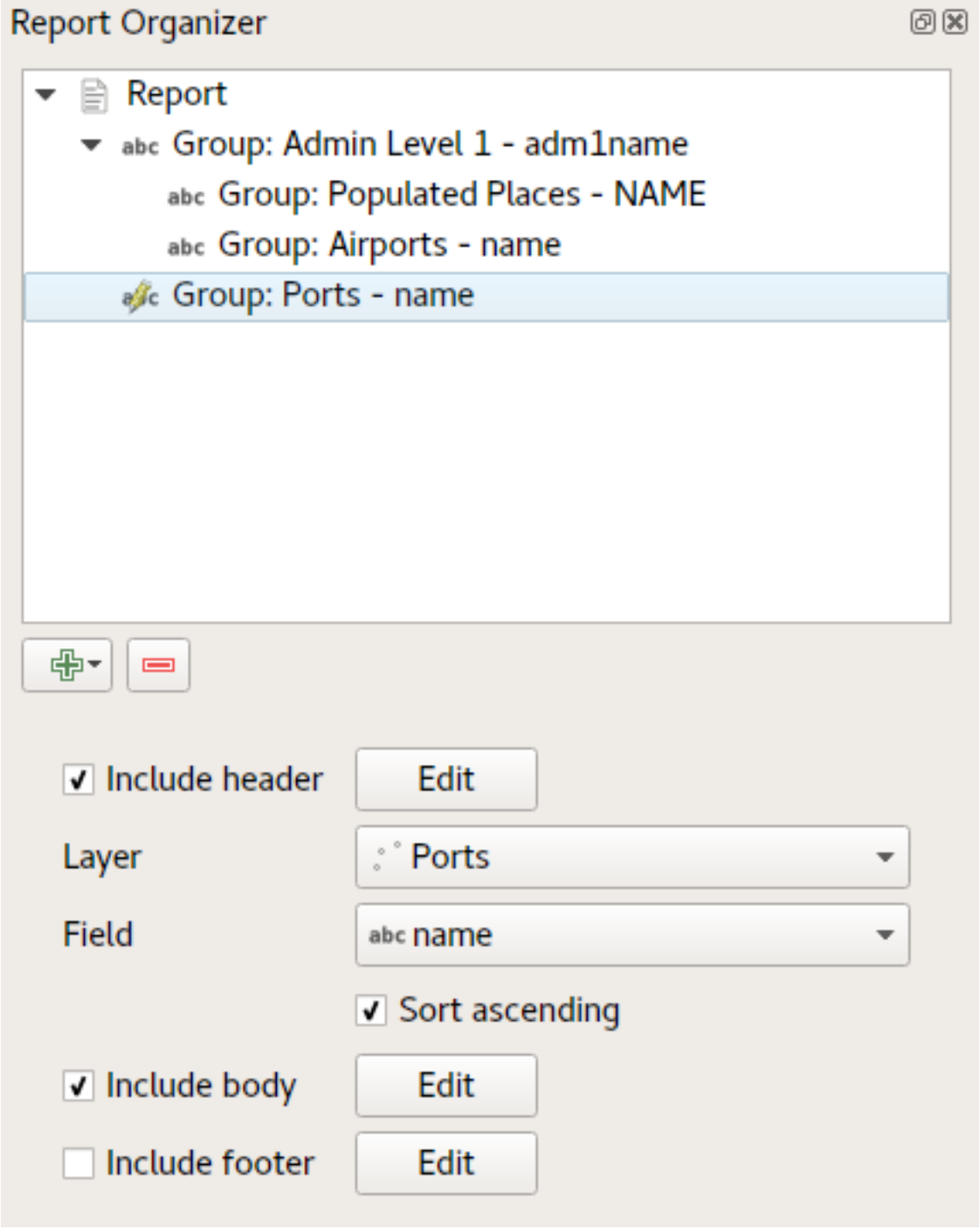
```
if($id=@atlas_featureid, 2.0, 0.1)
```

The report feature will get a 2 units wide polygon outline, while all other features will get a 0.1 units wide line. It is also possible to data define the colour (non-transparent dark magenta for the report feature and semi-transparent light gray for the other features):

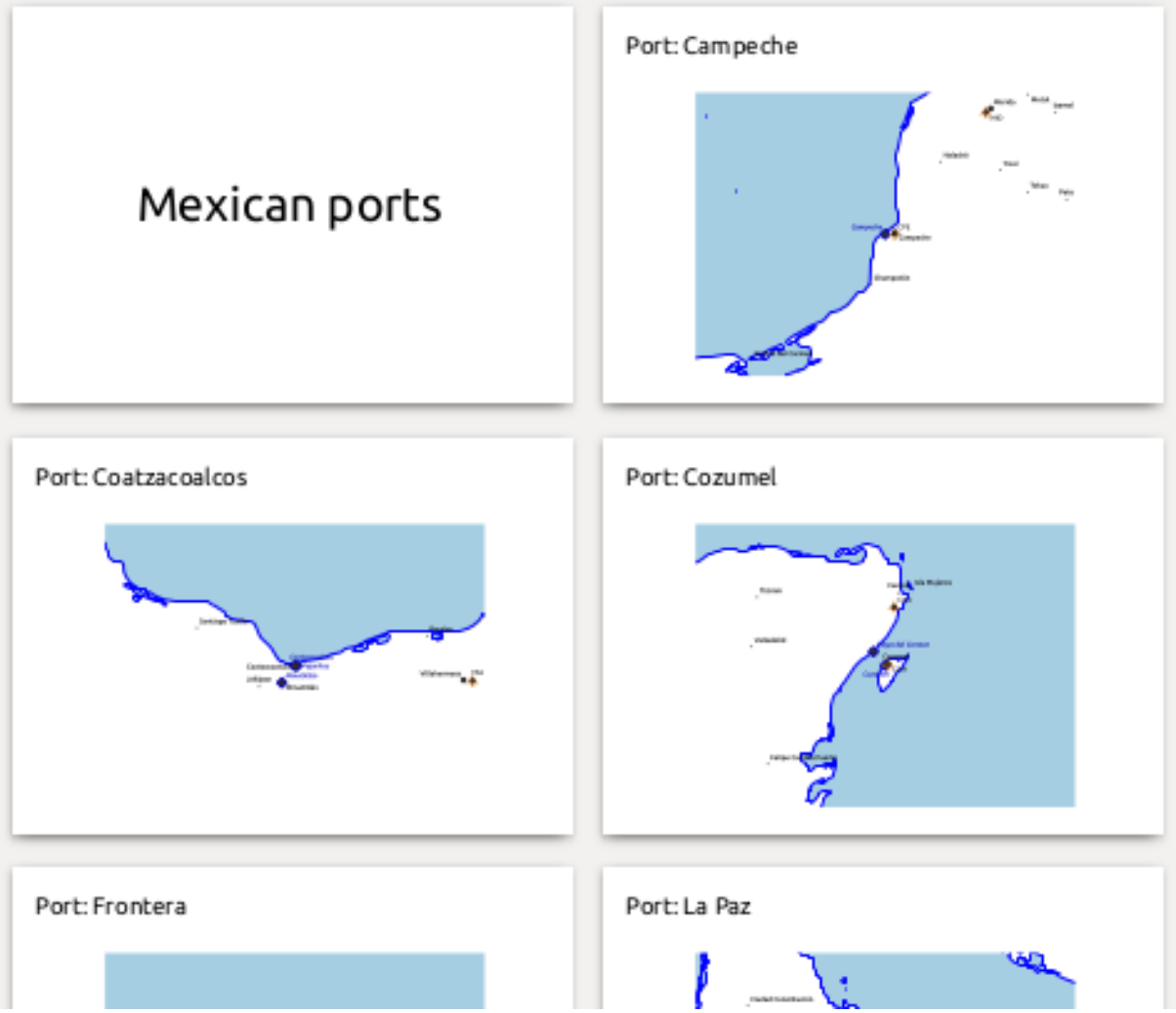
```
if($id=@atlas_featureid, '#FF880088', '#88CCCCCC')
```

More level 1 groups

Combining nested and consecutive sections, together with section headers and footers allows for tons of flexibility. For instance, in the below report we add another field group as a child of the main report for the *Ports* layer. Now, after listing the states together with their populated places and airports, we'll get a summary list of all the ports in the region:



This results in the last part of our report exporting as:



18.4.4 Export settings

When you export a report (*Report* ► *Export Report as Images... / SVG... / PDF...*), you will be asked for a file name, and then you get the opportunity to tune the export settings to get the most appropriate output.

As you see, reports in QGIS are extremely powerful and flexible!

> **Note**
>
> The current information was adapted from a North Road blog, Exploring Reports in QGIS 3.0 - the Ultimate Guide!

CHAPTER 19

Working with OGC / ISO protocols

The Open Geospatial Consortium (OGC) is an international organization with membership of more than 300 commercial, governmental, nonprofit and research organizations worldwide. Its members develop and implement standards for geospatial content and services, GIS data processing and exchange.

Describing a basic data model for geographic features, an increasing number of specifications are developed by OGC to serve specific needs for interoperable location and geospatial technology, including GIS. Further information can be found at https://www.ogc.org/.

Important OGC specifications supported by QGIS are:

- **WMS** --- Web Map Service (*WMS/WMTS Client*)
- **WMTS** --- Web Map Tile Service (*WMS/WMTS Client*)
- **WFS** --- Web Feature Service (*WFS and WFS-T Client*)
- **WFS-T** --- Web Feature Service - Transactional (*WFS and WFS-T Client*)
- **WCS** --- Web Coverage Service (*WCS Client*)
- **WPS** --- Web Processing Service
- **CSW** --- Catalog Service for the Web
- **SFS** --- Simple Features for SQL (*PostgreSQL Database*)
- **GML** --- Geography Markup Language
- **SensorThings API** --- SensorThings API (*SensorThings*)

OGC services are increasingly being used to exchange geospatial data between different GIS implementations and data stores. QGIS can deal with the above specifications as a client, being **SFS** (through support of the PostgreSQL data provider, see section *PostgreSQL Database*).

You can also share your maps and data through the WMS, WMTS, WFS, WFS-T and WCS protocols using a webserver with QGIS Server, UMN MapServer or GeoServer installed.

19.1 WMS/WMTS Client

19.1.1 Overview of WMS Support

QGIS currently can act as a WMS client that understands WMS 1.1, 1.1.1 and 1.3 servers. In particular, it has been tested against publicly accessible servers such as DEMIS.

A WMS server acts upon requests by the client (e.g., QGIS) for a raster map with a given extent, set of layers, symbolization style, and transparency. The WMS server then consults its local data sources, rasterizes the map, and sends it back to the client in a raster format. For QGIS, this format would typically be JPEG or PNG.

WMS is generically a REST (Representational State Transfer) service rather than a full-blown Web service. As such, you can actually take the URLs generated by QGIS and use them in a web browser to retrieve the same images that QGIS uses internally. This can be useful for troubleshooting, as there are several brands of WMS server on the market and they all have their own interpretation of the WMS standard.

WMS layers can be added quite simply, as long as you know the URL to access the WMS server, you have a serviceable connection to that server, and the server understands HTTP as the data transport mechanism.

Additionally, QGIS will cache your WMS responses (i.e. images) for 24h as long as the GetCapabilities request is not triggered. The GetCapabilities request is triggered every time the *Connect* button in the *WMS/WMTS* dialog is used to retrieve the WMS server capabilities. This is an automatic feature meant to optimize project loading time. If a project is saved with a WMS layer, the corresponding WMS tiles will be loaded from the cache the next time the project is opened as long as they are not older than 24h.

19.1.2 Overview of WMTS Support

QGIS can also act as a WMTS client. WMTS is an OGC standard for distributing tile sets of geospatial data. This is a faster and more efficient way of distributing data than WMS because with WMTS, the tile sets are pre-generated, and the client only requests the transmission of the tiles, not their production. A WMS request typically involves both the generation and transmission of the data. A well-known example of a non-OGC standard for viewing tiled geospatial data is Google Maps.

In order to display the data at a variety of scales close to what the user might want, the WMTS tile sets are produced at several different scale levels and are made available for the GIS client to request them.

This diagram illustrates the concept of tile sets:

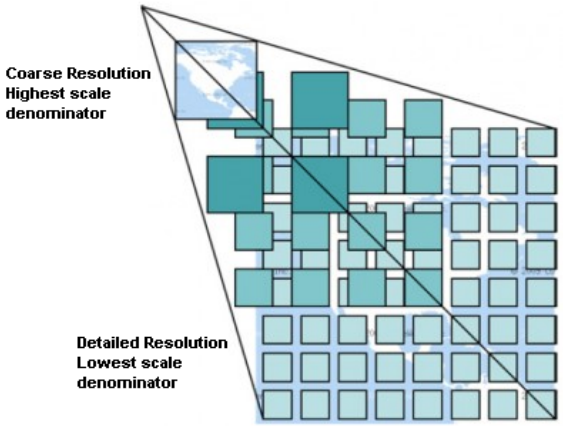


Fig. 19.1: Concept of WMTS tile sets

The two types of WMTS interfaces that QGIS supports are via Key-Value-Pairs (KVP) and RESTful. These two interfaces are different, and you need to specify them to QGIS differently.

1. In order to access a **WMTS KVP** service, a QGIS user must open the WMS/WMTS interface and add the following string to the URL of the WMTS tile service:

```
"?SERVICE=WMTS&REQUEST=GetCapabilities"
```

An example of this type of address is:

```
https://opencache.statkart.no/gatekeeper/gk/gk.open_wmts?service=WMTS&
↪request=GetCapabilities
```

For testing the topo2 layer in this WMTS works nicely. Adding this string indicates that a WMTS web service is to be used instead of a WMS service.

2. The **RESTful WMTS** service takes a different form, a straightforward URL. The format recommended by the OGC is:

```
{WMTSBaseURL}/1.0.0/WMTSCapabilities.xml
```

This format helps you to recognize that it is a RESTful address. A RESTful WMTS is accessed in QGIS by simply adding its address in the WMS setup in the URL field of the form. An example of this type of address for the case of an Austrian basemap is:

```
https://maps.wien.gv.at/basemap/1.0.0/WMTSCapabilities.xml
```

Note

You can still find some old services called WMS-C. These services are quite similar to WMTS (i.e., same purpose but working a little bit differently). You can manage them the same as you do WMTS services. Just add `?tiled=true` at the end of the url. See https://wiki.osgeo.org/wiki/Tile_Map_Service_Specification for more information about this specification.

When you read WMTS, you can often think WMS-C also.

19.1.3 Selecting WMS/WMTS Servers

The first time you use the WMS/WMTS feature in QGIS, there are no servers defined.

You then need to create connections to the server you are targeting:

1. Go to the *WMS/WMTS* tab of the *Data Source Manager* dialog, either by:
   - clicking the Open Data Source Manager button (or pressing `Ctrl+L`) and enabling the tab
   - clicking the Add WMS layer button on the *Manage Layers* toolbar
   - or selecting *Layer* ► *Add Layer* ► *Add WMS/WMTS Layer...* menu
2. Press *New* from the *Layers* tab. The *Create a New WMS/WMTS Connection...* dialog appears.

Tip

Right-click the *WMS/WMTS* entry from within the *Browser panel* and select *New Connection...* also opens the *Create a New WMS/WMTS Connection...* dialog.

3. Then enter the parameters to connect to your desired WMS server, as listed below:

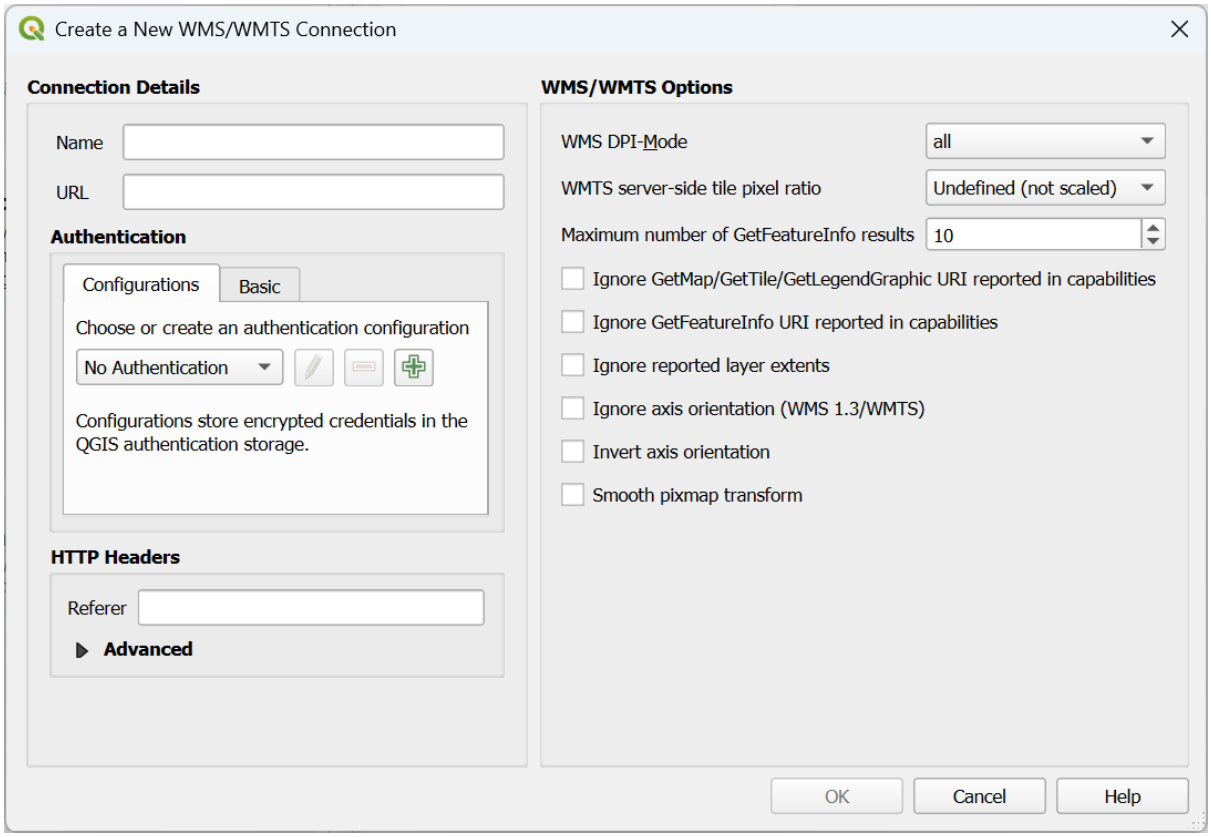


Fig. 19.2: Creating a connection to a WMS server

- *Name*: A name for the connection. This name will be used in the Server Connections drop-down box so that you can distinguish it from other WMS servers.
- *URL*: URL of the server providing the data. This must be a resolvable host name -- the same format as you would use to open a telnet connection or ping a host, i.e. the base URL only. For example, you shouldn't have fragments such as `request=GetCapabilities` or `version=1.0.0` in your URL.
- *Authentication* (optional): using a *stored configuration* or a basic authentication with *Username* and *Password*.

> **Warning**
>
> Entering **username** and **password** in the *Authentication* tab will keep unprotected credentials in the connection configuration. Those **credentials will be visible** if, for instance, you shared the project file with someone. Therefore, it's advisable to save your credentials in an *Authentication configuration* instead (*Configurations* tab). See 认证系统 for more details.

- HTTP *Referer*
- *WMS DPI-Mode*: Available options are **all**, **off**, **QGIS**, **UMN** and **GeoServer**
- *WMTS server-side tile pixel ratio*: When rendering WMTS layers, allows to scale up or down the tiles based on the device screen DPI. Available options are *Undefined (not scaled)*, *Standard (96 DPI)* and *High (192 DPI)*.
- *Maximum number of GetFeatureInfo results*: specifies a default value for the maximum number of results returned per layer by a GetFeatureInfo request using this connection (see FEATURE_COUNT parameter). Default value is `10`. Set to `0` to use server default value (usually `1`): no FEATURE_COUNT parameter will be added to the request.

- ☐ *Ignore GetMap/GetTile/GetLegendGraphic URI reported in capabilities*: if checked, use given URI from the *URL* field above.
- ☐ *Ignore GetFeatureInfo URI reported in capabilities*: if checked, use given URI from the *URL* field above.
- ☐ *Ignore reported layer extents*: because the extent reported by raster layers may be smaller than the actual area which can be rendered (notably for WMS servers with symbology which takes more space than the data extent), check this option to avoid cropping raster layers to their reported extents, resulting in truncated symbols on the borders of these layers.
- ☐ *Ignore axis orientation (WMS 1.3/WMTS)*
- ☐ *Invert axis orientation*
- ☐ *Smooth pixmap transformation*

4. 按 确定按钮

Once the new WMS/WMTS server connection has been created, it will be preserved for future QGIS sessions. Note that it is also possible to *Load* the connection parameters from a `.XML` file or *Save* them to a `.XML` file.

If you need to set up a proxy server to be able to receive WMS services from the internet, you can add your proxy server in the options. Choose *Settings* ► *Options* and click on the *Network* tab. There, you can add your proxy settings and enable them by setting ☑ *Use proxy for web access*. Make sure that you select the correct proxy type from the *Proxy type* [...▼] drop-down menu.

19.1.4 Loading WMS/WMTS Layers

Once you have successfully filled in your parameters, you can use the *Connect* button to retrieve the capabilities of the selected server. This includes the image encoding, layers, layer styles and projections. Since this is a network operation, the speed of the response depends on the quality of your network connection to the WMS server. While downloading data from the WMS server, the download progress is visualized in the lower left corner of the main QGIS dialog.

Your screen should now look a bit like Fig. 19.3, which shows the response provided by a WMS server.

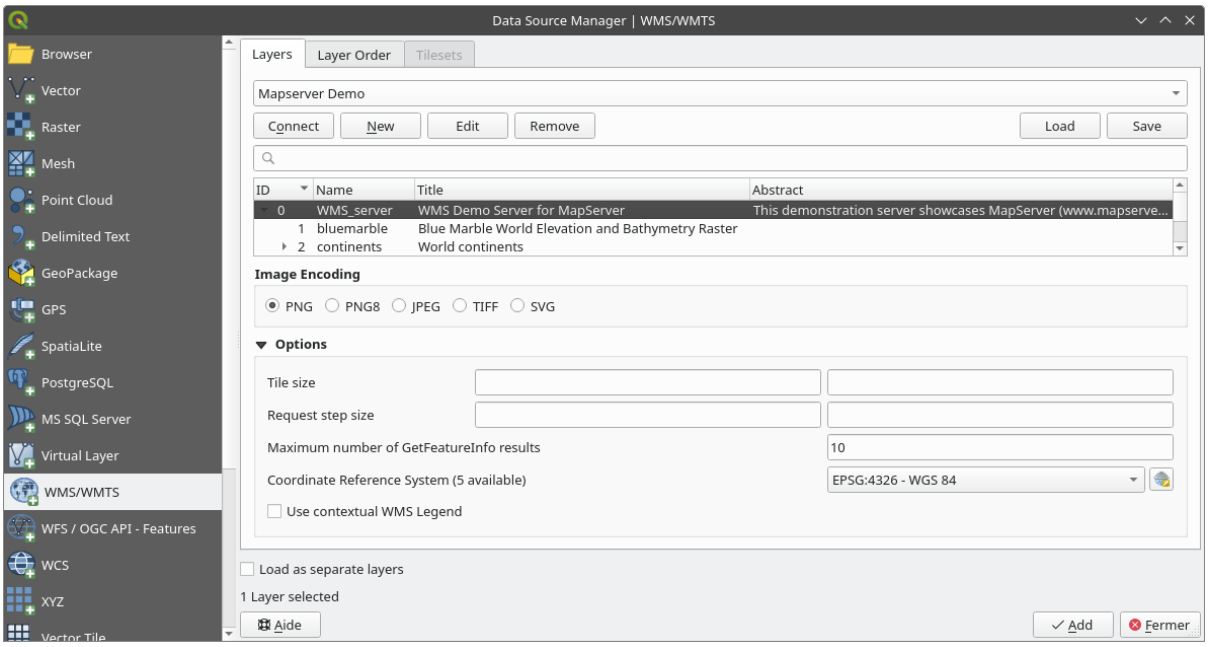


Fig. 19.3: Dialog for adding a WMS server, with filter on available layers

The upper part of the *Layers* tab of the dialog shows a tree structure that can include layer groups embedding layers with their associated image style(s) served by the server. Each item can be identified by:

- an *ID*
- *名称*
- a *Title*
- and an *Abstract*.

The list can be filtered using the widget in the top right corner.

Image Encoding

The *Image encoding* section lists the formats that are supported by both the client and server. Choose one depending on your image accuracy requirements.

> **Tip**
>
> **Image Encoding**
>
> You will typically find that a WMS server offers you the choice of JPEG or PNG image encoding. JPEG is a lossy compression format, whereas PNG faithfully reproduces the raw raster data.
>
> Use JPEG if you expect the WMS data to be photographic in nature and/or you don't mind some loss in picture quality. This trade-off typically reduces by five times the data transfer requirement compared with PNG.
>
> Use PNG if you want precise representations of the original data and you don't mind the increased data transfer requirements.

Options

The Options area of the dialog provides means to configure the WMS requests. You can define:

- *Tile size* if you want to set tile sizes (e.g., 256x256) to split up the WMS request into multiple requests.
- *Request step size*: if you want to reduce the effect of cut labels at tile borders, increasing the step size creates larger requests, fewer tiles and fewer borders. The default value is 2000.
- The *Maximum number of GetFeatureInfo results*: specifies the maximum number of results returned by a GetFeatureInfo request, for the layer(s) being loaded. Allows to override the *connection's default value* for specific layers.
- Each WMS layer can be presented in multiple CRSs, depending on the capability of the WMS server. If you select a WMS from the list, a field with the default projection provided by the web server appears. Press the Select CRS widget to replace the default projection of the WMS with another CRS supported by the WMS server.

  A dialog similar to the one shown in Fig. 6.4 will appear. The main difference with the WMS version of the dialog is that only those CRSs supported by the WMS server will be shown.
- Finally you can activate *Use contextual WMS Legend* if the WMS Server supports this feature. Then only the relevant legend for your current map view extent will be shown and thus will not include legend items for items you can't see in the current map.

You can select several layers at once, but only one image style per layer. When several layers are selected, they will be combined at the WMS server and transmitted to QGIS in one go, as a single layer. The default name is a slash (/) separated list of their original title. You can however opt to *Load as separate layers*, in which case each layer is added to the map canvas using its default name from the queried service.

Layer Order

The *Layer Order* tab lists the selected layers available from the current connected WMS server.

WMS layers rendered by a server are overlaid in the order listed in the *Layers* tab, from top to bottom of the list. If you want to change the overlay order, you can use the *Up* and *Down* buttons of the *Layer Order* tab.

Transparency

The *Global transparency* setting from the *Layer Properties* is hard coded to be always on, where available.

19.1.5 图块集

When using WMTS (Cached WMS) services you are able to browse through the *Tilesets* tab given by the server. Additional information like tile size, formats and supported CRS are listed in this table.

```
# example of WMTS service

https://opencache.statkart.no/gatekeeper/gk/gk.open_wmts?service=WMTS&
↪request=GetCapabilities
```

Selecting a layer to load, it is also possible to apply an *Interpretation method*, converting it into a raster layer of single band float type, ready for styling using QGIS usual *raster renderers*.

In combination with this feature, you can use the tile scale slider by selecting *View* ► *Panels* (or *Settings* ► *Panels*), then choosing *Tile Scale Panel*. This gives you the available scales from the tile server with a nice slider docked in.

19.1.6 Using the Identify Tool

Once you have added a WMS server, and if any layer from a WMS server is queryable, you can then use the Identify tool to select a pixel on the map canvas. A query is made to the WMS server for each selection made. The results of the query are returned in plain text. The formatting of this text is dependent on the particular WMS server used.

Format selection

If multiple output formats are supported by the server, a combo box with supported formats is automatically added to the identify results dialog and the selected format may be stored in the project for the layer.

GML format support

The Identify tool supports WMS server response (GetFeatureInfo) in GML format (it is called Feature in the QGIS GUI in this context). If "Feature" format is supported by the server and selected, results of the Identify tool are vector features, as from a regular vector layer. When a single feature is selected in the tree, it is highlighted in the map and it can be copied to the clipboard and pasted to another vector layer. See the example setup of the UMN Mapserver below to support GetFeatureInfo in GML format.

```
# in layer METADATA add which fields should be included and define geometry␣
↪(example):

"gml_include_items"   "all"
"ows_geometries"      "mygeom"
"ows_mygeom_type"     "polygon"

# Then there are two possibilities/formats available, see a) and b):

# a) basic (output is generated by Mapserver and does not contain XSD)
# in WEB METADATA define formats (example):
"wms_getfeatureinfo_formatlist" "application/vnd.ogc.gml,text/html"

# b) using OGR (output is generated by OGR, it is sent as multipart and contains␣
↪XSD)
```

(continues on next page)

(continued from previous page)

```
# in MAP define OUTPUTFORMAT (example):
OUTPUTFORMAT
    NAME "OGRGML"
    MIMETYPE "ogr/gml"
    DRIVER "OGR/GML"
    FORMATOPTION "FORM=multipart"
END

# in WEB METADATA define formats (example):
"wms_getfeatureinfo_formatlist" "OGRGML,text/html"
```

19.1.7 Viewing Properties

Once you have added a WMS server, you can view its properties by right-clicking on it in the legend and selecting *Properties*. The WMS/WMTS layer properties is much like the raster layer properties so you will find detailed description at *Raster Properties Dialog*. However, there are some differences, which will be explained below.

Information properties

Metadata Tab

The tab *Metadata* displays a wealth of information about the WMS server, generally collected from the capabilities statement returned from that server. Many definitions can be gleaned by reading the WMS standards (see OPEN-GEOSPATIAL-CONSORTIUM in 文献和网络参考), but here are a few handy definitions:

- **Server Properties**
  - **WMS Version** --- The WMS version supported by the server.
  - **Image Formats** --- The list of MIME-types the server can respond with when drawing the map. QGIS supports whatever formats the underlying Qt libraries were built with, which is typically at least `image/png` and `image/jpeg`.
  - **Identity Formats** --- The list of MIME-types the server can respond with when you use the Identify tool. Currently, QGIS supports the `text-plain` type.
- **Layer Properties**
  - **Selected** --- Whether or not this layer was selected when its server was added to this project.
  - **Visible** --- Whether or not this layer is selected as visible in the legend (not yet used in this version of QGIS).
  - **Can Identify** --- Whether or not this layer will return any results when the Identify tool is used on it.
  - **Can be Transparent** --- Whether or not this layer can be rendered with transparency. This version of QGIS will always use transparency if this is `Yes` and the image encoding supports transparency.
  - **Can Zoom In** --- Whether or not this layer can be zoomed in by the server. This version of QGIS assumes all WMS layers have this set to `Yes`. Deficient layers may be rendered strangely.
  - **Cascade Count** --- WMS servers can act as a proxy to other WMS servers to get the raster data for a layer. This entry shows how many times the request for this layer is forwarded to peer WMS servers for a result.
  - **Fixed Width, Fixed Height** --- Whether or not this layer has fixed source pixel dimensions. This version of QGIS assumes all WMS layers have this set to nothing. Deficient layers may be rendered strangely.
  - **WGS 84 Bounding Box** --- The bounding box of the layer, in WGS 84 coordinates. Some WMS servers do not set this correctly (e.g., UTM coordinates are used instead). If this is the case, then the initial view of this layer may be rendered with a very 'zoomed-out' appearance by QGIS. The WMS webmaster should be informed of this error, which they may know as the WMS XML elements `LatLonBoundingBox`, `EX_GeographicBoundingBox` or the CRS:84 `BoundingBox`.

- **Available in CRS** --- The projections that this layer can be rendered in by the WMS server. These are listed in the WMS-native format.
- **Available in style** --- The image styles that this layer can be rendered in by the WMS server.

Temporal properties

Raster *temporal properties* (namely *Dynamic Temporal Control*) can be set for WMS and WMTS layers. By default, when a time-dimension enabled WMS or WMTS layer is added to the project, it is indicated in the *Layers* panel with the Temporal Layer icon next to it. Its *Temporal* properties default to the *Automatic* temporal mode, meaning that the layer will follow the temporal controller's current time range by default.

You can then opt to show a specific static time value for the layer by unchecking *Dynamic Temporal Control* and picking an option under *Static WMS-T Temporal Range*:

- *Server default*
- *Predefined date* with a server exposing data for non-contiguous temporal ranges or *Predefined range* with a server exposing a range of available dates. A *Start date* and *End date* are necessary in the latter case. Their expected formatting can be deduced from the reference time option (see below). depending on whether the provider has data for contiguous period or not
- *Follow project's temporal range* as defined in the project's properties dialog

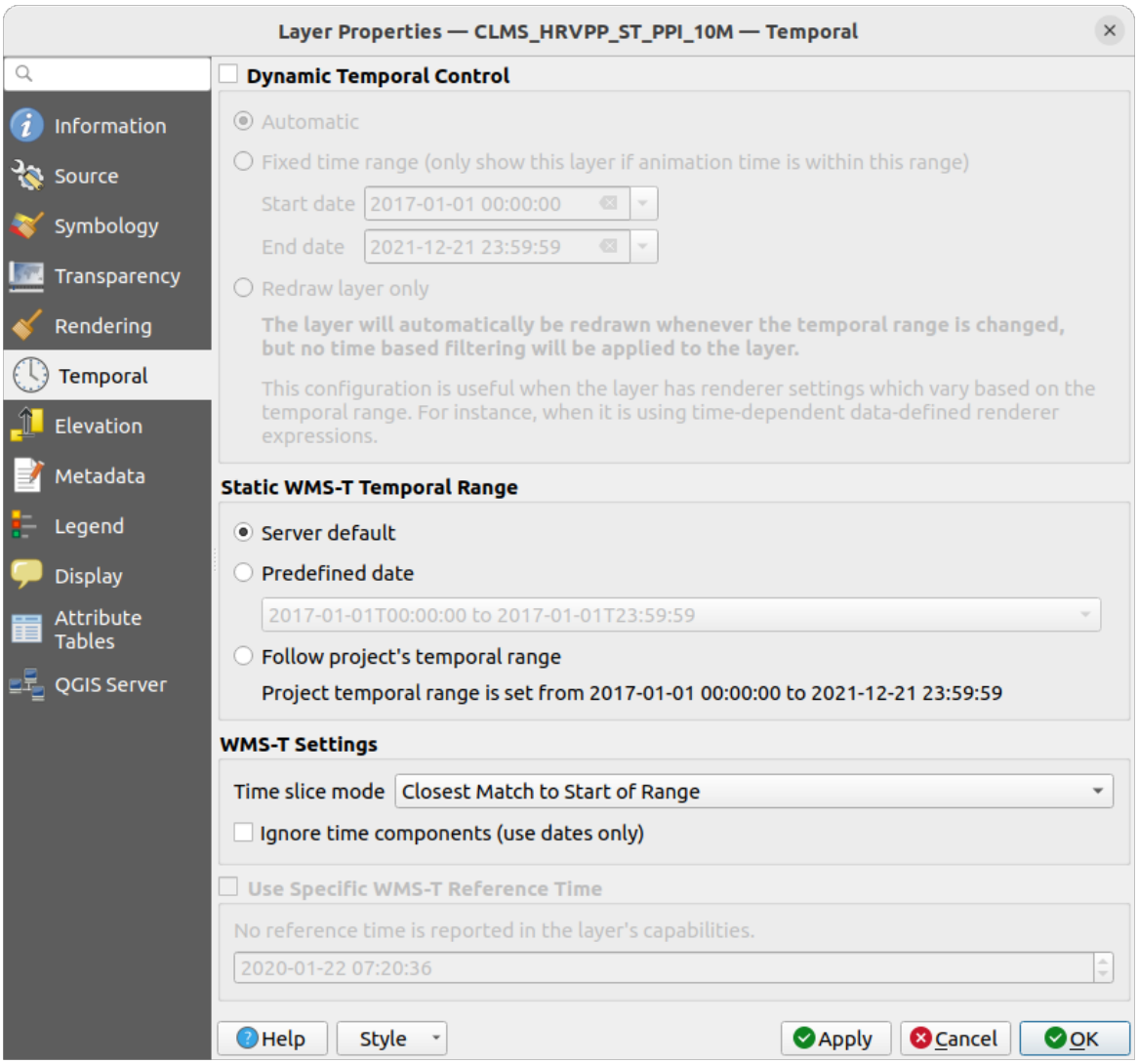


Fig. 19.4: Temporal properties of a WMTS layer

Whatever temporal data control is in use, there are some *WMS-T Settings* to help display the correct temporal data:

- *Time slice mode* which can be:
  - *Use whole temporal range*
  - *Match to start of range*
  - *Match to end of range*
  - *Closest match to start of range*
  - *Closest match to end of range*
- *Ignore time components (use dates only)*: If checked, the time component of temporal queries will be discarded and only the date component will be used in server requests.

You can also *Use Specific WMS-T Reference Time* picked from times reported in the layer's capabilities. Convenient for servers which expose a non-contiguous set of date time instances (instead of a range of dates).

QGIS Server properties

In addition to the raster layer *properties*, publishing a WMS/WMTS layer with QGIS Server will display following options:

- *WMS Print layer*: Allows to set alternative WMS layer that should be used for printing (in the GetProjectSettings reply). Convenient for WMTS layers that are generally not suitable for printing.
- ☑ *Publish WMS/WMTS data source uri*: Allows a web client to fetch the WMS/WMTS data directly
- ☑ *Advertise as background layer*

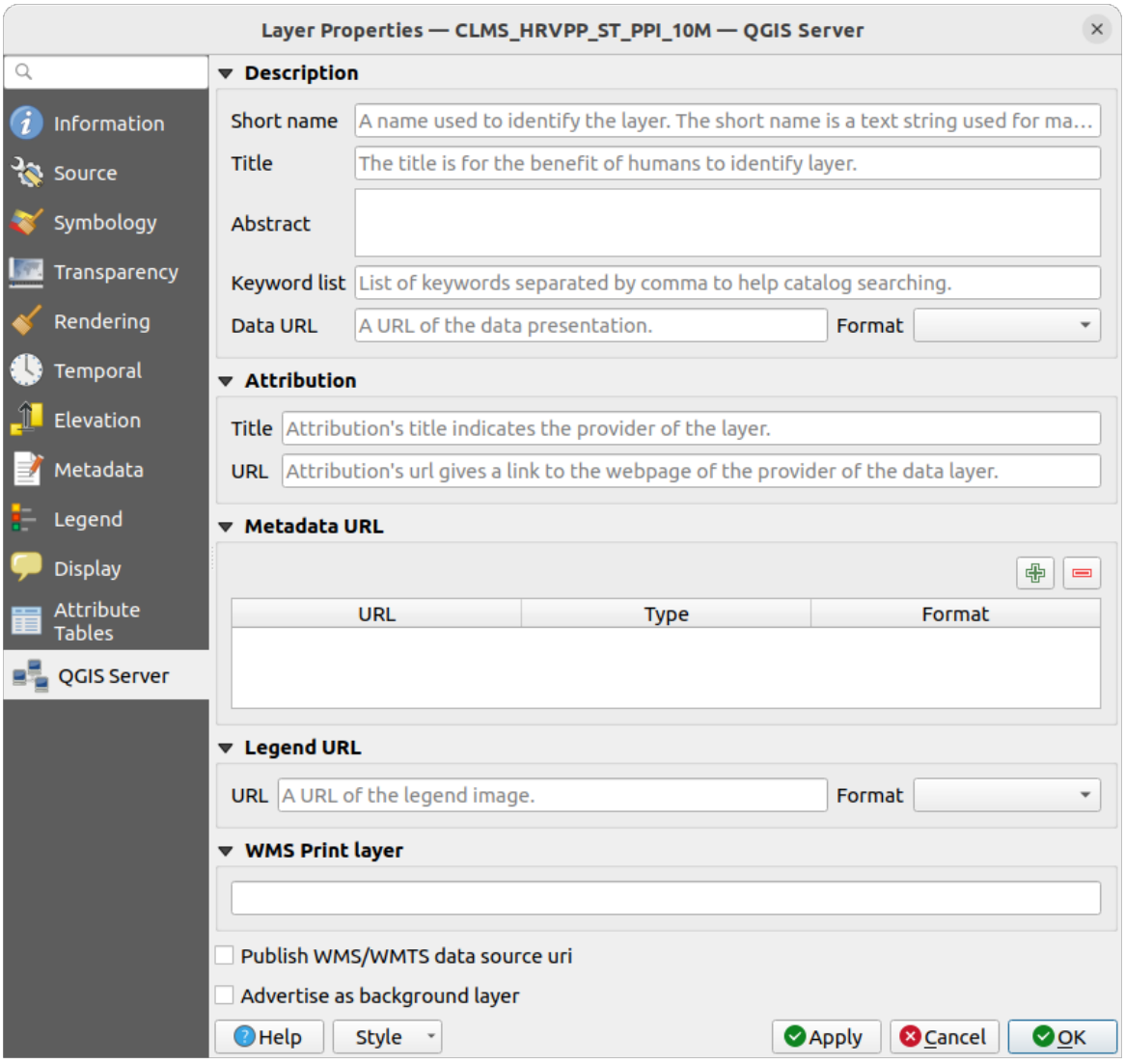


Fig. 19.5: QGIS Server properties of a WMS/WMTS layer

19.1.8 Show WMS legend graphic in table of contents and layout

The QGIS WMS data provider is able to display a legend graphic in the table of contents' layer list and in the print layout. The WMS legend will be shown only if the WMS server has GetLegendGraphic capability and the layer has getCapability url specified, so you additionally have to select a styling for the layer.

If a legendGraphic is available, it is shown below the layer. It is little and you have to click on it to open it in real dimension (due to QgsLegendInterface architectural limitation). Clicking on the layer's legend will open a frame with the legend at full resolution.

In the print layout, the legend will be integrated at it's original (downloaded) dimension. Resolution of the legend graphic can be set in the item properties under *Legend* ► *WMS LegendGraphic* to match your printing requirements.

The legend will display contextual information based on your current scale. The WMS legend will be shown only if the WMS server has GetLegendGraphic capability and the layer has getCapability url specified, so you have to select a styling.

19.2 WCS Client

A Web Coverage Service (WCS) provides access to raster data in forms that are useful for client-side rendering, as input into scientific models, and for other clients. The WCS may be compared to the WFS and the WMS. As WMS and WFS service instances, a WCS allows clients to choose portions of a server's information holdings based on spatial constraints and other query criteria.

QGIS has a native WCS provider and supports both version 1.0 and 1.1 (which are significantly different), but currently it prefers 1.0, because 1.1 has many issues (i.e., each server implements it in a different way with various particularities).

The native WCS provider handles all network requests and uses all standard QGIS network settings (especially proxy). It is also possible to select cache mode ('always cache', 'prefer cache', 'prefer network', 'always network'), and the provider also supports selection of time position, if temporal domain is offered by the server.

Loading a WCS Layer

To be able to load a WCS Layer, first create a connection to the WCS server:

1. Open the *Data Source Manager* dialog by pressing the Open Data Source Manager button
2. Enable the *WCS* tab
3. Click on *New...* to open the *Create a New WCS Connection* dialog

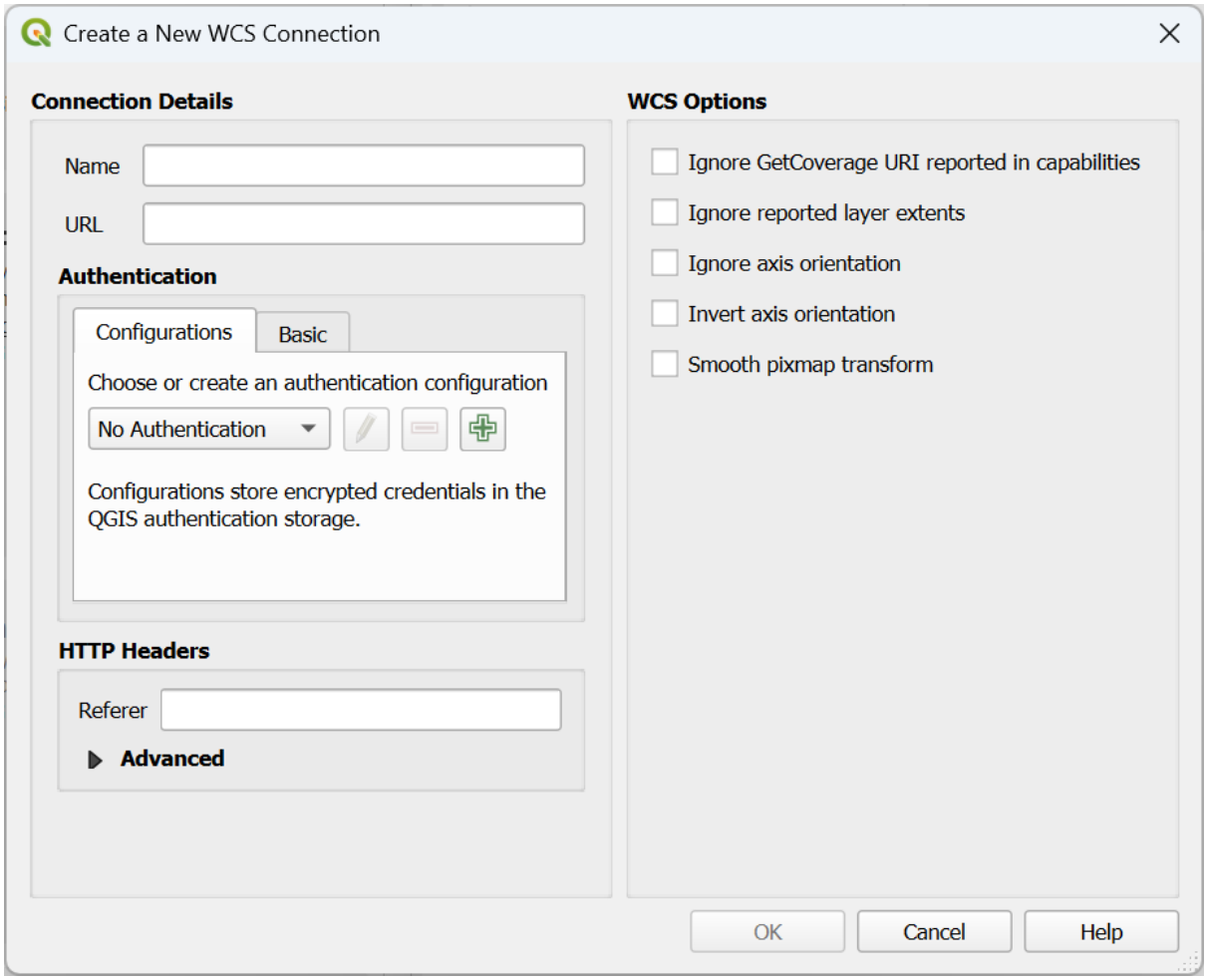


Fig. 19.6: Creating a connection to a WCS server

- *Name*: A name for the connection. This name will be used in the Server Connections drop-down box so that you can distinguish it from other WCS servers.
- *URL*: URL of the server providing the data. This must be a resolvable host name -- the same format as you would use to open a telnet connection or ping a host, i.e. the base URL only. For example, you shouldn't have fragments such as `request=GetCapabilities` or `version=1.0.0` in your URL.
- *Authentication* (optional): using a *stored configuration* or a basic authentication with *Username* and *Password*.

> **Warning**
>
> Entering **username** and **password** in the *Authentication* tab will keep unprotected credentials in the connection configuration. Those **credentials will be visible** if, for instance, you shared the project file with someone. Therefore, it's advisable to save your credentials in an *Authentication configuration* instead (*Configurations* tab). See 认证系统 for more details.

- HTTP *Referer*
- ☐ *Ignore GetCoverage URI reported in capabilities*: if checked, use given URI from the *URL* field above.
- ☐ *Ignore reported layer extents*: because the extent reported by raster layers may be smaller than the actual area which can be rendered (notably for WCS servers with symbology which takes more space than the data extent), check this option to avoid cropping raster layers to their reported extents, resulting in truncated symbols on the borders of these layers.

- ☐ *Ignore axis orientation*
- ☐ *Invert axis orientation*
- ☐ *Smooth pixmap transformation*

4. Press *OK* to create the connection.

Note that any proxy settings you may have set in your preferences are also recognized. Also note that it is possible to *Load* the connection parameters from a `.XML` file or *Save* them to a `.XML` file.

Now we are ready to load WCS layers from the above connection.

19.3 WFS and WFS-T Client

In QGIS, a WFS layer behaves pretty much like any other vector layer. You can identify and select features, and view the attribute table. QGIS supports WFS 1.0.0, 1.1.0, 2.0 and OGC API - Features (OAPIF), including editing (through WFS-T). QGIS also supports background download and progressive rendering, on-disk caching of downloaded features and version autodetection.

Layers of servers implementing OGC API - Features - Part 4: Create, Replace, Update and Delete can be turned into editing mode to allow creating, modifying and deleting features. Note that each created/modified/deleted feature requires a dedicated network request, so performance might suffer in case of simultaneous modification of hundreds or more features at a time.

In general, adding a WFS layer is very similar to the procedure used with WMS. There are no default servers defined, so you have to add your own. You can find WFS servers by using the *MetaSearch plugin* or your favourite web search engine. There are a number of lists with public URLs, some of them maintained and some not.

Loading a WFS Layer

As an example, we use the Gateway Geomatics WFS server and display a layer.

```
https://demo.gatewaygeomatics.com/cgi-bin/wfs_gateway?REQUEST=GetCapabilities&
↪VERSION=1.0.0&SERVICE=WFS
```

To be able to load a WFS Layer, first create a connection to the WFS server:

1. Open the *Data Source Manager* dialog by pressing the Open Data Source Manager button
2. Enable the *WFS / OGC API - Features* tab
3. Click on *New...* to open the *Create a New WFS Connection* dialog
4. In the *Connection Details* group, enter `Gateway Geomatics` as name
5. Enter the URL (see above)
6. Optionally, enter the *Authentication* settings

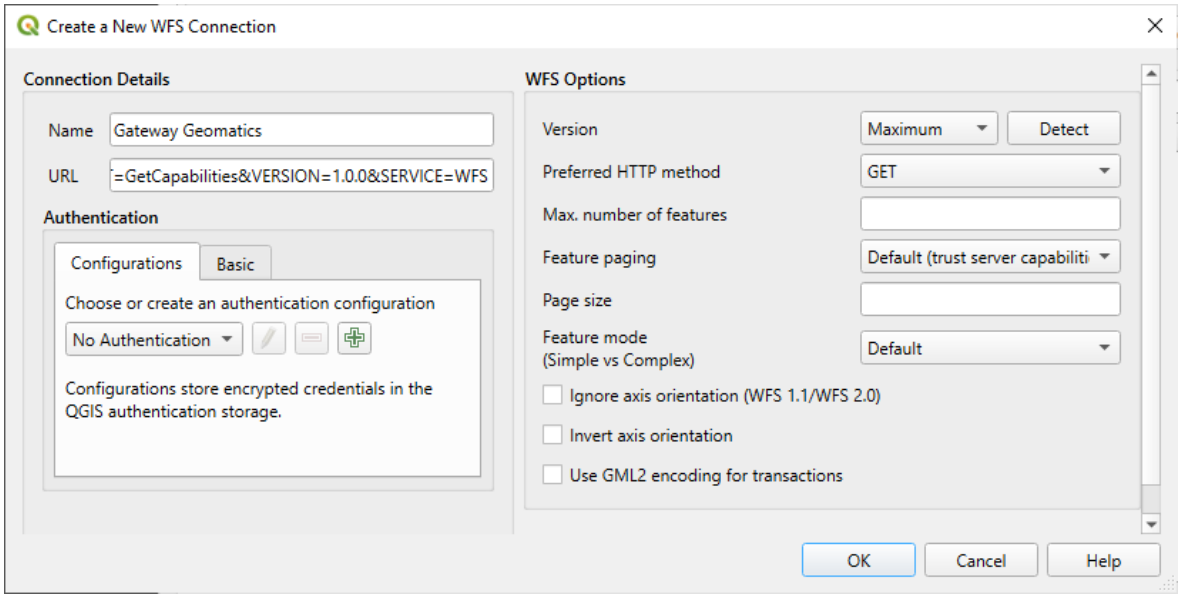


Fig. 19.7: Creating a connection to a WFS server

Note

In case of an OGC API - Features (OAPIF), the URL to provide should be the landing page, i.e., the main page from which it is possible to navigate to all the available service endpoints.

7. In the *WFS Options* group, you can:
   - Indicate the WFS version of the server. If unknown, press the *Detect* button to automatically retrieve it.
   - Select the *Preferred HTTP method* to use for requests. The default is *GET*, but you can also select *POST*.
   - Define the *Maximum number of features* retrieved in a single GetFetFeature request. If empty, no limit is set.
   - *Feature mode (simple vs complex)*: helps determine how to handle servers serving data with simple or complex GML schemas. Available options are:
     - *Default*: if the server supports editing capabilities, this is the same as using *Simple Features*. If not, it fallbacks to using *Complex Features* mode.
     - *Simple Features*: returns data as simple features only
     - *Complex Features*: uses the *OGR GMLAS <https://gdal.org/en/stable/drivers/vector/gmlas.html>*, potentially returning complex features, depending on the data schema, where nested data structures are exposed by QGIS as JSON.
   - Depending on the WFS version, indicate whether to:
     - Enable *Feature paging* and specify the maximum number of features to retrieve with *Page size*. If no limit is defined, then the server default is applied.
     - Force to *Ignore axis orientation (WFS 1.1/WFS 2.0)*
     - *Invert axis orientation*.
     - *Use GML2 encoding for transactions*.

Warning

> Entering **username** and **password** in the *Authentication* tab will keep unprotected credentials in the connection configuration. Those **credentials will be visible** if, for instance, you shared the project file with someone. Therefore, it's advisable to save your credentials in an *Authentication configuration* instead (*Configurations* tab). See 认证系统 for more details.

8. Press *OK* to create the connection.

Note that any proxy settings you may have set in your preferences are also recognized. Also note that it is possible to *Load* the connection parameters from a `.XML` file or *Save* them to a `.XML` file.

Now we are ready to load WFS layers from the above connection.

1. Choose 'Gateway Geomatics' from the *Server Connections* drop-down list.
2. Click *Connect*
3. Select the *Parks* layer in the list
4. You can also choose whether to:
   - *Only request features overlapping the view extent*
   - *Change...* the layer's CRS to any other supported by the service
   - or build a query to specify particular features to retrieve from the service: double-click on the row of the layer to open the *SQL Query Composer* dialog. That dialog provides widgets to write an advanced SQL query relying on available tables and columns of the service, with sorting and filtering and a bunch of SQL functions, spatial predicates and operators.

     The query you build will appear after validation in the *SQL* column within the *WFS / OGC API - Features* table, and the filtered layer will display the icon next to it in the *Layers* panel. It is thus possibe to adjust the query at any moment.
5. Click *Add* to add the layer to the map, using its default name from the queried service.

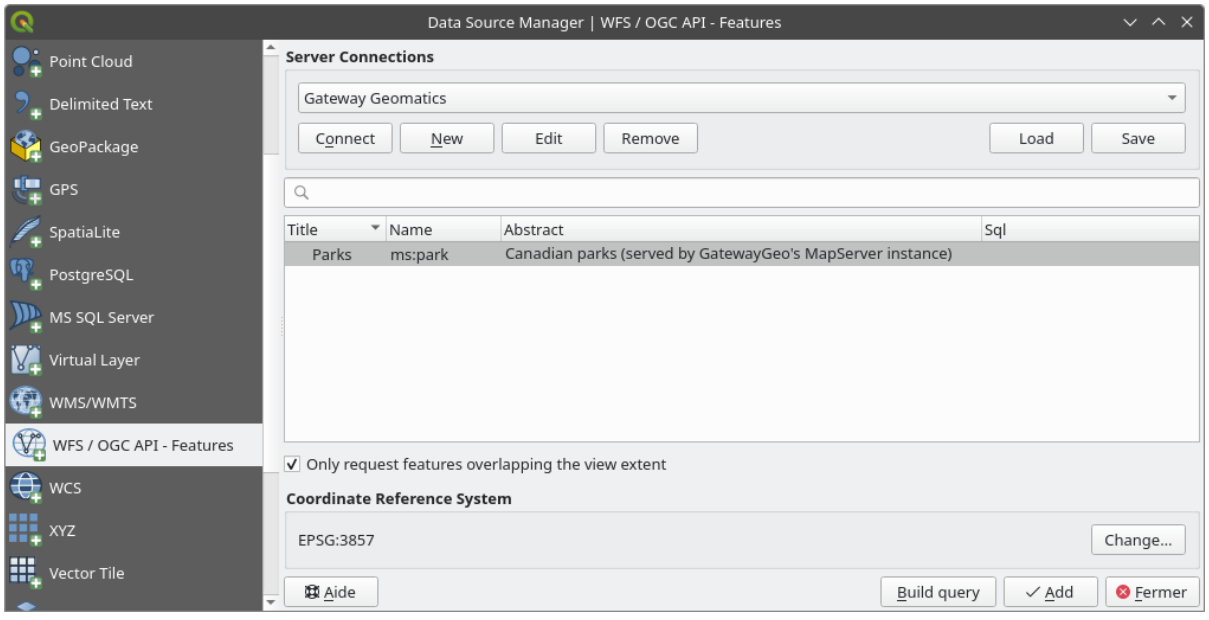


Fig. 19.8: Adding a WFS layer

You'll notice the download progress is visualized in the lower left of the QGIS main window. Once the layer is loaded, you can identify and select a couple of features and view the attribute table.

19.4 SensorThings

QGIS supports connections to OGC SensorThings API, a standard providing an open and unified framework to interconnect IoT sensing devices, data, and applications over the Web. It is an open standard addressing the syntactic and semantic interoperability of the Internet of Things. It is based on the Observations and Measurements data model, a standardized model for observations, and for features involved in sampling when making observations.

19.4.1 Setting connection

To add SensorThings data to QGIS use the *SensorThings* tab in the *Data Source Manager* dialog.

To establish a new connection, press *New* (or *New SensorThings Connection* from the Browser panel) and provide *Name* and *URL*. Advanced options, such as *authentication* and a *Referer*, can also be configured.

Press *OK* to establish the connection. Then you will be able to:

- *Edit* the SensorThings connection settings
- *Remove* the SensorThings connection

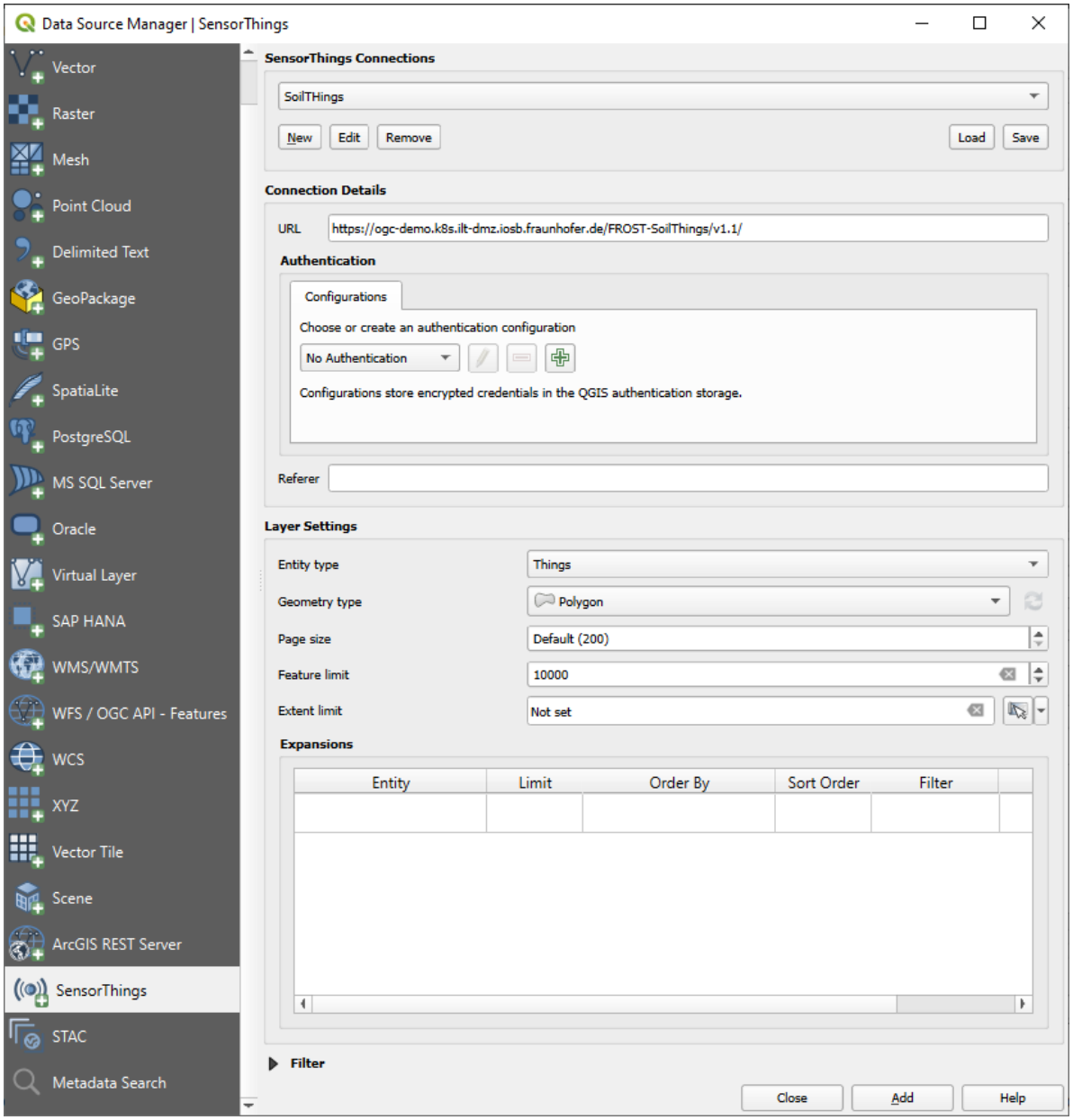


Fig. 19.9: SensorThings Connection dialog

Configurations can be saved to an `.XML` file (*Save*) through the *SensorThings* entry in *Data Source Manager* dialog or its contextual menu in the *Browser* panel (*Save Connections*). Likewise, configurations can be added from a file (*Load*).

19.4.2 Loading SensorThings data

Relations between layers (so-called entities) stored in a SensorThings dataset are expressed in the diagram below.

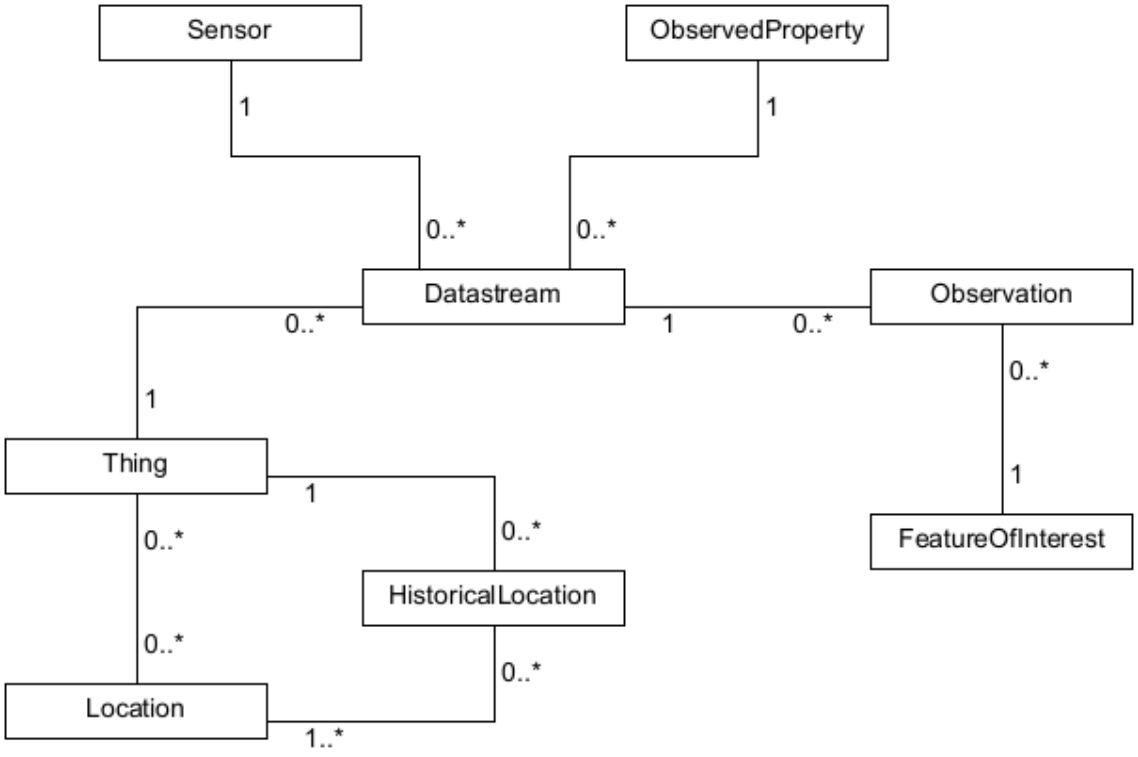


Fig. 19.10: Data model Observations and Measurements (Source: Sensor Web Tutorial by SIST network)

Any type of entity can be loaded in QGIS, but not all are spatial data. To load an entity, there are *Layer Settings* that can be configured:

- *Entity Type*: the entity to load from the data model as layer in QGIS
- *Geometry Type*: the geometry type of the selected entity to load. Press Check available types to limit the list to the actually supported geometry types.
- *Page Size*
- *Feature Limit* sets a maximum number of features to request from the service
- *Extent Limit* sets a maximum extent limit for the layer, so that only features within the extent are requested
- *Expansions*: The data model of SensorThings provides a mechanism of expansion of the results to related entities, similar to how tables are joined together in a relational database. Using this approach, you can expand the selected layer to include data from other items. This will flatten the relationship, creating as many parent features as children, and additional properties are added as columns in the attribute table.

Use *Filter* to build a query to filter the data, using SensorThings filter syntax.

Note

The above settings and filtering options are also available for update in the layer properties dialog, *Source* tab, once loaded in QGIS.

Press *Add* to load the selected entity type as layer in QGIS.

19.4.3 Working with a vector layer from SensorThings

A SensorThings layer is loaded in QGIS as a vector layer. As such, it displays the same tabs in the *layer properties* and allows same feature interactions using the selection or identify tools. There are however some specificities you should consider while working with SensorThings data.

Because of the data model, the result property of a SensorThings Observation is a string field. In case you want to use its numerical representation in for example a graduated style, use an expression to convert the value to real and try() in case this fails (e.g., `try( to_real("Observation_result"), Null)`).

In case you want to create a chart of the observations at one or more locations, you can install the QGIS plugin *Data Plotly*.

1. Now select the observations at a point location in the map view.
2. Open the plotly panel and activate the *Use only selected features* checkbox.
3. Select on the x-column a date-time property and on the y-column the *Observation_result*. This will plot the observations at that location over time.
4. Verify to filter by a single Observed Property.
5. Notice that the chart changes as soon as you select other locations on the map.

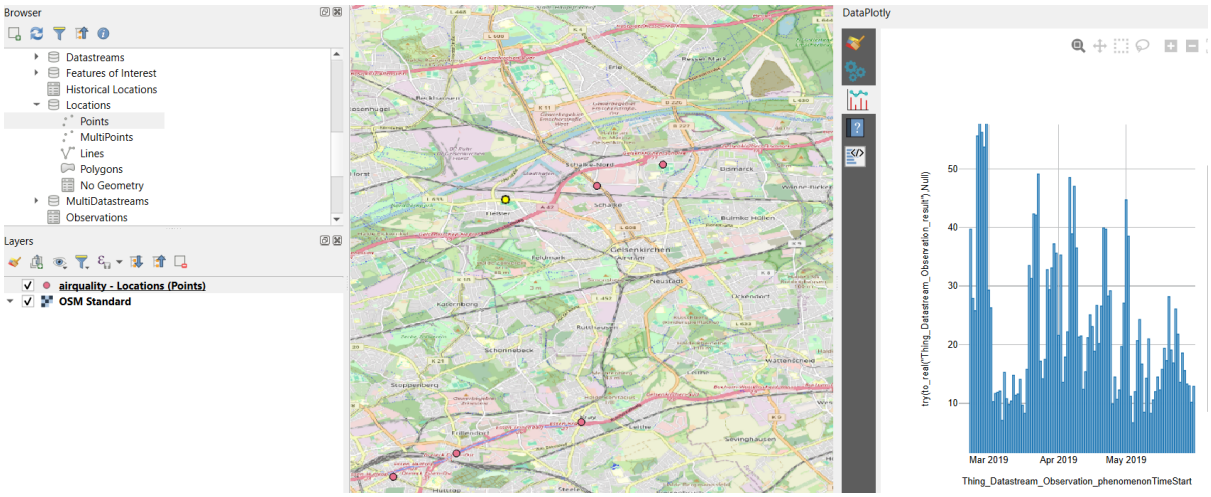


Fig. 19.11: Use Data plotly to plot the air quality observations at a location

19.5 STAC (SpatioTemporal Asset Catalogs)

STAC (SpatioTemporal Asset Catalog) is a specification for describing geospatial data in a consistent and accessible way. It defines a standard structure for organizing and indexing spatial-temporal assets such as satellite imagery, drone photos, or sensor data so that you can search, preview, and use these datasets across different platforms and tools.

STAC enables QGIS users to connect to and browse geospatial datasets, either from static catalogs or dynamic APIs, and download or stream assets such as imagery or other spatial data.

More information at https://stacspec.org/en.

Key components of STAC in QGIS include:

- **STAC Items**: The basic unit of a STAC catalog. Each item represents one geospatial asset or a group of related assets (e.g. image bands) at a specific time and location. It includes metadata such as geometry, datetime, links to assets (e.g., TIFFs, JSON, COG), and properties.
- **STAC Collections**: A grouping of items that share common characteristics and metadata (e.g. satellite mission). Collections may define spatial and temporal extents, licensing, and keywords.
- **STAC Catalogs**: A hierarchical container that organizes items and/or collections. Catalogs allow navigation of STAC datasets but do not necessarily include search capabilities.
- **STAC API**: An implementation of the STAC specification that allows querying and filtering of STAC items using spatial and temporal based filters, as well as searching within specific collections. STAC APIs follow the OGC API - Features pattern and support dynamic access to datasets.

There is an important distinction between static STAC catalogs and STAC API endpoints:

- **Static STAC Catalogs**: are collections of JSON files without search capabilities. They can be browsed via the Browser panel.
- **STAC API**: provide search capabilities and can be accessed in QGIS through both the Browser panel and the Data Source Manager.

19.5.1 Setting connection

STAC connections can be added in QGIS using either the *Browser panel* or the *Data Source Manager*:

- **Browser Panel:** In the *Browser panel*, right-click on the *STAC* entry and select *New STAC Connection...*. In the dialog that appears, enter a *Name* for the connection, the *URL* of the STAC catalog and optionally fill in *Authentication* credentials and a *Referer*. Then click *OK*.

  Use this method for browsing static STAC catalogs that do not support search or filtering.

- **Data Source Manager:** For STAC APIs you can use *Data Source Manager* dialog.

  Open the *Data Source Manager*, choose the *STAC* tab and click the *New* button. Fill in the *Name* and *URL* fields, and (optional) the *Authentication* credentials and a *Referer*. Press *OK* and then *Connect* to establish the connection, after that you will be able to:

  - *Edit* the STAC connection settings
  - *Remove* the STAC connection

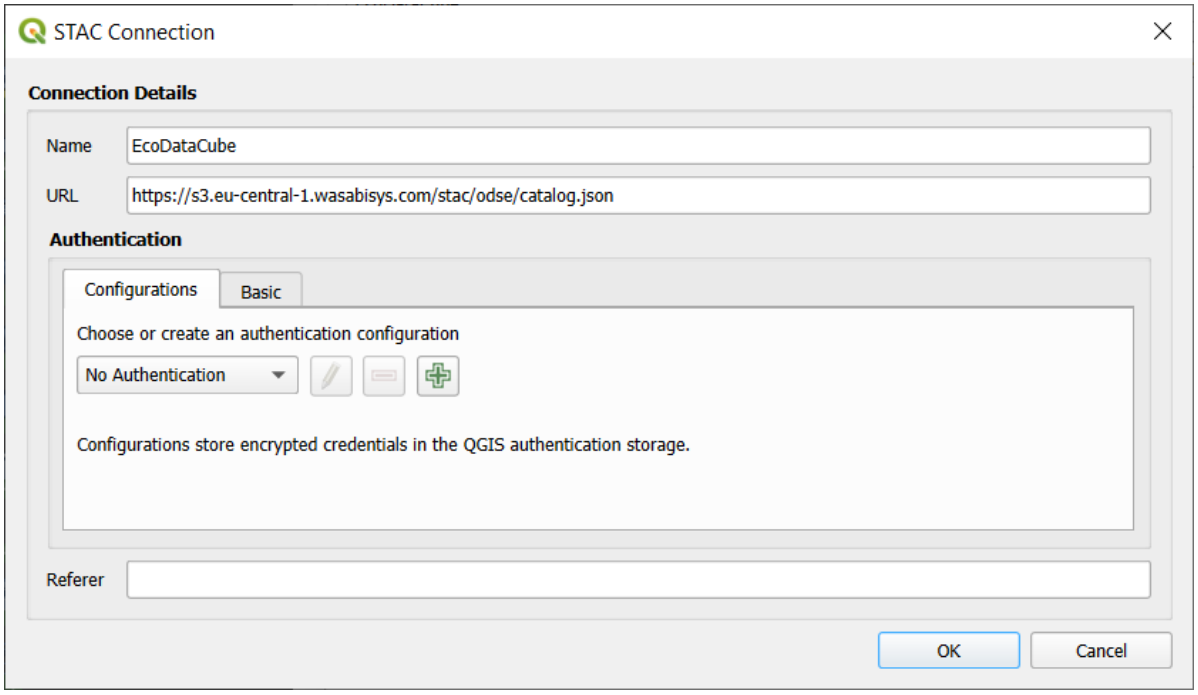


Fig. 19.12: Creating a connection to a STAC server

19.5.2 Browsing STAC Catalogs

Static STAC catalogs are displayed as hierarchical structures within the *Browser panel*. Once connected, the STAC catalog appears under *STAC* in the *Browser* panel. You can expand the catalog node to see its *Collections*. Expanding a collection reveals the individual *Items* it contains.

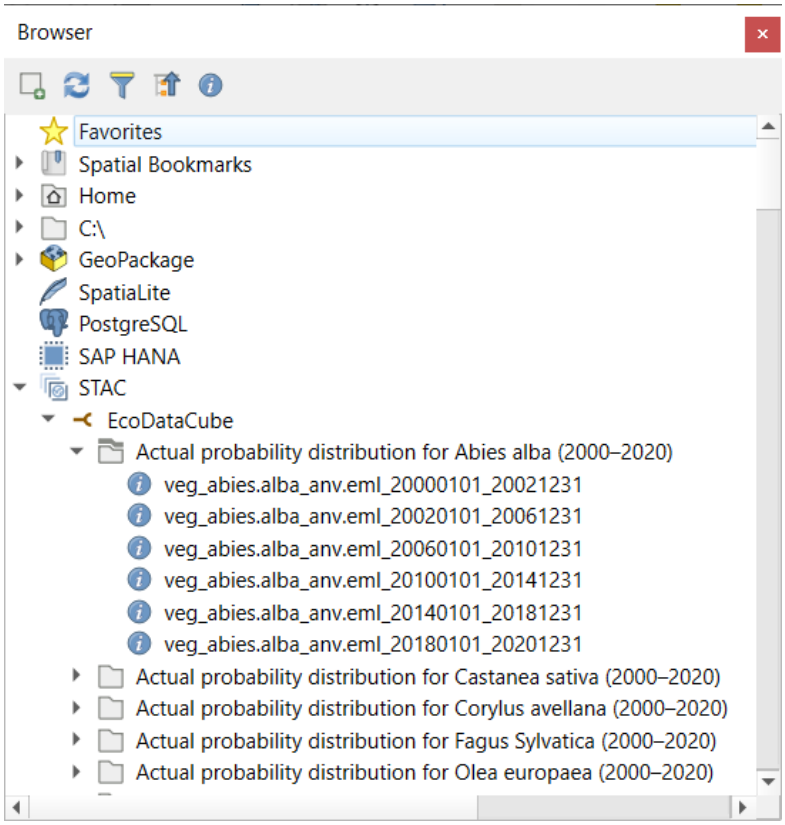


Fig. 19.13: STAC connection expanded in the Browser, showing Collections and Items

Right-click any STAC *Item* and choose *Details...* to view its metadata. The details panel shows the item's JSON content and a map of its coverage. If an item's asset is a cloud-optimized format (e.g. a `.COG`), you can add it directly to the map canvas. Otherwise, *Download Assets...* to save it locally before use.

19.5.3 Filtering and Searching STAC Items

For STAC API endpoints, QGIS supports spatial and temporal filtering, as well as limiting searches to specific collections.

To search for items:

1. Open *Data Source Manager* and go to the *STAC*
2. Select the STAC connection you created earlier and click *Filters*⋯

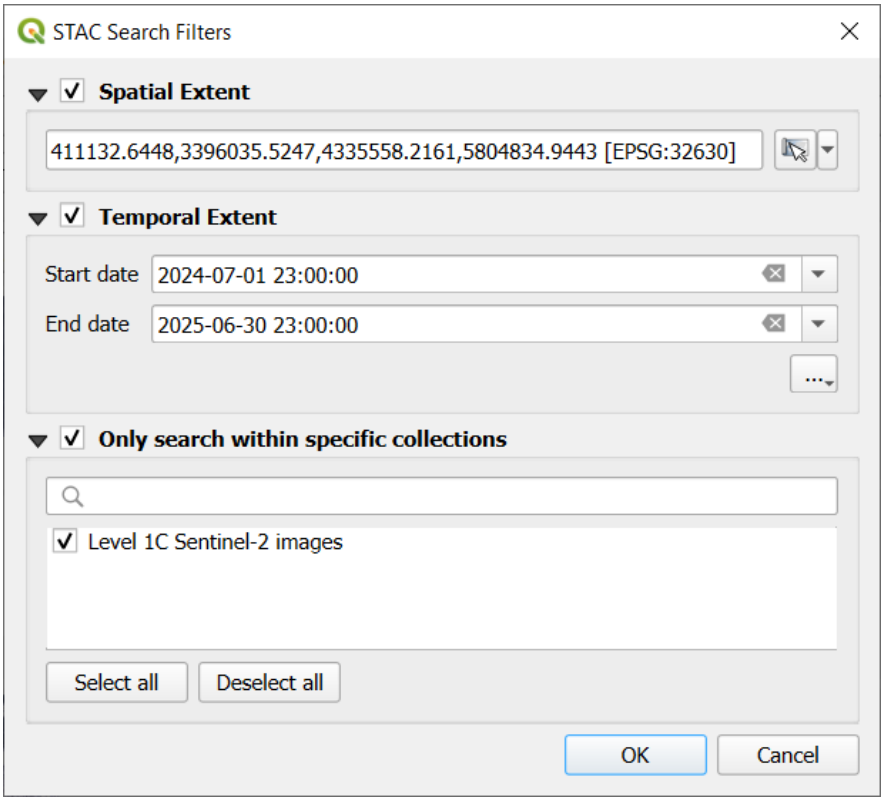


Fig. 19.14: Define spatial and temporal filters for a STAC catalog search

3. In the filter dialog, you can:
   - Enable *Spatial Extent* to restrict results to a bounding box or drawn area.
   - Enable *Temporal Extent* and define a date or range.
   - Limit to specific collections using *Only search within specific collections*
4. After applying filters, results will be listed in the main *Data Source Manager* panel.
5. Optionally check *Show Footprints* to display the footprints of the results on the map.

Right-click a result item to access actions:

- *Zoom to Item*
- *Pan to Item*
- *Download Assets*
- *Details...*

If the item's asset requires download, use the *Download Assets* option.

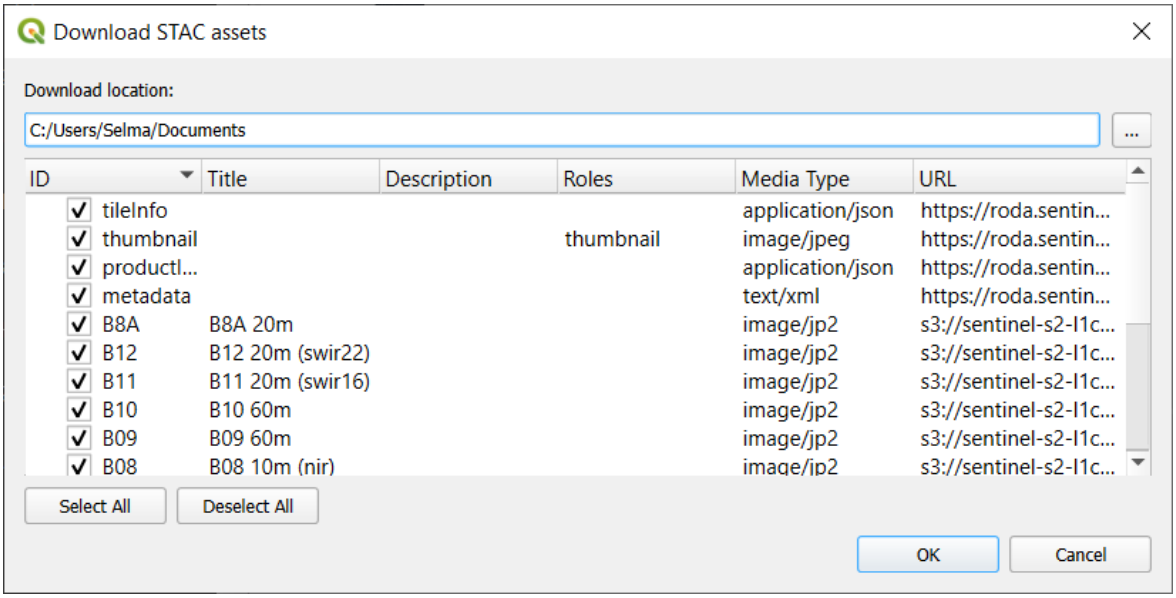


Fig. 19.15: Download STAC item assets

Downloaded assets include the main dataset and any auxiliary files such as thumbnails or style files. After download, use standard QGIS tools (e.g., *Add Raster Layer...*) to load and display the data.

19.6 ArcGIS REST Server

An ArcGIS REST Server can host many different types of web services (feature service, map service, image service, ...). Read *使用 ArcGIS REST 服务器* for instructions to connect to an ArcGIS REST server.

When loading layers form an ArcGIS Feature Service, symbols are automatically converted to QGIS-native symbology, allowing QGIS to visually reproduce the same appearance as ArcGIS web maps.

Layers from an ArcGIS Feature Service can be editable in QGIS if:

1. The ArcGIS service has editing enabled
2. Your credentials grant write access
3. And the remote layer supports editing operations (Create, Update, or Delete capabilities). If these options are disabled on the server, the layer will be read-only in QGIS.

Editing these layers follow the *same procedure* as for other vector layers. To update your map with the latest changes, use *View* ► *Refresh* or press `F5`. If multiple users edit the same layer concurrently, it is recommended to refresh your data before performing edits to avoid conflicts.

CHAPTER 20

使用 GPS 数据

20.1 Introducing GNSS/GPS Data

20.1.1 What is GPS?

GPS, the **Global Positioning System**, is a satellite-based system that allows anyone with a GPS receiver to find their exact position anywhere in the world. GPS is used as an aid in navigation, for example in airplanes, in boats and by hikers. The GPS receiver uses the signals from the satellites to calculate its latitude, longitude and (sometimes) elevation. Most receivers also have the capability to store:

- locations (known as **waypoints**)
- sequences of locations that make up a planned **route**
- and a **track** log of the receiver's movement over time.

Waypoints, routes and tracks are the three basic feature types in GPS data. QGIS displays waypoints in point layers, while routes and tracks are displayed in linestring layers.

> **Note**
>
> QGIS supports also GNSS receivers. But we keep using the term GPS in this documentation.

20.1.2 Transferring or loading GPS data

Loading a GPX file

有几十种不同的文件格式用于存储 GPS 数据。QGIS 使用的格式称为 GPX (GPS 交换格式)，这是一种标准的交换格式，可以在同一文件中包含任意数量的航点、路径和轨迹。

To load a `GPX` file:

1. Open the *GPS* tab in the *Data Source Manager* dialog, i.e.:
   - Click the Open Data Source Manager button on the toolbar (or press `Ctrl+L`) and enable the target tab
   - Or select *Layer* ► *Add Layer* ► *Add GPX Layer...*
2. Use the ... Browse button next to the *GPX dataset* option to select the GPX file
3. Use the check boxes to select the *Feature types* you want to load from the file. Each feature type (*Waypoints*, *Tracks* or *Routes*) will be loaded in a separate layer.

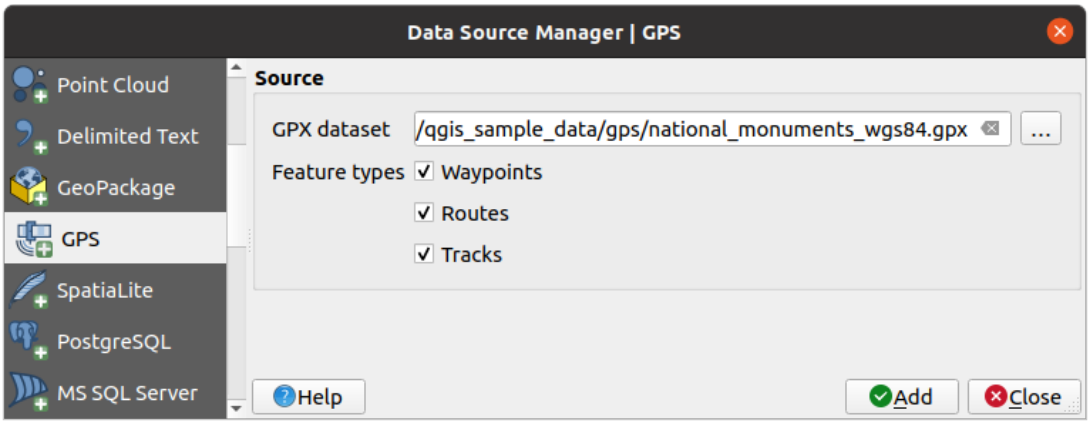


Fig. 20.1: 加载 GPS 数据对话框

Loading to or from a device

There are lots of different types of GPS devices and formats. Since QGIS uses GPX files, you need a way to convert other GPS file formats to GPX. QGIS can do that using the free program GPSBabel. GPSBabel can help you convert waypoints, tracks, and routes between popular GPS receivers such as Garmin or Magellan and mapping programs like Google Earth or Basecamp. Literally hundreds of GPS receivers and programs are supported. It can also transfer GPS data between your computer and a GPS device.

Under *Settings* ► *Options* ► *GPS* ► *GPSBabel*, QGIS allows you to define your own device type and set parameters of conversion that could later be used by the *Processing GPS algorithms*.

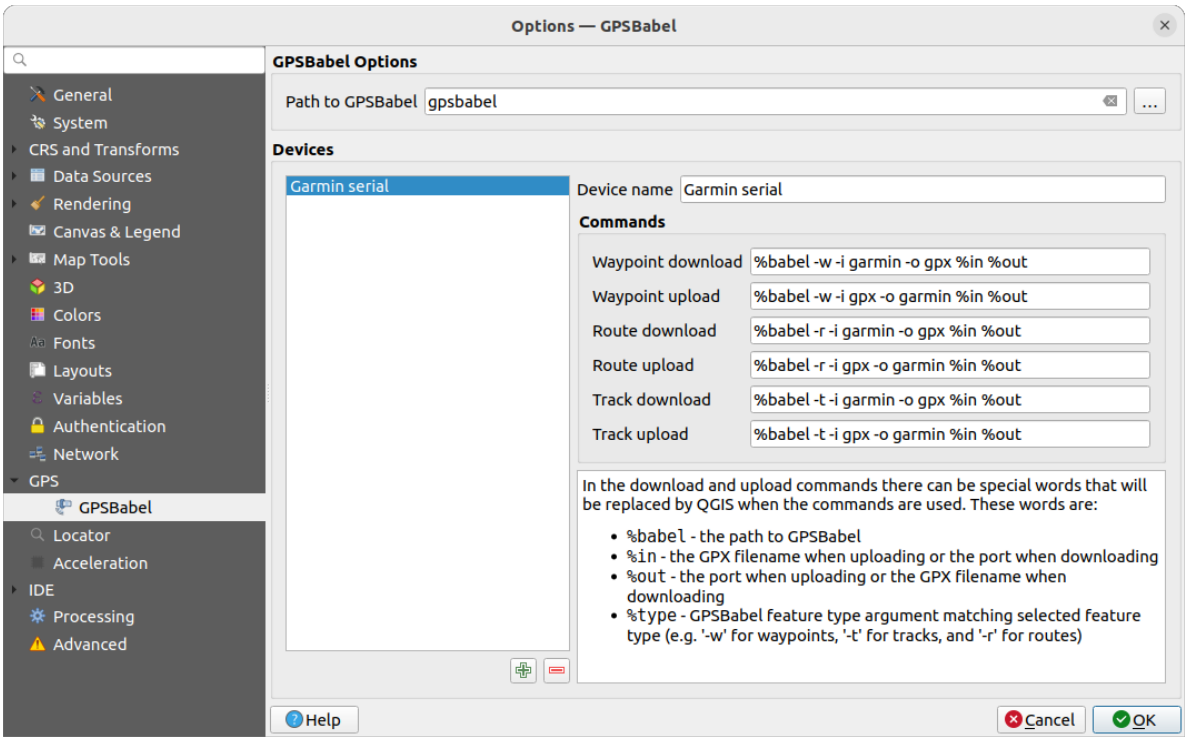


Fig. 20.2: GPS Babel settings

1. First you have to define the *Path to GPSBabel* binaries.
2. Then you may want to add your device. You can update devices list using Add new device or Remove device button.
3. For each device:
   - you provide a *Device name*
   - you configure different *Commands* QGIS will use while interacting with it, such as:
     - *Waypoint download* from the device
     - *Waypoint upload* to the device
     - *Route download* from the device
     - *Route upload* to the device
     - *Track download* from the device
     - *Track upload* to the device

   While the commands are usually GPSBabel commands, you can also use any other command line program that can create a GPX file. QGIS will replace the keywords `%type`, `%in`, and `%out` when it runs the command.

   As an example, if you create a device type with the download command `gpsbabel %type -i garmin -o gpx %in %out` and then use it to download waypoints from port `/dev/ttyS0` to the file `output.gpx`, QGIS will replace the keywords and run the command `gpsbabel -w -i garmin -o gpx /dev/ttyS0 output.gpx`.

   Read the GPSBabel manual for the command line options that may be specific to your use case.

Once you have created a new device type, it will appear in the device lists for the GPS download and upload algorithms.

> **Note**
>
> GPS units allow you to store data in different coordinate systems. When downloading a GPX file (from your GPS unit or a web site) and then loading it in QGIS, be sure that the data stored in the GPX file uses WGS 84 (latitude/longitude). QGIS expects this, and it is the official GPX specification. See GPX 1.1 Schema Documentation.

20.2 Live GPS tracking

QGIS can help you do field mapping with a GPS receiver. Such a live tracking operation is done using the *GPS Toolbar*. Some *device configuration* may be needed before connecting QGIS and the GPS receiver.

20.2.1 GPS toolbar

The *GPS Toolbar* provides the main tools to control a live tracking session. It can be activated from *View* ► *Toolbars* ► *GPS*. It follows the state of the project, GPS, and current GPS track, and enables actions only when they make sense. Once a device is detected by QGIS, you will be able to interact with it:

- Connect to GPS
- Disconnect from GPS
- Recenter map on GPS location: the map will immediately recenter on the current GPS position regardless of the *automatic recentering parameter*.
- Set destination layer for GPS digitized features: By default, QGIS will *Follow active layer* for feature digitizing, meaning that GPS digitizing tools will adapt to the layer selected in *Layers* panel and created features will be stored in it accordingly. While this may be useful in some contexts, it also implies careful interaction with other layers to avoid storing inadvertently features in undesired layers. This option allows you to explicitly indicate a layer for the data storage during the live tracking session, and you could switch depending on your needs. Exposed tools will adapt as well to the selected layer type.

  The GPS destination layer is made automatically editable when a feature is being created, and the user is notified accordingly.
- actions for feature creation:
  - Add vertex to GPS track using GPS current location
  - Create a new point feature at current GPS location
  - Create a new line feature using the current GPS track
  - Create a new polygon feature using the current GPS track
  - Reset track
- Show GPS information: opens the *GPS Information* panel
- a display box for quick access to some GPS information components you may need:
  - Show location
  - Show altitude (Geoid)
  - Show altitude (WGS 84 ellipsoid)

  - Show ground speed
  - Show bearing
  - Show total track length
  - Show distance from start of track

- The Settings button opens a drop-down menu with common settings which are expected to be modified mid-session:
  - *Show location marker*
  - *Show bearing line*
  - *Rotate map to match GPS direction*
  - Options to control map recentering:
    * *Always recenter map* when the GPS position is offset from the map canvas center by a certain distance (as a ratio of the map canvas extent)
    * *Recenter map when leaving extent*
    * *Never recenter*
  - *Automatically add track vertices* whenever new locations are received from the GPS device.
  - *Automatically save added features*: features created from GPS locations are immediately committed to their target layers (skipping the usual layer edit buffer)
  - *Time stamp destination* to adjust the field for time fix storage
  - *Log to GeoPackage/SpatiaLite...*: When activated, the user will be prompted to select an existing GeoPackage/SpatiaLite file or enter a new file name. A `gps_points` and `gps_tracks` table will be created in the file with a predefined structure.

    All incoming GPS messages will be logged to the `gps_points` layer, along with speed, bearing, altitude and accuracy information from the GPS.

    When the GPS is disconnected (or QGIS closed), the entire recorded GPS track will be added to the `gps_tracks` table (along with some calculated information like track length, start and end times).
  - *Log NMEA sentences...*: enables logging all raw NMEA strings from the device to a text file
  - *GPS settings...* to access the GPS *global options* dialog

> **Tip**
>
> **Live status bar information on position**
>
> When a GPS device is connected and the user moves the cursor over the map canvas, a live status bar message displays the distance and bearing from the cursor to the GPS position. Project distance and bearing settings are respected in this display. On a touch screen device, use a tap-and-hold event to trigger the live status bar message.

20.2.2 GPS Information Panel

For full monitoring of a live GPS tracking in QGIS, you may need to enable the *GPS Information Panel* (*View* ► *Panels* ► or press `Ctrl+0`).

In the top right corner of the *GPS Information Panel*, you press *Connect* to initiate connection between QGIS and a plugged-in GPS receiver, or *Disconnect* them.

In the top left of the panel, following buttons are accessible:

- Position: live details on the GPS position and sensors
- Signal: signal strength of satellite connections
- Settings: drop-down menu to *live tracking options* you may need to modify during a session

Position and additional attributes

In the Position tab, if the GPS is receiving signals from satellites, you will see your position in latitude, longitude and altitude together with additional attributes.

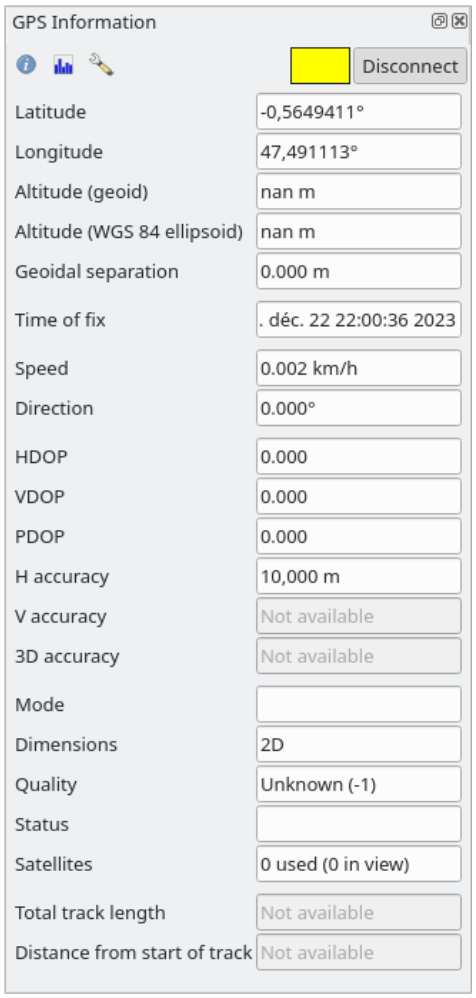


Fig. 20.3: GPS tracking position and additional attributes

- *Latitude*
- *Longitude*

- *Altitude (Geoid)*: Altitude/elevation above or below the mean sea level
- *Altitude (WGS 84 ellipsoid)*: Altitude/elevation above or below the WGS-84 Earth ellipsoid
- *Geoidal separation*: Difference between the WGS-84 Earth ellipsoid and mean sea level (geoid), − means mean sea level is below ellipsoid
- *Time of fix*
- *Speed*: Ground speed
- *Direction*: Bearing measured in degrees clockwise from true north to the direction of travel
- *HDOP*: Horizontal dilution of precision
- *VDOP*: Vertical dilution of precision
- *PDOP*: Dilution of precision
- *H accuracy*: Horizontal accuracy in meters
- *V accuracy*: Vertical accuracy in meters
- *3D accuracy*: 3D Root Mean Square (RMS) in meters
- *Mode*: GPS receiver configuration 2D/3D mode, can be `automatic` or `manual`
- *Dimensions*: Position fix dimension, can be `2D`, `3D` or `No fix`
- *Quality*: Positioning quality indicator
- *Status*: Position fix status, can be `Valid` or `Invalid`
- *Satellites*: Count of satellites used in obtaining the fix
- *Total track length*: Total distance of current GPS track
- *Distance from start of track*: Direct distance from first vertex in current GPS track to latest vertex

Signal

With Signal tab, you can see the signal strength of the satellites you are receiving signals from.

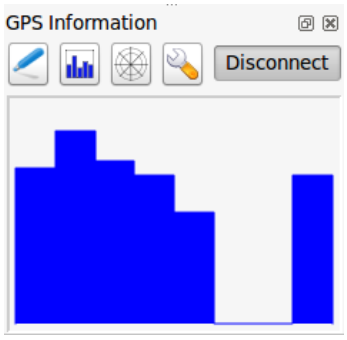


Fig. 20.4: GPS tracking signal strength

20.2.3 Connect to a Bluetooth GPS for live tracking

With QGIS you can connect a Bluetooth GPS for field data collection. To perform this task you need a GPS Bluetooth device and a Bluetooth receiver on your computer.

At first you must let your GPS device be recognized and paired to the computer. Turn on the GPS, go to the Bluetooth icon on your notification area and search for a New Device.

On the right side of the Device selection mask make sure that all devices are selected so your GPS unit will probably appear among those available. In the next step a serial connection service should be available, select it and click on *Configure* button.

Remember the number of the COM port assigned to the GPS connection as resulting by the Bluetooth properties.

After the GPS has been recognized, make the pairing for the connection. Usually the authorization code is `0000`.

Now open *GPS information* panel and switch to GPS options screen. Select the COM port assigned to the GPS connection and click the *Connect*. After a while a cursor indicating your position should appear.

If QGIS can't receive GPS data, then you should restart your GPS device, wait 5-10 seconds then try to connect again. Usually this solution work. If you receive again a connection error make sure you don't have another Bluetooth receiver near you, paired with the same GPS unit.

20.2.4 Examples of GPS devices connection in QGIS

Using GPSMAP 60cs

MS Windows

Easiest way to make it work is to use a middleware (freeware, not open) called GPSGate.

Launch the program, make it scan for GPS devices (works for both USB and BT ones) and then in QGIS:

1. In *Settings* ► *Options* ► *GPS*, detect the plugged-in device. You can use the *Autodetect* mode.
2. Press *Connect* in the *GPS Information* panel

Ubuntu/Mint GNU/Linux

As for Windows the easiest way is to use a server in the middle, in this case GPSD, so

1. Install the program

```
sudo apt install gpsd
```

2. Then load the `garmin_gps` kernel module

```
sudo modprobe garmin_gps
```

3. And then connect the unit and check with `dmesg` the actual device being used by the unit, for example `/dev/ttyUSB0`.
4. Now you can launch gpsd

```
gpsd /dev/ttyUSB0
```

5. And finally connect with the QGIS live tracking tool.

Using BTGP-38KM datalogger (only Bluetooth)

Using GPSD (under Linux) or GPSGate (under Windows) is effortless.

Using BlueMax GPS-4044 datalogger (both BT and USB)

MS Windows

The live tracking works for both USB and BT modes, by using GPSGate or even without it, just use the *Autodetect* mode, or point the tool to the right port.

Ubuntu/Mint GNU/Linux

For USB

The live tracking works both with GPSD

```
gpsd /dev/ttyACM3
```

or without it, by connecting the QGIS live tracking tool directly to the device (for example `/dev/ttyACM3`).

For Bluetooth

The live tracking works both with GPSD

```
gpsd /dev/rfcomm0
```

or without it, by connecting the QGIS live tracking tool directly to the device (for example `/dev/rfcomm0`).

CHAPTER 21

认证系统

21.1 Authentication System Overview

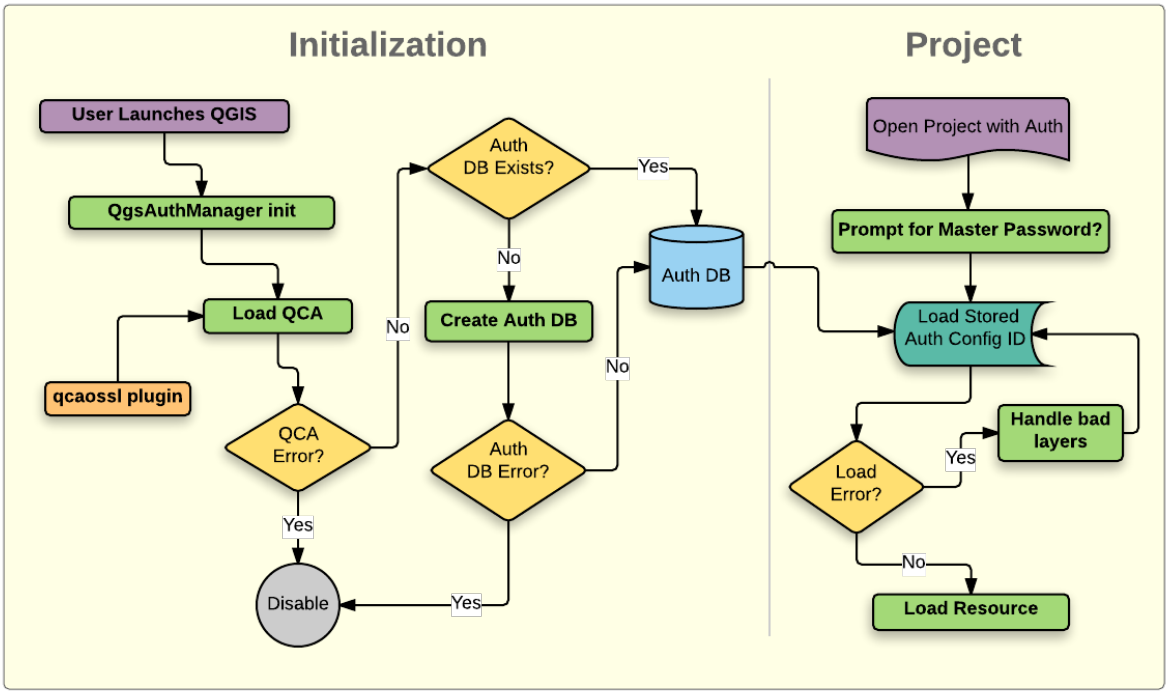


Fig. 21.1: Anatomy of authentication system

21.1.1 Authentication database

The authentication system stores authentication configurations in an SQLite database file located, by default, at `<profile directory>/qgis-auth.db`.

This authentication database can be moved between QGIS installations without affecting other current QGIS user preferences, as it is completely separate from normal QGIS settings. A configuration ID (a random 7-character alphanumeric string) is generated when initially storing a configuration to the database. This represents the configuration, thereby allowing the ID to be stored in plain text application components, (such as project, plugin, or settings files) without disclosure of its associated credentials.

Note

The parent directory of the *qgis-auth.db* can be set using the following environment variable, `QGIS_AUTH_DB_DIR_PATH`, or set on the command line during launch with the `--authdbdirectory` option.

21.1.2 Custom authentication databases

QGIS can be configured to use a custom authentication database instead of the above mentioned default SQLite one: any database suppported by the Qt SQL module can be used (e.g. PostgreSQL, MySQL, etc), provided that the corresponding Qt SQL driver is available in the system.

This can be useful in scenarios where a user wants to share the same authentication database between multiple QGIS installations, or when a user wants to use a different authentication database than the default SQLite one or when a centralized authentication database is used by QGIS server.

The only way to configure a custom authentication database is by setting the `QGIS_AUTH_DB_URI` environment variable to the URI of the connection, the URI is in the form of `driver://username:password@hostname:port/database?options`.

Where:

- `driver` is the name of the Qt SQL driver to use, e.g. `QPSQL` for PostgreSQL, `QMYSQL` for MySQL, etc.
- `username` is the username to use to connect to the database
- `password` is the password to use to connect to the database
- `hostname` is the hostname of the database server
- `port` is the port of the database server
- `database` is the name of the database to use
- `options` are the options to pass to the driver, e.g. `sslmode=require` for PostgreSQL

Note

The `schema` can be specified in the URI options, e.g. `QPSQL://username:password@hostname:port/database?schema=schema_name`

The database must exist before starting QGIS, and the user must have the necessary permissions to connect to the database and to create the required tables if they do not exist.

Warning

The password in the URI is stored in plain text in the environment variable, so it is recommended to use a passwordless user or a user with limited permissions to connect to the database.

Warning

Any database not based on SQLite is considered to be read-only (this can be changed by Python plugins if necessary).

This is an advanced feature, designed to allow one or more custom authentication databases or even custom Python implementations of credentials storages to be used by QGIS.

The system is also designed to allow for multiple authentication databases to be used but there is currently no user facing interface to manage multiple credential storages so its usage requires manual configuration and management, typically from a Python plugin.

21.1.3 Master password

To store or access sensitive information within the database, a user must define a *master password*. A new master password is requested and verified when initially storing any encrypted data to the database. When sensitive information is accessed, the user is prompted for the master password. The password is then cached for the remainder of the session (until application is quit), unless the user manually chooses an action to clear its cached value. Some instances of using the authentication system do not require input of the master password, such as when selecting an existing authentication configuration, or applying a configuration to a server configuration (such as when adding a WMS layer).

You can choose to save the password in the `Wallet/Keyring` of your computer.

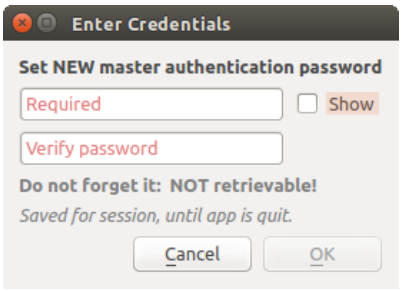


Fig. 21.2: Input new master password

Note

A path to a file containing the master password can be set using the following environment variable, `QGIS_AUTH_PASSWORD_FILE`.

Managing the master password

Once set, the master password can be reset; the current master password will be needed prior to resetting. During this process, there is an option to generate a complete backup of the current database.

Fig. 21.3: Resetting master password

If the user forgets the master password, there is no way to retrieve or override it. There is also no means of retrieving encrypted information without knowing the master password.

If a user inputs their existing password incorrectly three times, the dialog will offer to erase the database.

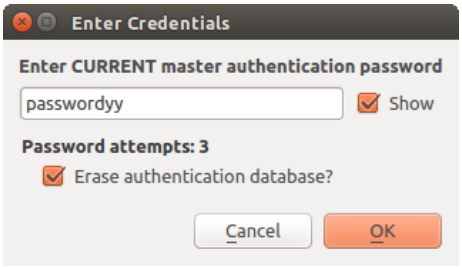


Fig. 21.4: Password prompt after three invalid attempts

21.1.4 Authentication Configurations

You can manage authentication configurations from *Configurations* in the *Authentication* tab of the QGIS Options dialog (*Settings* ► *Options*).

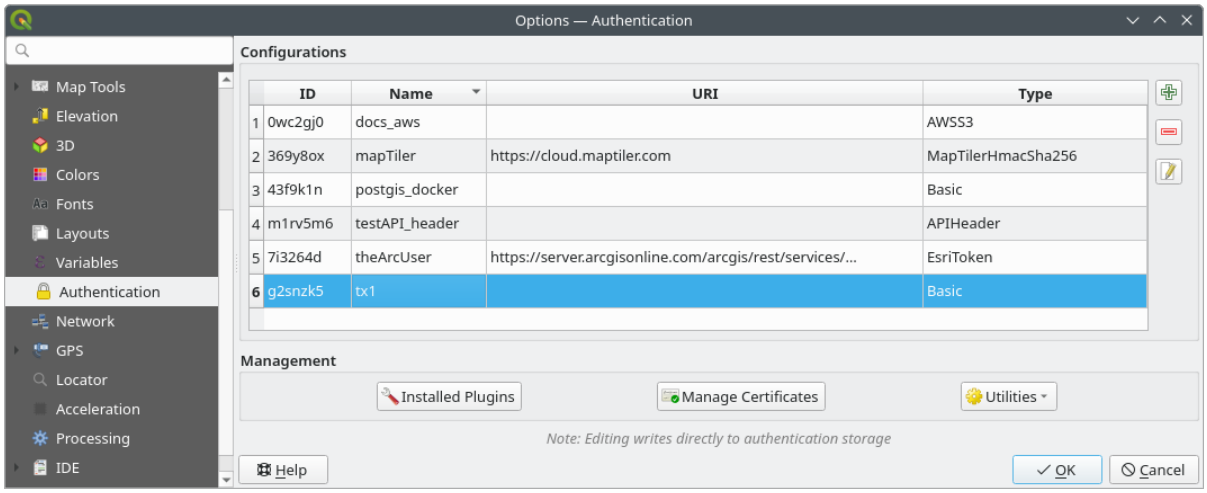


Fig. 21.5: Configurations editor

To create a new authentication configuration:

1. Press the Add a new authentication configuration
2. Provide a *Name* and optional *Resource* URL
3. A seven-characters *ID* will be assigned to the configuration for identification purpose. It can be customized after you pressed the Unlock to edit ID button. Be aware that updating that ID may break things.
4. Select the *target method* and fill the corresponding connection details
5. Once done, click *Save* to save the configuration. The saved configuration appears in the list of authentication configurations and can be used with supported connections.

The saved configuration can later be edited using Edit selected configuration or removed using Delete selected configuration.

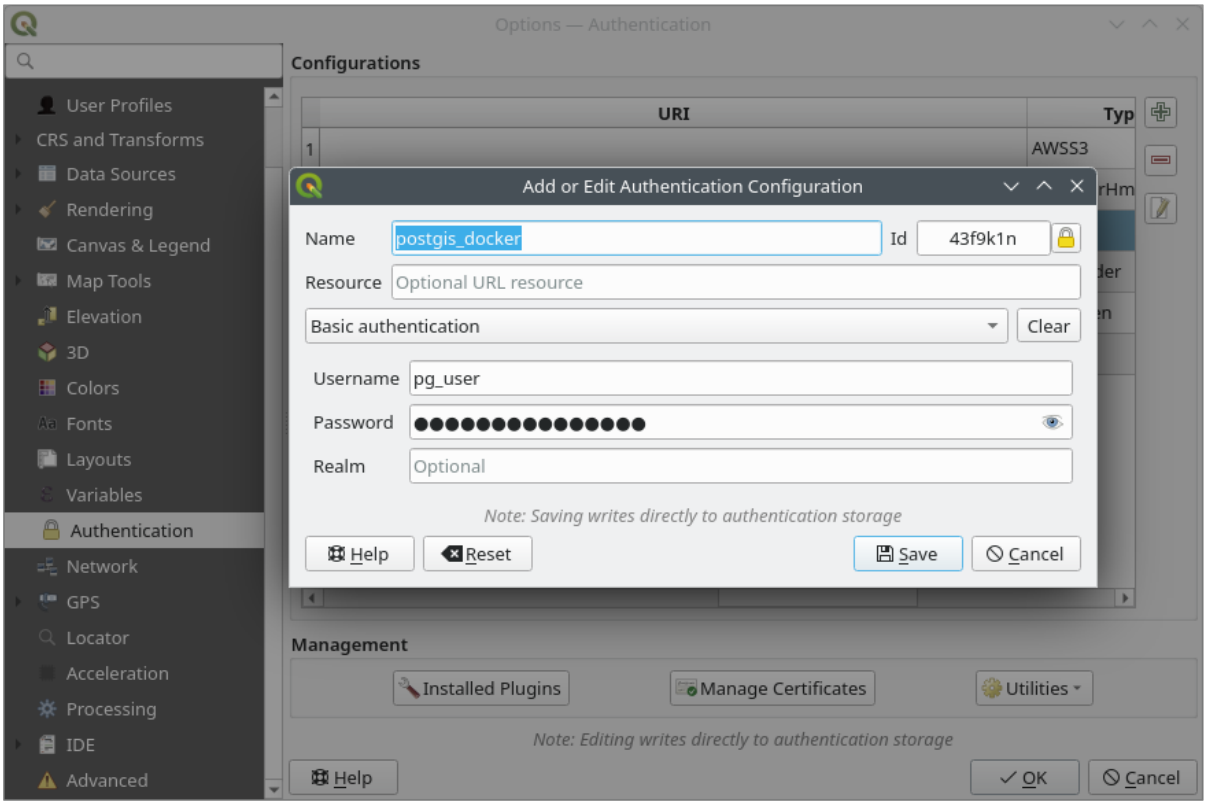


Fig. 21.6: Adding config from within Configuration editor

The same type of operations for authentication configuration management (Add, Edit and Remove) can be done when configuring a given service connection, such as configuring an OWS service connection. For that, there are action buttons within the *configuration selector* for fully managing configurations found within the authentication database. In this case, there is no need to go to the *configurations* in *Authentication* tab of QGIS options unless you need to do more comprehensive configuration management.

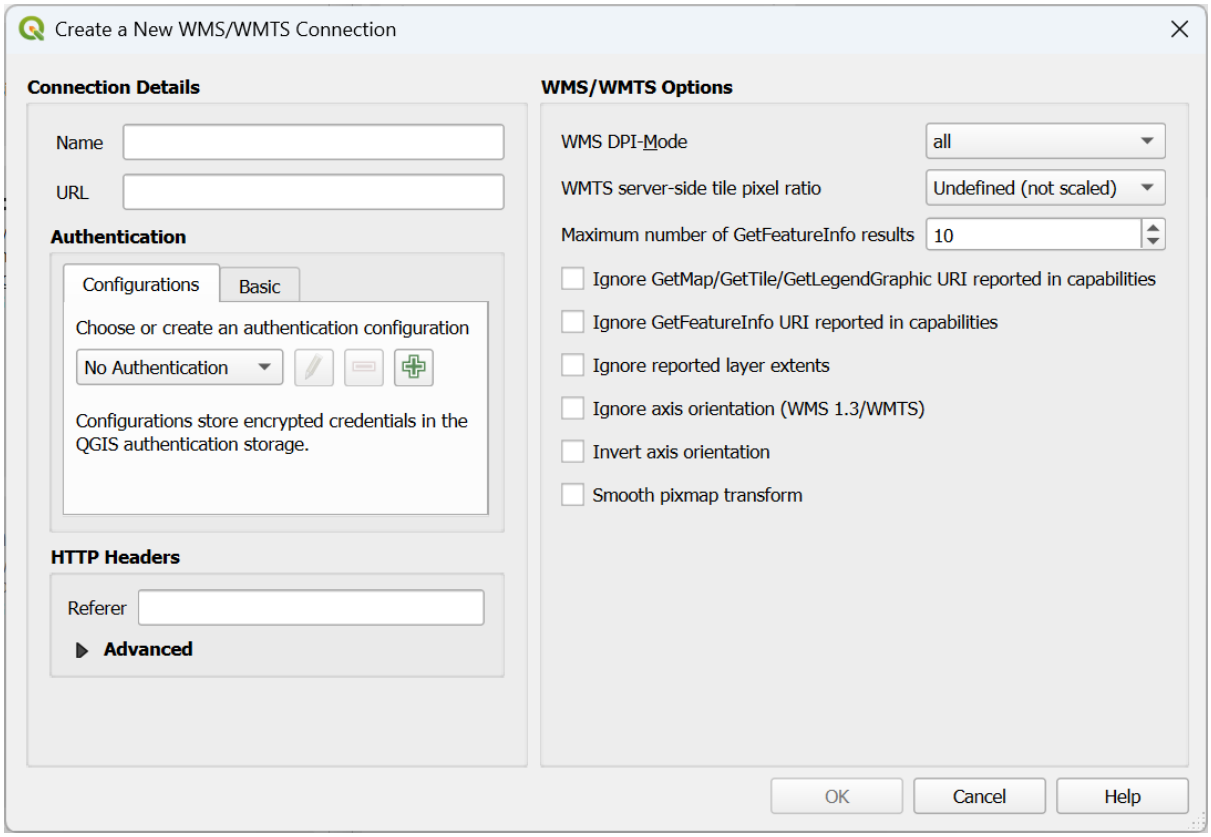


Fig. 21.7: WMS connection dialog showing authentication configuration widget

When creating or editing an authentication configuration, the info required is a name, an authentication method and any other info that the authentication method requires (see more about the available authentication types in *Authentication Methods*).

21.1.5 Authentication Methods

Available authentications are provided by C++ plugins much in the same way data provider plugins are supported by QGIS. The method of authentication that can be selected is relative to the access needed for the resource/provider, e.g. HTTP(S) or database, and whether there is support in both QGIS code and a plugin. As such, some authentication method plugins may not be applicable everywhere an authentication configuration selector is shown. A list of available authentication method plugins and their compatible resource/providers can be accessed going to *Settings* ► *Options* and, in the *Authentication* tab, click the *Installed Plugins* button.

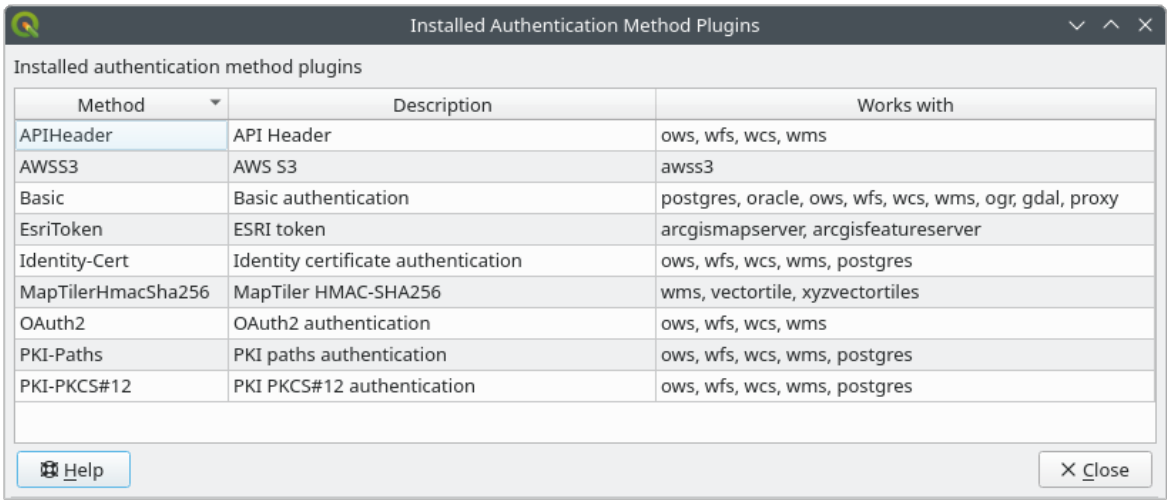


Fig. 21.8: Available method plugins list

Plugins can be created for new authentication methods that do not require QGIS to be recompiled. Since the support for plugins is currently C++-only, QGIS will need to be restarted for the new dropped-in plugin to become available to the user. Ensure your plugin is compiled against the same target version of QGIS if you intend to add it to an existing target install.

API Header Authentication

The API Header authentication method allows you to connect to services that require custom HTTP headers, such as API keys. For example, if a WMS service requires an API key sent as a header (e.g. `X-API-KEY`), you can use this method to provide it.

You can add one or more headers with the button; provide a *Header Name* and *Header Value* for each entry. Use to delete the selected header or the *Clear* button to remove all entries.

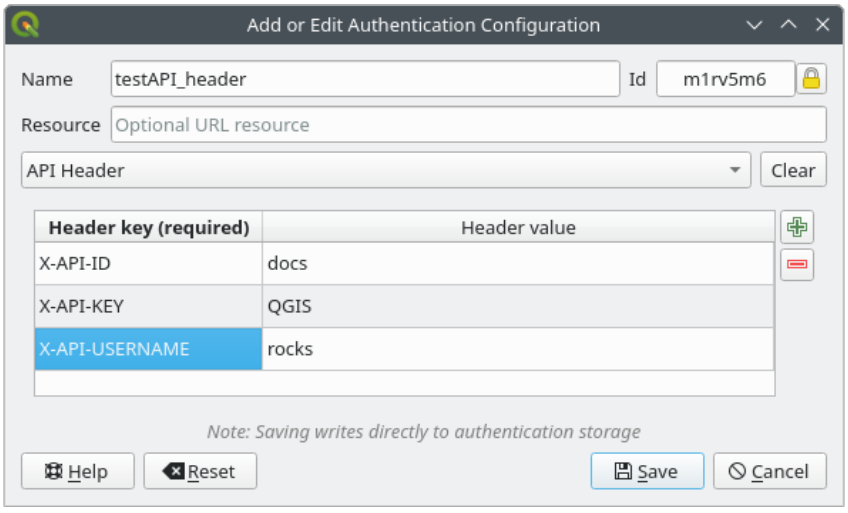


Fig. 21.9: API Header authentication configs

AWS S3 Authentication

The AWS S3 authentication method allows you to connect to Amazon S3 resources by providing a *Name*, *Resource*, *Username*, *Password*, and *Region*.

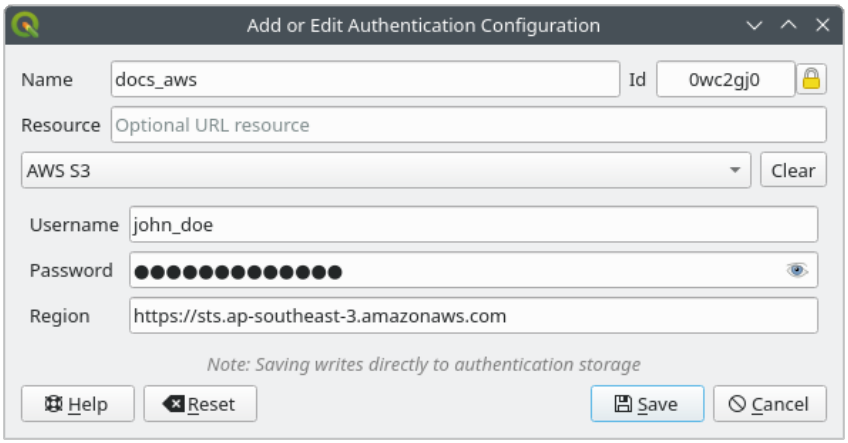


Fig. 21.10: AWS S3 authentication configs

Basic Authentication

The Basic authentication method is used for services that require standard HTTP authentication with a *Username*, *Password* and *Realm*.

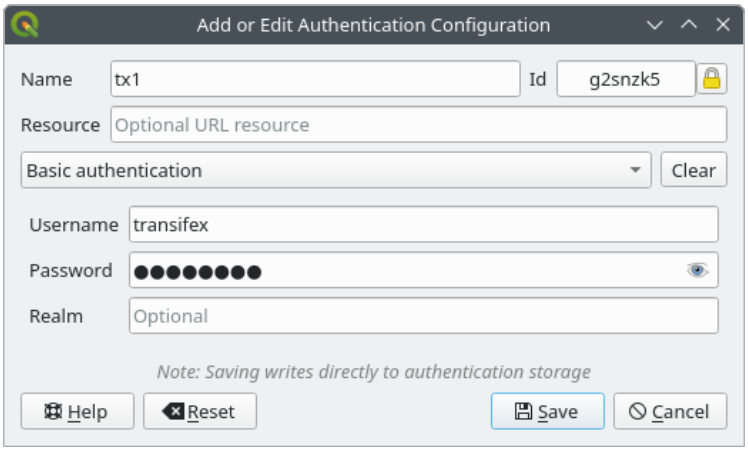


Fig. 21.11: Basic HTTP authentication configs

ESRI Token Authentication

The ESRI token authentication method is used for *ArcGIS REST Servers* that require token-based authentication.

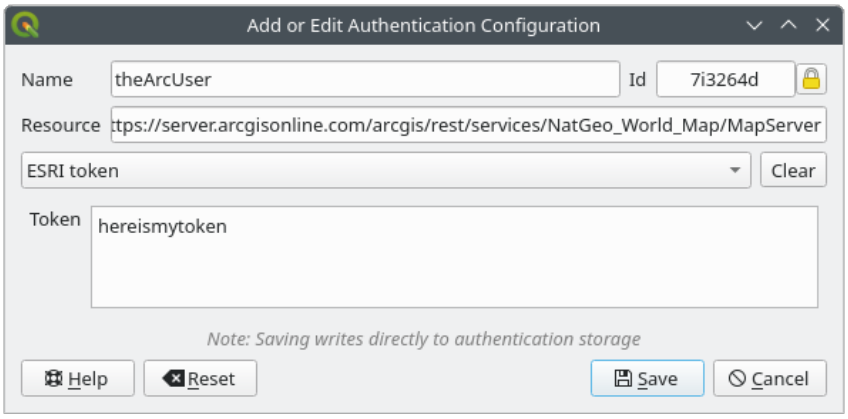


Fig. 21.12: ESRI Token authentication configs

Identity certificate authentication

The Identity certificate authentication method allows you to connect using a client *identity certificate*.

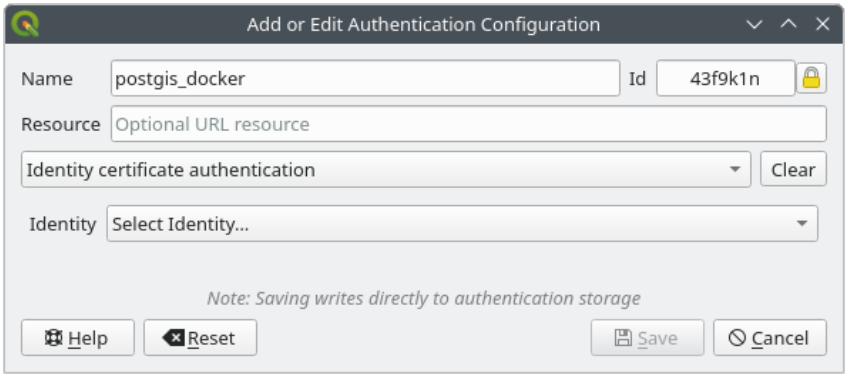


Fig. 21.13: Identity-cert authentication configs

MapTiler HMAC-SHA256 Authentication

The MapTiler HMAC-SHA256 authentication method is used to connect to MapTiler services that require HMAC-SHA256 authorization.

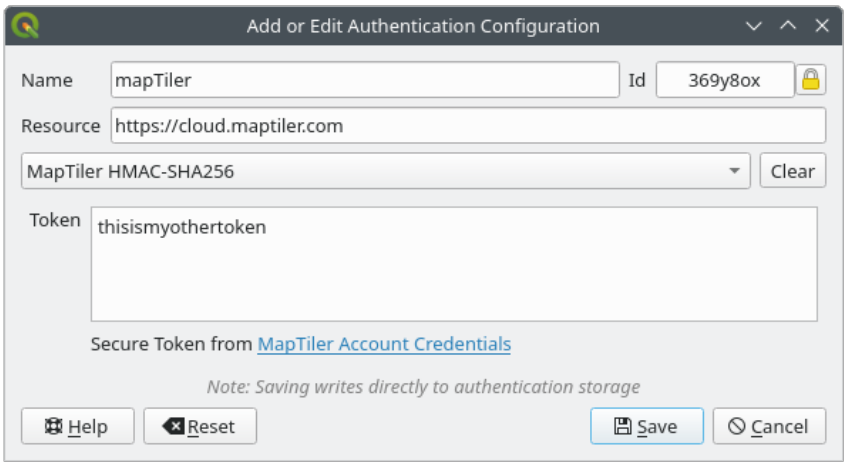


Fig. 21.14: MapTiler HmacSha256 authentication configs

OAuth2 Authentication

The OAuth2 authentication method is used to connect to services that require OAuth2 2.0 authorization, allowing secure access using client credential and token-based authentication.

It supports using `localhost` as the redirect host, providing compatibility with services that do not accept `http://127.0.0.1` redirects (for example, Microsoft SharePoint).

An advanced OAuth2 configuration option, *Extra token header(s)*, allows attaching additional tokens returned by the OAuth2 token endpoint as HTTP(S) request headers.

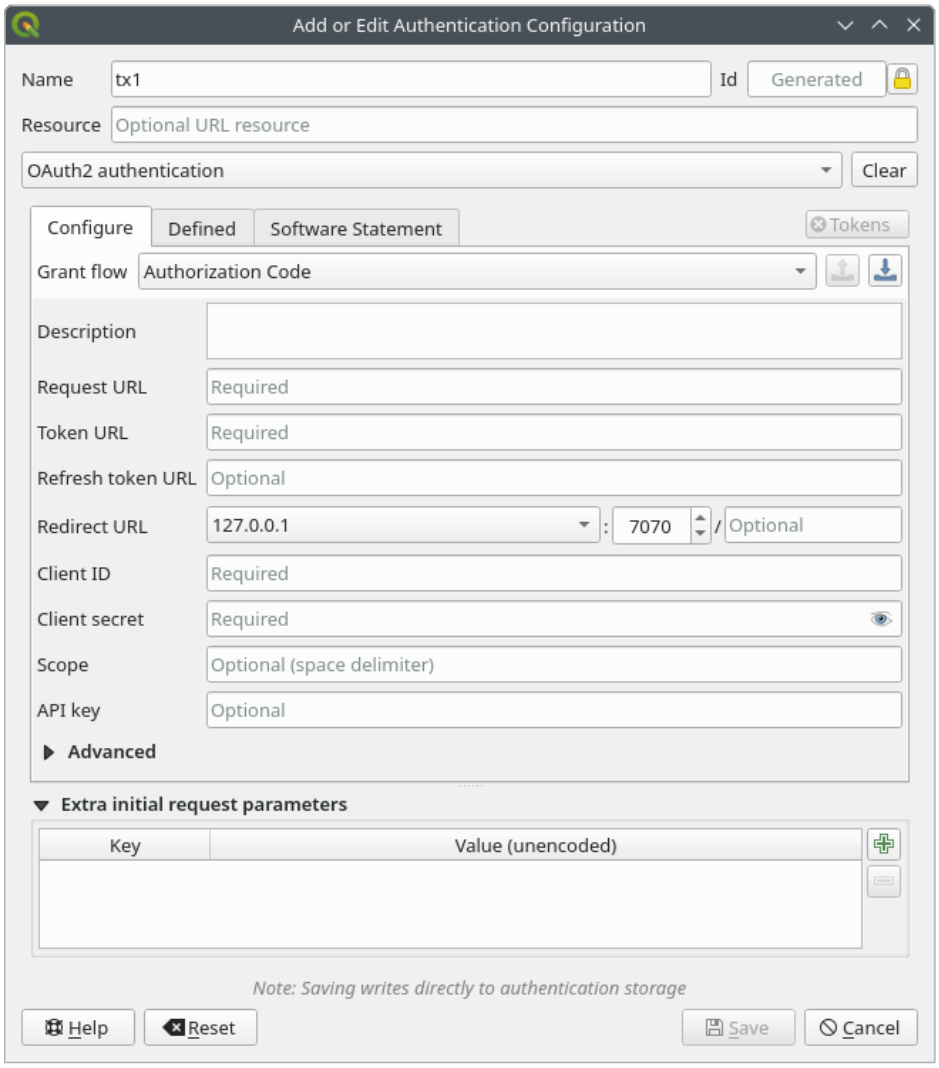


Fig. 21.15: OAuth2 authentication configs

PKI-Paths Authentication

The PKI-Paths authentication method allows you to connect using separate certificate and key files stored on your system.

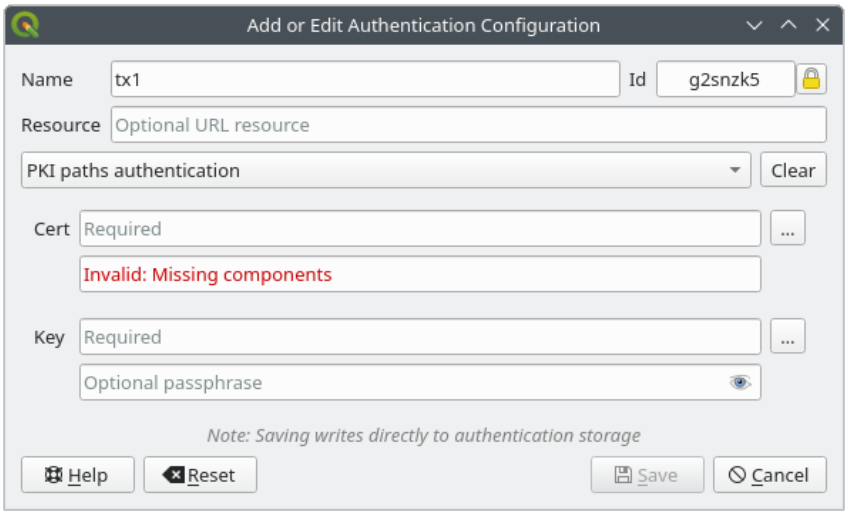


Fig. 21.16: PKI paths authentication configs

PKI-PKCS#12 Authentication

The PKI-PKCS#12 authentication method allows you to connect using a single bundle file containing both the certificate and private key, with an optional passphrase for the key.

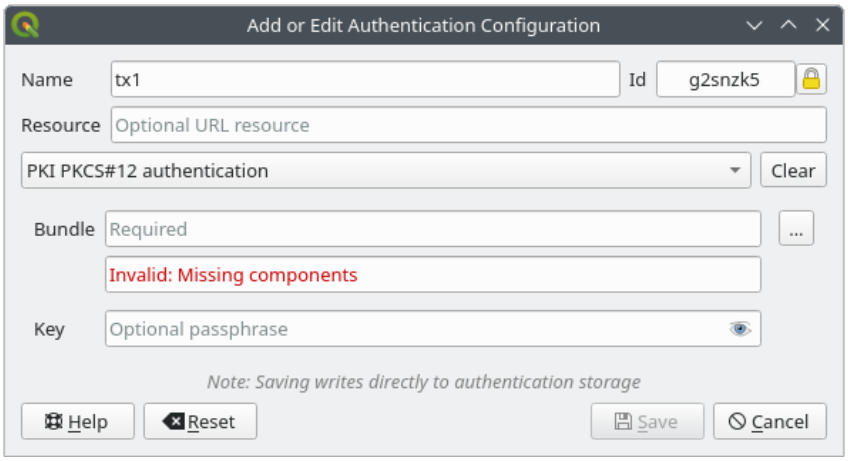


Fig. 21.17: PKI PKCS#12 file paths authentication configs

21.1.6 Master Password and Auth Config Utilities

Under the Options menu (*Settings* ► *Options*) in the *Authentication* tab, there are several utility actions to manage the authentication database and configurations:

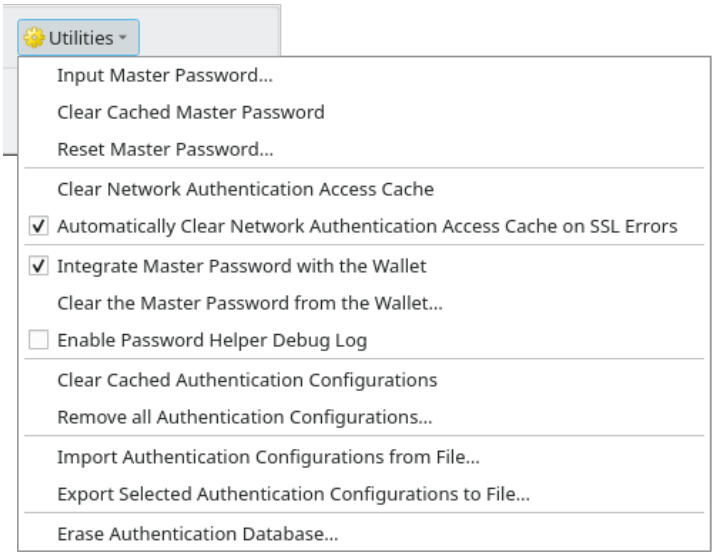


Fig. 21.18: Utilities menu

- *Input master password*···: opens the master password input dialog, independent of performing any authentication database command
- *Clear cached master password*: unsets the master password if it has been set
- *Reset master password*···: opens a dialog to change the master password (the current password must be known) and optionally back up the current database
- *Clear network authentication access cache*: clears the authentication cache of all connections
- *Automatically clear network authentication access cache on SSL errors*: the connection cache stores all authentication data for connections, also when the connection fails. If you change authentication configurations or certification authorities, you should clear the authentication cache or restart QGIS. When this option is checked, the authentication cache will be automatically cleared every time an SSL error occurs and you choose to abort the connection
- *Integrate master password with the Wallet*: adds the master password to your personal Wallet/Keyring
- *Clear the master password from the Wallet*···: deletes the master password from your Wallet/Keyring
- *Enable password helper debug log*: enables a debug tool that will contain all the log information of the authentication methods
- *Clear cached authentication configurations*: clears the internal lookup cache for configurations, used to speed up network connections. This does not clear QGIS' s core network access manager' s cache, which requires a relaunch of QGIS.
- *Remove all authentication configurations*···: clears the database of all configuration records, without removing other stored records.
- *Import authentication configurations from file*···: imports from an `.XML` file details for creating custom authentication configurations.
- *Export selected authentication configurations to file*···: exports the selected items to a possibly encrypted `.XML` file.

- *Erase authentication database…*: schedules a backup of the current database and complete rebuild of the database table structure. The actions are scheduled for a later time, to ensure that other operations, like project loading, do not interrupt the operation or cause errors due to a temporarily missing database.

21.1.7 Using authentication configurations

Typically, an authentication configuration is selected in a configuration dialog for a network services (such as WMS). However, the selector widget can be embedded anywhere authentication is needed or in non-core functionality, like in third-party PyQGIS or C++ plugins.

When using the selector, *No authentication* is displayed in the pop-up menu control when nothing is selected, when there are no configurations to choose from, or when a previously assigned configuration can no longer be found in the database. Use the drop-down menu to select an existing authentication configuration or press

to create a configuration you could use. More details at 认证.

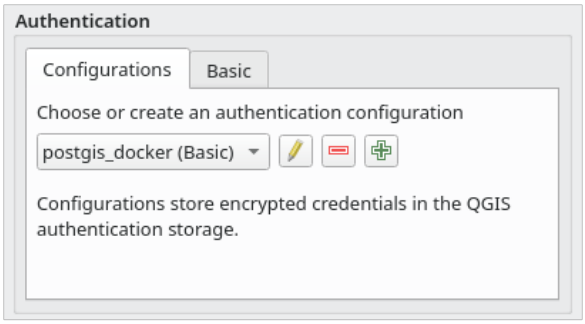


Fig. 21.19: Authentication configuration selector with selected config

21.1.8 Python bindings

All classes and public functions have sip bindings, except QgsAuthCrypto, since management of the master password hashing and auth database encryption should be handled by the main app, and not via Python.

Once the master password is entered, the API is open to access authentication configs in the authentication database, similar to how Firefox works. However, no wall against PyQGIS access has been defined. This may lead to issues where a user downloads/installs a malicious PyQGIS plugin or standalone app that gains access to authentication credentials.

21.2 User Authentication Workflows

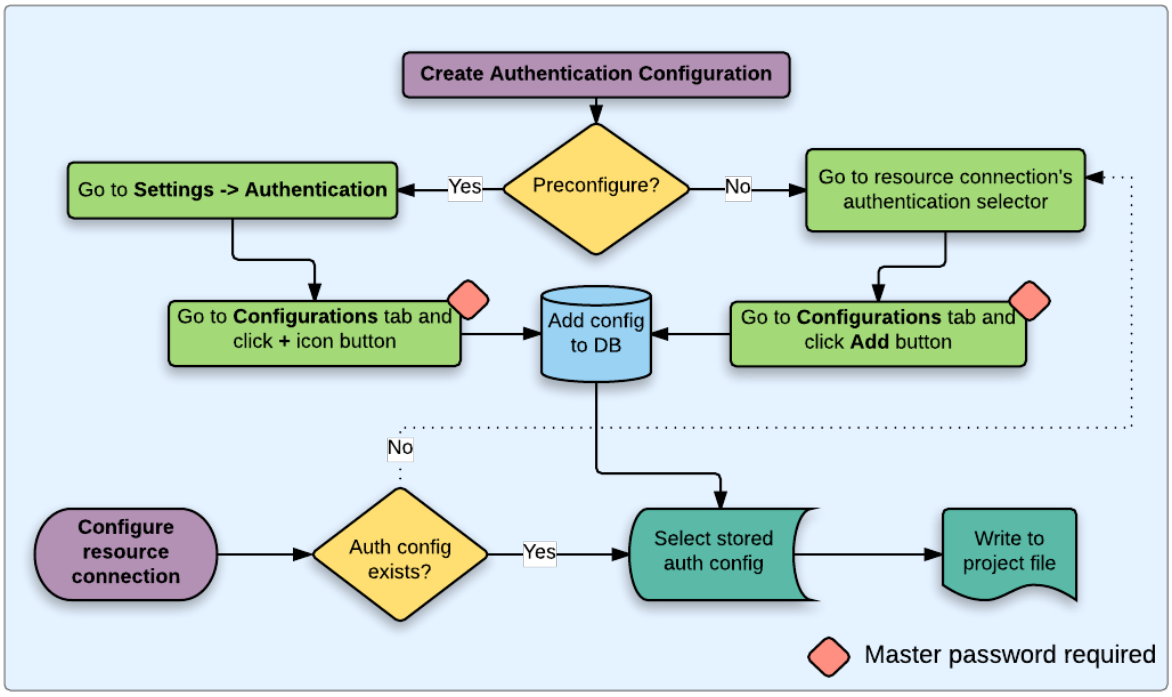


Fig. 21.20: Generic user workflow

21.2.1 HTTP(S) authentication

One of the most common resource connections is via HTTP(S), e.g. web mapping servers, and authentication method plugins often work for these types of connections. Method plugins have access to the HTTP request object and can manipulate both the request as well as its headers. This allows for many forms of internet-based authentication. When connecting via HTTP(S) using the standard username/password authentication method will attempt HTTP BASIC authentication upon connection.

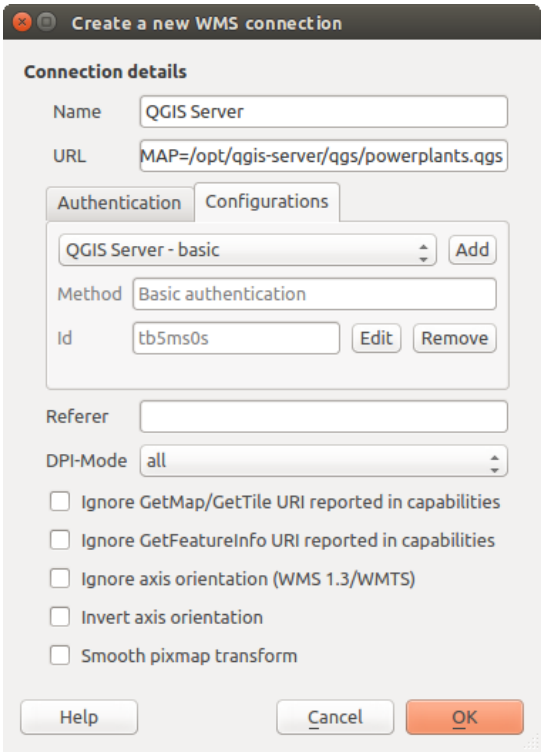


Fig. 21.21: Configuring a WMS connection for HTTP BASIC

21.2.2 Database authentication

Connections to database resources are generally stored as `key=value` pairs, which will expose usernames and (optionally) passwords, if *not* using an authentication configuration. When configuring with the auth system, the `key=value` will be an abstracted representation of the credentials, e.g. `authfg=81t21b9`.

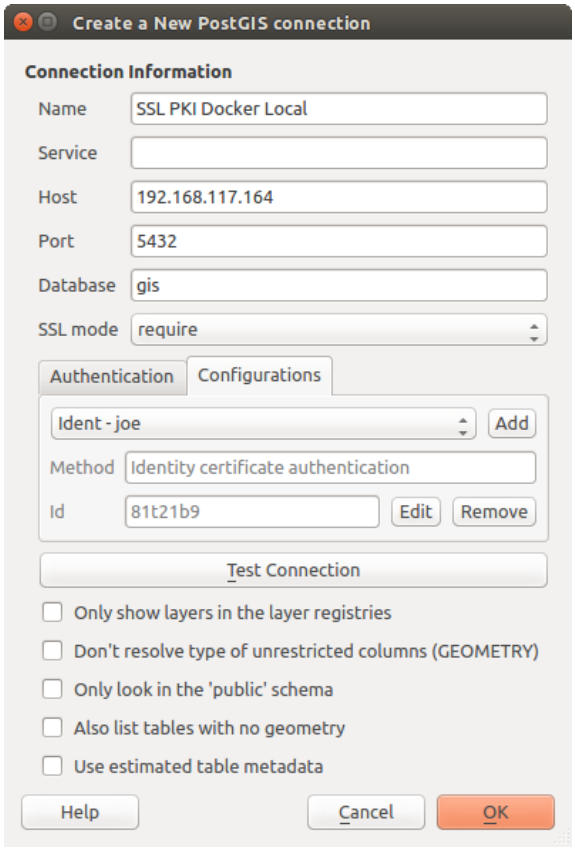


Fig. 21.22: Configuring a Postgres SSL-with-PKI connection

21.2.3 PKI authentication

When configuring PKI components within the authentication system, you have the option of importing components into the database or referencing component files stored on your filesystem. The latter may be useful if such components change frequently, or where the components will be replaced by a system administrator. In either instance you will need to store any passphrase needed to access private keys within the database.

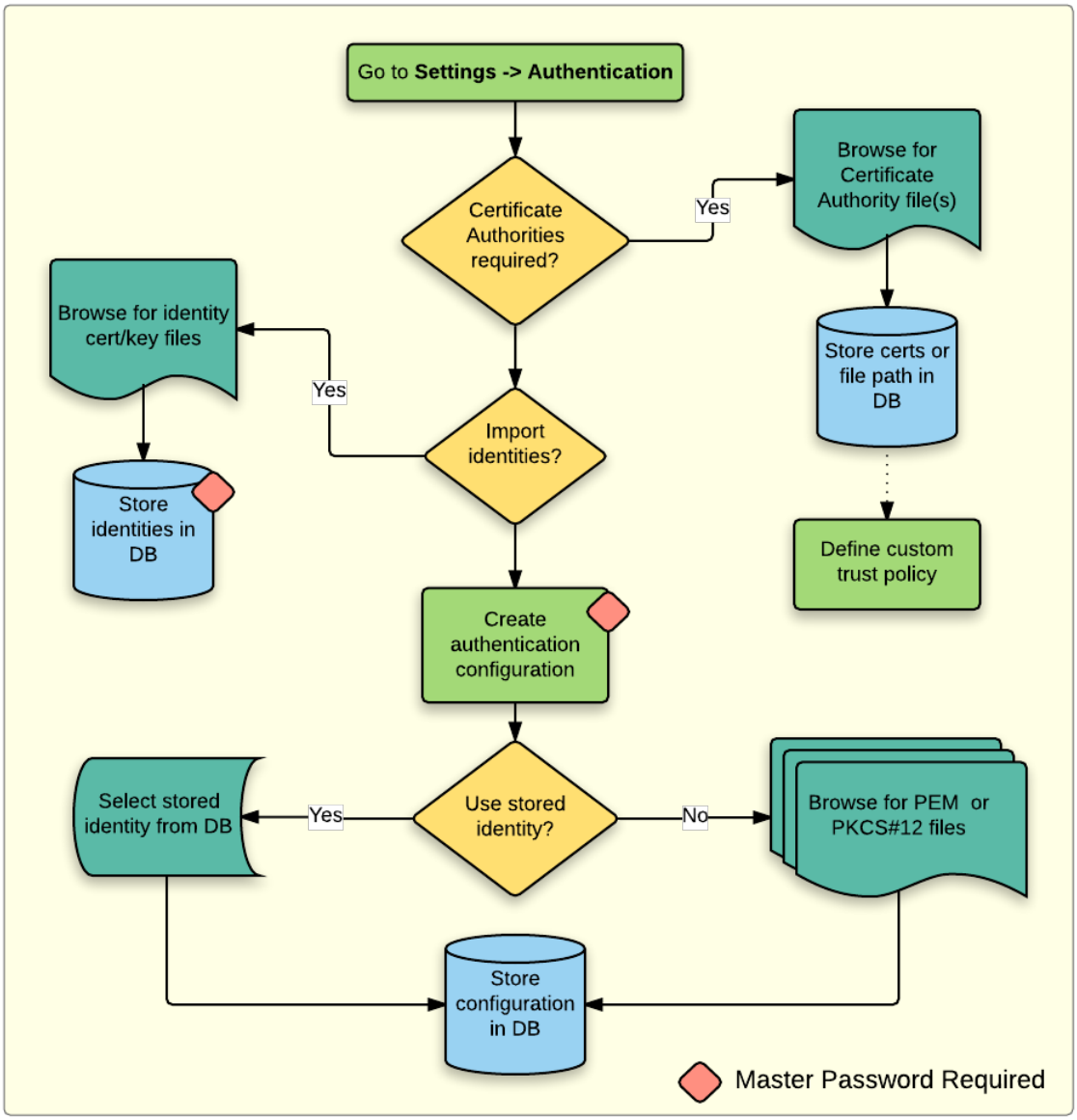


Fig. 21.23: PKI configuration workflow

All PKI components can be managed in separate editors within the **Certificate Manager**, which can be accessed in the *Authentication* tab in QGIS *Options* dialog (*Settings* ► *Options*) by clicking the *Manage Certificates* button.

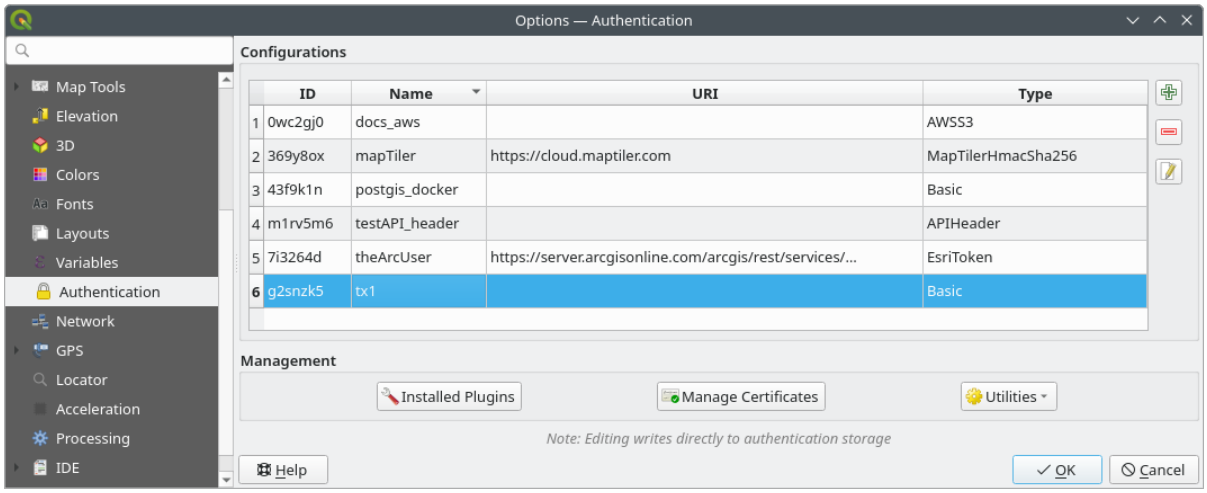


Fig. 21.24: Opening the Certificate Manager

In the *Certificate Manager*, there are editors for **Identities**, **Servers** and **Authorities**. Each of these are contained in their own tabs, and are described below in the order they are encountered in the workflow chart above. The tab order is relative to frequently accessed editors once you are accustomed to the workflow.

> **Note**
>
> Because all authentication system edits write immediately to the authentication database, there is no need to click the *Options* dialog *OK* button for any changes to be saved. This is unlike other settings in the Options dialog.

Authorities

You can manage available Certificate Authorities (CAs) from the **Authorities** tab in the **Certificate manager** from the **Authentication** tab of the QGIS **Options** dialog.

As referenced in the workflow chart above, the first step is to import or reference a file of CAs. This step is optional, and may be unnecessary if your PKI trust chain originates from root CAs already installed in your operating system (OS), such as a certificate from a commercial certificate vendor. If your authenticating root CA is not in the OS's trusted root CAs, it will need to be imported or have its file system path referenced. (Contact your system administrator if unsure.)

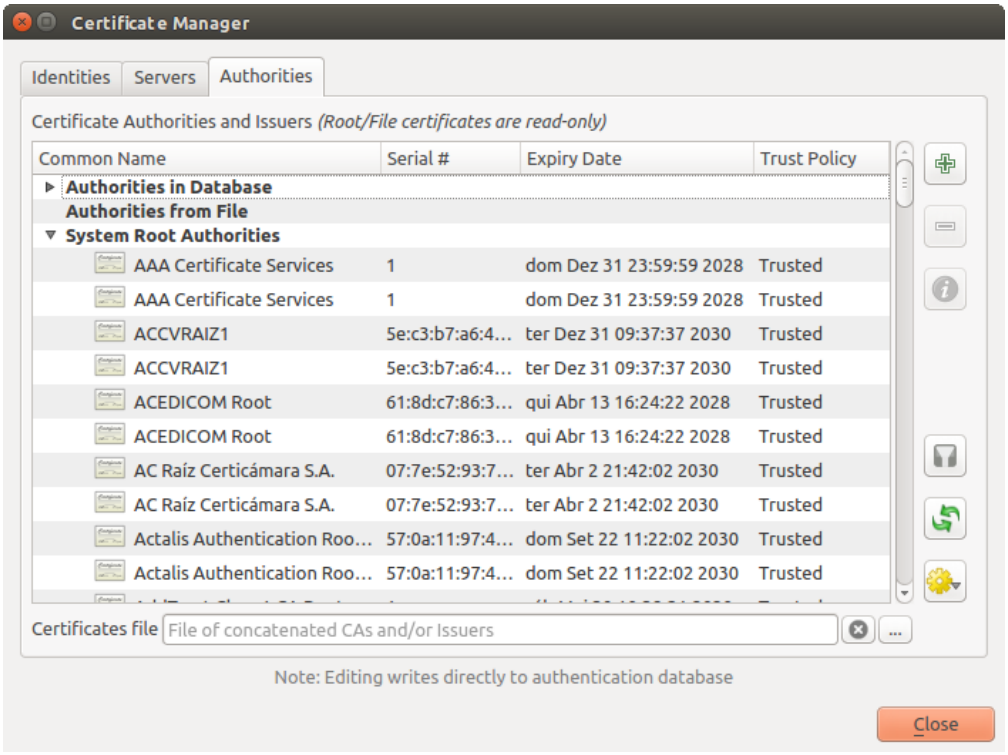


Fig. 21.25: Authorities editor

By default, the root CAs from your OS are available; however, their trust settings are not inherited. You should review the certificate trust policy settings, especially if your OS root CAs have had their policies adjusted. Any certificate that is expired will be set to untrusted and will not be used in secure server connections, unless you specifically override its trust policy. To see the QGIS-discoverable trust chain for any certificate, select it and click the Show information for certificate.

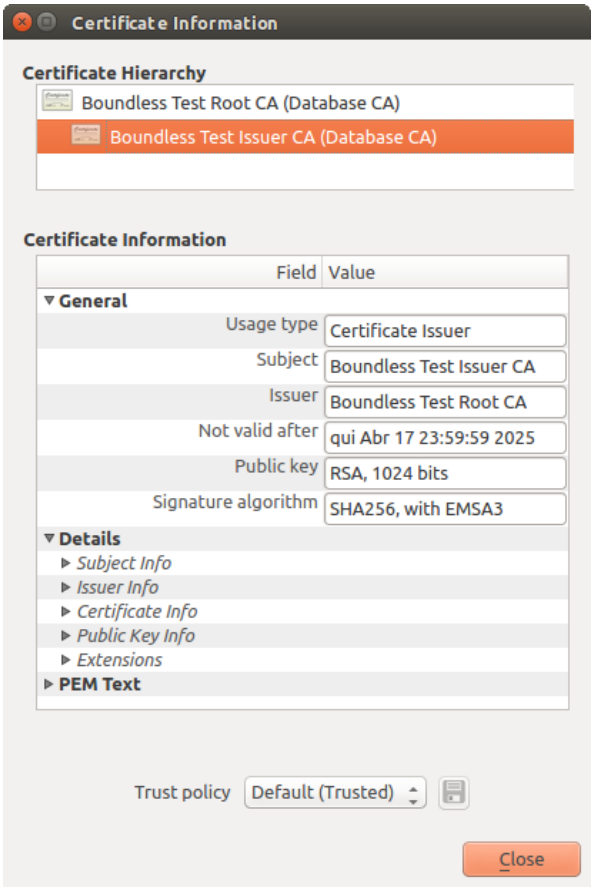


Fig. 21.26: Certificate info dialog

You can edit the *Trust policy* for any selected certificate within the chain. Any change in trust policy to a selected certificate will not be saved to the database unless the Save certificate trust policy change to database button is clicked *per* selected certification. Closing the dialog will **not** apply the policy changes.

You can review the filtered CAs, both intermediate and root certificates, that will be trusted for secure connections or change the default trust policy by clicking the **Options** button.

Warning

Changing the default trust policy may result in problems with secure connections.

You can import CAs or save a file system path from a file that contains multiple CAs, or import individual CAs. The standard PEM format for files that contain multiple CA chain certifications has the root cert at the bottom of the file and all subsequently signed child certificates above, towards the beginning of the file.

The CA certificate import dialog will find all CA certificates within the file, regardless of order, and also offers the option to import certificates that are considered invalid (in case you want to override their trust policy). You can override the trust policy upon import, or do so later within the **Authorities** editor.

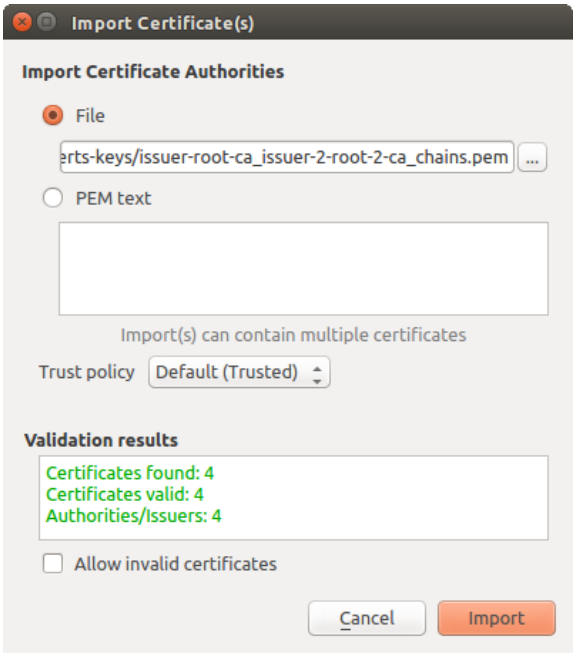


Fig. 21.27: Import certificates dialog

Note

If you are pasting certificate information into the *PEM text* field, note that encrypted certificates are not supported.

Identities

You can manage available client identity bundles from the *Identities* tab in the *Certificate manager* from the **Authentication** tab of the QGIS **Options** dialog. An identity is what authenticates you against a PKI-enabled service and usually consists of a client certificate and private key, either as separate files or combined into a single "bundled" file. The bundle or private key is often passphrase-protected.

Once you have any Certificate Authorities (CAs) imported you can optionally import any identity bundles into the authentication database. If you do not wish to store the identities, you can reference their component file system paths within an individual authentication configuration.

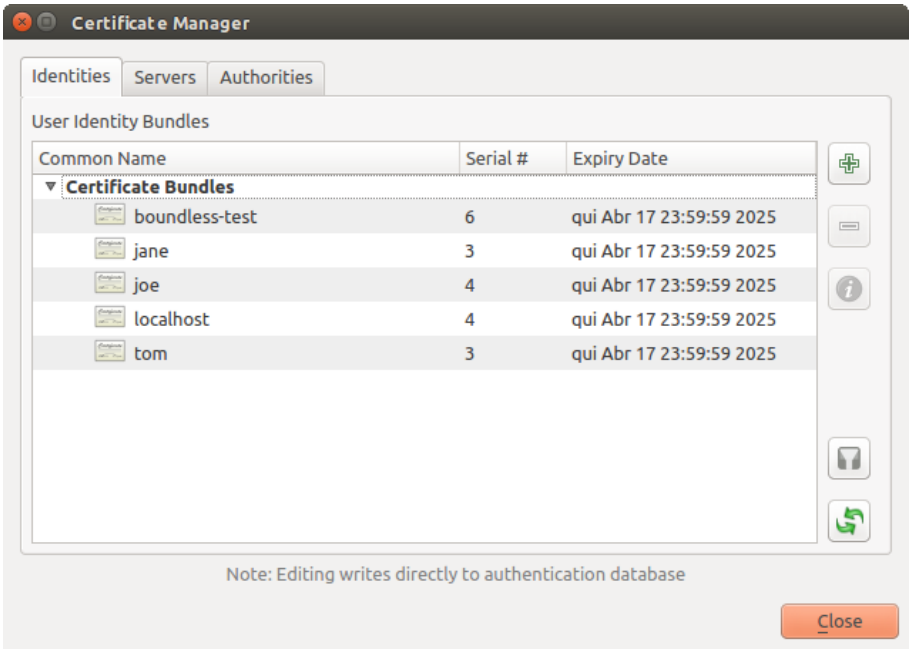


Fig. 21.28: Identities editor

When importing an identity bundle, it can be passphrase-protected or unprotected, and can contain CA certificates forming a trust chain. Trust chain certifications will not be imported here; they can be added separately under the *Authorities* tab.

Upon import the bundle's certificate and private key will be stored in the database, with the key's storage encrypted using the QGIS master password. Subsequent usage of the stored bundle from the database will only require input of the master password.

Personal identity bundles consisting of PEM/DER (.pem/.der) and PKCS#12 (.p12/.pfx) components are supported. If a key or bundle is passphrase-protected, the password will be required to validate the component prior to import. Likewise, if the client certificate in the bundle is invalid (for example, its effective date has not yet started or has elapsed) the bundle can not be imported.

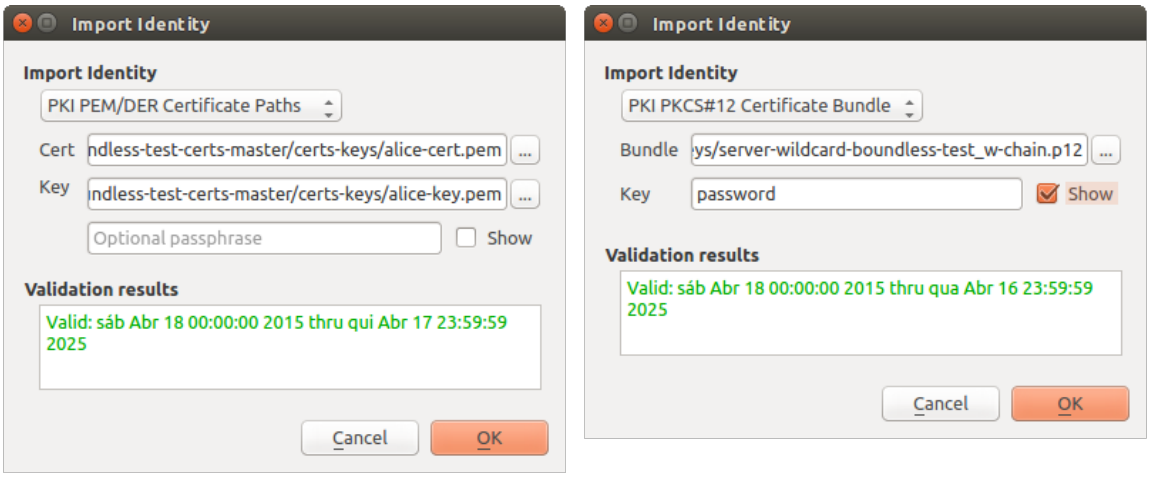


Fig. 21.29: PEM/DER and PKCS#12 identities import

21.2.4 Handling bad layers

Occasionally, the authentication configuration ID that is saved with a project file is no longer valid, possibly because the current authentication database is different than when the project was last saved, or due to a credentials mismatch. In such cases the *Handle bad layers* dialog will be presented upon QGIS launch.

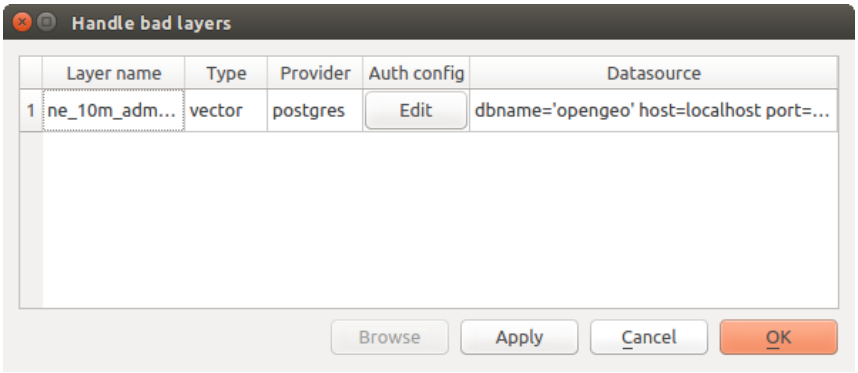


Fig. 21.30: Handle bad layers with authentication

If a data source is found to have an authentication configuration ID associated with it, you will be able to edit it. Doing so will automatically edit the data source string, much in the same way as opening the project file in a text editor and editing the string.

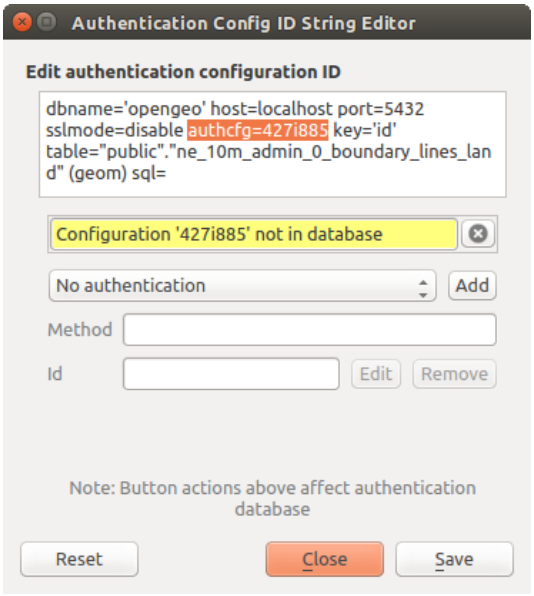


Fig. 21.31: Edit bad layer's authentication config ID

21.2.5 Changing authentication config ID

Occasionally, you will need to change the authentication configuration ID that is associated with accessing a resource. There are instances where this is useful:

- **Resource auth config ID is no longer valid**: This can occur when you have switched auth databases add need to *align* a new configuration to the ID already associated with a resource.
- **Shared project files**: If you intended to share projects between users, e.g. via a shared file server, you can *predefine* a 7-character (containing **a-z** and/or **0-9**) that is associated with the resource. Then, individual users change the ID of an authentication configuration that is specific to their credentials of the resource. When the project is opened, the ID is found in the authentication database, but the credentials are different per user.

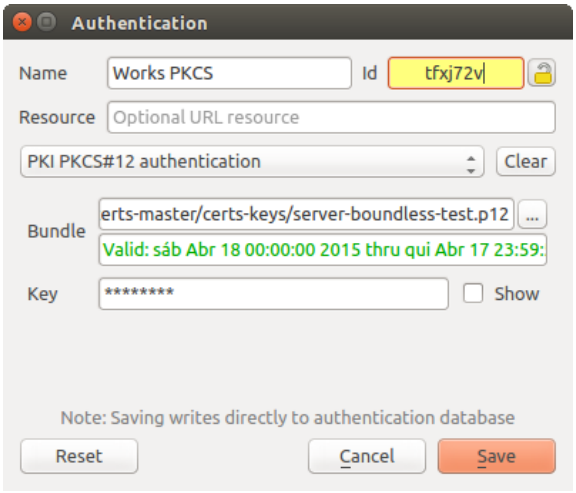


Fig. 21.32: Changing a layer's authentication config ID (unlocked yellow text field)

Warning

Changing the auth config ID is considered an advanced operation and should only be done with full knowledge as to why it is necessary. This is why there is a lock button that needs clicked, to unlock the ID's text field prior to editing the ID.

21.2.6 QGIS Server support

When using a project file, with layers that have authentication configurations, as a basis for a map in QGIS Server, there are a couple of additional setup steps necessary for QGIS to load the resources:

- Authentication database needs to be available
- Authentication database's master password needs to be available

When instantiating the authentication system, Server will create or use `qgis-auth.db` file in the active *user profile*, or the directory defined by the `QGIS_AUTH_DB_DIR_PATH` environment variable. It may be that the Server's user has no HOME directory, in which case, use the environment variable to define a directory that the Server's user has read/write permissions and is not located within the web-accessible directories.

To pass the master password to Server, write it to the first line of file at a path on the file system readable by the Server processes user and defined using the `QGIS_AUTH_PASSWORD_FILE` environment variable. Ensure to limit the file as only readable by the Server's process user and to not store the file within web-accessible directories.

Note

`QGIS_AUTH_PASSWORD_FILE` variable will be removed from the Server environment immediately after accessing.

21.2.7 SSL server exceptions

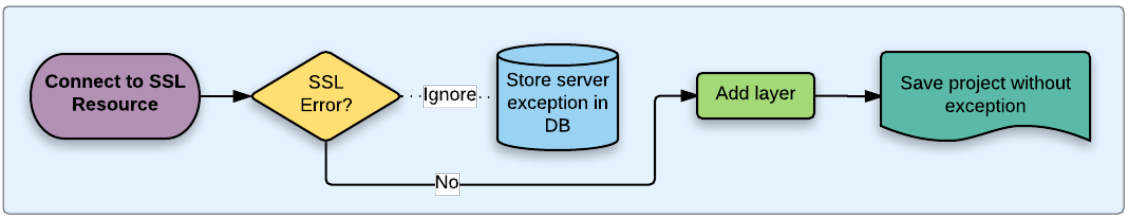


Fig. 21.33: SSL server exception

You can manage SSL server configurations and exceptions from the **Servers** tab in the **Authentication** section of the QGIS **Options** dialog.

Sometimes, when connecting to an SSL server, there are errors with the SSL "handshake" or the server's certificate. You can ignore those errors or create an SSL server configuration as an exception. This is similar to how web browsers allow you to override SSL errors, but with more granular control.

Warning

You should not create an SSL server configuration unless you have complete knowledge of the entire SSL setup between the server and client. Instead, report the issue to the server administrator.

Note

Some PKI setups use a completely different CA trust chain to validate client identities than the chain used to validate the SSL server certificate. In such circumstances, any configuration created for the connecting server will not necessarily fix an issue with the validation of your client identity, and only your client identity's issuer or server administrator can fix the issue.

You can pre-configure an SSL server configuration by clicking the button. Alternatively, you can add a configuration when an SSL error occurs during a connection and you are presented with an **SSL Error** dialog (where the error can be ignored temporarily or saved to the database and ignored):

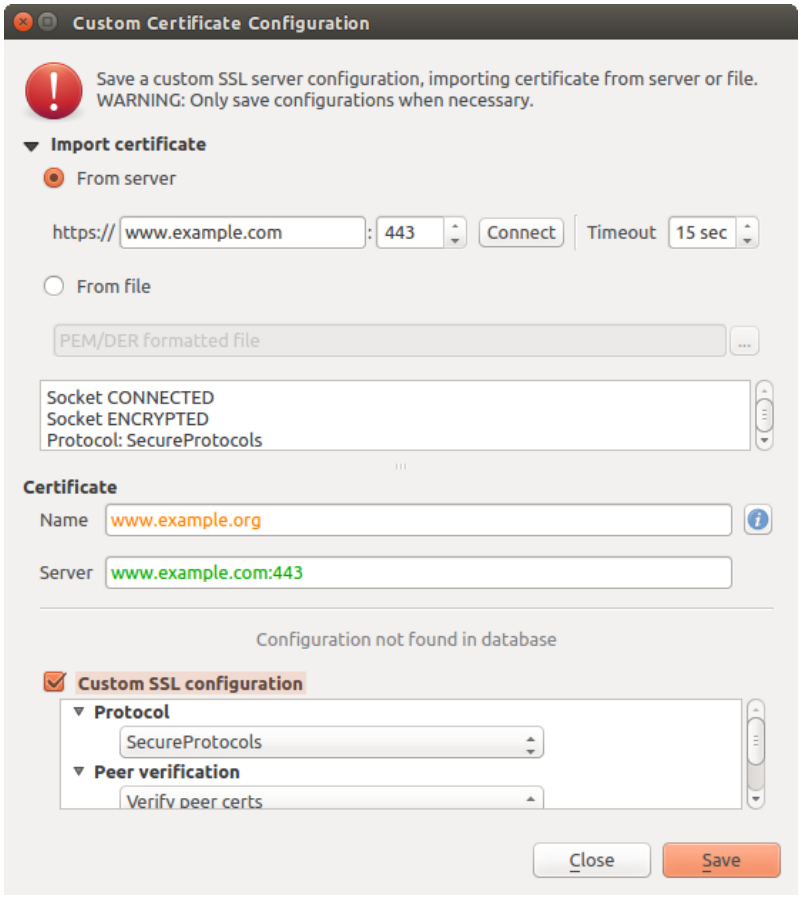


Fig. 21.34: Manually adding configuration

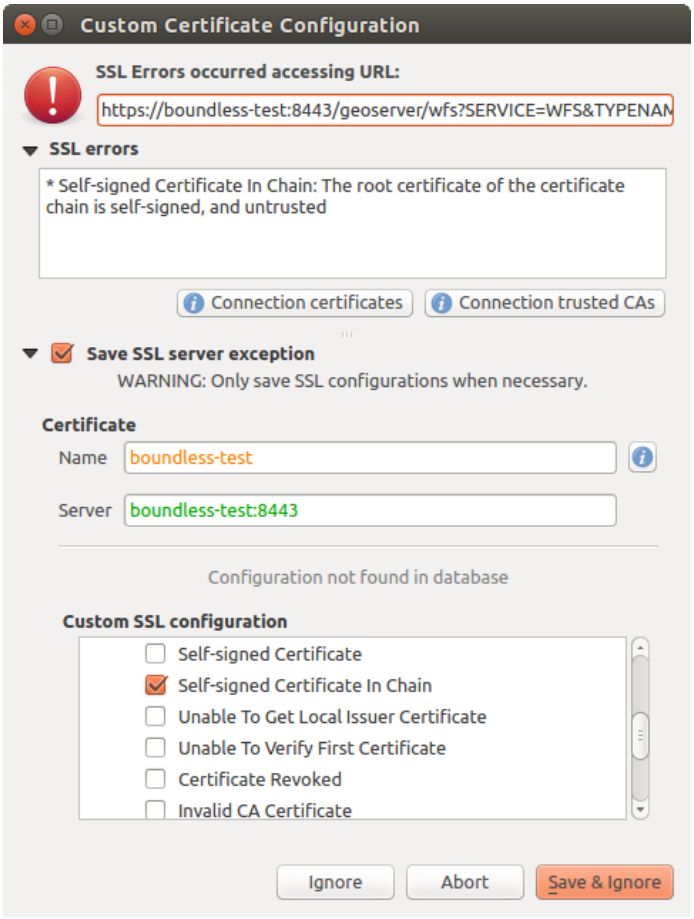


Fig. 21.35: Adding configuration during SSL error

Once an SSL configuration is saved to the database, it can be edited or deleted.

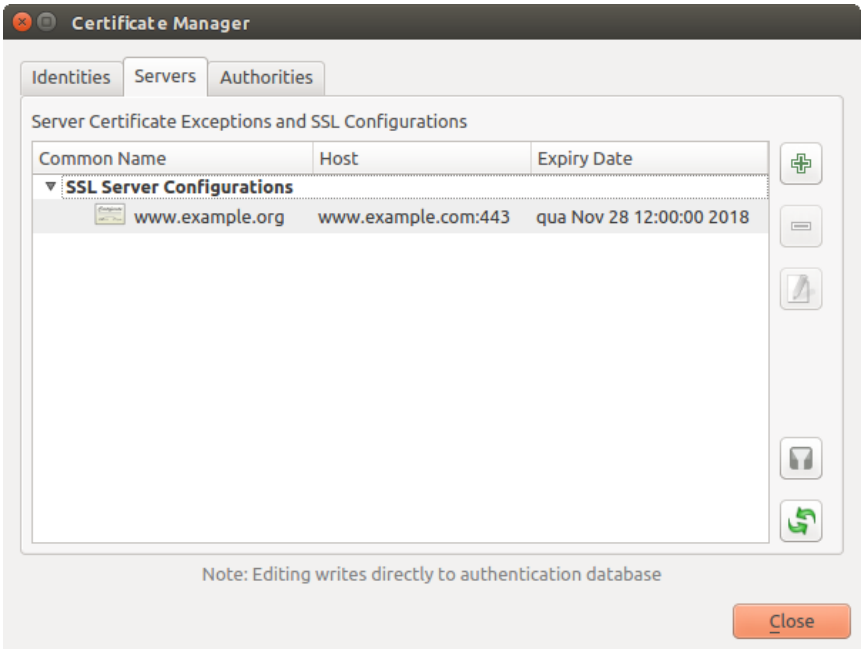


Fig. 21.36: Existing SSL configuration

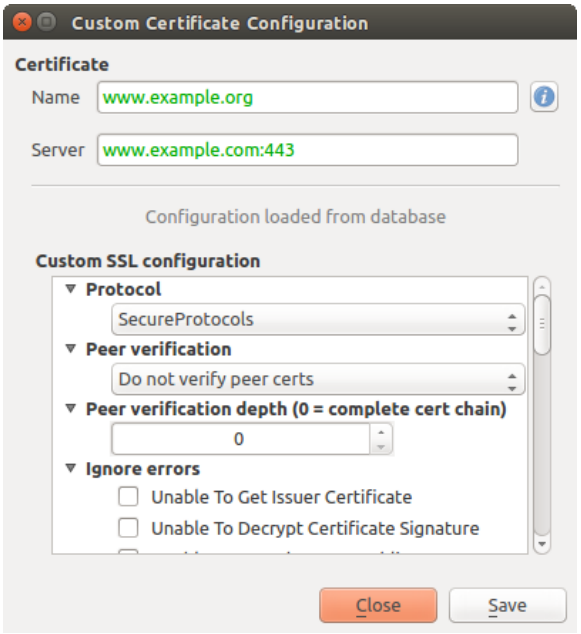


Fig. 21.37: Editing an existing SSL configuration

If you want to pre-configure an SSL configuration and the import dialog is not working for your server's connection, you can manually trigger a connection via the **Python Console** by running the following code (replace `https://bugreports.qt-project.org` with the URL of your server):

```
from qgis.PyQt.QtNetwork import QNetworkRequest
from qgis.PyQt.QtCore import QUrl
from qgis.core import QgsNetworkAccessManager

req = QNetworkRequest(QUrl('https://bugreports.qt-project.org'))
reply = QgsNetworkAccessManager.instance().get(req)
```

This will open an SSL error dialog if any errors occur, where you can choose to save the configuration to the database.

CHAPTER 22

GRASS GIS Integration

GRASS integration provides access to GRASS GIS databases and functionalities (see GRASS-PROJECT in 文献和网络参考). The integration consists of two parts: provider and plugin. The provider allows to browse, manage and visualize GRASS raster and vector layers. The plugin can be used to create new GRASS locations and mapsets, change GRASS region, create and edit vector layers and analyze GRASS 2-D and 3-D data with more than 400 GRASS modules. In this section, we'll introduce the provider and plugin functionalities and give some examples of managing and working with GRASS data.

The provider supports GRASS version 6 and 7, the plugin supports GRASS 6 and 7 (starting from QGIS 2.12). QGIS distribution may contain provider/plugin for either GRASS 6 or GRASS 7 or for both versions at the same time (binaries have different file names). Only one version of the provider/plugin may be loaded on runtime however.

22.1 Demo dataset

As an example, we will use the QGIS Alaska dataset (see section 下载示例数据). It includes a small sample GRASS `LOCATION` with three vector layers and one raster elevation map. Create a new folder called `grassdata`, download the QGIS 'Alaska' dataset `qgis_sample_data.zip` from https://qgis.org/downloads-list/#data and unzip the file into `grassdata`.

More sample GRASS `LOCATIONs` are available at the GRASS website.

22.2 Loading GRASS raster and vector layers

If the provider is loaded in QGIS, the location item with GRASS icon is added in the browser tree under each folder item which contains GRASS location. Go to the folder `grassdata` and expand location `alaska` and mapset `demo`.

You can load GRASS raster and vector layers like any other layer from the browser either by double click on layer item or by dragging and dropping to map canvas or legend.

Tip

GRASS Data Loading

If you don't see GRASS location item, verify in *Help* ► *About* ► *Providers* if GRASS vector provider is loaded.

22.3 Importing data into a GRASS LOCATION via drag and drop

This section gives an example of how to import raster and vector data into a GRASS mapset.

1. In QGIS browser navigate to the mapset you want to import data into.
2. In QGIS browser find a layer you want to import to GRASS, note that you can open another instance of the browser (*Browser Panel (2)*) if source data are too far from the mapset in the tree.
3. Drag a layer and drop it on the target mapset. The import may take some time for larger layers, you will see animated icon in front of new layer item until the import finishes.

When raster data are in different CRS, they can be reprojected using an *Approximate* (fast) or *Exact* (precise) transformation. If a link to the source raster is created (using `r.external`), the source data are in the same CRS and the format is known to GDAL, the source data CRS will be used. You can set these options in the *Browser* tab in *GRASS Options*.

If a source raster has more bands, a new GRASS map is created for each layer with **.<band number>** suffix and group of all maps with icon is created. External rasters have a different icon .

22.4 Managing GRASS data in QGIS Browser

- Copying maps: GRASS maps may be copied between mapsets within the same location using drag and drop.
- Deleting maps: Right click on a GRASS map and select *Delete* from context menu.
- Renaming maps: Right click on a GRASS map and select *Rename* from context menu.

22.5 GRASS Options

GRASS options may be set in *GRASS Options* dialog, which can be opened by right clicking on the location or mapset item in the browser and then choosing *GRASS Options*.

22.6 Starting the GRASS plugin

To use GRASS functionalities in QGIS, you must select and load the GRASS plugin using the Plugin Manager. To do this, go to the menu *Plugins* ► *Manage and Install Plugins...*, select *GRASS* and click *OK*.

The following main features are provided with the GRASS menu (*Plugins* ► *GRASS*) when you start the GRASS plugin:

- Open Mapset
- New Mapset
- Close Mapset
- Open GRASS Tools
- Display Current GRASS Region

- GRASS Options

22.7 Opening GRASS mapset

A GRASS mapset must be opened to get access to GRASS Tools in the plugin (the tools are disabled if no mapset is open). You can open a mapset from the browser: right click on mapset item and then choose *Open mapset* from context menu.

22.8 GRASS LOCATION and MAPSET

GRASS data are stored in a directory referred to as GISDBASE. This directory, often called `grassdata`, must be created before you start working with the GRASS plugin in QGIS. Within this directory, the GRASS GIS data are organized by projects stored in subdirectories called `LOCATIONs`. Each `LOCATION` is defined by its coordinate system, map projection and geographical boundaries. Each `LOCATION` can have several `MAPSETs` (subdirectories of the `LOCATION`) that are used to subdivide the project into different topics or sub-regions, or as workspaces for individual team members (see Neteler & Mitasova 2008 in 文献和网络参考). In order to analyze vector and raster layers with GRASS modules, you generally have to import them into a GRASS `LOCATION`. (This is not strictly true -- with the GRASS modules `r.external` and `v.external` you can create read-only links to external GDAL-supported datasets without importing them. This is not the usual way for beginners to work with GRASS, therefore this functionality will not be described here.)

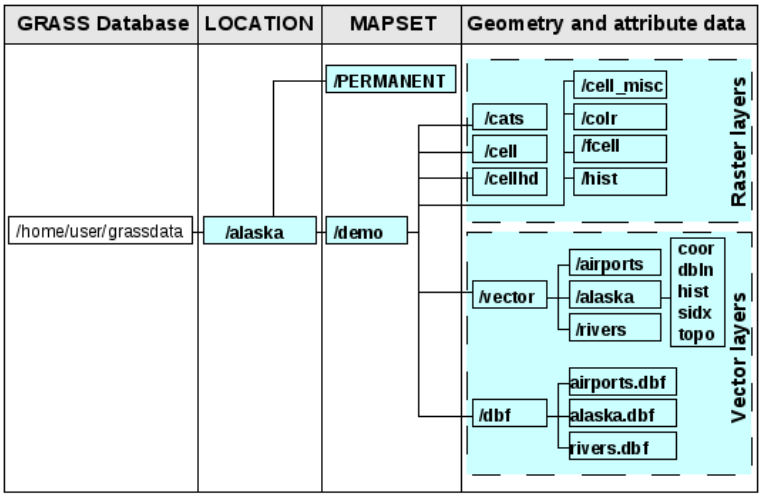


Fig. 22.1: GRASS data in the alaska LOCATION

22.9 Importing data into a GRASS LOCATION

See section *Importing data into a GRASS LOCATION via drag and drop* to find how data can be easily imported by dragging and dropping in the browser.

This section gives an example of how to import raster and vector data into the 'alaska' GRASS `LOCATION` provided by the QGIS 'Alaska' dataset in traditional way, using standard GRASS modules. Therefore, we use the landcover raster map `landcover.img` and the vector GML file `lakes.gml` from the QGIS 'Alaska' dataset (see 下载示例数据).

1. Start QGIS and make sure the GRASS plugin is loaded.
2. In the GRASS toolbar, click the Open MAPSET icon to bring up the *MAPSET* wizard.

3. Select as GRASS database the folder `grassdata` in the QGIS Alaska dataset, as `LOCATION` 'alaska', as `MAPSET` 'demo' and click *OK*.
4. Now click the Open GRASS tools icon. The GRASS Toolbox (see section *The GRASS Toolbox*) dialog appears.
5. To import the raster map `landcover.img`, click the module `r.in.gdal` in the *Modules Tree* tab. This GRASS module allows you to import GDAL-supported raster files into a GRASS `LOCATION`. The module dialog for `r.in.gdal` appears.
6. Browse to the folder `raster` in the QGIS 'Alaska' dataset and select the file `landcover.img`.
7. As raster output name, define `landcover_grass` and click *Run*. In the *Output* tab, you see the currently running GRASS command `r.in.gdal -o input=/path/to/landcover.img output=landcover_grass`.
8. When it says **Successfully finished**, click *View Output*. The `landcover_grass` raster layer is now imported into GRASS and will be visualized in the QGIS canvas.
9. To import the vector GML file `lakes.gml`, click the module `v.in.ogr` in the *Modules Tree* tab. This GRASS module allows you to import OGR-supported vector files into a GRASS `LOCATION`. The module dialog for `v.in.ogr` appears.
10. Browse to the folder `gml` in the QGIS 'Alaska' dataset and select the file `lakes.gml` as OGR file.
11. As vector output name, define `lakes_grass` and click *Run*. You don't have to care about the other options in this example. In the *Output* tab you see the currently running GRASS command `v.in.ogr -o dsn=/path/to/lakes.gml output=lakes\_grass`.
12. When it says **Succesfully finished**, click *View Output*. The `lakes_grass` vector layer is now imported into GRASS and will be visualized in the QGIS canvas.

22.9.1 Creating a new GRASS LOCATION

As an example, here is the sample GRASS `LOCATION alaska`, which is projected in the Albers Equal Area projection using feet as units. This sample GRASS `LOCATION alaska` will be used for all examples and exercises in the following GRASS-related sections. It is useful to download and install the dataset on your computer (see 下载示例数据).

1. Start QGIS and make sure the GRASS plugin is loaded.
2. Visualize the `alaska.shp` shapefile (see section 从文件加载图层) from the QGIS Alaska dataset (see 下载示例数据).
3. In the GRASS toolbar, click on the New mapset icon to bring up the *MAPSET* wizard.
4. Select an existing GRASS database (GISDBASE) folder `grassdata`, or create one for the new `LOCATION` using a file manager on your computer. Then click *Next*.
5. We can use this wizard to create a new `MAPSET` within an existing `LOCATION` (see section *Adding a new MAPSET*) or to create a new `LOCATION` altogether. Select *Create new location* (see Fig. 22.2).
6. Enter a name for the `LOCATION` -- we used 'alaska' -- and click *Next*.
7. Define the projection by clicking on the radio button *Projection* to enable the projection list.
8. We are using Albers Equal Area Alaska (feet) projection. Since we happen to know that it is represented by the EPSG ID 2964, we enter it in the search box. (Note: If you want to repeat this process for another `LOCATION` and projection and haven't memorized the EPSG ID, click on the CRS Status icon in the lower right-hand corner of the status bar (see section 使用投影)).
9. In *Filter*, insert 2964 to select the projection.
10. 单击 下一步。

11. To define the default region, we have to enter the LOCATION bounds in the north, south, east, and west directions. Here, we simply click on the button *Set Current QGIS Extent*, to apply the extent of the loaded layer `alaska.shp` as the GRASS default region extent.
12. 单击 下一步。
13. We also need to define a MAPSET within our new LOCATION (this is necessary when creating a new LOCATION). You can name it whatever you like - we used 'demo'. GRASS automatically creates a special MAPSET called PERMANENT, designed to store the core data for the project, its default spatial extent and coordinate system definitions (see Neteler & Mitasova 2008 in 文献和网络参考).
14. Check out the summary to make sure it's correct and click *Finish*.
15. The new LOCATION, 'alaska', and two MAPSETs, 'demo' and 'PERMANENT', are created. The currently opened working set is 'demo', as you defined.
16. Notice that some of the tools in the GRASS toolbar that were disabled are now enabled.

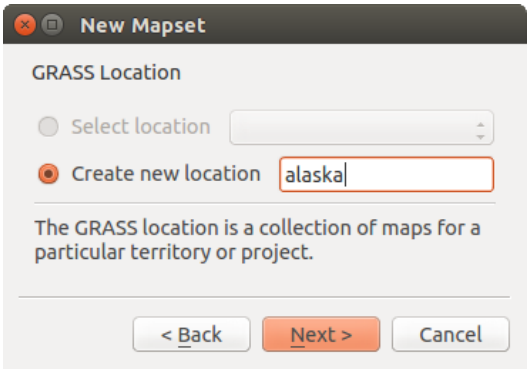


Fig. 22.2: Creating a new GRASS LOCATION or a new MAPSET in QGIS

If that seemed like a lot of steps, it's really not all that bad and a very quick way to create a LOCATION. The LOCATION 'alaska' is now ready for data import (see section *Importing data into a GRASS LOCATION*). You can also use the already-existing vector and raster data in the sample GRASS LOCATION 'alaska', included in the QGIS 'Alaska' dataset 下载示例数据, and move on to section *The GRASS vector data model*.

22.9.2 Adding a new MAPSET

A user has write access only to a GRASS MAPSET which he or she created. This means that besides access to your own MAPSET, you can read maps in other users' MAPSETs (and they can read yours), but you can modify or remove only the maps in your own MAPSET.

All MAPSETs include a WIND file that stores the current boundary coordinate values and the currently selected raster resolution (see Neteler & Mitasova 2008 in 文献和网络参考, and section *The GRASS region tool*).

1. Start QGIS and make sure the GRASS plugin is loaded.
2. In the GRASS toolbar, click on the New mapset icon to bring up the *MAPSET* wizard.
3. Select the GRASS database (GISDBASE) folder `grassdata` with the LOCATION 'alaska', where we want to add a further MAPSET called 'test'.
4. 单击 下一步。
5. We can use this wizard to create a new MAPSET within an existing LOCATION or to create a new LOCATION altogether. Click on the radio button *Select location* (see Fig. 22.2) and click *Next*.
6. Enter the name `test` for the new MAPSET. Below in the wizard, you see a list of existing MAPSETs and corresponding owners.
7. Click *Next*, check out the summary to make sure it's all correct and click *Finish*.

22.10 The GRASS vector data model

It is important to understand the GRASS vector data model prior to digitizing. In general, GRASS uses a topological vector model. This means that areas are not represented as closed polygons, but by one or more boundaries. A boundary between two adjacent areas is digitized only once, and it is shared by both areas. Boundaries must be connected and closed without gaps. An area is identified (and labelled) by the **centroid** of the area.

Besides boundaries and centroids, a vector map can also contain points and lines. All these geometry elements can be mixed in one vector and will be represented in different so-called 'layers' inside one GRASS vector map. So in GRASS, a layer is not a vector or raster map but a level inside a vector layer. This is important to distinguish carefully. (Although it is possible to mix geometry elements, it is unusual and, even in GRASS, only used in special cases such as vector network analysis. Normally, you should prefer to store different geometry elements in different layers.)

It is possible to store several 'layers' in one vector dataset. For example, fields, forests and lakes can be stored in one vector. An adjacent forest and lake can share the same boundary, but they have separate attribute tables. It is also possible to attach attributes to boundaries. An example might be the case where the boundary between a lake and a forest is a road, so it can have a different attribute table.

The 'layer' of the feature is defined by the 'layer' inside GRASS. 'Layer' is the number which defines if there is more than one layer inside the dataset (e.g., if the geometry is forest or lake). For now, it can be only a number. In the future, GRASS will also support names as fields in the user interface.

Attributes can be stored inside the GRASS `LOCATION` as dBase, SQLite3 or in external database tables, for example, PostgreSQL, MySQL, Oracle, etc.

Attributes in database tables are linked to geometry elements using a 'category' value.

'Category' (key, ID) is an integer attached to geometry primitives, and it is used as the link to one key column in the database table.

> **Tip**
>
> **Learning the GRASS Vector Model**
>
> The best way to learn the GRASS vector model and its capabilities is to download one of the many GRASS tutorials where the vector model is described more deeply. See https://grass.osgeo.org/learn/manuals/ for more information, books and tutorials in several languages.

22.11 Creating a new GRASS vector layer

To create a new GRASS vector layer, select one of following items from mapset context menu in the browser:

- New Point Layer
- New Line Layer
- New Polygon Layer

and enter a name in the dialog. A new vector map will be created and layer will be added to canvas and editing started. Selecting type of the layer does not restrict geometry types which can be digitized in the vector map. In GRASS, it is possible to organize all sorts of geometry types (point, line and polygon) in one vector map. The type is only used to add the layer to the canvas, because QGIS requires a layer to have a specific type.

It is also possible to add layers to existing vector maps selecting one of the items described above from context menu of existing vector map.

In GRASS, it is possible to organize all sorts of geometry types (point, line and area) in one layer, because GRASS uses a topological vector model, so you don't need to select the geometry type when creating a new GRASS vector. This is different from shapefile creation with QGIS, because shapefiles use the Simple Feature vector model (see section 新建矢量图层).

22.12 Digitizing and editing a GRASS vector layer

GRASS vector layers can be digitized using the standard QGIS digitizing tools. There are however some particularities, which you should know about, due to

- GRASS topological model versus QGIS simple feature
- complexity of GRASS model
  - multiple layers in single maps
  - multiple geometry types in single map
  - geometry sharing by multiple features from multiple layers

The particularities are discussed in the following sections.

Save, discard changes, undo, redo

> **Warning**
>
> All the changes done during editing are immediately written to vector map and related attribute tables.

Changes are written after each operation, it is however, possible to do undo/redo or discard all changes when closing editing. If undo or discard changes is used, original state is rewritten in vector map and attribute tables.

There are two main reasons for this behaviour:

- It is the nature of GRASS vectors coming from conviction that user wants to do what he is doing and it is better to have data saved when the work is suddenly interrupted (for example, blackout)
- Necessity for effective editing of topological data is visualized information about topological correctness, such information can only be acquired from GRASS vector map if changes are written to the map.

Toolbar

The 'Digitizing Toolbar' has some specific tools when a GRASS layer is edited:

Table 22.1: GRASS Digitizing Tools

Icon	工具	目标
	New Point	Digitize new point
	New Line	Digitize new line
	New Boundary	Digitize new boundary
	New Centroid	Digitize new centroid (label existing area)
	New Closed Boundary	Digitize new closed boundary

> **Tip**
>
> **Digitizing polygons in GRASS**
>
> If you want to create a polygon in GRASS, you first digitize the boundary of the polygon. Then you add a centroid (label point) into the closed boundary. The reason for this is that a topological vector model links the attribute information of a polygon always to the centroid and not to the boundary.

Category

Category, often called cat, is sort of ID. The name comes from times when GRASS vectors had only singly attribute ”category”. Category is used as a link between geometry and attributes. A single geometry may have multiple categories and thus represent multiple features in different layers. Currently it is possible to assign only one category per layer using QGIS editing tools. New features have automatically assigned new unique category, except boundaries. Boundaries usually only form areas and do not represent linear features, it is however possible to define attributes for a boundary later, for example in different layer.

New categories are always created only in currently being edited layer.

It is not possible to assign more categories to geometry using QGIS editing, such data are properly represented as multiple features, and individual features, even from different layers, may be deleted.

Attributes

Attributes of currently edited layer can only be modified. If the vector map contains more layers, features of other layers will have all attributes set to ’<not editable (layer #)>’ to warn you that such attribute is not editable. The reason is, that other layers may have and usually have different set of fields while QGIS only supports one fixed set of fields per layer.

If a geometry primitive does not have a category assigned, a new unique category is automatically assigned and new record in attribute table is created when an attribute of that geometry is changed.

Tip

If you want to do bulk update of attributes in table, for example using ’Field Calculator’ (*Using the Field Calculator*), and there are features without category which you don’t want to update (typically boundaries), you can filter them out by setting ’Advanced Filter’ to `cat is not null`.

Editing style

The topological symbology is essential for effective editing of topological data. When editing starts, a specialized ’GRASS Edit’ renderer is set on the layer automatically and original renderer is restored when editing is closed. The style may be customized in layer properties ’Style’ tab. The style can also be stored in project file or in separate file as any other style. If you customize the style, do not change its name, because it is used to reset the style when editing is started again.

Tip

Do not save project file when the layer is edited, the layer would be stored with ’Edit Style’ which has no meaning if layer is not edited.

The style is based on topological information which is temporarily added to attribute table as field ’topo_symbol’. The field is automatically removed when editing is closed.

Tip

Do not remove ’topo_symbol’ field from attribute table, that would make features invisible because the renderer is based on that column.

捕捉

To form an area, vertices of connected boundaries must have **exactly** the same coordinates. This can be achieved using snapping tool only if canvas and vector map have the same CRS. Otherwise, due conversion from map coordinates to canvas and back, the coordinate may become slightly different due to representation error and CRS transformations.

Tip

Use layer’s CRS also for canvas when editing.

Limitations

Simultaneous editing of multiple layers within the same vector at the same time is not supported. This is mainly due to the impossibility of handling multiple undo stacks for a single data source.

On Linux and macOS only one GRASS layer can be edited at time. This is due to a bug in GRASS which does not allow to close database drivers in random order. This is being solved with GRASS developers.

Tip

GRASS Edit Permissions

You must be the owner of the GRASS `MAPSET` you want to edit. It is impossible to edit data layers in a `MAPSET` that is not yours, even if you have write permission.

22.13 The GRASS region tool

The region definition (setting a spatial working window) in GRASS is important for working with raster layers. Vector analysis is by default not limited to any defined region definitions. But all newly created rasters will have the spatial extension and resolution of the currently defined GRASS region, regardless of their original extension and resolution. The current GRASS region is stored in the `$LOCATION/$MAPSET/WIND` file, and it defines north, south, east and west bounds, number of columns and rows, horizontal and vertical spatial resolution.

It is possible to switch on and off the visualization of the GRASS region in the QGIS canvas using the Display current GRASS region button.

The region can be modified in 'Region' tab in 'GRASS Tolls' dock widget. Type in the new region bounds and resolution, and click *Apply*. If you click on *Select the extent by dragging on canvas* you can select a new region interactively with your mouse on the QGIS canvas dragging a rectangle.

The GRASS module `g.region` provides a lot more parameters to define an appropriate region extent and resolution for your raster analysis. You can use these parameters with the GRASS Toolbox, described in section *The GRASS Toolbox*.

22.14 The GRASS Toolbox

The Open GRASS Tools box provides GRASS module functionalities to work with data inside a selected GRASS `LOCATION` and `MAPSET`. To use the GRASS Toolbox you need to open a `LOCATION` and `MAPSET` that you have write permission for (usually granted, if you created the `MAPSET`). This is necessary, because new raster or vector layers created during analysis need to be written to the currently selected `LOCATION` and `MAPSET`.

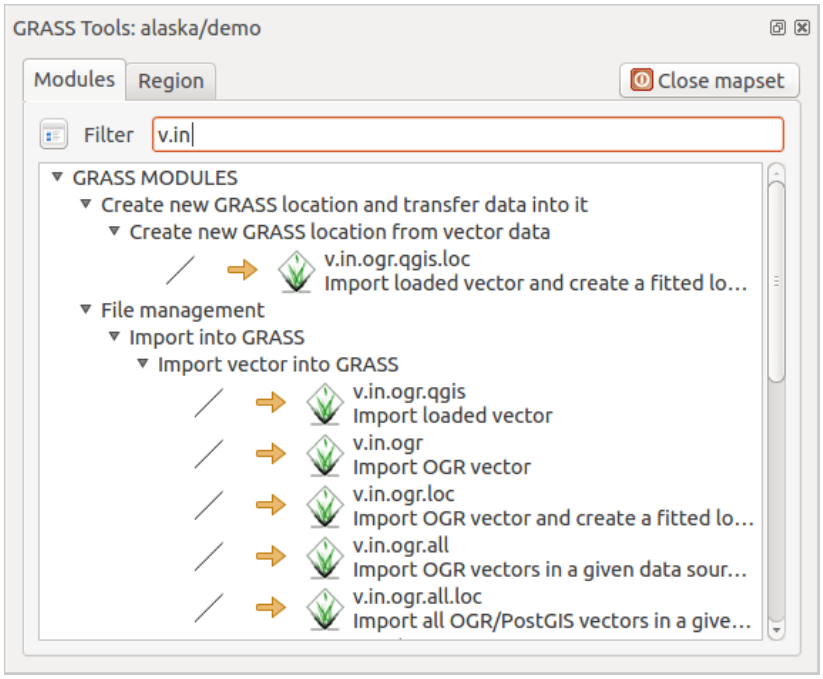


Fig. 22.3: GRASS Toolbox and Module Tree

22.14.1 Working with GRASS modules

The GRASS plugin provides a user-friendly toolbox with about 200 of the available GRASS modules and functionalities.

A complete list of GRASS modules available in the graphical Toolbox in QGIS version 3.44 is available in the GRASS wiki at https://grasswiki.osgeo.org/wiki/GRASS-QGIS_relevant_module_list.

It is also possible to customize the GRASS Toolbox content. This procedure is described in section *Customizing the GRASS Toolbox*.

As shown in Fig. 22.3, you can look for the appropriate GRASS module using the thematically grouped *Modules Tree* or the searchable *Modules List* tab.

By clicking on a graphical module icon, a new tab will be added to the Toolbox dialog, providing three new sub-tabs: *Options*, *Output* and *Manual*.

Options

The *Options* tab provides a simplified module dialog where you can usually select a raster or vector layer visualized in the QGIS canvas and enter further module-specific parameters to run the module.

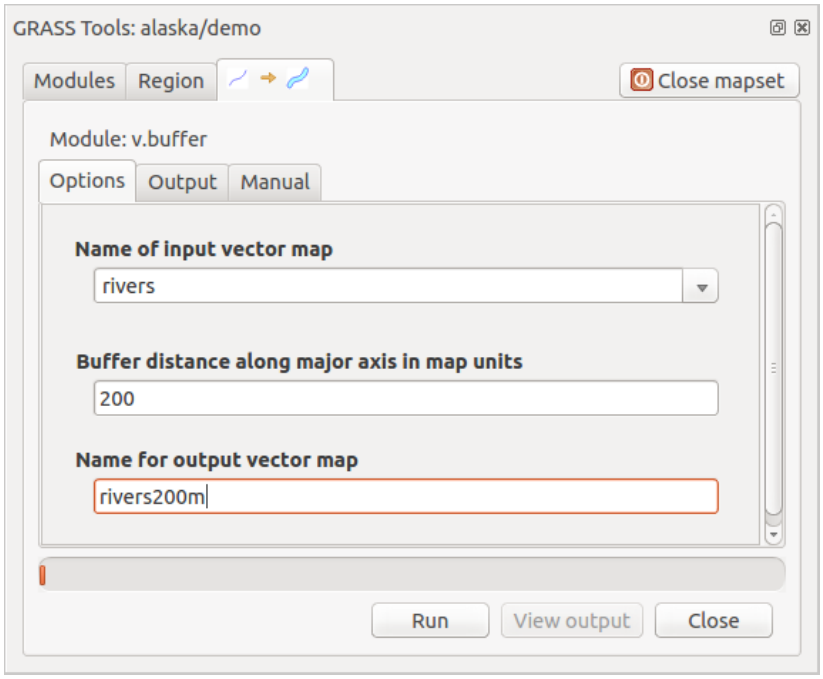


Fig. 22.4: GRASS Toolbox Module Options

The provided module parameters are often not complete to keep the dialog simple. If you want to use further module parameters and flags, you need to start the GRASS shell and run the module in the command line.

A new feature since QGIS 1.8 is the support for a *Show Advanced Options* button below the simplified module dialog in the *Options* tab. At the moment, it is only added to the module `v.in.ascii` as an example of use, but it will probably be part of more or all modules in the GRASS Toolbox in future versions of QGIS. This allows you to use the complete GRASS module options without the need to switch to the GRASS shell.

Output

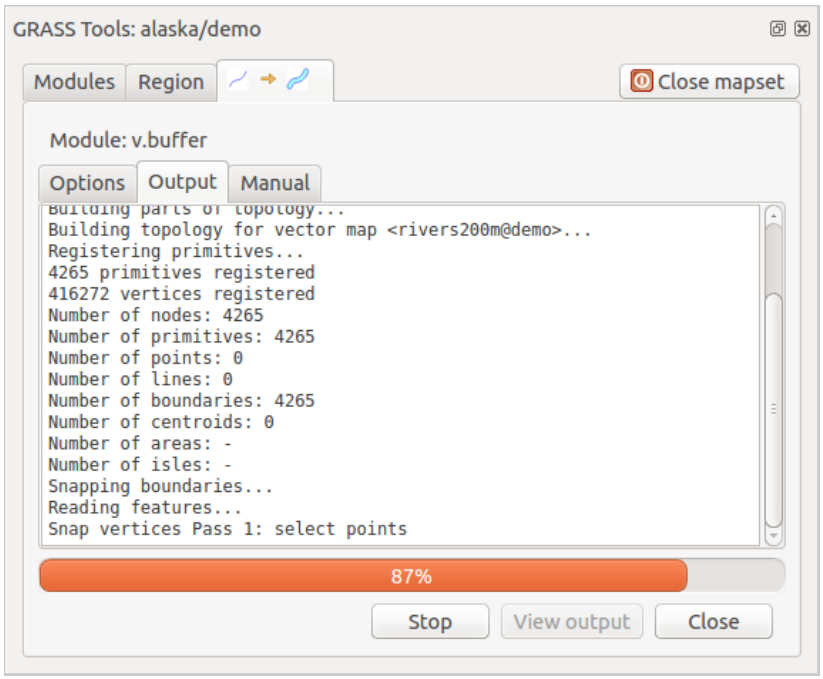


Fig. 22.5: GRASS Toolbox Module Output

The *Output* tab provides information about the output status of the module. When you click the *Run* button, the module switches to the *Output* tab and you see information about the analysis process. If all works well, you will finally see a `Successfully finished` message.

Manual

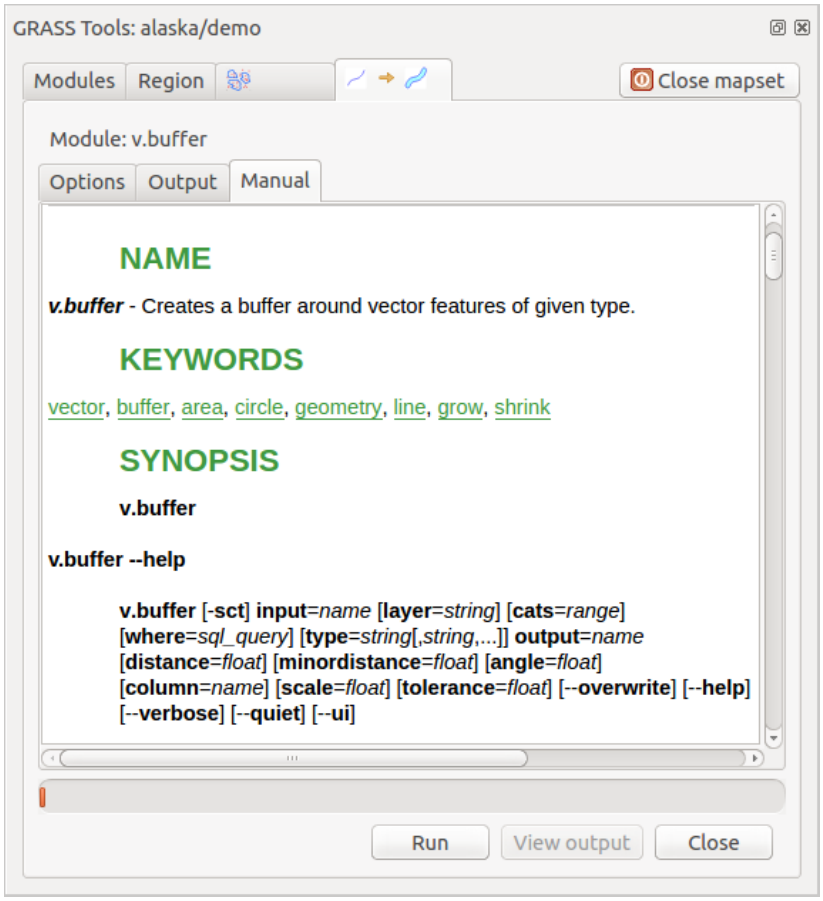


Fig. 22.6: GRASS Toolbox Module Manual

The *Manual* tab shows the HTML help page of the GRASS module. You can use it to check further module parameters and flags or to get a deeper knowledge about the purpose of the module. At the end of each module manual page, you see further links to the `Main Help index`, the `Thematic index` and the `Full index`. These links provide the same information as the module `g.manual`.

Tip

Display results immediately

If you want to display your calculation results immediately in your map canvas, you can use the 'View Output' button at the bottom of the module tab.

22.14.2 GRASS module examples

The following examples will demonstrate the power of some of the GRASS modules.

Creating contour lines

The first example creates a vector contour map from an elevation raster (DEM). Here, it is assumed that you have the Alaska `LOCATION` set up as explained in section *Importing data into a GRASS LOCATION*.

- First, open the location by clicking the Open mapset button and choosing the Alaska location.
- Now open the Toolbox with the Open GRASS tools button.
- In the list of tool categories, double-click *Raster* ► *Surface Management* ► *Generate vector contour lines*.
- Now a single click on the tool **r.contour** will open the tool dialog as explained above (see *Working with GRASS modules*).
- In the *Name of input raster map* enter `gtopo30`.
- Type into the *Increment between Contour levels* the value 100. (This will create contour lines at intervals of 100 meters.)
- Type into the *Name for output vector map* the name `ctour_100`.
- Click *Run* to start the process. Wait for several moments until the message `Successfully finished` appears in the output window. Then click *View Output* and *Close*.

Since this is a large region, it will take a while to display. After it finishes rendering, you can open the layer properties window to change the line color so that the contours appear clearly over the elevation raster, as in 矢量属性对话框.

Next, zoom in to a small, mountainous area in the center of Alaska. Zooming in close, you will notice that the contours have sharp corners. GRASS offers the **v.generalize** tool to slightly alter vector maps while keeping their overall shape. The tool uses several different algorithms with different purposes. Some of the algorithms (i.e., Douglas Peuker and Vertex Reduction) simplify the line by removing some of the vertices. The resulting vector will load faster. This process is useful when you have a highly detailed vector, but you are creating a very small-scale map, so the detail is unnecessary.

> **Tip**
>
> **The simplify tool**
>
> Note that QGIS has a *Vector* ► *Geometry Tools* ► *Simplify geometries* tool that works just like the GRASS **v.generalize** Douglas-Peuker algorithm.

However, the purpose of this example is different. The contour lines created by `r.contour` have sharp angles that should be smoothed. Among the **v.generalize** algorithms, there is Chaiken's, which does just that (also Hermite splines). Be aware that these algorithms can **add** additional vertices to the vector, causing it to load even more slowly.

- Open the GRASS Toolbox and double-click the categories *Vector* ► *Develop map* ► *Generalization*, then click on the **v.generalize** module to open its options window.
- Check that the 'ctour_100' vector appears as the *Name of input vector*.
- From the list of algorithms, choose Chaiken's. Leave all other options at their default, and scroll down to the last row to enter in the field *Name for output vector map* 'ctour_100_smooth', and click *Run*.
- The process takes several moments. Once `Successfully finished` appears in the output windows, click *View Output* and then *Close*.
- You may change the color of the vector to display it clearly on the raster background and to contrast with the original contour lines. You will notice that the new contour lines have smoother corners than the original while staying faithful to the original overall shape.

Fig. 22.7: GRASS module v.generalize to smooth a vector map

Tip

Other uses for r.contour

The procedure described above can be used in other equivalent situations. If you have a raster map of precipitation data, for example, then the same method will be used to create a vector map of isohyetal (constant rainfall) lines.

Creating a Hillshade 3-D effect

Several methods are used to display elevation layers and give a 3-D effect to maps. The use of contour lines, as shown above, is one popular method often chosen to produce topographic maps. Another way to display a 3-D effect is by hillshading. The hillshade effect is created from a DEM (elevation) raster by first calculating the slope and aspect of each cell, then simulating the sun's position in the sky and giving a reflectance value to each cell. Thus, you get sun-facing slopes lighted; the slopes facing away from the sun (in shadow) are darkened.

- Begin this example by loading the `gtopo30` elevation raster. Start the GRASS Toolbox, and under the Raster category, double-click to open *Spatial analysis* ► *Terrain analysis*.
- Then click **r.shaded.relief** to open the module.
- Change the *azimuth angle* 1,00 270 to 315.
- Enter `gtopo30_shade` for the new hillshade raster, and click *Run*.
- When the process completes, add the hillshade raster to the map. You should see it displayed in grayscale.
- To view both the hillshading and the colors of the `gtopo30` together, move the hillshade map below the `gtopo30` map in the table of contents, then open the *Properties* window of `gtopo30`, switch to the *Transparency* tab and set its transparency level to about 25%.

You should now have the `gtopo30` elevation with its colormap and transparency setting displayed **above** the grayscale hillshade map. In order to see the visual effects of the hillshading, turn off the `gtopo30_shade` map, then turn it back on.

Raster statistics in a vector map

The next example shows how a GRASS module can aggregate raster data and add columns of statistics for each polygon in a vector map.

- Again using the Alaska data, refer to *Importing data into a GRASS LOCATION* to import the `shapefiles/trees.shp` file into GRASS.
- Now an intermediate step is required: centroids must be added to the imported trees map to make it a complete GRASS area vector (including both boundaries and centroids).
- From the Toolbox, choose *Vector* ► *Manage features*, and open the module **v.centroids**.
- Enter as the *output vector map* 'forest_areas' and run the module.
- Now load the `forest_areas` vector and display the types of forests - deciduous, evergreen, mixed - in different colors: In the layer *Properties* window, *Symbology* tab, choose from *Legend type* 'Unique value' and set the *Classification field* to 'VEGDESC'. (Refer to the explanation of the symbology tab in *Symbology Properties* of the vector section.)
- Next, reopen the GRASS Toolbox and open *Vector* ► *Vector update* by other maps.
- Click on the **v.rast.stats** module. Enter `gtopo30` and `forest_areas`.
- Only one additional parameter is needed: Enter *column prefix* `elev`, and click *Run*. This is a computationally heavy operation, which will run for a long time (probably up to two hours).
- Finally, open the `forest_areas` attribute table, and verify that several new columns have been added, including `elev_min`, `elev_max`, `elev_mean`, etc., for each forest polygon.

22.14.3 Customizing the GRASS Toolbox

Nearly all GRASS modules can be added to the GRASS Toolbox. An XML interface is provided to parse the pretty simple XML files that configure the modules' appearance and parameters inside the Toolbox.

A sample XML file for generating the module `v.buffer` (v.buffer.qgm) looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE qgisgrassmodule SYSTEM "http://mrcc.com/qgisgrassmodule.dtd">

<qgisgrassmodule label="Vector buffer" module="v.buffer">
        <option key="input" typeoption="type" layeroption="layer" />
        <option key="buffer"/>
        <option key="output" />
</qgisgrassmodule>
```

The parser reads this definition and creates a new tab inside the Toolbox when you select the module.

CHAPTER 23

QGIS 处理架构

23.1 简介

This chapter introduces the QGIS processing framework, a geoprocessing environment that can be used to call native and third-party algorithms from QGIS, making your spatial analysis tasks more productive and easy to accomplish.

As a *Core plugin*, Processing is installed by default but you need to activate it:

1. Go to *Plugins* ► *Manage and install plugins...*
2. Click on the *Installed* tab at the left
3. Check the box next to the *Processing* entry
4. Close the dialog.

   A *Processing* menu is now available in the top menu bar. From there you can reach the main components of this framework.

In the following sections, we will review how to use the graphical elements of this framework and make the most out of each one of them.

There are four basic elements in the framework GUI, which are used to run algorithms for different purposes. Choosing one tool or another will depend on the kind of analysis that is to be performed and the particular characteristics of each user and project. All of them (except for the batch processing interface, which is called from the toolbox or the algorithm execution dialog, as we will see) can be accessed from the *Processing* menu item (you will see more entries; the remaining ones are not used to execute algorithms and will be explained later in this chapter).

- The *Toolbox*: The main element of the GUI, it is used to execute a single algorithm or run a batch process based on that algorithm.

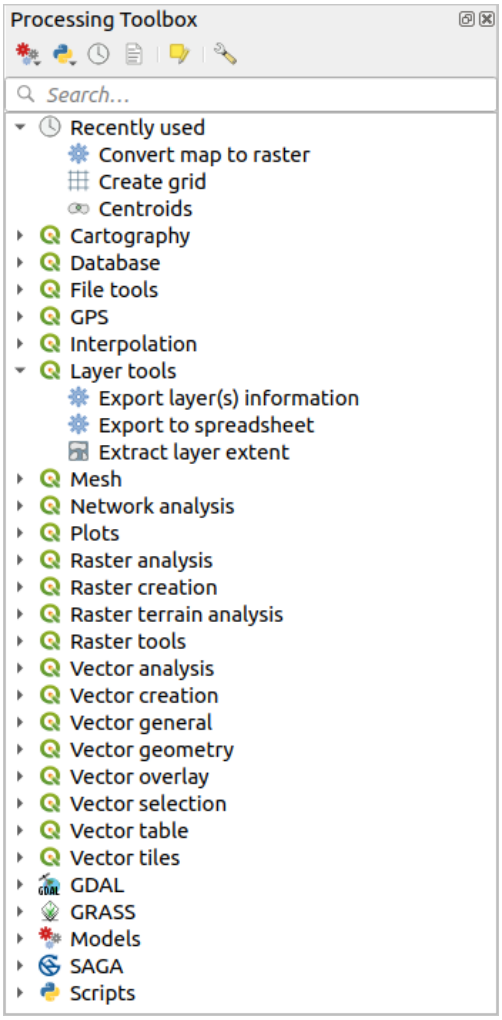


Fig. 23.1: 处理工具箱

- The *Model Designer*: Several algorithms can be combined graphically using the modeler to define a workflow, creating a single process that involves several subprocesses.

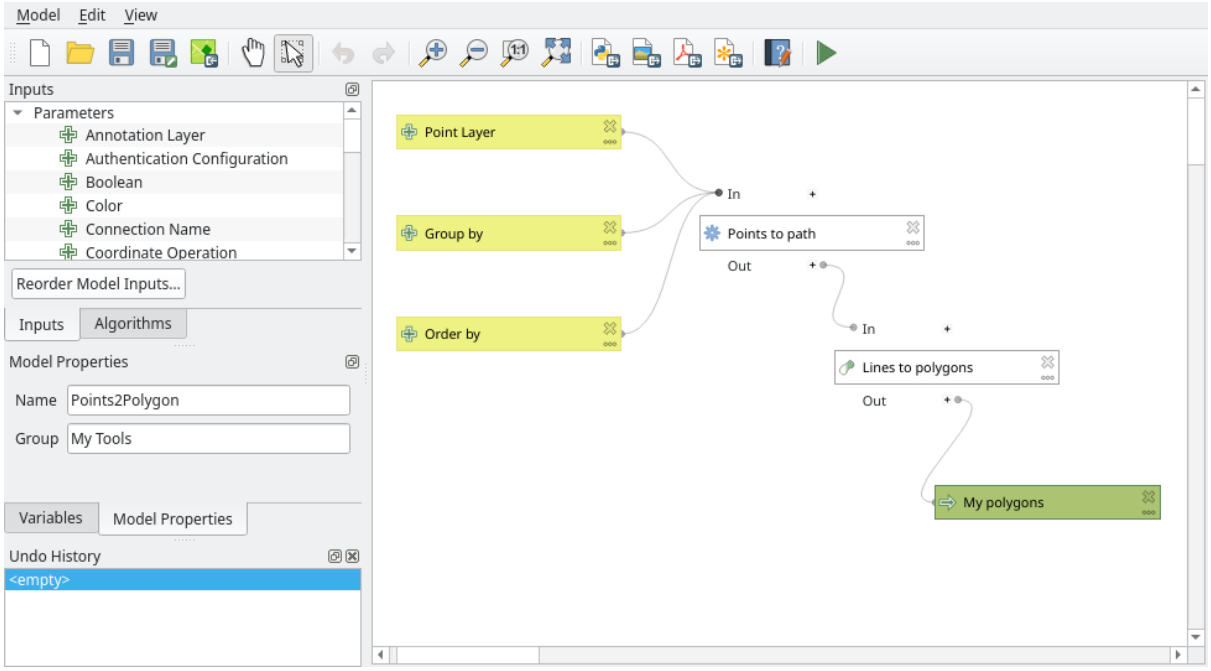


Fig. 23.2: 处理模型

- The *History* manager: All actions performed using any of the aforementioned elements are stored in a history file and can be later easily reproduced using the history manager.

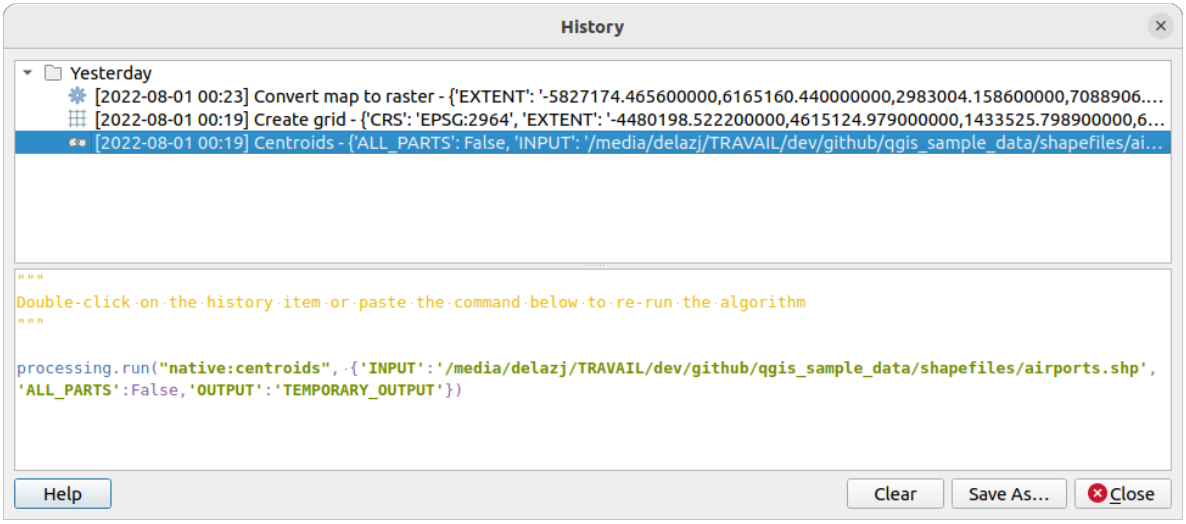


Fig. 23.3: Processing History

- The *Batch Processing* interface: This interface allows you to execute batch processes and automate the execution of a single algorithm on multiple datasets.

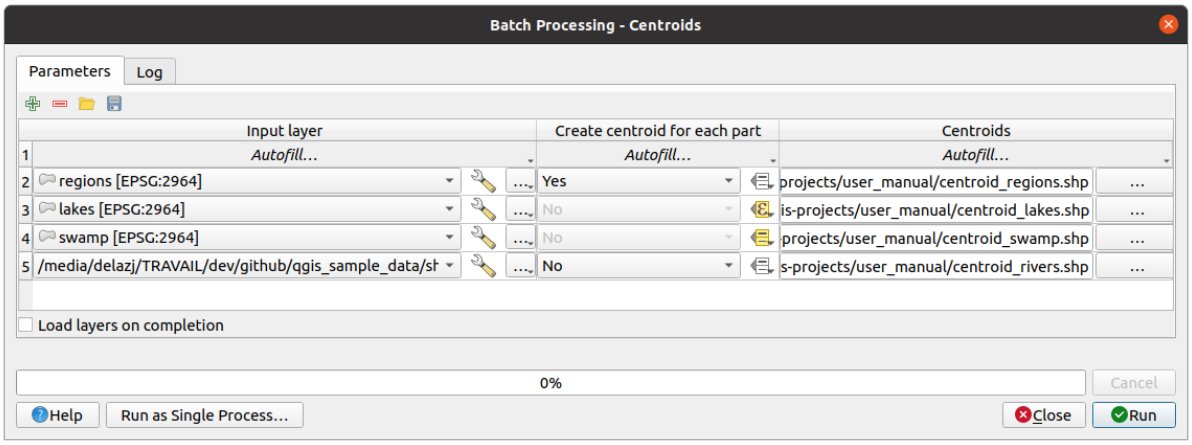


Fig. 23.4: Batch Processing interface

In the following sections, we will review each one of these elements in detail.

23.2 Configuring the Processing Framework

The Processing Options menu (*Settings*► *Options* ► *Processing* tab) allows you to configure how algorithms work. Configuration parameters are structured in separate blocks that you can select on the left-hand side of the dialog.

23.2.1 General

The *General* block contains the default settings to control how the algorithm dialog as well as input or output parameters should behave. Some of the settings can however be overridden at the algorithm level, per *algorithm run* or for *individual parameters*.

- *Default output raster layer extension* is by default `tif`
- *Default output vector layer extension* is by default `gpkg`
- *Invalid features filtering* when executing algorithm:
  - *Do not filter (better performance)*: all the features (with valid and invalid geometries) are processed, but the result may be erroneous depending on how the geometry invalidity affects the operations
  - *Skip (ignore) features with invalid geometries*, meaning that only a subset of your dataset (the valid geometry features) will be processed
  - *Stop algorithm execution when a geometry is invalid*: you'll need to track and fix the invalid geometries if you want the algorithm to process the whole layer. Algorithms like *Check validity* or *Fix geometries* can help you achieve this.

  The *Invalid features filtering* setting can be overridden on a per-input basis, at algorithm runtime.
- *Keep dialog open after running algorithm*. Once an algorithm has finished execution and its output layers are loaded into the QGIS project, the algorithm dialog is closed. If you want to keep it open (to run the algorithm again with different parameters, or to better check the output that is written to the log tab), check this option.
- *Max Threads*
- *Output folder* for non temporary outputs: If no folder path is provided for the Processing execution outputs, this is the folder in which they will be saved. Default is `processing` under the user home directory (e.g., `C:\Users\<YOURUSERNAME>\processing\` on Windows).

- *Override temporary output folder path*: Temporary outputs are saved by default in the `tmp` folder on the machine. This option helps you set a different place for storage.
- *Pre-execution script* and *Post-execution script*. These parameters point to files that contain scripts written using the processing scripting functionality, explained in the section covering scripting and the console.
- *Prefer output filename for layer names*. The name of each resulting layer created by an algorithm is defined by the algorithm itself. In some cases, a fixed name might be used, meaning that the same output name will be used, no matter which input layer is used. In other cases, the name might depend on the name of the input layer or some of the parameters used to run the algorithm. If this checkbox is checked, the name will be taken from the output filename instead. Notice that, if the output is saved to a temporary file, the filename of this temporary file is usually a long and meaningless one intended to avoid collision with other already existing filenames.
- *Results group name*. If you want to obtain all processing result layers in a group in the *Layers* panel, set a group name for this parameter. The group may exist already or not. QGIS will add all output layers to such a group. By default, this parameter is empty, so all output layers are added to different places in the *Layers* panel, depending on the item that is active when running an algorithm. Note that output layers will be loaded to the *Layers* panel only if *Open output file after running algorithm* is checked in the algorithm dialog.
- *Show algorithms with known issues*: By default, QGIS avoids display of broken algorithms (generally from third-party providers). If checked, they will be available in the Processing toolbox, with a warning icon and a tooltip explaining they have issues. Use at your own risks.
- *Show feature count for output vector layers*. Since calculating the feature count might take some time in certain data formats, this option is off by default.
- *Show layer CRS definition in selection boxes*
- *Show tooltip when there are disabled providers*
- *Style for line layers*, *Style for point layers*, *Style for polygons layers* and *Style for raster layers* are used for setting the default rendering style for output layers (that is, layers generated by processing algorithms). Just create the style you want using QGIS, save it to a file, and then enter the path to that file in the settings so the algorithms can use it. Whenever a layer is loaded by Processing and added to the QGIS canvas, it will be rendered with that style.

  Rendering styles can be configured individually for each algorithm and each one of its outputs. Just right-click on the name of the algorithm in the toolbox and select *Edit rendering styles for outputs*. You will see a dialog like the one shown next.

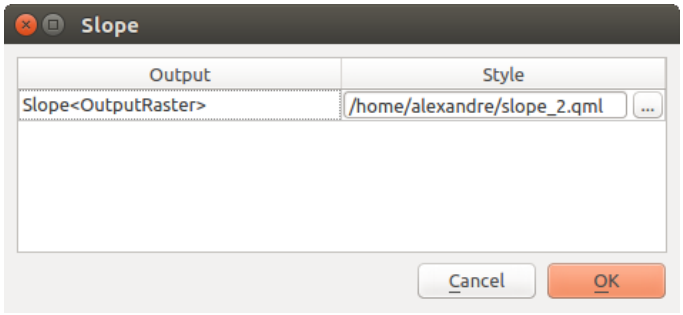


Fig. 23.5: Rendering Styles

  Select the style file (`.qml`) that you want for each output and press *OK*.
- *Warn before executing if parameter CRS's do not match*: By default, QGIS native algorithms (i.e. the ones listed under the *Menus* ► *QGIS (native C++)* group) transparently reproject the input layers to the first one's CRS before execution. Check this option to get a notification from the other tools that do not suppport reprojection, when the inputs CRS are not identical. Third-party providers are not concerned.

23.2.2 Menus

The *Menus* block controls whether an algorithm, script or model (built-in or provided by plugins) should be made available through a dedicated menu or toolbar (along with the Processing Toolbox). For each item of each provider, you can:

- *Add button in toolbar*, making it available in the *Processing Algorithms* toolbar
- assign an *Icon* to the algorithm
- set a *Menu path*: the algorithm will then be available through an existing or a custom menu, e.g. `Vect&or/MyTopAlgorithms` will add a shortcut in a *MyTopAlgorithms* sub-menu of the default *Vector* menu.

Restart QGIS to apply the settings. At any time, your changes can be *Reset to defaults*.

23.2.3 Models and Scripts

In the *Models* and *Scripts* blocks, a default path, set under the active *User profile* folder, is provided for storing models and scripts respectively. As for the other options, you can modify the path, e.g., to lead to a remote or shared folder for all your users.

23.2.4 Providers

You will also find a block for algorithm *Providers*. This is the place where installed providers expose their settings.

By default, QGIS is installed with two algorithm providers:

- *GDAL* whose algorithms you may *Activate* (or not) in this dialog. Read more about the *GDAL algorithms*.
- *GRASS* whose algorithms require the installation of the *GRASS GIS Processing Provider* from the *Plugins Manager*. You could then set some dedicated settings such as:
  - *For raster layers, use r.external (faster) instead of r.in.gdal*
  - *For vector layers, use v.external (faster) instead of v.in.ogr*
  - *Location of GRASS docs*: by default, and depending on your OS, QGIS utilizes a hardcoded list of paths for finding the local GRASS documentation and opening it whenever you hit the *Help* button of a GRASS algorithm dialog. Filling this option will override the default path, allowing you to use a custom location, or remote documentation.
  - *Log console output*
  - *Log execution commands*

Installed plugins that provide Processing algorithms may also have their provider listed in this group, with custom settings.

23.3 The Toolbox

The *Processing Toolbox* is the main element of the processing GUI, and the one that you are more likely to use in your daily work. It shows the list of all available **algorithms** grouped in different blocks called *Providers*, and custom **models** and **scripts** you can add to extend the set of tools. Hence the toolbox is the access point to run them, whether as a single process or as a batch process involving several executions of the same algorithm on different sets of inputs.

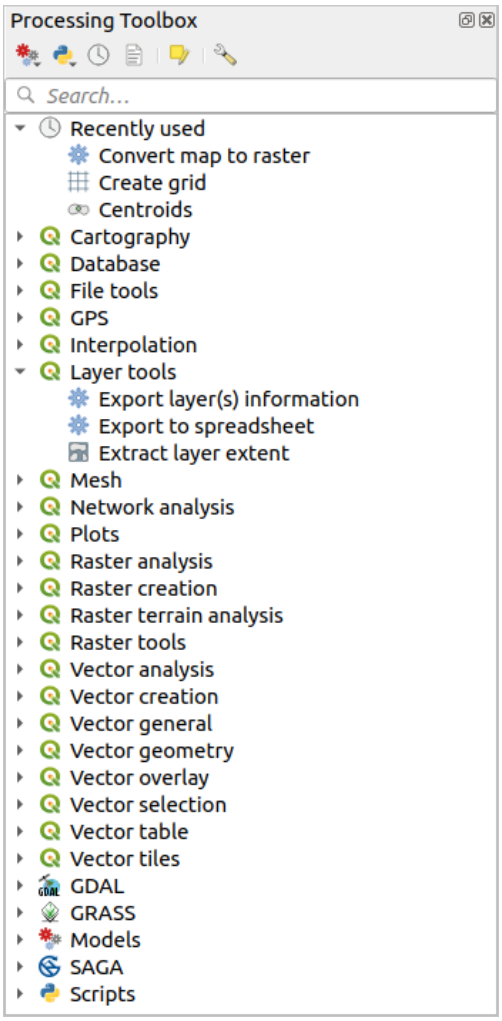


Fig. 23.6: 处理工具箱

Providers can be (de)activated in the *Processing settings dialog*. By default, only providers that do not rely on third-party applications (that is, those that only require QGIS elements to be run) are active. Algorithms requiring external applications might need additional configuration. Configuring providers is explained in a *later chapter* in this manual.

In the upper part of the toolbox dialog, you will find a set of tools to:

- work with Models: *Create New Model...*, *Open Existing Model...* and *Add Model to Toolbox...*;
- work with Scripts: *Create New Script...*, *Create New Script from Template...*, *Open Existing Script...* and *Add Script to Toolbox...*;
- open the History panel;
- open the Results Viewer panel;

- toggle the toolbox to the *in-place modification mode* using the Edit Features In-Place button: only the algorithms that are suitable to be executed on the active layer without outputting a new layer are displayed;
- open the Options dialog.

Below this toolbar is a *Search...* box to help you easily find the tools you need. You can enter any word or phrase on the text box. Notice that, as you type, the number of algorithms, models or scripts in the toolbox is reduced to just those that contain the text you have entered in their names or keywords.

The top of the algorithm list shows the most Recently Used tools and the Favorites tools, making it convenient to reexecute them. The Recently Used list is there by default, showing the algorithms you have recently executed. The Favorites list is created when you manually add an algorithm to it.

You can right-click on any algorithm in the toolbox and choose from the contextual menu to:

- *Execute...* to run the algorithm.
- *Execute as Batch Process...* to run the algorithm in batch mode.
- *Edit Rendering Styles for Outputs...* to customize the rendering styles for the algorithm's outputs.
- *Add to Favorites* to add the algorithm to the Favorites list.
- *Remove from Favorites* to remove the algorithm from the Favorites list.

Unlike the Recently Used list, the Favorites list remains static and is not affected by which algorithms were executed.

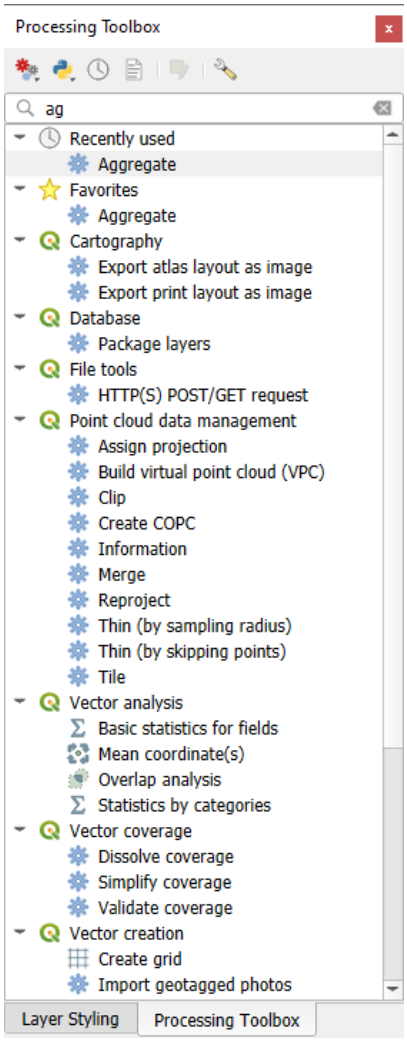


Fig. 23.7: Processing Toolbox showing search results

To execute a tool, just double-click on its name in the toolbox.

23.3.1 The algorithm dialog

Once you double-click on the name of the algorithm that you want to execute, a dialog similar to that in the Fig. 23.8 below is shown (in this case, the dialog corresponds to the `Centroids` algorithm). Dialog title will include the group name from which the algorithm originates.

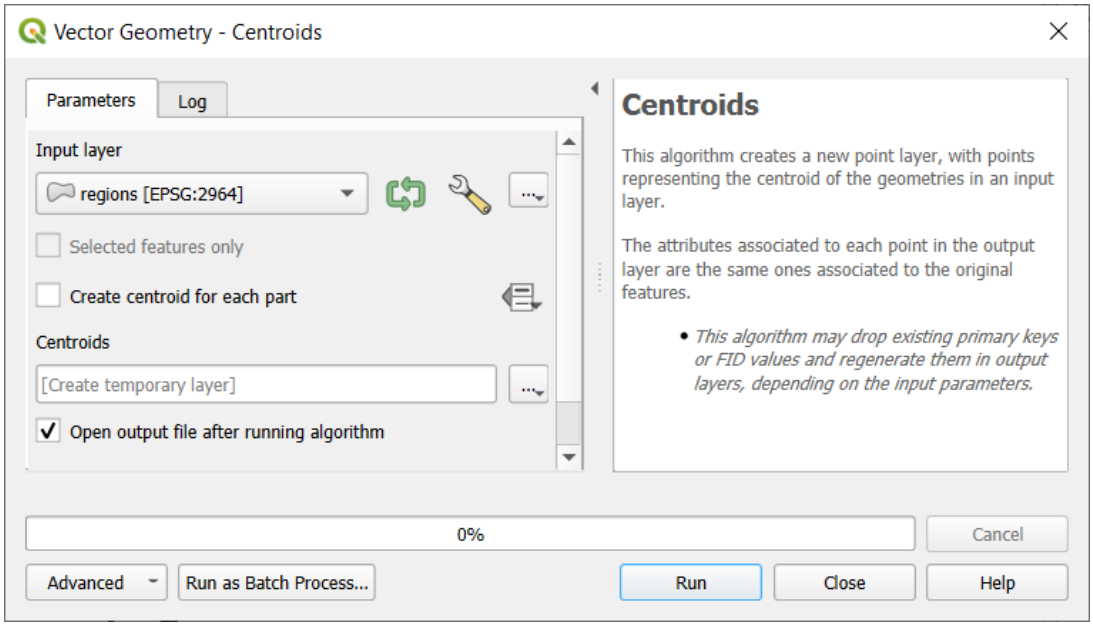


Fig. 23.8: Algorithm Dialog - Parameters

The dialog shows two tabs (*Parameters* and *Log*) on the left part, the algorithm description on the right, and a set of buttons at the bottom.

Parameter types

The *Parameters* tab is used to set the input values that the algorithm needs to be executed. It shows a list of input values and configuration parameters to be set. It of course has a different content, depending on the requirements of the algorithm to be executed, and is created automatically based on those requirements.

> **Tip**
>
> **Setting your own default values for algorithm parameters**
>
> Algorithm dialogs open with some parameters prefilled with values from QGIS installation. It is however possible to set *your own default values* for specific algorithm parameters so that they are used at algorithm startup.

Although the number and type of parameters depend on the characteristics of the algorithm, the structure is similar for all of them. The parameters found in the table can be of one of the following types.

- A **vector layer**, to select from a list of all vector layers available (currently opened) in QGIS. You can also use unloaded layers: press the ... button on the widget right-hand side, and select:
  - *Select file...*: selects file on disk using the Operating System file explorer
  - *Browse for layer...*: opens the *Browser panel*, allowing to take the layers directly from database sources (PostgreSQL, SQL Server, Oracle, ...), web services (WFS, AFS, ...) or files on disk.

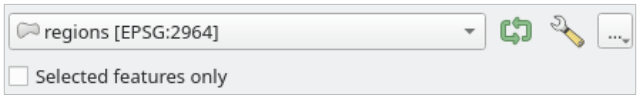


Fig. 23.9: Vector input widget

> **Tip**
>
> **Drag-and drop as convenient alternative for layer selection**
>
> Besides selecting from the drop-down menu or browsing sources, you can also drag and drop a layer from the *Layers* or *Browser* panel, or from the Operating System file explorer directly into the combobox.

> **Note**
>
> By default, the layer widget shows the CRS of the layer along with its name. If you do not want to see this additional information, you can disable this functionality in the Processing Settings dialog, unchecking the *General* ► *Show layer CRS definition in selection boxes* option.

The vector input widget also has following features:

- an iterator button: If toggled, the algorithm will be executed iteratively on each one of its features, instead of just once for the whole layer, producing as many outputs as times the algorithm is executed. This allows for automating the process when all features in a layer have to be processed separately. If the algorithm contains several input vectors you can iterate over, the iteration will be processed only on the first toggled parameter, in the order parameters are declared in the algorithm.
- Advanced options button to adjust settings to use for that specific parameter. These settings concern:
  - *Invalid feature filtering*: allows the *default method* for handling features with invalid geometries to be overridden
  - *Limit features processed*: optional limit on number of features processed from the source
  - *Feature filter*: allows to enter an expression to subset the layer dynamically when running the tool, avoiding the need for separate steps to set layer filters or create layer subsets.

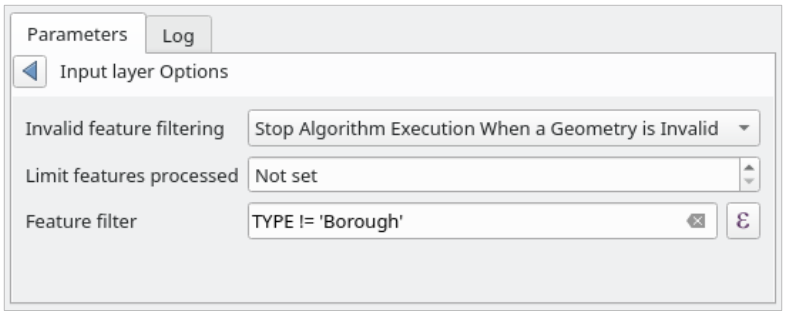


Fig. 23.10: Advanced options for vector input widget

  - It is also possible to limit the algorithm execution on the vector layer to its *Selected features only*.
- A **table**, to select from a list of all available in QGIS. Non-spatial tables are loaded into QGIS like vector layers, and use the *same widget*.
- A **raster layer**, to select from a list of all raster layers available in QGIS. The selector contains as well a ... button on its right-hand side, to let you select filenames that represent layers currently not loaded in QGIS.

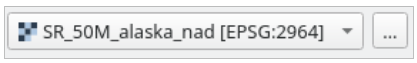


Fig. 23.11: Raster input widget

- An **option**, to choose from a selection list of possible options.

- A **numerical value**, to be introduced in a spin box. In some contexts (when the parameter applies at the feature level and not at the layer's), you will find a Data-defined override button by its side, allowing you to open the *expression builder* and enter a mathematical expression to generate variable values for the parameter. Some useful variables related to data loaded into QGIS can be added to your expression, so you can select a value derived from any of these variables, such as the cell size of a layer or the northernmost coordinate of another one.

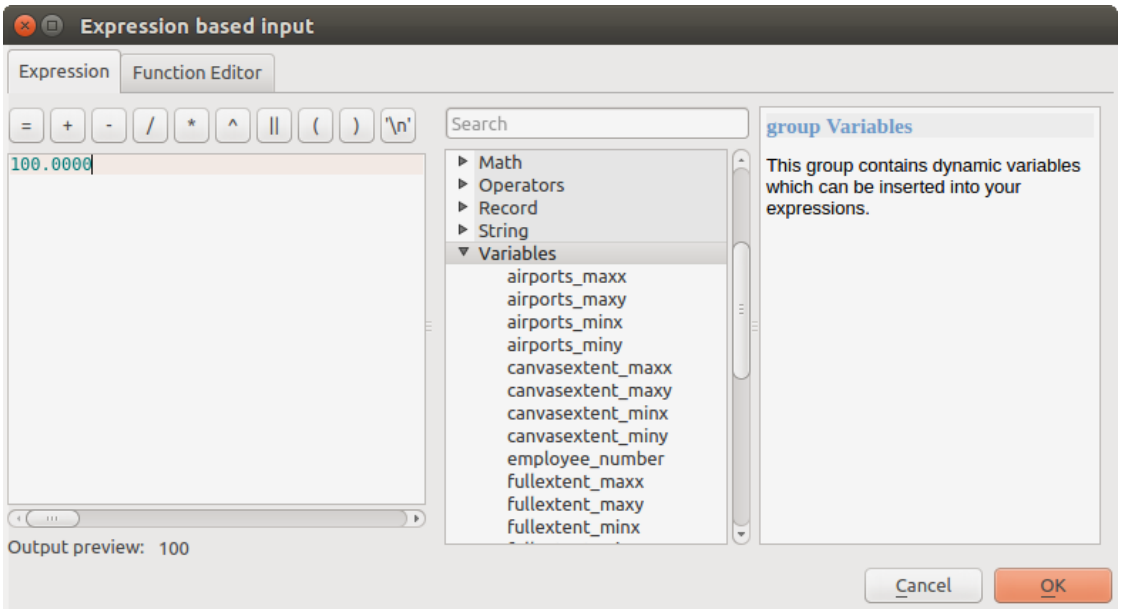


Fig. 23.12: 基于表达式的输入

- A **range**, with min and max values to be introduced in two text boxes.
- A **text string**, to be introduced in a text box.
- A **field**, to choose from the attributes table of a vector layer or a single table selected in another parameter.
- A **coordinate reference system**. You can select it among the recently used ones from the drop-down list or from the *CRS selection* dialog that appears when you click on the button on the right-hand side.

> **Tip**
>
> **Drag-and-drop as convenient shortcut for CRS selection**
>
> You can also drag a layer from the Layers panel and drop it onto the CRS widget. The widget will automatically set its CRS to match the dropped layer's.

- An **extent**, a text box defining a rectangle through its corners coordinate in the format `xmin, xmax, ymin, ymax`. Press the Set to current map canvas extent button to use the map canvas extent. Clicking the arrow on the right-hand side of the value selector, a pop-up menu will appear, giving you options to:
  - *Calculate from layer* ►: fills the text box with the coordinates of the bounding box of a layer to select among the loaded ones
  - *Calculate from layout map* ►: fills the text box with the coordinates of a map item selected from a layout in the current project
  - *Calculate from bookmark* ►: fills the text box with the coordinates of a saved bookmark
  - *Use current map canvas extent*

– *Draw on canvas*: the parameters window will hide itself, so you can click and drag onto the canvas. Once you have defined the extent rectangle, the dialog will reappear, containing the values in the extent text box.

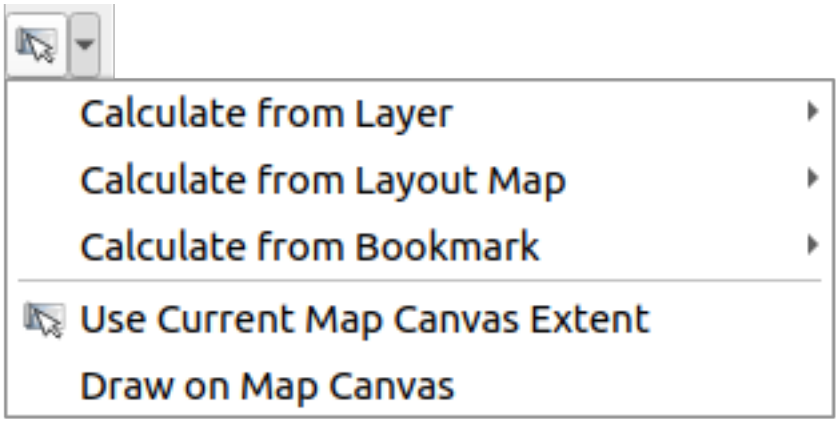


Fig. 23.13: Extent selector

- A **list of elements** (whether raster or vector layers, tables, fields) to select from. Click on the ... button at the left of the option to see a dialog like the following one. Multiple selection is allowed and when the dialog is closed, number of selected items is displayed in the parameter text box widget.

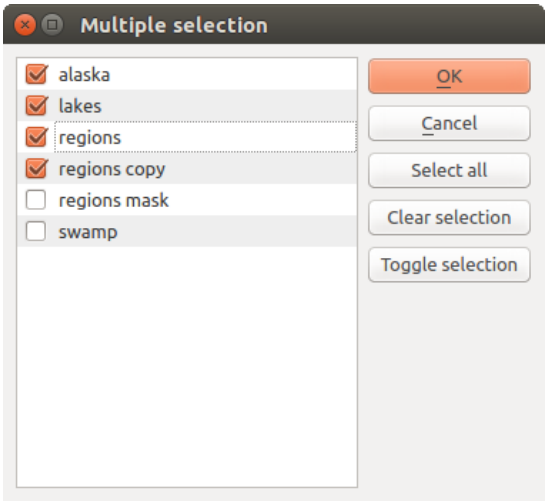


Fig. 23.14: Multiple Selection

- A **small table** to be edited by the user. These are used to define parameters like lookup tables or convolution kernels, among others.

  Click on the button on the right side to see the table and edit its values.

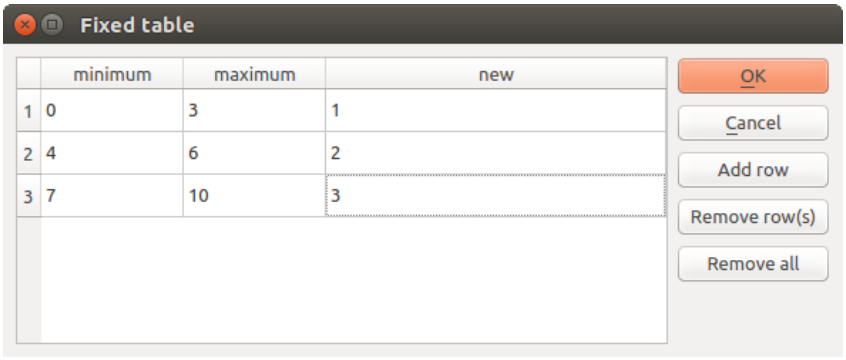


Fig. 23.15: Fixed Table

Depending on the algorithm, the number of rows can be modified or not by using the buttons on the right side of the window.

Note

Some algorithms require many parameters to run, e.g. in the *Raster calculator* you have to specify manually the cell size, the extent and the CRS. You can avoid to choose all the parameters manually when the algorithm has the `Reference layers` parameter. With this parameter you can choose the reference layer and all its properties (cell size, extent, CRS) will be used.

Logging the execution

Along with the *Parameters* tab, there is another tab named *Log* (see Fig. 23.16 below). Information provided by the algorithm during its execution is written in this tab, allowing you to track the execution as well as being aware and having more details about the algorithm as it runs. You can directly click on the names of output files, folders, or HTML files listed in the log. Doing so will open the folder containing the generated file and automatically select it. Information on algorithm execution is also output in the *View* ► *Panels* ► *Log Messages Panel*.

Notice that not all algorithms write information to the *Log* tab, and many of them might run silently without producing any output other than the final files. Check the *Log Messages Panel* in that case.

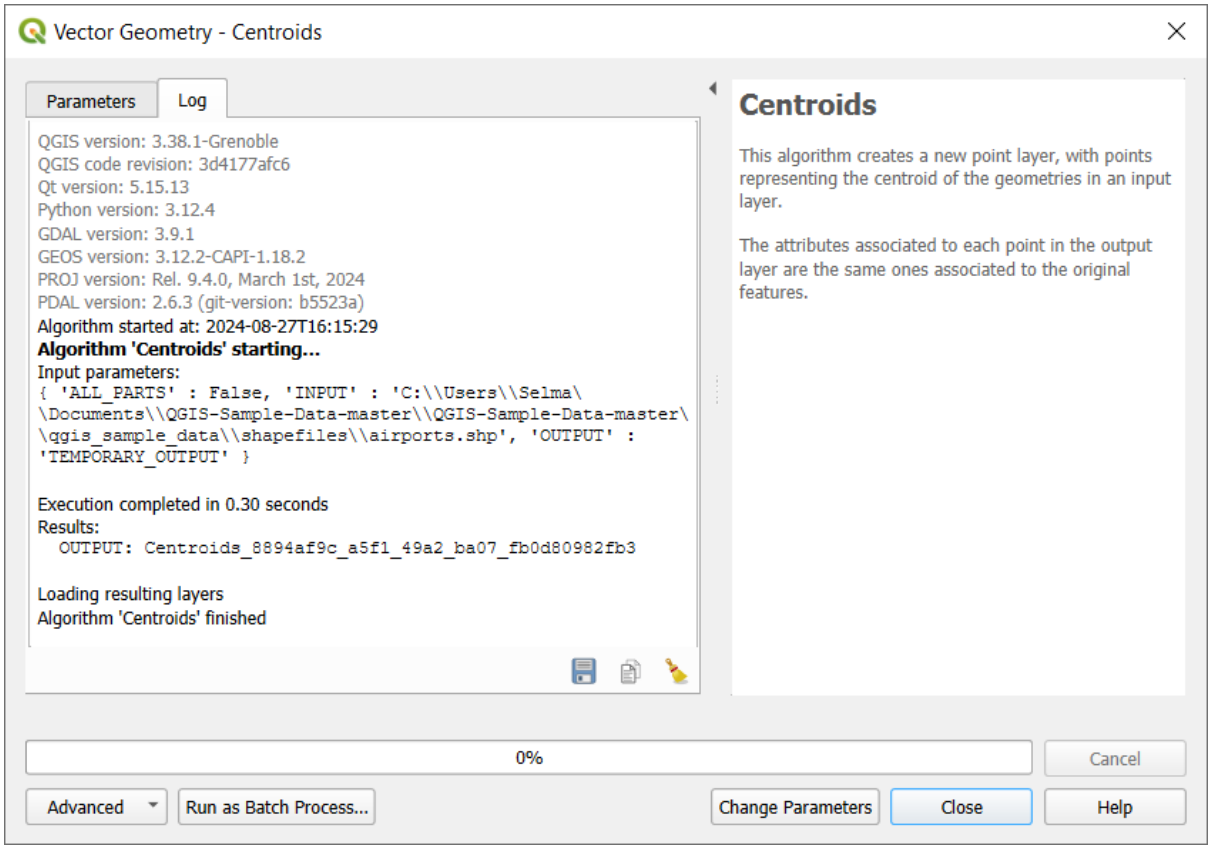


Fig. 23.16: Algorithm Dialog - Log

At the bottom of the *Log* tab you will find buttons to *Save Log to File*, *Copy Log to Clipboard* and *Clear Log*. These are particularly handy when you have checked the *Keep dialog open after running algorithm* in the *General* part of the Processing options.

Other tools

On the right hand side of the dialog you will find a short description of the algorithm, which will help you understand its purpose and its basic ideas. If such a description is not available, the description panel will not be shown.

For a more detailed help file, which might include description of every parameter it uses, or examples, you will find a *Help* button at the bottom of the dialog bringing you to the *Processing algorithms documentation* or to the provider documentation (for some third-party providers).

The *Advanced* ► menu provides functions to reuse the configuration defined in the dialog without running the algorithm:

- *Algorithm Settings...*: allows to override processing settings for the current algorithm execution. More details at *Override algorithm settings*.
- *Copy as Python Command*: allows for easy copying of the equivalent *PyQGIS command* to run the tool using the parameters defined in the dialog
- *Copy as qgis_process Command*: allows for easy generation of *qgis_process command*, including its environment settings like the distance units, area units, ellipsoid, and any tricky parameter values like GeoPackage outputs with specific layers
- *Copy as JSON*: all the settings of the command are copied in a `JSON` format, ready to be consumed by qgis_process. This is a convenient way to see the expected format of the commands, even for complex

parameters (like TIN interpolation parameters). You can store these easily and then restore them later by pasting the values.

- *Paste Settings* in a `JSON` format

The *Run as Batch Process...* button triggers the *batch processing mode* allowing to configure and run multiple instances of the algorithm with a variety of parameters. A *Run as Single Process...* button helps you switch back from the batch mode.

When an algorithm execution finishes (either successfully or not), a new button *Change Parameters* is shown as long as the *Log* tab is active.

Override algorithm settings

Triggered from within the *Advanced* drop-down menu at the bottom of an algorithm dialog, the *Algorithm Settings...* shows a panel allowing users to control general processing settings which apply to that algorithm execution only. It is intended to be a place where a user can override their *global processing settings* on an ad-hoc basis without having to change their usual default settings.

Settings that can be overridden are:

- *Invalid feature filtering*: unlike the existing per-parameter setting override for this, setting the handling method here will apply to **ALL inputs** for the algorithm
- *Calculation settings*, such as *Distance units* and *Area units* to use for distance/area measurements
- *Environment settings*, such as *Temporary folder* and *Number of threads to use*

A note on projections

Processing algorithm execution are always performed in the input layer coordinate reference system (CRS). Due to QGIS's on-the-fly reprojecting capabilities, although two layers might seem to overlap and match, that might not be true if their original coordinates are used without reprojecting them onto a common coordinate system. Whenever you use more than one layer as input to a *QGIS native algorithm*, whether vector or raster, the layers will all be reprojected to match the coordinate reference system of the first input layer.

This is however less true for most of the external applications whose algorithms are exposed through the processing framework as they assume that all of the layers are already in a common coordinate system and ready to be analyzed.

By default, the parameters dialog will show a description of the CRS of each layer along with its name, making it easy to select layers that share the same CRS to be used as input layers. If you do not want to see this additional information, you can disable this functionality in the Processing settings dialog, unchecking the *Show layer CRS definition in selection boxes* option.

If you try to execute an algorithm using as input two or more layers with unmatching CRSs, a warning dialog will be shown. This occurs thanks to the *Warn before executing if layer CRS's do not match* option.

You still can execute the algorithm, but be aware that in most cases that will produce wrong results, such as empty layers due to input layers not overlapping.

> **Tip**
>
> **Use Processing algorithms to do intermediate reprojection**
>
> When an algorithm can not successfully perform on multiple input layers due to unmatching CRSs, use QGIS internal algorithm such as *Reproject layer* to perform layers' reprojection to the same CRS before executing the algorithm using these outputs.

23.3.2 Data objects generated by algorithms

Data objects generated by an algorithm can be of different types. They can be a layer (vector with or without geometry, raster, mesh, point cloud, ...), stored as a plain file on disk or in a database, text-based files such as `qml` style file or graphics `.html` file, a folder, ...

The parameters dialog will contain a widget corresponding to the needed output, where you can type the output channel to use for saving it. An output channel contains the information needed to save the resulting object somewhere. In the most usual case, you will save it to a file, but in the case of vector layers, and when they are generated by native algorithms (algorithms not using external applications) you can also save to a PostgreSQL, GeoPackage or SpatiaLite database, or a memory layer.

The **output** parameter for setting the result of the algorithm execution provides a text box for filling the path to the output result, and a drop-down ... button for additional options (the availability depends on the algorithm you run):

- *Skip Output*: if you are not interested in a given output of the algorithm
- *Create Temporary Layer* (`TEMPORARY_OUTPUT`): the output is stored in a vector *temporary scratch layer*.
- *Save to File…*: you will be prompted with a save file dialog, where you can select the desired file path. Supported file extensions are shown in the file format selector of the dialog, depending on the kind of output and the algorithm.

  The format of the output is defined by the filename extension. The supported formats depend on what is supported by the algorithm itself. To select a format, just select the corresponding file extension (or add it, if you are directly typing the file path instead). If the extension of the file path you entered does not match any of the supported formats, a default extension will be appended to the file path, and the file format corresponding to that extension will be used to save the layer or table. Default extensions are `.dbf` for tables, `.tif` for raster layers and `.gpkg` for vector layers. These can be modified in the setting dialog, selecting any other of the formats supported by QGIS.

  You can set a default folder for output data objects. Go to the settings dialog (you can open it from the *Settings* ► *Options* ► *Processing* menu), and in the *General* group, you will find a parameter named *Output folder*. This output folder is used as the default path in case you type just a filename with no path (i.e., `myfile.shp`) when executing an algorithm.
- *Save to a Temporary File*: if you do not enter any filename in the output text box (or select the corresponding option in the context menu), the result will be saved as a temporary file in the corresponding default file format, and it will be deleted once you exit QGIS (take care with that, in case you save your project and it contains temporary layers). The file is saved in the default `TEMP` folder on the machine, if *not overridden*.
- *Save to Geopackage…*: the output layer will be saved as a named table of a new or existing GeoPackage database
- *Save to Database Table…*: the output layer will be saved as a new table in a database such as GeoPackage, SpatiaLite, PostgreSQL or MS SQL Server. A connection should already exist. You'll need to provide the name of the new table and depending on the type of target database, select a schema,
- *Append to Layer…*: adds features output by the algorithm to an existing layer. Upon selecting the target layer, a field mapping panel opens, allowing to manually set how fields from the output layer are mapped to the target layer's fields.

- *Save to Directory*: Use a new or existing directory for storing the algorithm outputs
- *Save to a Temporary Directory*: creates a directory within the set `TEMP` folder for storing the algorithm outputs. The files will be deleted once you exit QGIS.

When running an algorithm that uses a vector layer in iterative mode, the entered file path is used as the base path for all generated files, which are named using the base name and appending a number representing the index of the iteration. The file extension (and format) is used for all such generated files.

Algorithms also generate graphics and text as HTML files. These results are shown at the end of the algorithm execution in the *Results Viewer* dialog. This dialog will keep the results produced by any algorithm during the current session, and can be shown at any time by selecting *Processing* ► *Results Viewer* from the QGIS main menu.

For layer-based outputs, it is possible to *Open output file after running algorithm* directly in QGIS, as a new entry in the *Layers* panel. By default, the files are opened.

23.4 The history manager

23.4.1 The processing history

Every time you execute an algorithm, information about the process is stored in the history manager. The date and time of the execution are saved, along with the parameters used, making it is easy to track and control all the work that has been developed using the Processing framework, and to reproduce it.

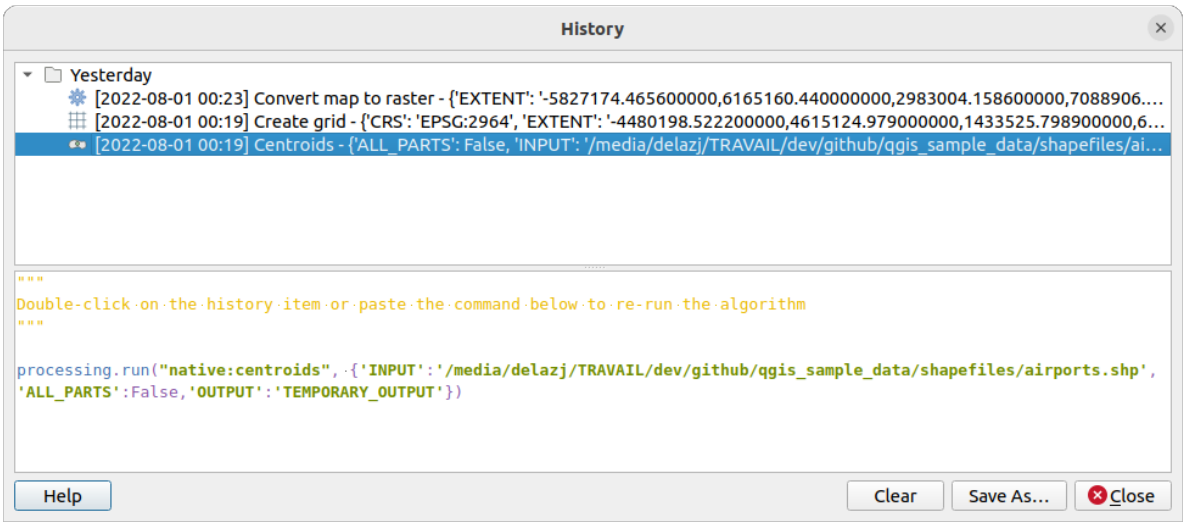


Fig. 23.17: 历史

Process information is kept as a command-line expression, even if the algorithm was launched from the toolbox. This makes it useful for those learning how to use the command-line interface, since they can call an algorithm using the toolbox and then check the history manager to see how it could be called from the command line.

Right-click on a row and you should be able to:

- *Copy as Python Command*: allows for easy copying of the equivalent *PyQGIS command* run from the dialog. Same as the code displayed below the commands list.
- *Copy as qgis_process Command*: allows for easy generation of *qgis_process command*, including its environment settings like the distance units, area units, ellipsoid, and any tricky parameter values like GeoPackage outputs with specific layers
- *Copy as JSON*: all the settings of the command are copied in a `JSON` format, ready to be consumed by qgis_process. This is a convenient way to see the expected format of the commands, even for complex parameters (like TIN interpolation parameters). You can store these easily and then restore them later by pasting the values over an algorithm dialog.
- *Create Test...* using the concerned algorithm and parameters, following instructions at Processing README file.

Apart from browsing the entries in the registry, you can also re-execute processes by simply double-clicking on the entry. The algorithm dialog then opens with parameters already set, and you can change any of them to fit your needs and re-run the algorithm.

The *History* dialog also provides a convenient way to contribute to the consolidation of the testing infrastructure of QGIS Processing algorithms and scripts.

23.4.2 The processing log

The history dialog only contains the execution calls, but not the information produced by the algorithm when executed. That information is written to the QGIS log (*View* ► *Panels* ► *Log Messages Panel*).

Third-party algorithms are usually executed by using their command-line interfaces, which communicate with the user via the console. Although that console is not shown, usually a full dump of it is written to the log each time you run one of those algorithms. To avoid cluttering the log with that information, you can disable it for each provider in the settings dialog.

Some algorithms, even if they can produce a result with the given input data, output comments or additional information to log when they detect potential problems with the data, in order to warn you. Make sure you check those messages in the log if you get unexpected results.

23.5 The model designer

The *model designer* allows you to create complex models using a simple and easy-to-use interface. When working with a GIS, most analysis operations are not isolated, rather part of a chain of operations. Using the model designer, that chain of operations can be wrapped into a single process, making it convenient to execute later with a different set of inputs. No matter how many steps and different algorithms it involves, a model is executed as a single algorithm, saving time and effort.

The model designer can be opened from the Processing menu (*Processing* ► *Model Designer*).

23.5.1 The model designer interface

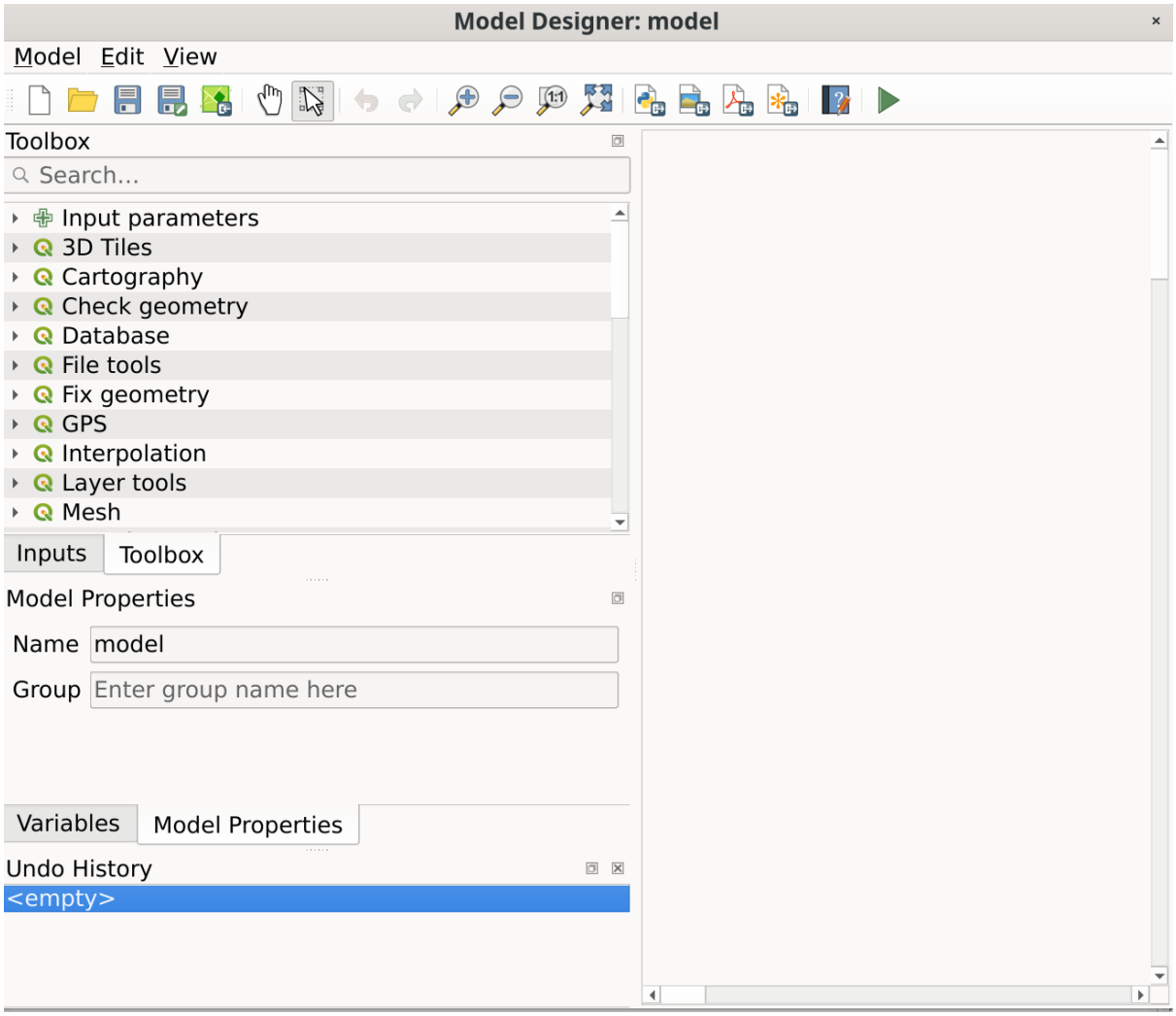


Fig. 23.18: Model designer

In its main part, the modeler has a working canvas where the structure of the model and the workflow it represents can be constructed.

At the top of the dialog, different menus and the *Navigation* toolbar give access to a variety of tools.

Model menu

标注	快捷键	Navigation Toolbar	描述
Validate Model			Checks whether the algorithms and inputs used in the model exist. Convenient before releasing a model.
Run Model...	F5	☑	Executes the model
Run Selected Steps...	Shift+F5		Runs only the selected steps in a model, allowing to run a subset of the model
Reorder Model Inputs...			Sets the order in which inputs are presented to the user in the algorithm dialog.
Reorder Output Layers...			Sets a specific order which the outputs from the model must use when loading the results into a project.
Open Model...	Ctrl+O	☑	Opens a `.model3` file for edit or execution
Save Model	Ctrl+S	☑	Saves the model to disk as a `.model3` file
Save Model as...	Ctrl+Shi	☑	Saves the model to disk as a new `.model3` file
Save Model in project		☑	Embeds the model file in the project file, making it available when sharing the project file.
Edit Model Help...		☑	An interface to document the model, the algorithms, the parameters and outputs, as well as the author and versioning
导出 ►			
► *Export as Image...*		☑	Saves the model's graphical design to an image file format (for illustration purpose)
► *Export as PDF...*			Saves the model's graphical design to a `PDF` file format (for illustration purpose)
► *Export as SVG...*			Saves the model's graphical design to an `SVG` file format (for illustration purpose)
► *Export as Script Algorithm...*		☑	Generates a python script file including the model's instructions

Edit menu

标注	快捷键	Navigation Toolbar	描述
全选	Ctrl+A		Selects all the model components in the designer
Snap selected components to Grid			snaps and aligns the elements into a grid
重做	Ctrl+Y	☑	Rollback the latest canceled action. See also the *Undo/Redo* panel.
撤销	Ctrl+Z	☑	Cancel the previous change. See also the *Undo/Redo* panel.
剪切	Ctrl+X		Cuts a selection of components from the model.
复制	Ctrl+C		Copies a selection of components from the model.
粘贴	Ctrl+V		Pastes a cut or copied selection of components from a model to another or within the same model. The selected components keep their original properties and comments.
Delete selected components	Del		Removes a component from the model.
Add Group Box			Adds a box at the background of related components in order to visually group them. Particularly useful in big models to keep the workflow clean.

View menu

标注	快捷键	Navigation Toolbar	描述
Zoom To ►			Zooms to the selected group box extent
放大	Ctrl++	☑	
缩小	Ctrl+-	☑	
缩放到 *100%*	Ctrl+1	☑	
全图显示	Ctrl+0	☑	Displays all the components in the designer current canvas
Show Comments			Displays comments associated to every algorithm or input in the model designer
Enable Snapping			
切换面板可见性	Ctrl+Tab		Switches ON or OFF the *panels* in the designer

面板

The left part of the window is a section with five panels that can be used to add new elements to the model:

1. *Inputs*: all the *input parameters* that could shape your model
2. *Toolbox*: here you can find available *Processing algorithms*. It also includes input parameters. Use *Search...* box to find algorithms or inputs.
3. *Model Properties*: specify the name (required) of the model and the group in which it will be displayed in the *Processing Toolbox*
4. *Variables*: Models can contain dedicated *variables* that are unique and only available to them. These variables can be accessed by any expression used within the model. They are useful to control algorithms within a model and control multiple aspects of the model by changing a single variable. The variables can be viewed and modified in the *Variables* panel.
5. *Undo History*: this panel will register everything that happens in the modeler, making it easy to cancel things you did wrong.

About available algorithms

Some algorithms that can be executed from the *Processing* ► *Toolbox* do not appear in the list of available algorithms when you are designing a model. To be included in a model, an algorithm must have the correct semantic. If an algorithm does not have such a well-defined semantic (for instance, if the number of output layers cannot be known in advance), then it is not possible to use it within a model, and it will not appear in the list of algorithms that you can find in the modeler dialog. On the other hand some algorithms are specific to the modeler. Those algorithms are located within the group 'Modeler Tools'.

23.5.2 Creating a model

Creating a model involves two basic steps:

1. *Definition of necessary inputs*. These inputs will be added to the parameters window, so the user can set their values when executing the model. The model itself is an algorithm, so the parameters window is generated automatically as for all algorithms available in the Processing framework.
2. *Definition of the workflow*. Using the input data of the model, the workflow is defined by adding algorithms and selecting how they use the defined inputs or the outputs generated by other algorithms in the model.

Definition of inputs

The first step is to define the inputs for the model. They are found in the *Inputs* panel on the left side of the modeler window. Hovering with the mouse over the inputs will show a tooltip with additional information. For a full list of available parameters in modeler and their correspondance for scripting, please read 处理算法的输入与输出.

When double-clicking on an element, a dialog is shown that lets you define its characteristics. Depending on the parameter, the dialog will contain at least one element (the description, which is what the user will see when executing the model). For example, when adding a numerical value, as can be seen in the next figure, in addition to the description of the parameter, you have to set a default value and the range of valid values.

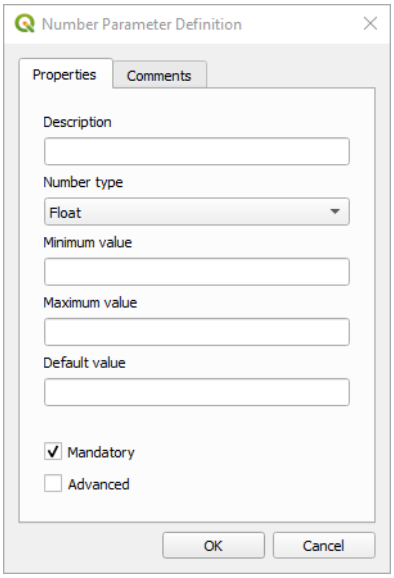


Fig. 23.19: Model Parameters Definition

You can define your input as mandatory for your model by checking the Mandatory option and by checking the Advanced checkbox you can set the input to be within the Advanced section. This is particularly useful when the model has many parameters and some of them are not trivial, but you still want to choose them.

For each added input, a new element is added to the modeler canvas.

Fig. 23.20: Model Parameters

You can also add inputs by dragging the input type from the list and dropping it at the position where you want it in the modeler canvas. If you want to change a parameter of an existing input, just double click on it, and the same dialog will pop up.

When using a model within another model, the inputs and outputs necessary will be displayed in the canvas.

Definition of the workflow

In the following example we will add two inputs and two algorithms. The aim of the model is to copy the elevation values from a DEM raster layer to a line layer using the Drape algorithm, and then calculate the total ascent of the line layer using the Climb Along Line algorithm.

In the *Inputs* tab, choose the two inputs as Vector Layer for the line and Raster Layer for the DEM. We are now ready to add the algorithms to the workflow.

Algorithms can be found in the *Toolbox* panel, grouped much in the same way as they are in the Processing toolbox.

Fig. 23.21: Model Inputs

To add an algorithm to a model, double-click on its name or drag and drop it, just like for inputs. As for the inputs you can change the description of the algorithm and add a comment. When adding an algorithm, an execution dialog will

appear, with a content similar to the one found in the execution panel that is shown when executing the algorithm from the toolbox. The following picture shows both the `Drape (set Z value from raster)` and the `Climb along line` algorithm dialogs.

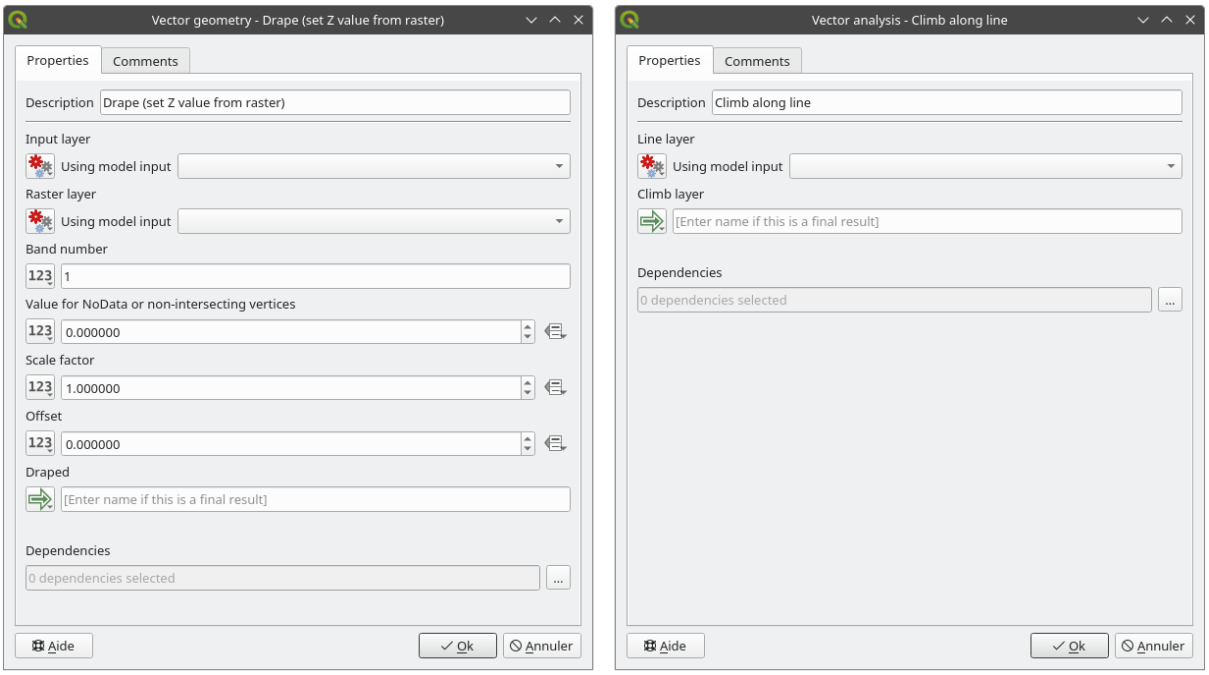


Fig. 23.22: Model Algorithm parameters

As you can see, there are however some differences. Each parameter has a drop-down menu next to it allowing to control how it will be served during the workflow:

- Value: allows you to assign a static value to the parameter. Depending on the parameter type, the widget will let you enter a number (`5.0`), a string (`mytext`), select layer(s) loaded in the QGIS project or from a folder, pick items from a list, ...
- Pre-calculated Value: opens the *Expression Builder* dialog and lets you define an expression to fill the parameter. Model inputs together with some other layer statistics are available as **variables** and are listed at the top of the Search dialog of the Expression Builder. The expression is evaluated once before the child algorithm is executed and used during the execution of that algorithm.
- Model Input: allows to use an input added to the model as a parameter. Once clicked, this option will list all the suitable inputs for the parameter.
- Algorithm Output: allows to use the output of another algorithm as an input of the current algorithm. As of model inputs, this option will list all the suitable inputs for the parameter.
- The **output parameter** also has the above options in its drop-down menu:
  - add static outputs for child algorithms, e.g. always saving a child algorithm's output to a predefined geopackage or postgres layer
  - use an expression based output values for child algorithms, e.g. generating an automatic file name based on today's date and saving outputs to that file
  - use a model input, e.g. the *File/Folder* model input to specify an output file or folder
  - use another algorithm output, e.g. the output of the *Create directory* algorithm (from *Modeler tools*)
  - an addditional Model Output option makes the output of the algorithm available in the model. If a layer generated by the algorithm is only to be used as input to another algorithm, don't edit that text box.

In the following picture you can see the two input parameters defined as `Model Input` and the temporary output layer:

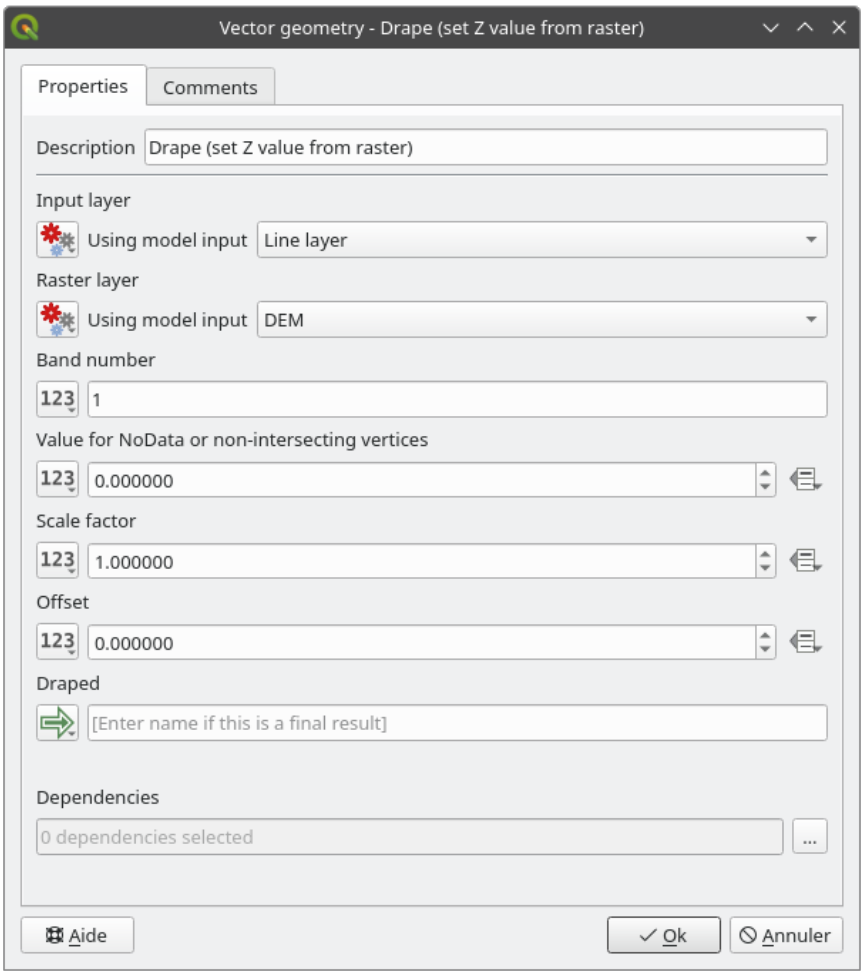


Fig. 23.23: Algorithm Input and Output parameters

You will also find an additional parameter named *Dependencies* that is not available when calling the algorithm from the toolbox. This parameter allows you to define the order in which algorithms are executed, by explicitly defining one algorithm as a *parent* of the current one. This will force the *parent* algorithm to be executed before the current one.

When you use the output of a previous algorithm as the input of your algorithm, that implicitly sets the previous algorithm as parent of the current one (and places the corresponding arrow in the modeler canvas). However, in some cases an algorithm might depend on another one even if it does not use any output object from it (for instance, an algorithm that executes a SQL sentence on a PostgreSQL database and another one that imports a layer into that same database). In that case, just select the previous algorithm in the *Dependencies* parameter and they will be executed in the correct order.

Once all the parameters have been assigned valid values, click on *OK* and the algorithm will be added to the canvas. It will be linked to the elements in the canvas (algorithms or inputs) that provide objects that are used as inputs for the algorithm.

Elements can be dragged to a different position on the canvas using the Select/Move/Link Item tool. This is useful to make the structure of the model clearer and more intuitive. You can also resize the elements, grasping their border. This is particularly useful if the description of the input or algorithm is long. With *View* ► *Enable snapping* option checked, items resizing or displacement can be bound to a virtual grid, for a more visually structured algorithm design.

Links between elements are updated automatically and you can see a + button at the top and at the bottom of each

algorithm. Clicking the button will list all the inputs and outputs of the algorithm so you can have a quick overview.

You can create or remove connections between model components using an interactive drag-and-drop interface. Each parameter and algorithm item on the model canvas includes small circular sockets, output sockets on the right and input sockets on the left. To connect components, click and drag from an output socket (either from a model input or an algorithm) to an input socket on another algorithm. A visual edge follows the cursor and completes the connection when released. To remove a connection, drag the edge away from the input socket. The algorithm dialogs update automatically to reflect changes made using the drag-and-drop interface.

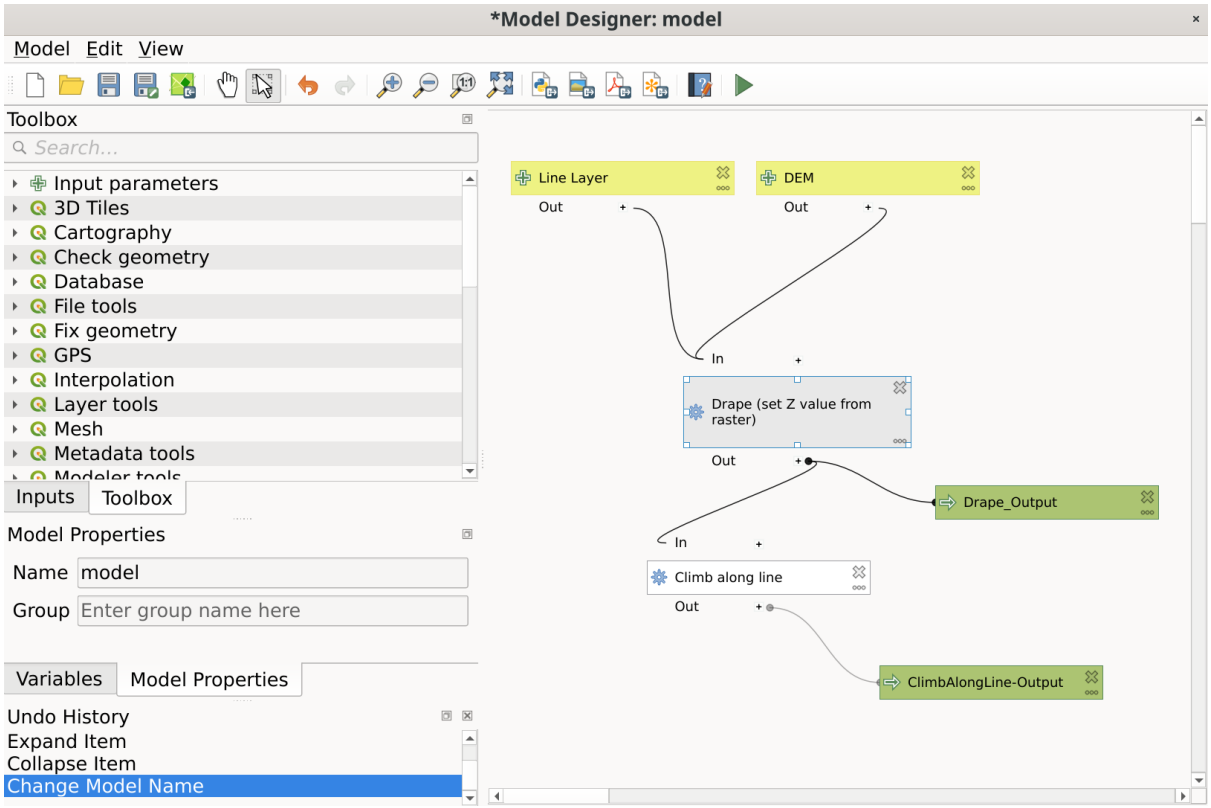


Fig. 23.24: A complete model

With the *Edit* ► *Add Group Box* tool, you can add a draggable *box* to the canvas. This feature is very useful in big models to group related elements in the modeler canvas and to keep the workflow clean. For example we might group together all the algorithms of the example:

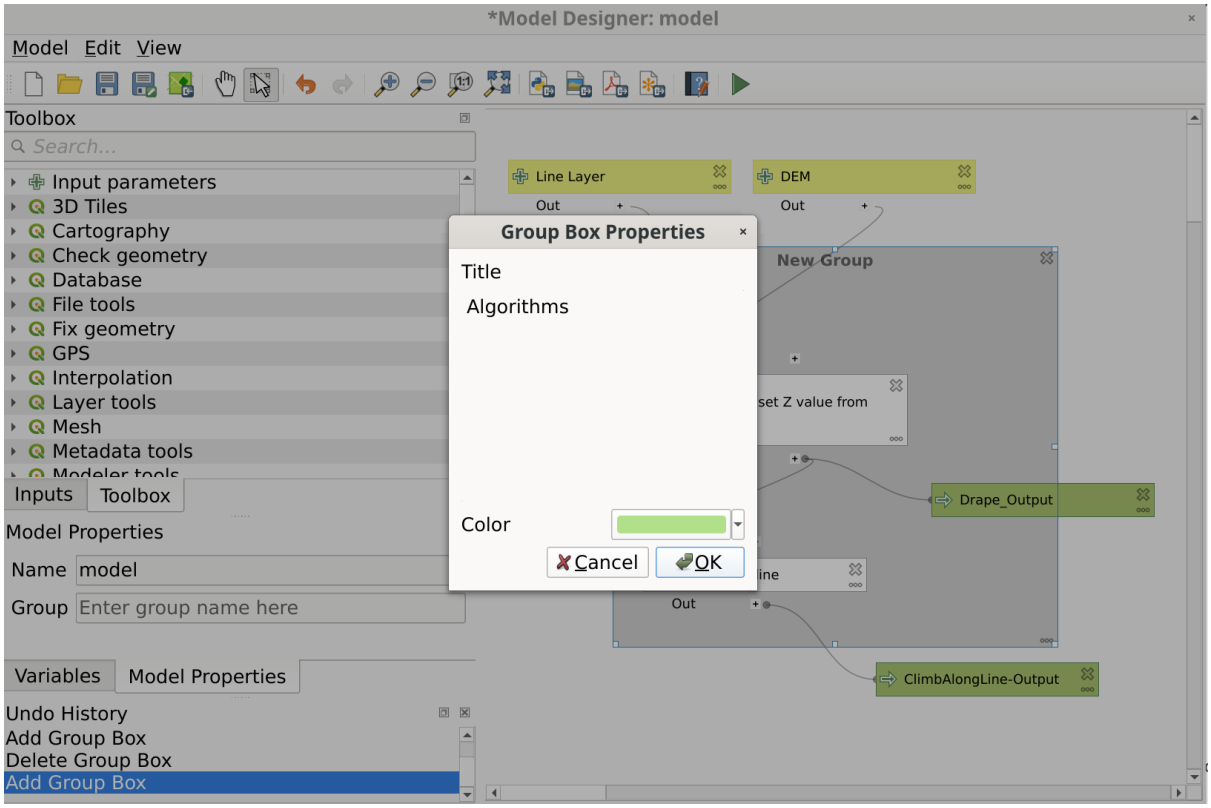


Fig. 23.25: Model Group Box

You can change the name and the color of the boxes. Group boxes are very useful when used together with *View* ► *Zoom To* ► tool, allowing you to zoom to a specific part of the model. You can also zoom in and out by using the mouse wheel.

You might want to change the order of the inputs and how they are listed in the main model dialog. At the bottom of the `Input` panel you will find the `Reorder Model Inputs...` button and by clicking on it a new dialog pops up allowing you to change the order of the inputs:

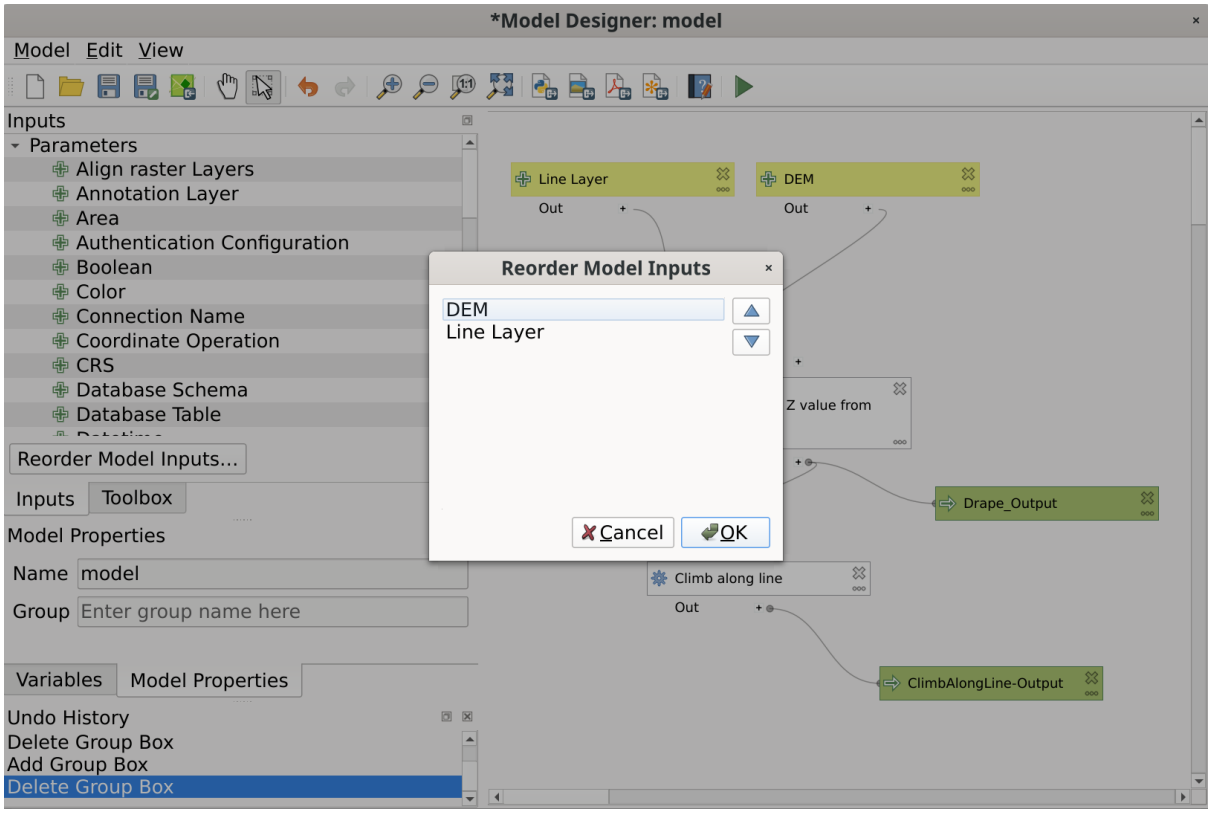


Fig. 23.26: Reorder Model Inputs

There is also the possibility to set a specific order which the outputs from the model must use when loading the results into a project. This gives the model creator a means of ensuring that layers are logically ordered on the canvas when running a model, such as placing a vector layer output over a raster layer output, or a point layer over a polygon layer. The model creator can also set an optional "Group name" for the outputs for automatically grouping outputs within the layer tree using a new group name or by adding them to an existing group. In the `Model` menu you will find the `Reorder Output Layers...` entry and by clicking on it a new dialog pops up allowing you to change the order of the output layers:

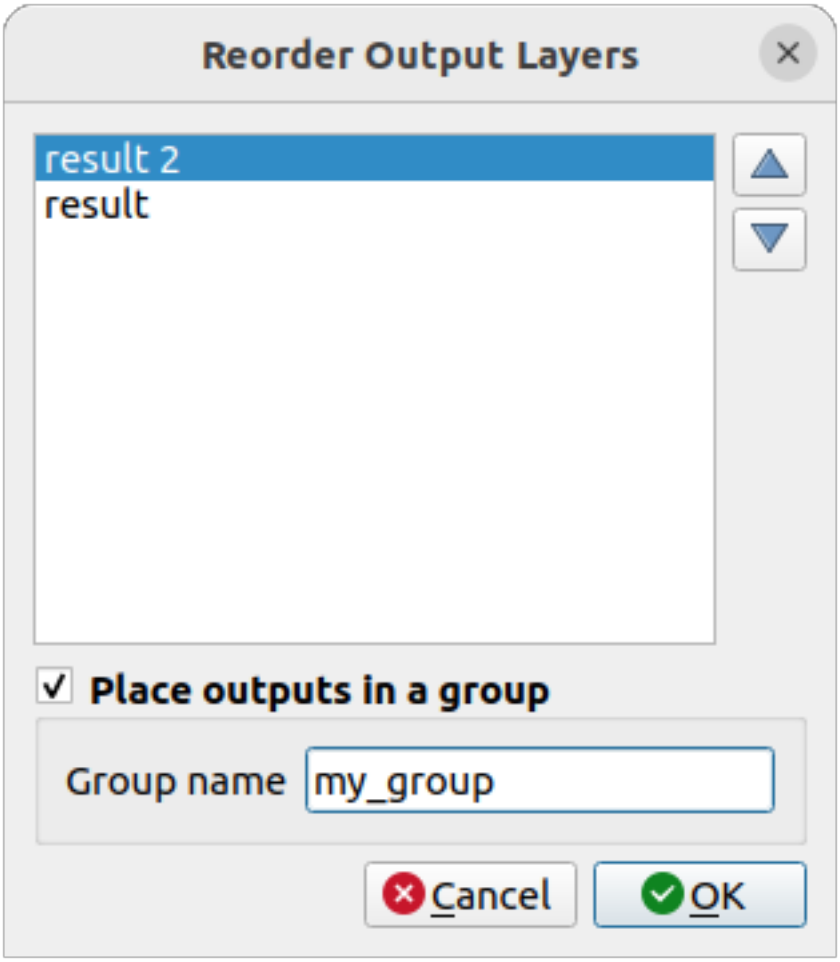


Fig. 23.27: Reorder Output Layers

Comments can also be added to inputs or algorithms present in the modeler. This can be done by going in the *Comment* tab of the item or with a right-click. In the same tab a color can be set manual for individual model comments. Comments are visible only in the modeler canvas and not in the final algorithm dialog; they can be hidden by deactivating *View* ► *Show Comments*.

Your model can be run in various ways:

- You can run the whole model by clicking on Run Model... from the toolbar, right-clicking on the model in the *Browser panel* or *Model* ► *Run Model...*: when using the editor to execute a model, any non-default values will be saved in the inputs. This means that executing the model at a later time from the editor will have the dialog prefilled with those values on any subsequent run.
- You can select elements of the model and run only that subset of the model: press the *Run Selected Steps...* option from the *Models* ► menu or from the contextual menu of a selected algorithm. The initial state will be taken from any previous executions of the model through the editor, so results from previous steps in the model are available for the selected steps. This makes it possible to fix parts of a large model, without having to constantly run the entire model to test. Especially useful when earlier steps in the model are time consuming.
- You can run a subset of the model, starting from a specific algorithm: right-click the algorithm and select *Run from Here...*. Likewise, values are taken from previous executions.
- In order to use the algorithm from the Processing Toolbox, it has to be saved and the modeler dialog closed, to allow the toolbox to refresh its contents.

Documenting your model

You need to document your model, and this can be done from the modeler itself. Click on the Edit model help button, and a dialog like the one shown next will appear.

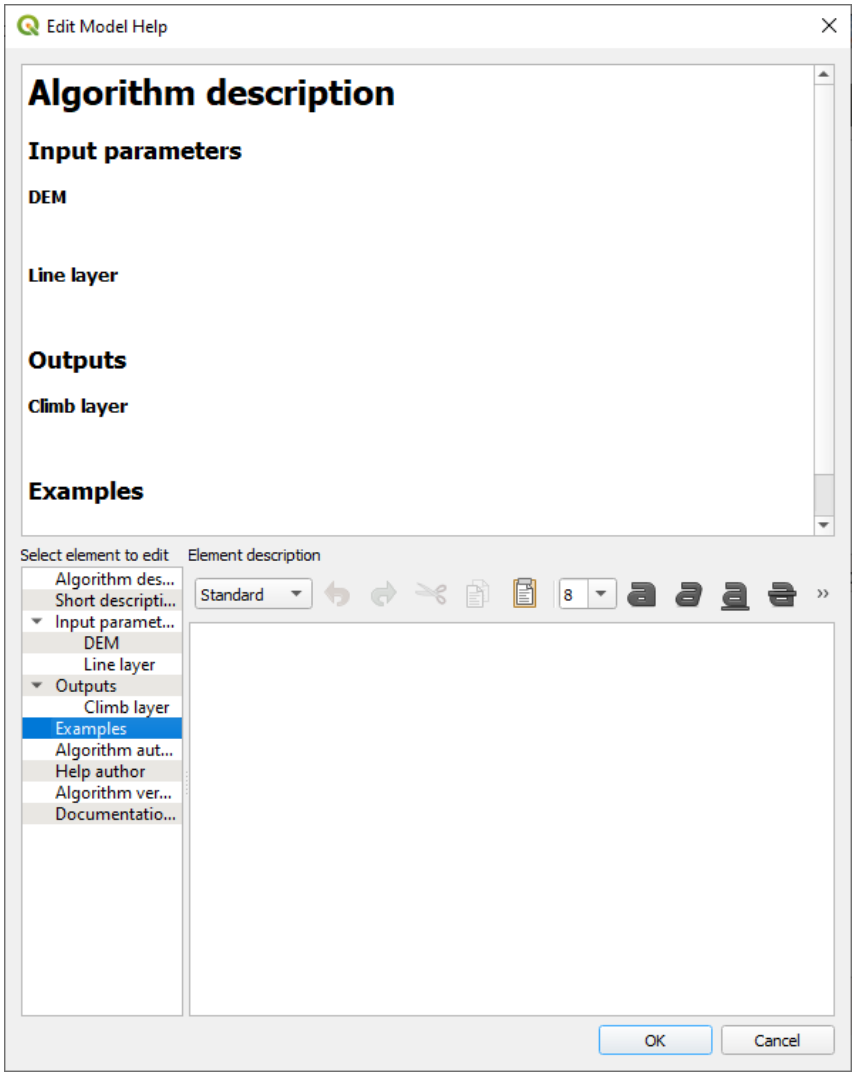


Fig. 23.28: Editing Help

On the right-hand side, you will see a simple HTML page, created using the description of the input parameters and outputs of the algorithm, along with some additional items like a general description of the model or its author. Also, there is an Example section where you can input your own custom examples to help explain the usage of the model. The first time you open the help editor, all these descriptions are empty, but you can edit them using the elements on the left-hand side of the dialog. Select an element on the upper part and then write its description in the text box below.

Model help is saved as part of the model itself.

23.5.3 Saving and loading models

Saving models

Use the

button to save the current model and the

button to open a previously saved model. Models are saved with the `.model3` extension. If the model has already been saved from the modeler window, you will not be prompted for a filename. Since there is already a file associated with the model, that file will be used for subsequent saves.

Before saving a model, you have to enter a name and a group for it in the text boxes in the upper part of the window.

Models saved in the `models` folder (the default folder when you are prompted for a filename to save the model) will appear in the toolbox in the corresponding branch. When the toolbox is invoked, it searches the `models` folder for files with the `.model3` extension and loads the models they contain. Since a model is itself an algorithm, it can be added to the toolbox just like any other algorithm.

Models can also be saved within the project file using the

button. Models saved using this method won't be written as `.model3` files on the disk but will be embedded in the project file.

Project models are available in the *Project models* menu of the toolbox and in the *Project* ► *Models* menu item.

The models folder can be set from the Processing configuration dialog, under the *Modeler* group.

Models loaded from the `models` folder appear not only in the toolbox, but also in the algorithms tree in the *Algorithms* tab of the modeler window. That means that you can incorporate a model as a part of a bigger model, just like other algorithms.

Models will show up in the *Browser* panel and can be run from there.

Exporting a model as a Python script

As we will see in a later chapter, Processing algorithms can be called from the QGIS Python console, and new Processing algorithms can be created using Python. A quick way to create such a Python script is to create a model and then export it as a Python file.

To do so, click on the

in the modeler canvas or right click on the name of the model in the Processing Toolbox and choose

.

Exporting a model as an image, PDF or SVG

A model can also be exported as an image, SVG or PDF (for illustration purposes) by clicking

,

or

.

23.5.4 Editing a model

You can edit the model you are currently creating, redefining the workflow and the relationships between the algorithms and inputs that define the model.

You can also edit existing model from the *Browser panel* choosing *Edit Model...* option.

If you right-click on an algorithm in the canvas, you will see a context menu like the one shown next:

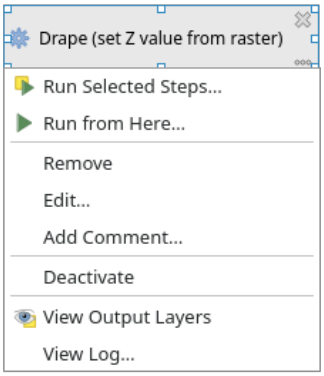


Fig. 23.29: Modeler Right Click

- Options for *running part of the model* have been exposed earlier.
- Selecting the *Remove* option will cause the selected algorithm to be removed. An algorithm can be removed only if there are no other algorithms depending on it. That is, if no output from the algorithm is used as input in a different one.
- Selecting the *Edit...* option will show the parameter dialog so you can change the inputs and parameter values. Not all input elements available in the model will appear as available inputs. Layers or values generated at a more advanced step in the workflow defined by the model will not be available if they cause circular dependencies.

  Select the new values and click on the *OK* button as usual. The connections between the model elements will change in the modeler canvas accordingly.
- The *Add comment...* allows you to add a comment to the algorithm to better describe the behavior.

Note

Right-clicking an input parameter will also allow you to *Remove*, *Edit*··· and *Add comment*···

- A model can be run partially by deactivating some of its algorithms. To do it, select the *Deactivate* option in the context menu that appears when right-clicking on an algorithm element. The selected algorithm, and all the ones in the model that depend on it will be displayed in grey and will not be executed as part of the model.

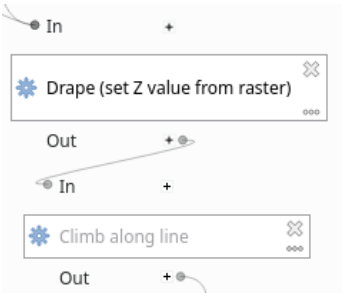


Fig. 23.30: Model with a deactivated algorithm

  When right-clicking on an algorithm that is not active, you will see an *Activate* menu option that you can use to reactivate it.
- When editing a model through the designer (and after having run that model), right-clicking any child step in the model allows to select *View Output Layers*. This will add the output layers from that step as new layers in the current QGIS project.

  This action is available for all child algorithms in the model, even if the model is not configured to use the outputs from those children as model outputs. This is designed as a helpful debugging action. If a model

fails (or gives unexpected results), you can then trace through the model and view the outputs for suspected problematic steps. It avoids the need to add temporary outputs to a model and re-run to test.

Additionally, this action is always available after running the model, even if the model itself failed (e.g., because of a misconfigured step later in the model).

- The *View Log* action helps you see the log for each child algorithm after you've closed down the algorithm dialog.

23.6 The batch processing interface

23.6.1 简介

All algorithms (including models) can be executed as a batch process. That is, they can be executed using not just a single set of inputs, but several of them, executing the algorithm as many times as needed. This is useful when processing large amounts of data, since it is not necessary to launch the algorithm many times from the toolbox.

To execute an algorithm as a batch process, right-click on its name in the toolbox and select the *Execute as batch process* option in the pop-up menu that will appear.

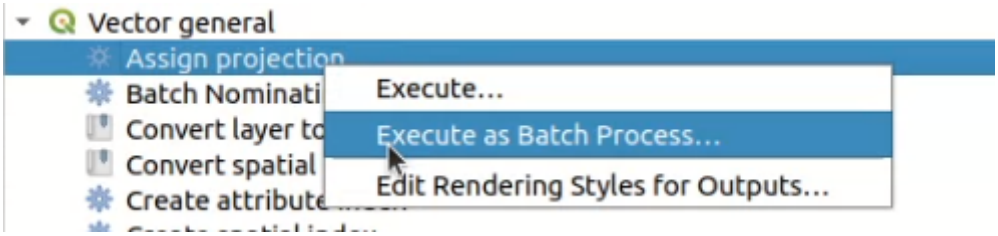


Fig. 23.31: Batch Processing from right-click

If you have the execution dialog of the algorithm open, you can also start the batch processing interface from there, clicking on the *Run as batch process...* button.

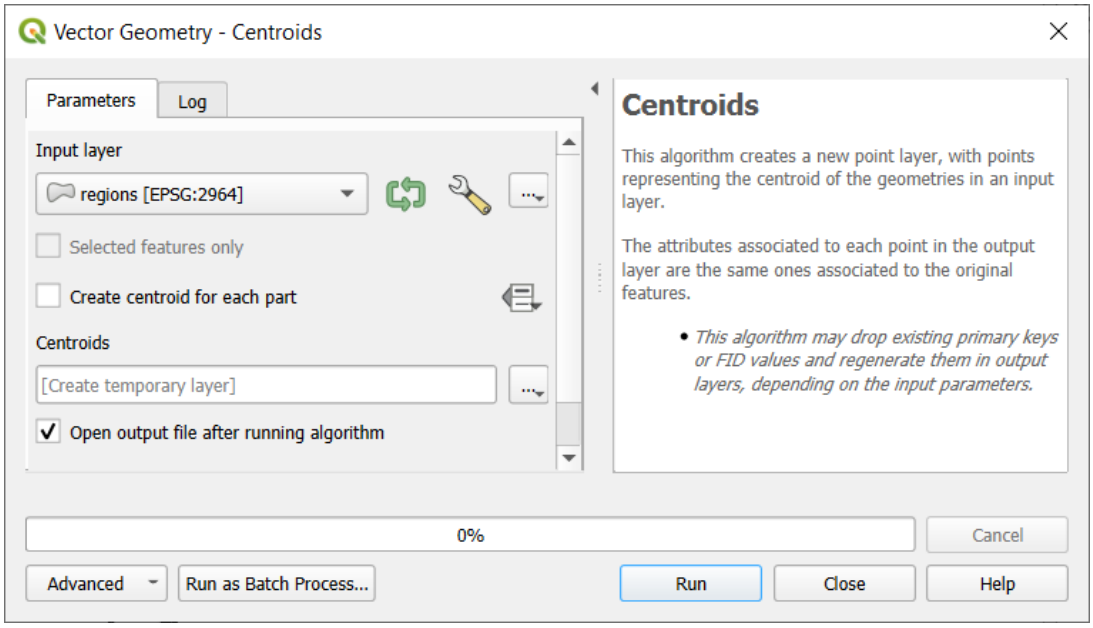


Fig. 23.32: Batch Processing From Algorithm Dialog

23.6.2 The parameters table

Executing a batch process is similar to performing a single execution of an algorithm. Parameter values have to be defined, but in this case we need not just a single value for each parameter, but a set of them instead, one for each time the algorithm has to be executed. Values are introduced using a table like the one shown next, where each row is an iteration and columns are the parameters of the algorithm.

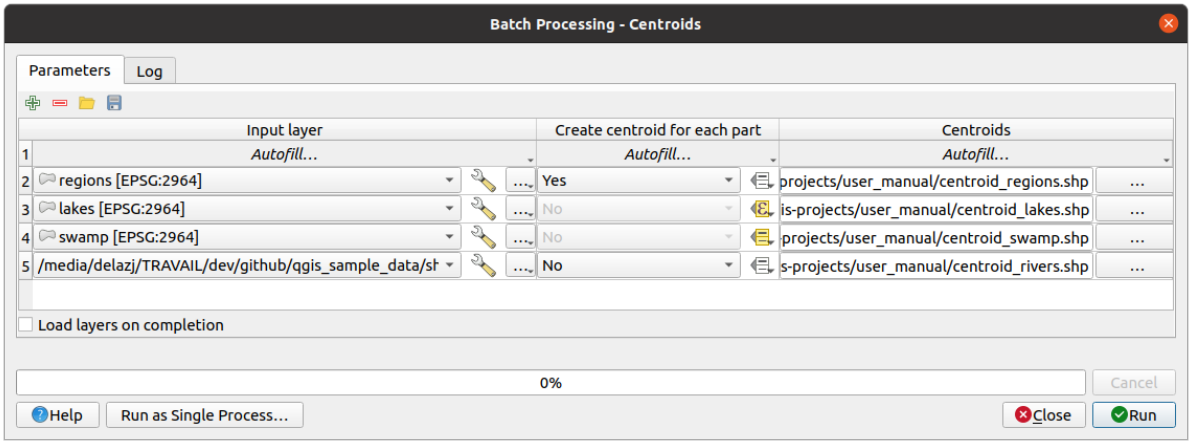


Fig. 23.33: Batch Processing

From the top toolbar, you can:

- Toggle advanced mode: Available only when the algorithm has parameters that are marked as advanced, this button allows to show or hide such parameters in the batch dialog.
- Add row: adds a new processing entry for configuration
- Remove row(s): remove selected rows from the table. Row selection is done by clicking the number at the left and allows *keyboard combination* for multi selection.
- Open a batch processing configuration file
- Save the batch processing configuration to a `.JSON` file that can be run afterwards

By default, the table contains just two rows:

- The first row displays in each cell an *Autofill...* ► drop-down menu with *options* to quickly fill the cells below. Available options depend on the parameter type.
- The second row (as well as each subsequent one) represents a single execution of the algorithm, and each cell contains the value of one of the parameters. It is similar to the parameters dialog that you see when executing an algorithm from the toolbox, but with a different arrangement.

At the bottom of the table, you can set whether to *Load layers on completion*.

Once the size of the table has been set, it has to be filled with the desired values.

23.6.3 Filling the parameters table

For most parameters, setting the value is trivial. The appropriate widget, same as in the *single process dialog*, is provided, allowing to just type the value, or select it from a list of possible values, depending on the parameter type. This also includes data-define widget, when compatible.

To automate the batch process definition and avoid filling the table cell by cell, you may want to press down the *Autofill...* menu of a parameter and select any of the following options to replace values in the column:

- *Fill Down* will take the input for the first process and enter it for all other processes.
- *Calculate by Expression...* will allow you to create a new QGIS expression to use to update all existing values within that column. Existing parameter values (including those from other columns) are available for use inside the expression via *variables*. E.g. setting the number of segments based on the buffer distance of each layer:

```
CASE WHEN @DISTANCE > 20 THEN 12 ELSE 8 END
```

- *Add Values by Expression...* will add new rows using the values from an expression which returns an array (as opposed to *Calculate by Expression...*, which works only on existing rows). The intended use case is to allow populating the batch dialog using complex numeric series. For example adding rows for a batch buffer using the expression `generate_series(100, 1000, 50)` for distance parameter results in new rows with values 100, 150, 200, ... 1000.
- When setting a file or layer parameter, more options are provided:
  - *Add Files by Pattern...*: adds new rows to the table for files matching a *File pattern* in a folder to *Look in*. E.g. `*.shp` will add to the list all the `SHP` files in the folder. Check *Search recursively* to also browse sub-folders.
  - *Select Files...* individually on disk
  - *Add All Files from a Directory...*
  - *Select from Open Layers...* in the active project

Output data parameter exposes the same capabilities as when executing the algorithm as a single process. Depending on the algorithm, the output can be:

- skipped, if the cell is left empty
- saved as a temporary layer: fill the cell with your chosen output name, select *Create Temporary Layer* from the ... drop-down, and remember to tick the *Load layers on completion* checkbox. For temporary layers, the value you provide will be used as the layer name. You can use the *Autofill* options to construct that name.
- saved as a plain file (`.SHP`, `.GPKG`, `.XML`, `.PDF`, `.JPG`,...): choose *Select File/Folder...* from the ... drop-down. For plain files, you can set the path using the *Autofill* options exposed beforehand.

  E.g. use *Calculate by Expression...* to set output file names to complex expressions like:

```
'/home/me/stuff/buffer_' || left(@INPUT, 30) || '_' || @DISTANCE || '.shp'
```

  You can also type the file path directly or use the file chooser dialog that appears when clicking on the accompanying *Select File/Folder...* button. Once you select the file, a new dialog is shown to allow for auto-completion of other cells in the same column (same parameter).

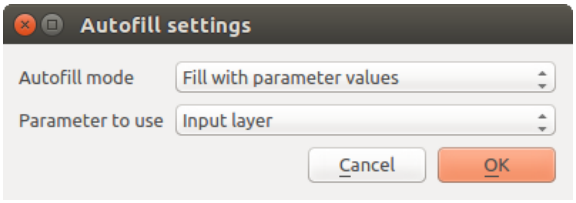


Fig. 23.34: Batch Processing Save

If the default value (*Do not autofill*) is selected, it will just put the selected filename in the selected cell from the parameters table. If any of the other options is selected, all the cells **below** the selected one will be automatically filled based on a defined criteria:

- *Fill with numbers*: incrementally appends a number to the file name
- *Fill with parameter values*: you can select a parameter whose value in the same row is appended to the file name. This is particularly useful for naming output data objects according to input ones.

- saved as a layer within a database container:

```
# Indicate a layer within a GeoPackage file
ogr:dbname='C:/Path/To/Geopackage.gpkg' table="New_Table" (geom)

# Use the "Calculate By Expression" to output to different layers in a␣
↪GeoPackage
'ogr:dbname=\'' || @project_folder || '/Buffers.gpkg\' table="' || @INPUT || '_
↪' || @DISTANCE || '" (geom)'
```

23.6.4 Executing the batch process

To execute the batch process once you have introduced all the necessary values, just click on *Run*. The *Log* panel is activated and displays details and steps of the execution process. Progress of the global batch task will be shown in the progress bar in the lower part of the dialog.

23.7 从控制台使用处理算法

控制台允许高级用户提升工作效率，并执行处理框架其他 GUI 元素无法完成的复杂操作。通过命令行接口可定义包含多个算法的模型，并添加循环、条件语句等额外操作，以创建更灵活、更强大的工作流。

QGIS 中没有独立的处理控制台，但所有处理命令均可通过 QGIS 内置的*Python 控制台* 使用。这意味着您可以将这些命令集成到控制台工作中，并将处理算法与该环境中所有其他功能（包括 QGIS API 的方法）连接起来。

您在 Python 控制台中执行的代码，即使未调用任何特定的处理方法，也可以转换为一个新算法，之后可像使用其他算法一样，从工具箱、图形建模器或其他组件中调用它。事实上，工具箱中的一些算法就是简单的脚本。

在本节中，我们将学习如何从 QGIS Python 控制台使用处理算法，以及如何使用 Python 编写算法。

23.7.1 从 Python 控制台调用算法

首先，您需要使用以下语句导入处理函数：

```
>>> from qgis import processing
```

现在，从控制台基本只能做一件（有趣）的事：执行算法。这是通过 `run()` 方法完成的，该方法以要执行的算法名称作为第一个参数，随后是根据算法需求而定的若干附加参数。因此，您首先需要知道要执行的算法名称。这并非工具箱中显示的名称，而是唯一的命令行名称。要查找算法的正确名称，可使用 `processingRegistry`。在控制台中输入以下代码：

```
>>> for alg in QgsApplication.processingRegistry().algorithms():
        print(alg.id(), "->", alg.displayName())
```

您将看到类似以下内容（为提高可读性已添加额外横线）：

```
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
...
```

这是所有可用算法 ID 的列表，按提供者名称和算法名称排序，并附带对应的名称。

一旦知道了算法的命令行名称，下一步就是确定执行它的正确语法，即了解调用 `run()` 方法时所需的参数。

有一个方法可详细描述算法，用于获取算法所需参数及其生成输出的列表。要获取此信息，可使用 `algorithmHelp(id_of_the_algorithm)` 方法。请使用算法的 ID，而非完整的描述性名称。

以 `native:buffer` 作为参数调用该方法（`qgis:buffer` 是 `native:buffer` 的别名，同样有效），您将获得如下描述：

```
>>> processing.algorithmHelp("native:buffer")
Buffer (native:buffer)

This algorithm computes a buffer area for all the features in an
input layer, using a fixed or dynamic distance.

The segments parameter controls the number of line segments to
use to approximate a quarter circle when creating rounded
offsets.

The end cap style parameter controls how line endings are handled
in the buffer.

The join style parameter specifies whether round, miter or
beveled joins should be used when offsetting corners in a line.

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.
```

(continues on next page)

(continued from previous page)

```
----------------
Input parameters
----------------

INPUT: Input layer

   Parameter type: QgsProcessingParameterFeatureSource

   Accepted data types:
           - str: layer ID
           - str: layer name
           - str: layer source
           - QgsProcessingFeatureSourceDefinition
           - QgsProperty
           - QgsVectorLayer

DISTANCE: Distance

   Parameter type: QgsProcessingParameterDistance

   Accepted data types:
           - int
           - float
           - QgsProperty

SEGMENTS: Segments

   Parameter type: QgsProcessingParameterNumber

   Accepted data types:
           - int
           - float
           - QgsProperty

END_CAP_STYLE: End cap style

   Parameter type: QgsProcessingParameterEnum

   Available values:
           - 0: Round
           - 1: Flat
           - 2: Square

   Accepted data types:
           - int
           - str: as string representation of int, e.g. '1'
           - QgsProperty

JOIN_STYLE: Join style

   Parameter type: QgsProcessingParameterEnum

   Available values:
           - 0: Round
           - 1: Miter
           - 2: Bevel

   Accepted data types:
           - int
           - str: as string representation of int, e.g. '1'
           - QgsProperty
```

(continues on next page)

(continued from previous page)

```
MITER_LIMIT: Miter limit

   Parameter type: QgsProcessingParameterNumber

   Accepted data types:
            - int
            - float
            - QgsProperty

DISSOLVE: Dissolve result

   Parameter type: QgsProcessingParameterBoolean

   Accepted data types:
            - bool
            - int
            - str
            - QgsProperty

OUTPUT: Buffered

   Parameter type: QgsProcessingParameterFeatureSink

   Accepted data types:
            - str: destination vector file, e.g. 'd:/test.shp'
            - str: 'memory:' to store result in temporary memory layer
            - str: using vector provider ID prefix and destination URI,
                   e.g. 'postgres:...' to store result in PostgreSQL table
            - QgsProcessingOutputLayerDefinition
            - QgsProperty

----------------
Outputs
----------------

OUTPUT:  <QgsProcessingOutputVectorLayer>
   Buffered
```

现在您已具备运行任意算法所需的一切。如前所述，可使用 `run()` 来运行算法。其语法如下：

```
>>> processing.run(name_of_the_algorithm, parameters)
```

其中 parameters 是一个参数字典，具体内容取决于您要运行的算法，且与 `algorithmHelp()` 方法给出的列表完全一致。

```
>>> 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'})
```

如果某个参数是可选的且您不想使用它，则不要将其包含在字典中。

如果未指定某个参数，则将使用其默认值。

根据参数类型不同，值的输入方式也不同。以下列表简要说明了各类输入参数的值输入方法：

- 栅格图层、矢量图层或表：只需使用一个字符串，指定数据对象在 QGIS 图层面板中的名称，或使用文件路径（若对应图层未打开，将自动打开但不会添加到地图画布）。如果您已有代表该图层的

QGIS 对象实例，也可直接将其作为参数传入。

- 枚举类型：若算法包含枚举参数，应使用整数值输入该参数。要了解可用选项，可使用上述的 `algorithmHelp()` 命令。例如，`native:buffer` 算法有一个名为 JOIN_STYLE 的枚举：

```
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
```

  本例中，该参数有三个选项。请注意，索引从 0 开始。

- 布尔值：使用 `True` 或 `False`。
- 多输入：值为一个字符串，各输入描述符以分号（`;`）分隔。与单图层或表类似，每个输入描述符可以是数据对象名称或其文件路径。
- 来自 XXX 的表字段：使用字符串指定要使用的字段名称。此参数区分大小写。
- 固定表格：输入所有表格值的列表，以逗号（`,`）分隔，并用引号（`"`）括起。值从上行开始，从左到右排列。您也可以使用一个二维数组来表示该表格。
- 坐标参考系（CRS）：输入所需 CRS 的 EPSG 代码编号。
- 范围（Extent）：必须使用一个字符串，其中包含以逗号（`,`）分隔的 `xmin`、`xmax`、`ymin` 和 `ymax` 值。

布尔值、文件、字符串和数值型参数无需额外说明。

字符串、布尔值或数值等输入参数具有默认值。若未提供相应参数，则使用默认值。

对于输出数据对象，请输入用于保存结果的文件路径，方式与工具箱中相同。若未指定输出对象，结果将保存到临时文件（若为可选输出则跳过）。文件扩展名决定文件格式。如果您输入了算法不支持的文件扩展名，将使用该输出类型的默认格式，并将对应扩展名附加到您提供的路径后。

与通过工具箱执行算法不同，若通过 Python 控制台使用 `run()` 方法执行算法，其输出不会自动添加到地图画布。该方法返回一个字典，键为一个或多个输出名称（即算法描述中所示名称），值为对应输出的文件路径：

```
>>> 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
```

随后，您可以像加载普通图层一样将输出加载到工程中：

```
>>> buffered_layer = myresult['OUTPUT']
>>> QgsProject.instance().addMapLayer(buffered_layer)
```

若要立即将处理结果加载到工程中，可使用 `runAndLoadResults()` 方法替代 `run()`。

```
>>> processing.runAndLoadResults("native:buffer", {parameters:values})
```

若要从控制台打开算法对话框，可使用 `createAlgorithmDialog` 方法。唯一必需的参数是算法名称，但您也可以定义参数字典，以便自动填充对话框：

```
>>> 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()
```

`execAlgorithmDialog` 方法会立即打开对话框：

```
>>> 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 创建脚本并从工具箱运行

您可以通过编写 Python 代码创建自己的算法。处理脚本需继承 `QgsProcessingAlgorithm`，因此需添加若干额外代码以实现必需的函数。在 Processing 工具箱顶部的 *Scripts* 下拉菜单中，您可以找到 *Create new script*（空白脚本）和*:guilabel:'Create New Script from Template'*（包含*:class:'QgsProcessingAlgorithm <qgis.core.QgsProcessingAlgorithm>* 必需函数代码的模板）。将打开 Processing 脚本编辑器，您应在其中编写代码。将脚本保存到 `scripts` 文件夹（打开保存文件对话框时的默认文件夹）并使用 `.py` 扩展名，即可生成对应的算法。

算法名称（即工具箱中显示的名称）需在代码中定义。

让我们看以下代码，它定义了一个处理算法：首先对用户指定的矢量图层进行平滑处理，然后以用户定义的缓冲距离执行缓冲操作。

```
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"

```

(continues on next page)

(continued from previous page)

```
    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}
```

完成必要导入后，需指定以下 `QgsProcessingAlgorithm` 函数：

- `name()`：算法的 ID（小写）。
- `displayName()`：算法的人类可读名称。
- `createInstance()`：创建算法类的新实例。
- `initAlgorithm()`：配置参数定义和输出定义。

  在此描述算法的参数和输出。本例中包括：输入的要素源、结果的要素接收器（sink），以及缓冲距离的数值。

- `processAlgorithm()`：执行实际工作。

  此处我们首先运行 `smoothgeometry` 算法对几何进行平滑处理，然后在其平滑后的输出上运行 `buffer` 算法。要在算法内部运行其他算法，必须将 `is_child_algorithm` 参数设为 `True`。您可以看到输入和输出参数是如何作为参数传递给 `smoothgeometry` 和 `buffer` 算法的。

输入和输出有多种参数类型可用。完整列表请参见处理算法的输入与输出。

构造函数的第一个参数是参数名称，第二个是参数描述（用于用户界面）。其余参数则根据具体参数类型而定。

可使用 `QgsProcessingAlgorithm` 的 `parameterAs` 函数将输入转换为 QGIS 类。例如，要将缓冲距离数值作为双精度浮点数获取：

```
self.parameterAsDouble(parameters, self.INPUT_BUFFERDIST, context)).
```

processAlgorithm 函数应返回一个字典，包含算法定义的每个输出的值。这使得其他算法（包括同一模型中的其他算法）能够访问这些输出。

良好的算法应定义并返回尽可能多的有意义输出。非要素类输出（如数字和字符串）在将算法作为更大模型的一部分运行时非常有用，因为这些值可用作模型中后续算法的输入参数。建议添加诸如处理的要素数量、遇到的无效要素数量、输出的要素数量等数值型输出。返回的输出越多，您的算法就越有用！

反馈机制

应使用传递给 processAlgorithm() 的 feedback 对象向用户提供反馈或交互。您可以使用该对象的 setProgress() 方法更新进度条（0 到 100），以告知用户算法的执行进度。当算法耗时较长时，这非常有用。

feedback 对象提供 isCanceled() 方法，应持续监控以支持用户取消算法执行。其 pushInfo() 方法可用于向用户发送信息，而 reportError() 则便于向用户报告非致命错误。

算法应避免使用其他用户反馈形式（如 print 语句或记录到 QgsMessageLog），而应始终使用 feedback 对象。这既支持详细日志记录，又具备线程安全性（鉴于算法通常在后台线程中运行，这一点尤为重要）。

错误处理

如果您的算法遇到阻止其执行的错误（如无效输入值或其他无法或不应恢复的情况），则应抛出 QgsProcessingException。例如::

```
if feature['value'] < 20:
  raise QgsProcessingException('Invalid input value {}, must be >= 20'.
↪format(feature['value']))
```

应尽量避免对非致命错误（例如要素几何为空）抛出 QgsProcessingException，而应通过 feedback.reportError() 报告错误并跳过该要素。这有助于使您的算法“模型友好”——当遇到非致命错误时，不会导致整个算法执行中断。

为脚本编写文档

与模型类似，您可以为脚本创建额外文档，以说明其功能及使用方法。

QgsProcessingAlgorithm 提供了 helpString()、shortHelpString() 和 helpUrl() 函数用于此目的。请指定或重写这些方法以向用户提供更多帮助。

shortDescription() 用于在工具箱中悬停于算法上时显示的工具提示。

23.7.3 执行前与执行后脚本钩子

脚本也可用作执行前和执行后钩子，分别在算法运行前和运行后执行。这可用于自动化每次执行算法时都应完成的任务。

语法与上述完全相同，但额外提供一个名为 alg 的全局变量，代表刚刚执行（或即将执行）的算法。

在处理选项对话框的 *General* 组中，您会找到两个条目：*Pre-execution script* 和 *Post-execution script*，可在其中分别输入要运行的脚本文件名。

23.8 Using processing from the command line

QGIS comes with a tool called `QGIS Processing Executor` which allows you to run Processing algorithms and models (built-in or provided by plugins) directly from the command line without starting QGIS Desktop itself.

From a command line tool, run `qgis_process` and you should get:

```
QGIS Processing Executor - 3.35.0-Master 'Master' (3.35.0-Master)
Usage: C:\OSGeo4W\apps\qgis-dev\bin\qgis_process.exe [--help] [--version] [--json]␣
↪[--verbose] [--no-python] [--skip-loading-plugins] [command] [algorithm id, path␣
↪to model file, or path to Python script] [parameters]

Options:

  --help or -h            Output the help
  --version or -v         Output all versions related to QGIS Process
  --json                  Output results as JSON objects
  --verbose               Output verbose logs
  --no-python             Disable Python support (results in faster startup)
  --skip-loading-plugins  Avoid loading enabled plugins (results in faster startup)

Available commands:

  plugins             list available and active plugins
  plugins enable      enables an installed plugin. The plugin name must be␣
↪specified, e.g. "plugins enable cartography_tools"
  plugins disable     disables an installed plugin. The plugin name must be␣
↪specified, e.g. "plugins disable cartography_tools"
  list                list all available processing algorithms
  help                show help for an algorithm. The algorithm id or a path to a␣
↪model file must be specified.
  run                 runs an algorithm. The algorithm id or a path to a model file␣
↪and parameter values must be specified.
                      Parameter values are specified after -- with PARAMETER=VALUE␣
↪syntax.
                      Ordered list values for a parameter can be created by␣
↪specifying the parameter multiple times,
                      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.
```

Note

Only installed plugins that advertise `hasProcessingProvider=yes` in their `metadata.txt` file are recognized and can be activated or loaded by qgis_process tool.

Hint

Before calling qgis_process on a system without window manager (e.g. a headless server), you should set:
`export QT_QPA_PLATFORM=offscreen`

The command `list` can be used to get a list of all available providers and algorithms.

```
qgis_process list
```

The command `help` can be used to get further information about commands or algorithms.

```
qgis_process help qgis:regularpoints
```

The command `run` can be used to run an algorithm or model. Specify the name of the algorithm or a path to a model as first parameter.

```
qgis_process run native:buffer -- INPUT=source.shp DISTANCE=2 OUTPUT=buffered.shp
```

Where a parameter accepts a list of values, set the same variable multiple times.

```
qgis_process run native:mergevectorlayers -- LAYERS=input1.shp LAYERS=input2.shp␣
↪OUTPUT=merged.shp
```

While running an algorithm a text-based feedback bar is shown, and the operation can be cancelled via `CTRL+C`.

The `run` command also supports further parameters.

- `--json` will format stdout output in a JSON structured way.
- `--ellipsoid` will set the ellipsoid to the specified one.
- `--distance_units` will use the specified distance units.
- `--area_units` will use the specified area units.
- `--project_path` will load the specified project for running the algorithm.

Complex input parameters, i.e. parameter types which are themselves specified as a dictionary type object for algorithms, are supported by qgis_process. To indicate that parameters will be specified via stdin, the qgis_process command must follow the format (with a trailing – in place of the usual arguments list).

```
qgis_process run algorithmId –
```

The JSON object must contain an ”inputs” key, which is a map of the input parameter values. E.g.

```
echo "{'inputs': {'INPUT': 'my_shape.shp', 'DISTANCE': 5}}" | qgis_process run␣
↪native:buffer –
```

Additionally, extra settings like the distance units, area units, ellipsoid and project path can be included in this JSON object:

```
{
 'ellipsoid': 'EPSG:7019',
 'distance_units': 'feet',
 'area_units': 'ha',
 'project_path': 'C:/temp/my_project.qgs'
 'inputs': {'DISTANCE': 5, 'SEGMENTS': 8 ... }
}
```

Specifying input parameters via stdin implies automatically the `JSON` output format for results.

23.9 Writing new Processing algorithms as Python scripts

There are two options for writing Processing algorithms using Python.

- *Extending* `QgsProcessingAlgorithm`
- *Using the @alg decorator*

Within QGIS, you can use *Create new script* in the *Scripts* menu at the top of the *Processing Toolbox* to open the *Processing Script Editor* where you can write your code. To simplify the task, you can start with a script template by using *Create new script from template* from the same menu. This opens a template that extends `QgsProcessingAlgorithm`.

If you save the script in the `scripts` folder (the default location) with a `.py` extension, the algorithm will become available in the *Processing Toolbox*.

23.9.1 Extending QgsProcessingAlgorithm

The following code

1. takes a vector layer as input
2. counts the number of features
3. does a buffer operation
4. creates a raster layer from the result of the buffer operation
5. returns the buffer layer, raster layer and number of features

```
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'
```

(continues on next page)

(continued from previous page)

```
    def displayName(self):
        """
        Returns the translated algorithm name.
        """
        return self.tr('Buffer and export to raster (extend)')

    def group(self):
        """
        Returns the name of the group this algorithm belongs to.
        """
        return self.tr('Example scripts')

    def groupId(self):
        """
        Returns the unique ID of the group this algorithm belongs
        to.
        """
        return 'examplescripts'

    def shortHelpString(self):
        """
        Returns a localised short help string for the algorithm.
        """
        return self.tr('Example algorithm short description')

    def initAlgorithm(self, config=None):
        """
        Here we define the inputs and outputs of the algorithm.
        """
        # 'INPUT' is the recommended name for the main input
        # parameter.
        self.addParameter(
            QgsProcessingParameterFeatureSource(
                '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'
            )
```

(continues on next page)

(continued from previous page)

```
        )
        self.addParameter(
            QgsProcessingParameterDistance(
                'CELLSIZE',
                self.tr('CELLSIZE'),
                defaultValue = 10.0,
                parentParameterName='INPUT'
            )
        )
        self.addOutput(
            QgsProcessingOutputNumber(
                'NUMBEROFFEATURES',
                self.tr('Number of features processed')
            )
        )

    def processAlgorithm(self, parameters, context, feedback):
        """
        Here is where the processing itself takes place.
        """
        # First, we get the count of features from the INPUT layer.
        # This layer is defined as a QgsProcessingParameterFeatureSource
        # parameter, so it is retrieved by calling
        # self.parameterAsSource.
        input_featuresource = self.parameterAsSource(parameters,
                                                     'INPUT',
                                                     context)
        numfeatures = input_featuresource.featureCount()

        # Retrieve the buffer distance and raster cell size numeric
        # values. Since these are numeric values, they are retrieved
        # using self.parameterAsDouble.
        bufferdist = self.parameterAsDouble(parameters, 'BUFFERDIST',
                                            context)
        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,
```

(continues on next page)

(continued from previous page)

```
            feedback=feedback)

        # Check for cancelation
        if feedback.isCanceled():
            return {}

        # Run the separate rasterization algorithm using the buffer result
        # as an input.
        rasterized_result = processing.run(
            'qgis:rasterize',
            {
                # Here we pass the 'OUTPUT' value from the buffer's result
                # dictionary off to the rasterize child algorithm.
                'LAYER': buffer_result['OUTPUT'],
                'EXTENT': buffer_result['OUTPUT'],
                'MAP_UNITS_PER_PIXEL': rastercellsize,
                # Use the original parameter value.
                'OUTPUT': parameters['OUTPUT']
            },
            is_child_algorithm=True,
            context=context,
            feedback=feedback)

        if feedback.isCanceled():
            return {}

        # Return the results
        return {'OUTPUT': rasterized_result['OUTPUT'],
                'BUFFER_OUTPUT': buffer_result['OUTPUT'],
                'NUMBEROFFEATURES': numfeatures}
```

Processing algorithm standard functions:

- **createInstance (mandatory)**
  Must return a new copy of your algorithm. If you change the name of the class, make sure you also update the value returned here to match!
- **name (mandatory)**
  Returns the unique algorithm name, used for identifying the algorithm.
- **displayName (mandatory)**
  Returns the translated algorithm name.
- **group**
  Returns the name of the group this algorithm belongs to.
- **groupId**
  Returns the unique ID of the group this algorithm belongs to.
- **shortHelpString**
  Returns a localised short help string for the algorithm.
- **initAlgorithm (mandatory)**
  Here we define the inputs and outputs of the algorithm.

  `INPUT` and `OUTPUT` are recommended names for the main input and main output parameters, respectively.

  If a parameter depends on another parameter, `parentParameterName` is used to specify this relationship (could be the field / band of a layer or the distance units of a layer).
- **processAlgorithm (mandatory)**
  This is where the processing takes place.

Parameters are retrieved using special purpose functions, for instance `parameterAsSource` and `parameterAsDouble`.

`processing.run` can be used to run other processing algorithms from a processing algorithm. The first parameter is the name of the algorithm, the second is a dictionary of the parameters to the algorithm. `is_child_algorithm` is normally set to `True` when running an algorithm from within another algorithm. `context` and `feedback` inform the algorithm about the environment to run in and the channel for communicating with the user (catching cancel request, reporting progress, providing textual feedback). When using the (parent) algorithm's parameters as parameters to "child" algorithms, the original parameter values should be used (e.g. `parameters['OUTPUT']`).

It is good practice to check the feedback object for cancelation as much as is sensibly possible! Doing so allows for responsive cancelation, instead of forcing users to wait for unwanted processing to occur.

The algorithm should return values for all the output parameters it has defined as a dictionary. In this case, that's the buffer and rasterized output layers, and the count of features processed. The dictionary keys must match the original parameter/output names.

23.9.2 The @alg decorator

Using the @alg decorator, you can create your own algorithms by writing the Python code and adding a few extra lines to supply additional information needed to make it a proper Processing algorithm. This simplifies the creation of algorithms and the specification of inputs and outputs.

One important limitation with the decorator approach is that algorithms created in this way will always be added to a user's Processing Scripts provider -- it is not possible to add these algorithms to a custom provider, e.g. for use in plugins.

The following code uses the @alg decorator to

1. use a vector layer as input
2. count the number of features
3. do a buffer operation
4. create a raster layer from the result of the buffer operation
5. returns the buffer layer, raster layer and number of features

```
from qgis import processing
from qgis.processing import alg
from qgis.core import QgsProject

@alg(name='bufferrasteralg', label='Buffer and export to raster (alg)',
     group='examplescripts', group_label='Example scripts')
# 'INPUT' is the recommended name for the main input parameter
@alg.input(type=alg.SOURCE, name='INPUT', label='Input vector layer')
# 'OUTPUT' is the recommended name for the main output parameter
@alg.input(type=alg.RASTER_LAYER_DEST, name='OUTPUT',
           label='Raster output')
@alg.input(type=alg.VECTOR_LAYER_DEST, name='BUFFER_OUTPUT',
           label='Buffer output')
@alg.input(type=alg.DISTANCE, name='BUFFERDIST', label='BUFFER DISTANCE',
           default=1.0)
@alg.input(type=alg.DISTANCE, name='CELLSIZE', label='RASTER CELL SIZE',
           default=10.0)
@alg.output(type=alg.NUMBER, name='NUMBEROFFEATURES',
            label='Number of features processed')

def bufferrasteralg(instance, parameters, context, feedback, inputs):
    """
    Description of the algorithm.
    (If there is no comment here, you will get an error)
```

(continues on next page)

(continued from previous page)

```
    """
    input_featuresource = instance.parameterAsSource(parameters,
                                                     'INPUT', context)
    numfeatures = input_featuresource.featureCount()
    bufferdist = instance.parameterAsDouble(parameters, 'BUFFERDIST',
                                            context)
    rastercellsize = instance.parameterAsDouble(parameters, 'CELLSIZE',
                                                context)
    if feedback.isCanceled():
        return {}
    buffer_result = processing.run('native:buffer',
                                   {'INPUT': parameters['INPUT'],
                                    'OUTPUT': parameters['BUFFER_OUTPUT'],
                                    'DISTANCE': bufferdist,
                                    'SEGMENTS': 10,
                                    'DISSOLVE': True,
                                    'END_CAP_STYLE': 0,
                                    'JOIN_STYLE': 0,
                                    'MITER_LIMIT': 10
                                    },
                                   is_child_algorithm=True,
                                   context=context,
                                   feedback=feedback)
    if feedback.isCanceled():
        return {}
    rasterized_result = processing.run('qgis:rasterize',
                                   {'LAYER': buffer_result['OUTPUT'],
                                    'EXTENT': buffer_result['OUTPUT'],
                                    'MAP_UNITS_PER_PIXEL': rastercellsize,
                                    'OUTPUT': parameters['OUTPUT']
                                   },
                                   is_child_algorithm=True, context=context,
                                   feedback=feedback)
    if feedback.isCanceled():
        return {}
    return {'OUTPUT': rasterized_result['OUTPUT'],
            'BUFFER_OUTPUT': buffer_result['OUTPUT'],
            'NUMBEROFFEATURES': numfeatures}
```

As you can see, it involves two algorithms ('native:buffer' and 'qgis:rasterize'). The last one ('qgis:rasterize') creates a raster layer from the buffer layer that was generated by the first one ('native:buffer').

The part of the code where this processing takes place is not difficult to understand if you have read the previous chapter. The first lines, however, need some additional explanation. They provide the information that is needed to turn your code into an algorithm that can be run from any of the GUI components, like the toolbox or the model designer.

These lines are all calls to the `@alg` decorator functions that help simplify the coding of the algorithm.

- The @alg decorator is used to define the name and location of the algorithm in the Toolbox.
- The @alg.input decorator is used to define the inputs of the algorithm.
- The @alg.output decorator is used to define the outputs of the algorithm.

For existing parameters and their correspondance, read 处理算法的输入与输出.

23.9.3 Handing algorithm output

When you declare an output representing a layer (raster or vector), the algorithm will try to add it to QGIS once it is finished.

- Raster layer output: QgsProcessingParameterRasterDestination / alg.RASTER_LAYER_DEST.
- Vector layer output: QgsProcessingParameterVectorDestination / alg.VECTOR_LAYER_DEST.

So even if the `processing.run()` method does not add the layers it creates to the user's current project, the two output layers (buffer and raster buffer) will be loaded, since they are saved to the destinations entered by the user (or to temporary destinations if the user does not specify destinations).

If a layer is created as output of an algorithm, it should be declared as such. Otherwise, you will not be able to properly use the algorithm in the modeler, since what is declared will not match what the algorithm really creates.

You can return strings, numbers and more by specifying them in the result dictionary (as demonstrated for "NUMBEROFFEATURES"), but they should always be explicitly defined as outputs from your algorithm. We encourage algorithms to output as many useful values as possible, since these can be valuable for use in later algorithms when your algorithm is used as part of a model.

23.9.4 Communicating with the user

If your algorithm takes a long time to process, it is a good idea to inform the user about the progress. You can use `feedback` (`QgsProcessingFeedback`) for this.

The progress text and progressbar can be updated using two methods: `setProgressText(text)` and `setProgress(percent)`.

You can provide more information by using `pushCommandInfo(text)`, `pushDebugInfo(text)`, `pushInfo(text)` and `reportError(text)`.

If your script has a problem, the correct way of handling it is to raise a `QgsProcessingException`. You can pass a message as an argument to the constructor of the exception. Processing will take care of handling it and communicating with the user, depending on where the algorithm is being executed from (toolbox, modeler, Python console, ...)

23.9.5 为脚本编写文档

You can document your scripts by overloading the `helpString()` and `helpUrl()` methods of `QgsProcessingAlgorithm`.

23.9.6 Flags

You can override the `flags()` method of `QgsProcessingAlgorithm` to tell QGIS more about your algorithm. You can for instance tell QGIS that the script shall be hidden from the modeler, that it can be canceled, that it is not thread safe, and more.

Tip

By default, Processing runs algorithms in a separate thread in order to keep QGIS responsive while the processing task runs. If your algorithm is regularly crashing, you are probably using API calls which are not safe to do in a background thread. Try returning the QgsProcessingAlgorithm.FlagNoThreading flag from your algorithm's flags() method to force Processing to run your algorithm in the main thread instead.

23.9.7 Best practices for writing script algorithms

Here's a quick summary of ideas to consider when creating your script algorithms and, especially, if you want to share them with other QGIS users. Following these simple rules will ensure consistency across the different Processing elements such as the toolbox, the modeler or the batch processing interface.

- Do not load resulting layers. Let Processing handle your results and load your layers if needed.
- Always declare the outputs your algorithm creates.
- Do not show message boxes or use any GUI element from the script. If you want to communicate with the user, use the methods of the feedback object (`QgsProcessingFeedback`) or throw a `QgsProcessingException`.

There are already many processing algorithms available in QGIS. You can find code on the QGIS repo.

23.10 处理算法的输入与输出

Here is the list of input and output types that are supported in Processing with their corresponding alg decorator constants, their name in the modeler designer if exposed.

23.10.1 输入类型

输入名称	类	Alg 常量	描述
注记层	`QgsProcessingParame`	`alg.ANNOTATION_LAYE`	An annotation layer
Area	`QgsProcessingParame`		A numeric parameter representing an area measure
Authentication Configuration	`QgsProcessingParame`	`alg.AUTH_CFG`	Allows users to select from available authentication configurations or create new authentication configurations
Raster band	`QgsProcessingParame`	`alg.BAND`	A band of a raster layer
布尔	`QgsProcessingParame`	`alg.BOOL`	A boolean value
Connection Name	`QgsProcessingParame`		A selection from available registered connections for a particular data provider
Color	`QgsProcessingParame`	`alg.COLOR`	A color
Coordinate Operation	`QgsProcessingParame`	`alg.COORDINATE_OPER`	A coordinate operation (for CRS transformations)
CRS	`QgsProcessingParame`	`alg.CRS`	A Coordinate Reference System
Database Schema	`QgsProcessingParame`	`alg.DATABASE_SCHEMA`	A database schema
Database Table	`QgsProcessingParame`	`alg.DATABASE_TABLE`	A database table
日期	`ProcessingDateTimeP Date`	`alg.DATE`	A pure date
Datetime	`ProcessingDateTimeP DateTime`	`alg.DATETIME`	A datetime
Distance	`QgsProcessingParame`	`alg.DISTANCE`	A double numeric parameter for distance values
DXF Layers	`QgsProcessingParame`		A list of vector layers to export as DXF file

continues on next page

Table 23.1 – continued from previous page

输入名称	类	Alg 常量	描述
Enum	QgsProcessingParame	alg.ENUM	An enumeration, allowing for selection from a set of predefined values
Expression	QgsProcessingParame	alg.EXPRESSION	An expression
Extent	QgsProcessingParame	alg.EXTENT	A spatial extent defined by xmin, xmax, ymin, ymax
	QgsProcessingParame	alg.FIELD	A field in the attribute table of a vector layer
Field Aggregates	QgsProcessingParame		A parameter for "aggregate" configurations, which consist of a definition of desired output fields, types, and aggregate used to populate them
Fields Mapper	QgsProcessingParame		A parameter for "field mapping" configurations, which consist of a definition of desired output fields, types, and expressions used to populate them
File/Folder	QgsProcessingParame	alg.FILE	A filename of an existing file
	QgsProcessingParame	alg.FILE_DEST	A filename for a newly created output file
	QgsProcessingParame	alg.FOLDER_DEST	A folder (destination folder)
Geometry	QgsProcessingParame	alg.GEOMETRY	A geometry
	ProcessingNumberPar Integer	alg.INT	An integer
打印布局	QgsProcessingParame	alg.LAYOUT	A layout
Print Layout Item	QgsProcessingParame	alg.LAYOUT_ITEM	A layout item
Map Layer	QgsProcessingParame	alg.MAPLAYER	A map layer
Map Theme	QgsProcessingParame	alg.MAP_THEME	A project map theme
Matrix	QgsProcessingParame	alg.MATRIX	A matrix
Mesh Dataset Groups	QgsProcessingParame		Dataset groups of mesh layer
Mesh Dataset Time	QgsProcessingParame		Dataset index from a mesh layer by time setting
网孔图层	QgsProcessingParame	alg.MESH_LAYER	A mesh layer
Multiple Input	QgsProcessingParame	alg.MULTILAYER	A set of layers
Number	ProcessingNumberPar Double	alg.NUMBER	A numerical value
点	QgsProcessingParame	alg.POINT	A point
Point Cloud Attribute	QgsProcessingParame		An attribute from a point cloud layer
	QgsProcessingParame	alg.POINTCLOUD_LAYE	A point cloud layer destination parameter, for specifying the destination path for a point cloud layer created by the algorithm
点云图层	QgsProcessingParame	alg.POINTCLOUD_LAYE	A point cloud layer
	QgsProcessingParame	alg.PROVIDER_CONNEC	An available connection for a database provider
范围	QgsProcessingParame	alg.RANGE	A number range

continues on next page

Table 23.1 – continued from previous page

输入名称	类	Alg 常量	描述
栅格图层	QgsProcessingParame	alg.RASTER_LAYER	A raster layer
	QgsProcessingParame	alg.RASTER_LAYER_DE	A raster layer destination parameter, for specifying the destination path for a raster layer created by the algorithm
Scale	QgsProcessingParame	alg.SCALE	A map scale
	QgsProcessingParame	alg.SINK	A feature sink
字符串	QgsProcessingParame	alg.STRING	A text string
TIN Creation Layers	QgsProcessingParame		Selection of multiple layers to create a TIN with vertices and/or break lines
时间	ProcessingDateTimeP Time	alg.TIME	A pure time
Vector Features	QgsProcessingParame	alg.SOURCE	A feature source
矢量图层	QgsProcessingParame	alg.VECTOR_LAYER	A vector layer
	QgsProcessingParame	alg.VECTOR_LAYER_DE	A vector layer destination parameter, for specifying the destination path for a vector layer created by the algorithm
Vector Tile Writer Layers	QgsProcessingParame		A vector tile layer destination parameter, for specifying the destination path for a vector tile layer created by the algorithm
Volume	QgsProcessingParame		A numeric parameter representing a volume measure

23.10.2 Output types

类	Alg 常量	描述
QgsProcessingOutputBoolean	alg.BOOL	A boolean value
QgsProcessingOutputNumber	alg.DISTANCE	A double numeric parameter for distance values
QgsProcessingOutputFile	alg.FILE	A filename of an existing file
QgsProcessingOutputFolder	alg.FOLDER	A folder
QgsProcessingOutputHtml	alg.HTML	HTML
QgsProcessingOutputNumber	alg.INT	A integer
QgsProcessingOutputLayerDefinition	alg.LAYERDEF	A layer definition
QgsProcessingOutputMapLayer	alg.MAPLAYER	A map layer
QgsProcessingOutputMultipleLayers	alg.MULTILAYER	A set of layers
QgsProcessingOutputNumber	alg.NUMBER	A numerical value
QgsProcessingOutputPointCloudLayer	alg.POINTCLOUD_LAYER	A point cloud layer
QgsProcessingOutputRasterLayer	alg.RASTER_LAYER	A raster layer
QgsProcessingOutputString	alg.STRING	A text string
QgsProcessingOutputVectorLayer	alg.VECTOR_LAYER	A vector layer
QgsProcessingOutputVectorTileLayer		A vector tile layer

23.11 Configuring external applications

The Processing framework can be extended using additional applications. Algorithms that rely on external applications are managed by their own algorithm providers. Additional providers can be found as separate plugins, and installed using the QGIS Plugin Manager.

This section will show you how to configure the Processing framework to include these additional applications, and it will explain some particular features of the algorithms based on them. Once you have correctly configured the system, you will be able to execute external algorithms from any component like the toolbox or the model designer, just like you do with any other algorithm.

By default, algorithms that rely on an external application not shipped with QGIS are not enabled. You can enable them in the Processing settings dialog if they are installed on your system.

23.11.1 A note for Windows users

If you are not an advanced user and you are running QGIS on Windows, you might not be interested in reading the rest of this chapter. Make sure you install QGIS in your system using the standalone installer. That will automatically install GRASS in your system and configure it so it can be run from QGIS. All the algorithms from this provider will be ready to be run without needing any further configuration. If installing with the OSGeo4W application, make sure that you also select GRASS for installation.

23.11.2 A note on file formats

When using external software, opening a file in QGIS does not mean that it can be opened and processed in that other software. In most cases, other software can read what you have opened in QGIS, but in some cases, that might not be true. When using databases or uncommon file formats, whether for raster or vector layers, problems might arise. If that happens, try to use well-known file formats that you are sure are understood by both programs, and check the console output (in the log panel) to find out what is going wrong.

You might for instance get trouble and not be able to complete your work if you call an external algorithm with a GRASS raster layers as input. For this reason, such layers will not appear as available to algorithms.

You should, however, not have problems with vector layers, since QGIS automatically converts from the original file format to one accepted by the external application before passing the layer to it. This adds extra processing time, which might be significant for large layers, so do not be surprised if it takes more time to process a layer from a DB connection than a layer from a Shapefile format dataset of similar size.

Providers not using external applications can process any layer that you can open in QGIS, since they open it for analysis through QGIS.

All raster and vector output formats produced by QGIS can be used as input layers. Some providers do not support certain formats, but all can export to common formats that can later be transformed by QGIS automatically. As for input layers, if a conversion is needed, that might increase the processing time.

23.11.3 A note on vector layer selections

External applications may also be made aware of the selections that exist in vector layers within QGIS. However, that requires rewriting all input vector layers, just as if they were originally in a format not supported by the external application. Only when no selection exists, or the *Use only selected features* option is not enabled in the processing general configuration, can a layer be directly passed to an external application.

In other cases, exporting only selected features is needed, which causes longer execution times.

23.11.4 Using third-party Providers

R scripts and libraries

To enable R in Processing you need to install the **Processing R Provider** plugin and configure R for QGIS.

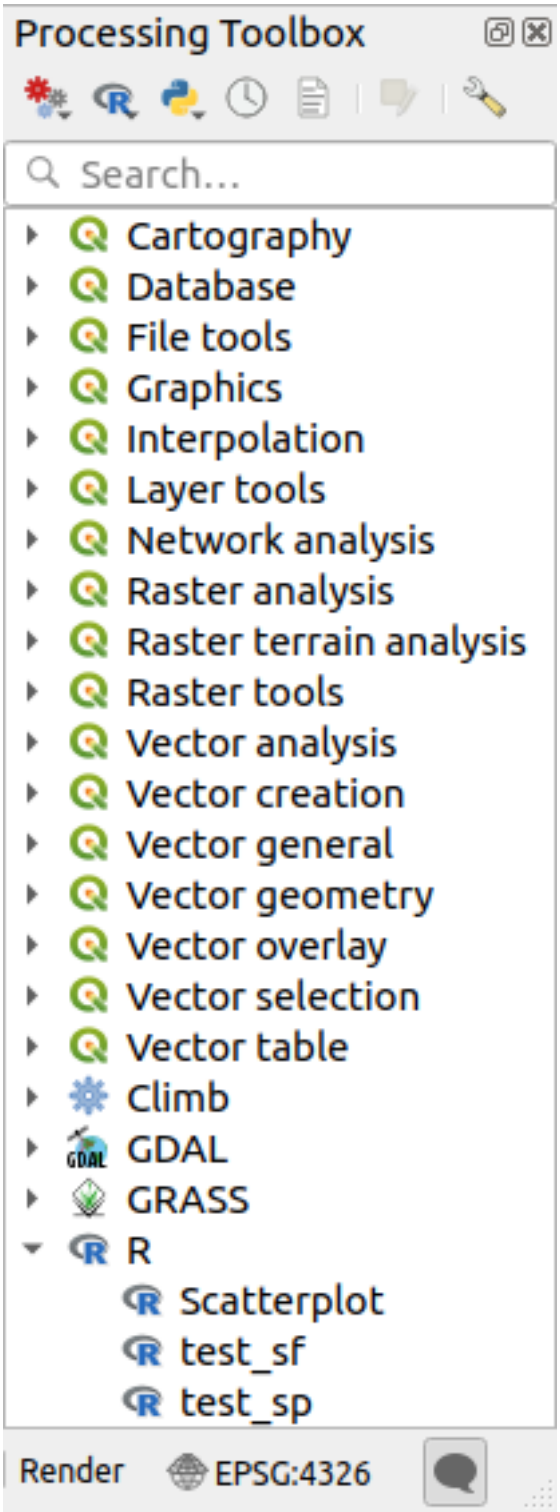
Configuration is done in *Provider* ► *R* in the *Processing* tab of *Settings* ► *Options*.

Depending on your operating system, you may have to use *R folder* to specify where your R binaries are located.

> **Note**
>
> On **Windows** the R executable file is normally in a folder (`R-<version>`) under `C:\Program Files\R\`. Specify the folder and **NOT** the binary!
>
> On **Linux** you just have to make sure that the R folder is in the PATH environment variable. If `R` in a terminal window starts R, then you are ready to go.

After installing the **Processing R Provider** plugin, you will find some example scripts in the *Processing Toolbox*:

- *Scatterplot* runs an R function that produces a scatter plot from two numerical fields of the provided vector layer.
- *test_sf* does some operations that depend on the `sf` package and can be used to check if the R package `sf` is installed. If the package is not installed, R will try to install it (and all the packages it depends on) for you, using the *Package repository* specified in *Provider* ► *R* in the Processing options. The default is https://cran.r-project.org/. Installing may take some time...
- *test_sp* can be used to check if the R package `sp` is installed. If the package is not installed, R will try to install it for you.



If you have R configured correctly for QGIS, you should be able to run these scripts.

Adding R scripts from the QGIS collection

R integration in QGIS is different from that of some other third party providers like SAGA in that there is not a predefined set of algorithms you can run (except for some example script that come with the *Processing R Provider* plugin).

A set of example R scripts is available in the QGIS Repository. Perform the following steps to load and enable them using the *QGIS Resource Sharing* plugin.

1. Add the *QGIS Resource Sharing* plugin (you may have to enable *Show also experimental plugins* in the Plugin Manager *Settings*)
2. Open it (*Plugins* ► *Resource Sharing* ► *Resource Sharing*)
3. Choose the *Settings* tab
4. Click *Reload repositories*
5. Choose the *All* tab
6. Select *QGIS R script collection* in the list and click on the *Install* button
7. The collection should now be listed in the *Installed* tab
8. Close the plugin
9. Open the *Processing Toolbox*, and if everything is OK, the example scripts will be present under R, in various groups (only some of the groups are expanded in the screenshot below).

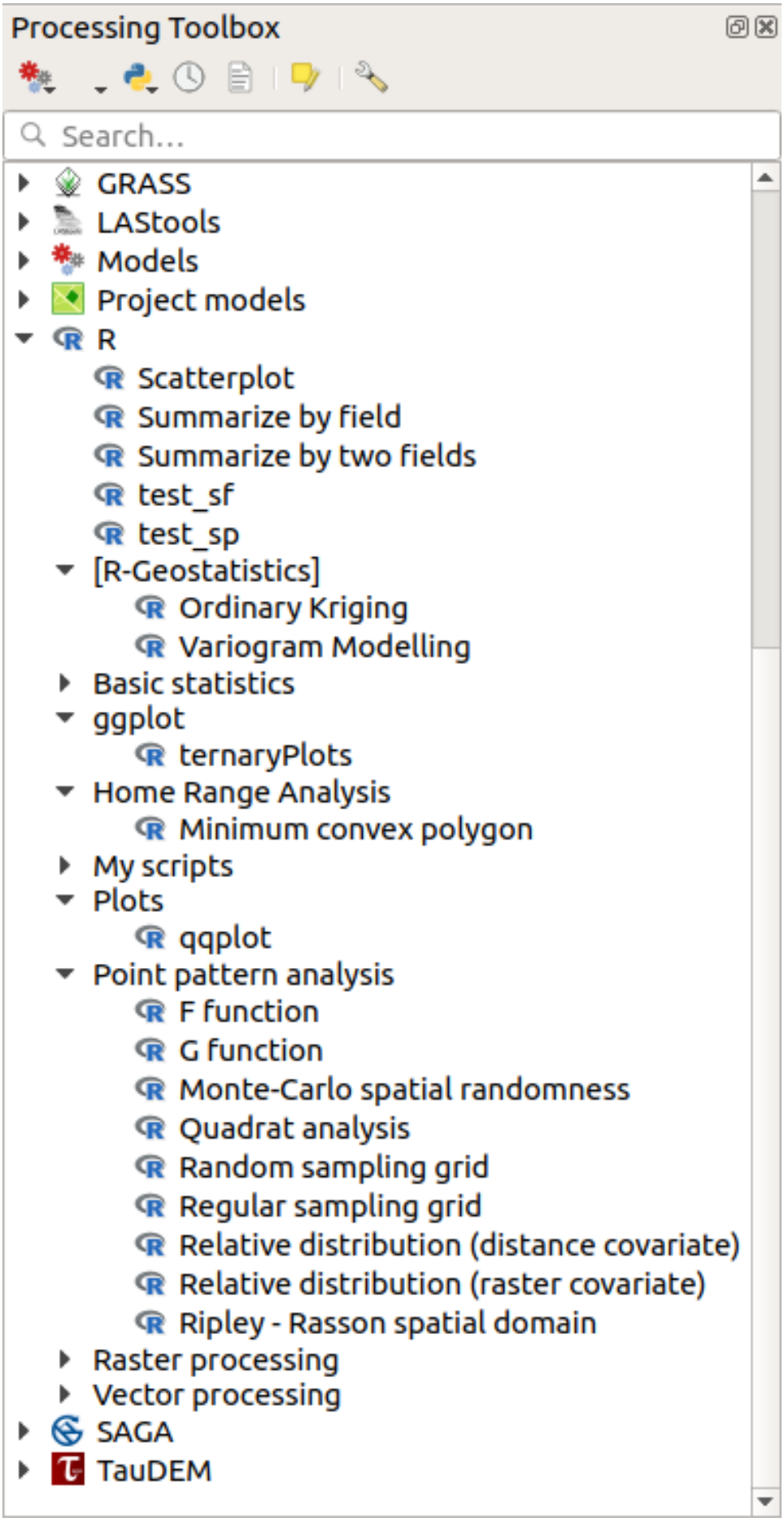
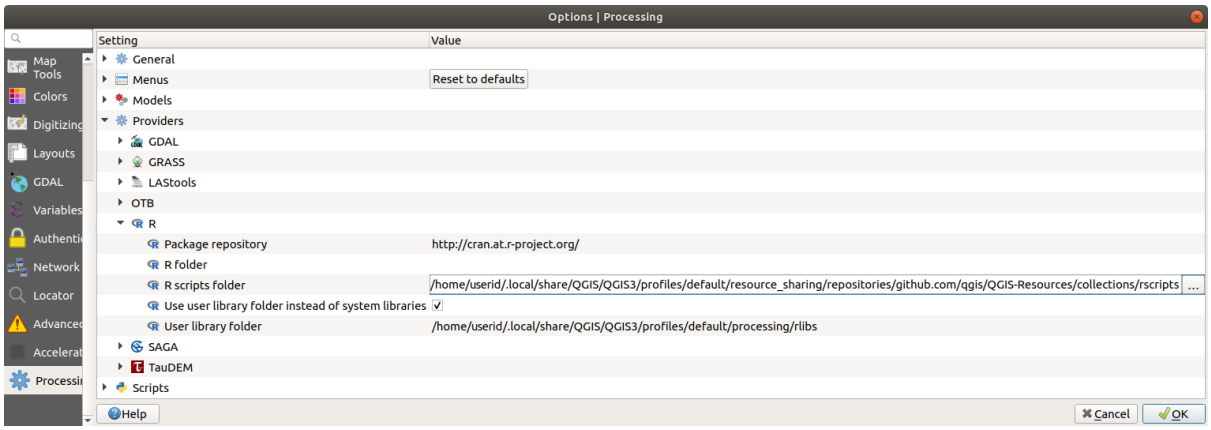


Fig. 23.35: The *Processing Toolbox* with some R scripts shown

The scripts at the top are the example scripts from the *Processing R Provider* plugin.

10. If, for some reason, the scripts are not available in the *Processing Toolbox*, you can try to:
    1. Open the Processing settings (*Settings* ► *Options* ► *Processing* tab)
    2. Go to *Providers* ► *R* ► *R scripts folder*
       - On Ubuntu, set the path to (or, better, include in the path):

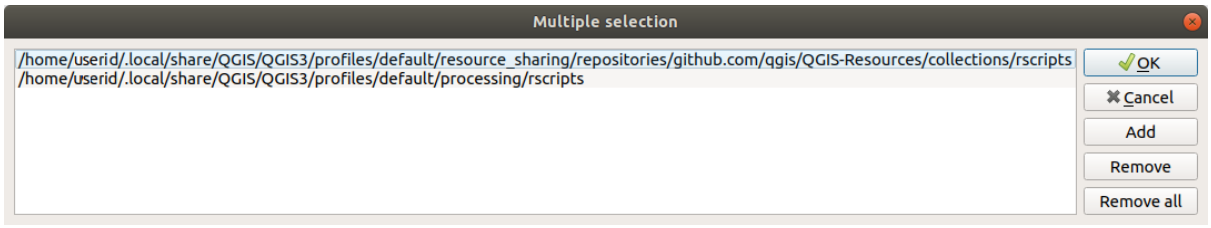
         /home/<user>/.local/share/QGIS/QGIS3/profiles/default/resource_sharing/repositories/github.com/qgis/QGIS-Resources/collections/rscripts



       - On Windows, set the path to (or, better, include in the path):

         C:\Users\<user>\AppData\Roaming\QGIS\QGIS3\profiles\default\resource_sharing\repositories\github.com\qgis\QGIS-Resources\collections\rscripts

       To edit, double-click. You can then choose to just paste / type the path, or you can navigate to the directory by using the ... button and press the *Add* button in the dialog that opens. It is possible to provide several directories here. They will be separated by a semicolon (";").



If you would like to get all the R scrips from the QGIS 2 on-line collection, you can select *QGIS R script collection (from QGIS 2)* instead of *QGIS R script collection*. You will probably find that scripts that depend on vector data input or output will not work.

Creating R scripts

You can write scripts and call R commands, as you would do from R. This section shows you the syntax for using R commands in QGIS, and how to use QGIS objects (layers, tables) in them.

To add an algorithm that calls an R function (or a more complex R script that you have developed and you would like to have available from QGIS), you have to create a script file that performs the R commands.

R script files have the extension `.rsx`, and creating them is pretty easy if you just have a basic knowledge of R syntax and R scripting. They should be stored in the R scripts folder. You can specify the folder (*R scripts folder*) in the *R* settings group in Processing settings dialog).

Let' s have a look at a very simple script file, which calls the R method `spsample` to create a random grid within the boundary of the polygons in a given polygon layer. This method belongs to the `maptools` package. Since almost all the algorithms that you might like to incorporate into QGIS will use or generate spatial data, knowledge of spatial packages like `maptools` and `sp`/`sf`, is very useful.

```
##Random points within layer extent=name
##Point pattern analysis=group
##Vector_layer=vector
##Number_of_points=number 10
##Output=output vector
library(sp)
spatpoly = as(Vector_layer, "Spatial")
pts=spsample(spatpoly,Number_of_points,type="random")
spdf=SpatialPointsDataFrame(pts, as.data.frame(pts))
Output=st_as_sf(spdf)
```

The first lines, which start with a double Python comment sign (`##`), define the display name and group of the script, and tell QGIS about its inputs and outputs.

Note

To find out more about how to write your own R scripts, have a look at the R Intro section in the training manual and consult the *QGIS R Syntax* section.

When you declare an input parameter, QGIS uses that information for two things: creating the user interface to ask the user for the value of that parameter, and creating a corresponding R variable that can be used as R function input.

In the above example, we have declared an input of type `vector`, named `Vector_layer`. When executing the algorithm, QGIS will open the layer selected by the user and store it in a variable named `Vector_layer`. So, the name of a parameter is the name of the variable that you use in R for accessing the value of that parameter (you should therefore avoid using reserved R words as parameter names).

Spatial parameters such as vector and raster layers are read using the `st_read()` (or `readOGR`) and `brick()` (or `readGDAL`) commands (you do not have to worry about adding those commands to your description file -- QGIS will do it), and they are stored as `sf` (or `Spatial*DataFrame`) objects.

Table fields are stored as strings containing the name of the selected field.

Vector files can be read using the `readOGR()` command instead of `st_read()` by specifying `##load_vector_using_rgdal`. This will produce a `Spatial*DataFrame` object instead of an `sf` object.

Raster files can be read using the `readGDAL()` command instead of `brick()` by specifying `##load_raster_using_rgdal`.

If you are an advanced user and do not want QGIS to create the object for the layer, you can use `##pass_filenames` to indicate that you prefer a string with the filename. In this case, it is up to you to open the file before performing any operation on the data it contains.

With the above information, it is possible to understand the first lines of the R script (the first line not starting with a Python comment character).

```
library(sp)
spatpoly = as(Vector_layer, "Spatial")
pts=spsample(polyg,numpoints,type="random")
```

The `spsample` function is provided by the *sp* library, so the first thing we do is to load that library. The variable `Vector_layer` contains an `sf` object. Since we are going to use a function (`spsample`) from the *sp* library, we must convert the `sf` object to a `SpatialPolygonsDataFrame` object using the `as` function.

Then we call the `spsample` function with this object and the `numpoints` input parameter (which specifies the number of points to generate).

Since we have declared a vector output named `Output`, we have to create a variable named `Output` containing an `sf` object.

We do this in two steps. First we create a `SpatialPolygonsDataFrame` object from the result of the function, using the *SpatialPointsDataFrame* function, and then we convert that object to an `sf` object using the `st_as_sf` function (of the *sf* library).

You can use whatever names you like for your intermediate variables. Just make sure that the variable storing your final result has the defined name (in this case `Output`), and that it contains a suitable value (an `sf` object for vector layer output).

In this case, the result obtained from the `spsample` method had to be converted explicitly into an `sf` object via a `SpatialPointsDataFrame` object, since it is itself an object of class `ppp`, which can not be returned to QGIS.

If your algorithm generates raster layers, the way they are saved will depend on whether or not you have used the `##dontuserasterpackage` option. If you have used it, layers are saved using the `writeGDAL()` method. If not, the `writeRaster()` method from the `raster` package will be used.

If you have used the `##pass_filenames` option, outputs are generated using the `raster` package (with `writeRaster()`).

If your algorithm does not generate a layer, but a text result in the console instead, you have to indicate that you want the console to be shown once the execution is finished. To do so, just start the command lines that produce the results you want to print with the > ('greater than') sign. Only output from lines prefixed with > are shown. For instance, here is the description file of an algorithm that performs a normality test on a given field (column) of the attributes of a vector layer:

```
##layer=vector
##field=field layer
##nortest=group
library(nortest)
>lillie.test(layer[[field]])
```

The output of the last line is printed, but the output of the first is not (and neither are the outputs from other command lines added automatically by QGIS).

If your algorithm creates any kind of graphics (using the `plot()` method), add the following line (`output_plots_to_html` used to be `showplots`):

```
##output_plots_to_html
```

This will cause QGIS to redirect all R graphical outputs to a temporary file, which will be opened once R execution has finished.

Both graphics and console results will be available through the processing results manager.

For more information, please check the R scripts in the official QGIS collection (you download and install them using the *QGIS Resource Sharing* plugin, as explained in *Adding R scripts from the QGIS collection*). Most of them are rather simple and will greatly help you understand how to create your own scripts.

Note

The `sf`, `rgdal` and `raster` libraries are loaded by default, so you do not have to add the corresponding `library()` commands. However, other libraries that you might need have to be explicitly loaded by typing: `library(ggplot2)` (to load the `ggplot2` library). If the package is not already installed on your machine, Processing will try to download and install it. In this way the package will also become available in R Standalone. **Be aware** that if the package has to be downloaded, the script may take a long time to run the first time.

R libraries installed when running sf_test

The R script `sp_test` tries to load the R packages `sp` and `raster`.

The R script *sf_test* tries to load `sf` and `raster`. If these two packages are not installed, R may try to load and install them (and all the libraries that they depend on).

The following R libraries end up in `~/.local/share/QGIS/QGIS3/profiles/default/processing/rscripts` after `sf_test` has been run from the Processing Toolbox on Ubuntu with version 2.0 of the *Processing R Provider* plugin and a fresh install of *R* 3.4.4 (*apt* package `r-base-core` only):

```
abind, askpass, assertthat, backports, base64enc, BH, bit, bit64, blob, brew,␣
↪callr, classInt, cli, colorspace, covr, crayon, crosstalk, curl, DBI, deldir,
desc, dichromat, digest, dplyr, e1071, ellipsis, evaluate, fansi, farver, fastmap,␣
↪gdtools, ggplot2, glue, goftest, gridExtra, gtable, highr, hms,
htmltools, htmlwidgets, httpuv, httr, jsonlite, knitr, labeling, later, lazyeval,␣
↪leafem, leaflet, leaflet.providers, leafpop, leafsync, lifecycle, lwgeom,
magrittr, maps, mapview, markdown, memoise, microbenchmark, mime, munsell, odbc,␣
↪openssl, pillar, pkgbuild, pkgconfig, pkgload, plogr, plyr, png, polyclip,
praise, prettyunits, processx, promises, ps, purrr, R6, raster, RColorBrewer, Rcpp,
↪ reshape2, rex, rgeos, rlang, rmarkdown, RPostgres, RPostgreSQL,
rprojroot, RSQLite, rstudioapi, satellite, scales, sf, shiny, sourcetools, sp,␣
↪spatstat, spatstat.data, spatstat.utils, stars, stringi, stringr, svglite,
sys, systemfonts, tensor, testthat, tibble, tidyselect, tinytex, units, utf8, uuid,
↪ vctrs, viridis, viridisLite, webshot, withr, xfun, XML, xtable
```

GRASS

Configuring GRASS is very easy. First, the path to the GRASS folder has to be defined, but only if you are running Windows.

By default, the Processing framework tries to configure its GRASS connector to use the GRASS distribution that ships along with QGIS. This should work without problems for most systems, but if you experience problems, you might have to configure the GRASS connector manually. Also, if you want to use a different GRASS installation, you can change the setting to point to the folder where the other version is installed. GRASS 7 is needed for algorithms to work correctly.

If you are running Linux, you just have to make sure that GRASS is correctly installed, and that it can be run without problem from a terminal window.

GRASS algorithms use a region for calculations. This region can be defined manually or automatically, taking the minimum extent that covers all the input layers used to execute the algorithm each time.

LAStools

To use LAStools in QGIS, you need to download and install LAStools on your computer and install the LAStools plugin (available from the official repository) in QGIS.

On Linux platforms, you will need Wine to be able to run some of the tools.

LAStools is activated and configured in the Processing options (*Settings* ► *Options*, *Processing* tab, *Providers* ► *LAStools*), where you can specify the location of LAStools (*LAStools folder*) and Wine (*Wine folder*). On Ubuntu, the default Wine folder is `/usr/bin`.

OTB Applications

OTB applications are fully supported within the QGIS Processing framework.

OTB (Orfeo ToolBox) is an image processing library for remote sensing data. It also provides applications that provide image processing functionalities. The list of applications and their documentation are available in OTB CookBook

> **Note**
>
> Note that OTB is not distributed with QGIS and needs to be installed separately. Binary packages for OTB can be found on the download page.

To configure QGIS processing to find the OTB library:

1. Open the processing settings: *Settings* ► *Options* ► *Processing*
2. You can see `OTB` under *Providers* menu:
    1. Expand the *OTB* entry
    2. Set the *OTB folder*. This is the location of your OTB installation.
    3. Set the *OTB application folder*. This is the location of your OTB applications (`<PATH_TO_OTB_INSTALLATION>/lib/otb/applications`)
    4. Click *OK* to save the settings and close the dialog.

If settings are correct, OTB algorithms will be available in the *Processing Toolbox*.

Documentation of OTB settings available in QGIS Processing

- **OTB folder**: This is the directory where OTB is available.
- **OTB application folder**: This is the location(s) of OTB applications.

  Multiple paths are allowed.
- **Logger level** (optional): Level of logger to use by OTB applications.

  The level of logging controls the amount of detail printed during algorithm execution. Possible values for logger level are `INFO`, `WARNING`, `CRITICAL`, `DEBUG`. This value is `INFO` by default. This is an advanced user configuration.
- **Maximum RAM to use** (optional): by default, OTB applications use all available system RAM.

  You can, however, instruct OTB to use a specific amount of RAM (in MB) using this option. A value of 256 is ignored by the OTB processing provider. This is an advanced user configuration.
- **Geoid file** (optional): Path to the geoid file.

  This option sets the value of the elev.dem.geoid and elev.geoid parameters in OTB applications. Setting this value globally enables users to share it across multiple processing algorithms. Empty by default.
- **SRTM tiles folder** (optional): Directory where SRTM tiles are available.

  SRTM data can be stored locally to avoid downloading of files during processing. This option sets the value of elev.dem.path and elev.dem parameters in OTB applications. Setting this value globally enables users to share it across multiple processing algorithms. Empty by default.

Compatibility and Troubleshoot

Starting from OTB 6.6.1, new releases of OTB are made compatible with at least the latest QGIS version available at that time.

If you have issues with OTB applications in QGIS Processing, please open an issue on the OTB bug tracker, using the `qgis` label.

Additional information about OTB and QGIS can be found in OTB Cookbook.

CHAPTER 24

数据处理提供程序和算法

这里记录了处理算法及其参数 (如用户界面所示)。

24.1 QGIS algorithm provider

QGIS algorithm provider implements various analysis and geoprocessing operations using mostly only QGIS API. So almost all algorithms from this provider will work ”out of the box” without any additional configuration.

This provider incorporates some algorithms from plugins and also adds its own algorithms.

24.1.1 3D Tiles

Convert B3DM to GLTF

Converts files from the legacy `.B3DM` format to `.GLTF` or `.GLB`.

参数

标注	名称	类型	描述
Input B3DM	`INPUT`	[3D Tile]	Input file to convert.
Output file	`OUTPUT`	[3D Tile] 默认: [保存到临时文件]	Specify the output 3D tile file. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Output file	`OUTPUT`	[3D Tile]	Output layer in `.GLTF` format.

Python 代码

Algorithm ID: `native:b3dmtogltf`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Convert GLTF to vector features

Converts `.GLTF` or `.GLB` file contents to standard vector layer formats.

参数

标注	名称	类型	描述
Input GLTF	`INPUT`	[3D Tile]	Input file to convert.
Output polygons 可选	`OUTPUT_POLYGON`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output vector layer to store polygon features. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。
Output lines 可选	`OUTPUT_LINES`	[vector: line] Default: `[Create temporary layer]`	Specify the output vector layer to store line features. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Output polygons	`OUTPUT_POLYGON`	[vector: polygon]	A vector layer storing the polygon features in input layer.
Output lines	`OUTPUT_LINES`	[vector: line]	A vector layer storing the line features in input layer.

Python 代码

Algorithm ID: `native:gltftovector`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.2 Cartography

Align points to features

Calculates the rotation required to align point features with their nearest feature from another reference layer. A new field is added to the output layer which is filled with the angle (in degrees, clockwise) to the nearest reference feature.

Optionally, the output layer's symbology can be set to automatically use the calculated rotation field to rotate marker symbols. If desired, a maximum distance to use when aligning points can be set, to avoid aligning isolated points to distant features.

> **Hint**
>
> This algorithm is designed for use cases like aligning building point symbols to follow the nearest road direction.

Allows *features in-place modification* of point features

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: point]	Point features to calculate the rotation for
Reference layer	`REFERENCE_LAYE`	[vector: geometry]	Layer to find the closest feature from for rotation calculation
Maximum distance to consider 可选	`MAX_DISTANCE`	[numeric: double] Default: Not set	If no reference feature is found within this distance, no rotation is assigned to the point feature.
Angle field name	`FIELD_NAME`	[string] Default: 'rotation'	Field in which to store the rotation value.
Automatically apply symbology	`APPLY_SYMBOLOG`	[boolean] Default: True	Rotates the symbol marker of the features using the angle field value

continues on next page

Table 24.5 – continued from previous page

标注	名称	类型	描述
Aligned layer	OUTPUT	[vector: point] 默认：[保存到临时文件]	Specify the rotated output vector layer. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Aligned layer	OUTPUT	[vector: point]	The point layer appended with a rotation field. If loaded to QGIS, it is applied by default the input layer symbology, with a data-defined rotation of its marker symbol.

Python 代码

Algorithm ID: `native:angletonearest`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Combine style databases

Combines multiple QGIS style databases into a single style database. If items of the same type with the same name exist in different source databases these will be renamed to have unique names in the output combined database.

See also

Create style database from project

参数

标注	名称	类型	描述
Input databases	INPUT	[file] [list]	Files containing QGIS style items

continues on next page

Table 24.7 – continued from previous page

标注	名称	类型	描述
Objects to combine	OBJECTS	[enumeration] [list]	Types of style items in the input databases you would like to put in the new database. These can be: • 0 --- *Symbols* • 1 --- *Color ramps* • 2 --- *Text formats* • 3 --- *Label settings*
Output style database	OUTPUT	[file] 默认: [保存到临时文件]	Output .XML file combining the selected style items. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Color ramp count	COLORRAMPS	[numeric: integer]	Number of color ramps
Label settings count	LABELSETTINGS	[numeric: integer]	Number of label settings
Output style database	OUTPUT	[file]	Output .XML file combining the selected style items
Symbol count	SYMBOLS	[numeric: integer]	Number of symbols
Text format count	TEXTFORMATS	[numeric: integer]	Number of text formats

Python 代码

Algorithm ID: native:combinestyles

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Create categorized renderer from styles

Sets a vector layer's renderer to a categorized renderer using matching symbols from a style database. If no style file is specified, symbols from the user's current *symbol library* are used instead.

A specified expression or field is used to create categories for the renderer. Each category is individually matched to the symbols which exist within the specified QGIS XML style database. Whenever a matching symbol name is found, the category's symbol will be set to this matched symbol.

If desired, outputs can also be tables containing lists of the categories which could not be matched to symbols, and symbols which were not matched to categories.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	Vector layer to apply a categorized style to
Categorize using expression	`FIELD`	[expression]	Field or expression to categorize the features
Style database (leave blank to use saved symbols)	`STYLE`	[file]	File (`.XML`) containing the symbols to apply to the input layer categories. The file can be obtained from the Style Manager *Share symbols* tool. If no file is specified, QGIS local symbols library is used.
Use case-sensitive match to symbol names	`CASE_SENSITIVE`	[boolean] Default: False	If True (checked), applies a case sensitive comparison between the categories and symbols names
Ignore non-alphanumeric characters while matching	`TOLERANT`	[boolean] Default: False	If True (checked), non-alphanumeric characters in the categories and symbols names will be ignored, allowing greater tolerance during the match.
Non-matching categories 可选	`NON_MATCHING_C`	[vector: table] Default: `[Skip output]`	Output table for categories which do not match any symbol in the database. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。
Non-matching symbol names 可选	`NON_MATCHING_S`	[vector: table] Default: `[Skip output]`	Output table for symbols from the provided style database which do not match any category. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Non-matching categories	`NON_MATCHING_C`	[vector: table]	Lists categories which could not be matched to any symbol in the provided style database
Non-matching symbol names	`NON_MATCHING_S`	[vector: table]	Lists symbols from the provided style database which could not match any category
Categorized layer	`OUTPUT`	[same as input]	The input vector layer with the categorized style applied. No new layer is output.

Python 代码

Algorithm ID: `native:categorizeusingstyle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Create style database from project

Extracts all style objects (symbols, color ramps, text formats and label settings) from a QGIS project.

The extracted symbols are saved to a QGIS style database (`XML` format), which can be managed and imported via the *Style Manager* dialog.

See also

Combine style databases

参数

标注	名称	类型	描述
Input project (leave blank to use current) 可选	`INPUT`	[file]	A QGIS project file to extract the style items from
Objects to extract	`OBJECTS`	[enumeration] [list]	Types of style items in the input project you would like to put in the new database. These can be: • 0 --- *Symbols* • 1 --- *Color ramps* • 2 --- *Text formats* • 3 --- *Label settings*
Output style database	`OUTPUT`	[file] 默认：[保存到临时文件]	Specify the output `.XML` file for the selected style items. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Color ramp count	`COLORRAMPS`	[numeric: integer]	Number of color ramps
Label settings count	`LABELSETTINGS`	[numeric: integer]	Number of label settings
Output style database	`OUTPUT`	[file]	Output `.XML` file for the selected style items

continues on next page

Table 24.12 – continued from previous page

标注	名称	类型	描述
Symbol count	SYMBOLS	[numeric: integer]	Number of symbols
Text format count	TEXTFORMATS	[numeric: integer]	Number of text formats

Python 代码

Algorithm ID: `native:stylefromproject`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Export atlas layout as image

Exports the atlas of a print layout as image files (e.g. PNG or JPEG images).

If a coverage layer is set, the selected layout's atlas settings exposed in this algorithm will be overwritten. In this case, an empty filter or sort by expression will turn those settings off.

参数

Basic parameters

标注	名称	类型	描述
Atlas layout	LAYOUT	[layout]	Layout to export
Coverage layer 可选	COVERAGE_LAYER	[vector: geometry]	Layer to use to generate the atlas
Filter expression	FILTER_EXPRESS	[expression]	Expression to use to filter out atlas features
Sort expression 可选	SORTBY_EXPRESS	[expression]	Expression to use to sort the atlas features
Reverse sort order	SORTBY_REVERSE	[boolean] Default: False	Determines if sorting should be inverted. Used when a sort expression is provided.
Output filename expression	FILENAME_EXPRE	[expression] Default: 'output_'||@atlas_featurer	Expression for use to generate filenames
Output folder	FOLDER	[folder]	Destination folder where the images will be generated

Advanced parameters

标注	名称	类型	描述
Map layers to assign to unlocked map item(s) 可选	LAYERS	[enumeration] [layer]	Layers to display in the map item(s) whose contents are not locked

continues on next page

Table 24.14 – continued from previous page

标注	名称	类型	描述
Image format	EXTENSION	[enumeration] Default: png	File format of the generated output(s). The list of available formats varies depending on OS and installed drivers.
DPI 可选	DPI	[numeric: double] Default: Not set	DPI of the output file(s). If not set, the value in the print layout settings will be used.
Generate world file	GEOREFERENCE	[boolean] Default: True	Determines if a world file should be generated
Export RDF metadata	INCLUDE_METADA	[boolean] Default: True	Determines if RDF metadata (title, author, ...) should be generated
Enable antialiasing	ANTIALIAS	[boolean] Default: True	Determines if antialiasing should be enabled

输出

标注	名称	类型	描述
Image file	OUTPUT	[file]	Image files generated by the atlas layout

Python 代码

Algorithm ID: `native:atlaslayouttoimage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Export atlas layout as PDF (multiple files)

Exports the atlas of a print layout to multiple PDF files.

If a coverage layer is set, the selected layout's atlas settings exposed in this algorithm will be overwritten. In this case, an empty filter or sort by expression will turn those settings off.

参数

Basic parameters

标注	名称	类型	描述
Atlas layout	LAYOUT	[layout]	Layout to export
Coverage layer 可选	COVERAGE_LAYER	[vector: geometry]	Layer to use to generate the atlas
Filter expression	FILTER_EXPRESS	[expression]	Expression to use to filter out atlas features
Sort expression 可选	SORTBY_EXPRESS	[expression]	Expression to use to sort the atlas features
Reverse sort order	SORTBY_REVERSE	[boolean] Default: False	Determines if sorting should be inverted. Used when a sort expression is provided.

continues on next page

Table 24.16 – continued from previous page

标注	名称	类型	描述
Output filename 可选	OUTPUT_FILENAM	[expression]	Name pattern of the PDF output files.
Output folder	OUTPUT_FOLDER	[folder]	Destination folder for the output PDF files.

Advanced parameters

标注	名称	类型	描述
Map layers to assign to unlocked map item(s) 可选	LAYERS	[enumeration] [layer]	Layers to display in the map item(s) whose contents are not locked
DPI 可选	DPI Default: Not set	[numeric: double]	DPI of the output file(s). If not set, the value in the print layout settings will be used.
Always export as vectors	FORCE_VECTOR	[boolean] Default: False	Determines if vectorial data should be left as vectors
Always export as raster	FORCE_RASTER	[boolean] Default: False	Forces all the items in the map to be rasterized. This parameter takes precedence over the FORCE_VECTOR parameter.
Append georeference information	GEOREFERENCE	[boolean] Default: True	Determines if a world file should be generated
Export RDF metadata	INCLUDE_METADA	[boolean] Default: True	Determines if RDF metadata (title, author, ...) should be generated
Disable tiled raster layer exports	DISABLE_TILED	[boolean] Default: False	Determines if raster should be tiled
Simplify geometries to reduce output file size	SIMPLIFY	[boolean] Default: True	Determines if geometries should be simplified to reduce output file size
Text export	TEXT_FORMAT	[enumeration] Default: 0	Determines if text should be exported as path or text objects. *Possible options* are: • 0 - Always export text as paths (recommended) • 1 - Always export texts as text objects • 2 - Prefer exporting texts as text objects
Image compression	IMAGE_COMPRESS	[enumeration] Default: 0	Determines compression level of the image and how suitable the file could be for printing outputs or post-production in external applications. Possible options are: • 0 - Lossy (JPEG) • 1 - Lossless

输出

标注	名称	类型	描述
PDF file	OUTPUT	[file]	PDF file corresponding to the exported atlas layout

Python 代码

Algorithm ID: native:atlaslayouttomultiplepdf

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Export atlas layout as PDF (single file)

Exports the atlas of a print layout as a single PDF file.

If a coverage layer is set, the selected layout's atlas settings exposed in this algorithm will be overwritten. In this case, an empty filter or sort by expression will turn those settings off.

参数

Basic parameters

标注	名称	类型	描述
Atlas layout	LAYOUT	[layout]	Layout to export
Coverage layer 可选	COVERAGE_LAYER	[vector: geometry]	Layer to use to generate the atlas
Filter expression	FILTER_EXPRESS	[expression]	Expression to use to filter out atlas features
Sort expression 可选	SORTBY_EXPRESS	[expression]	Expression to use to sort the atlas features
Reverse sort order	SORTBY_REVERSE	[boolean] Default: False	Determines if sorting should be inverted. Used when a sort expression is provided.
PDF file	OUTPUT	[file] Default: [Save to temporary file]	Name (including path) of the output file. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Map layers to assign to unlocked map item(s) 可选	LAYERS	[enumeration] [layer]	Layers to display in the map item(s) whose contents are not locked
DPI 可选	DPI Default: Not set	[numeric: double]	DPI of the output file(s). If not set, the value in the print layout settings will be used.
Always export as vectors	FORCE_VECTOR	[boolean] Default: False	Determines if vectorial data should be left as vectors
Always export as raster	FORCE_RASTER	[boolean] Default: False	Forces all the items in the map to be rasterized. This parameter takes precedence over the FORCE_VECTOR parameter.
Append georeference information	GEOREFERENCE	[boolean] Default: True	Determines if a world file should be generated
Export RDF metadata	INCLUDE_METADA	[boolean] Default: True	Determines if RDF metadata (title, author, ...) should be generated
Disable tiled raster layer exports	DISABLE_TILED	[boolean] Default: False	Determines if raster should be tiled
Simplify geometries to reduce output file size	SIMPLIFY	[boolean] Default: True	Determines if geometries should be simplified to reduce output file size
Text export	TEXT_FORMAT	[enumeration] Default: 0	Determines if text should be exported as path or text objects. *Possible options* are: • 0 - Always export text as paths (recommended) • 1 - Always export texts as text objects • 2 - Prefer exporting texts as text objects
Image compression	IMAGE_COMPRESS	[enumeration] Default: 0	Determines compression level of the image and how suitable the file could be for printing outputs or post-production in external applications. Possible options are: • 0 - Lossy (JPEG) • 1 - Lossless

输出

标注	名称	类型	描述
PDF file	OUTPUT	[file]	PDF file corresponding to the exported atlas layout

Python 代码

Algorithm ID: `native:atlaslayouttopdf`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Export print layout as image

Exports a print layout as an image file (e.g. PNG or JPEG images)

参数

Basic parameters

标注	名称	类型	描述
Print layout	`LAYOUT`	[layout]	Layout to export
Image file	`OUTPUT`	[file] Default: [Save to temporary file]	Name (including path) of the output file. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Map layers to assign to unlocked map item(s) 可选	`LAYERS`	[enumeration] [layer]	Layers to display in the map item(s) whose contents are not locked
DPI 可选	`DPI` Default: Not set	[numeric: double]	DPI of the output file(s). If not set, the value in the print layout settings will be used.
Generate world file	`GEOREFERENCE`	[boolean] Default: True	Determines if a world file should be generated
Export RDF metadata	`INCLUDE_METADA`	[boolean] Default: True	Determines if RDF metadata (title, author, ...) should be generated
Enable antialiasing	`ANTIALIAS`	[boolean] Default: True	Determines if antialiasing should be enabled

输出

标注	名称	类型	描述
Image file	OUTPUT	[file]	Image file corresponding to the exported print layout

Python 代码

Algorithm ID: native:printlayouttoimage

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Export print layout as PDF

Exports a print layout as a PDF file.

参数

Basic parameters

标注	名称	类型	描述
Print Layout	LAYOUT	[layout]	Layout to export
PDF file	OUTPUT	[file] Default: [Save to temporary file]	Name (including path) of the output file. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Map layers to assign to unlocked map item(s) 可选	LAYERS	[enumeration] [layer]	Layers to display in the map item(s) whose contents are not locked
DPI 可选	DPI Default: Not set	[numeric: double]	DPI of the output file(s). If not set, the value in the print layout settings will be used.
Always export as vectors	FORCE_VECTOR	[boolean] Default: False	Determines if vectorial data should be left as vectors
Always export as raster	FORCE_RASTER	[boolean] Default: False	Forces all the items in the map to be rasterized. This parameter takes precedence over the FORCE_VECTOR parameter.

continues on next page

Table 24.26 – continued from previous page

标注	名称	类型	描述
Append georeference information	GEOREFERENCE	[boolean] Default: True	Determines if a world file should be generated
Export RDF metadata	INCLUDE_METADA	[boolean] Default: True	Determines if RDF metadata (title, author, ...) should be generated
Disable tiled raster layer exports	DISABLE_TILED	[boolean] Default: False	Determines if raster should be tiled
Simplify geometries to reduce output file size	SIMPLIFY	[boolean] Default: True	Determines if geometries should be simplified to reduce output file size
Text export	TEXT_FORMAT	[enumeration] Default: 0	Determines if text should be exported as path or text objects. *Possible options* are: • 0 - Always export text as paths (recommended) • 1 - Always export texts as text objects • 2 - Prefer exporting texts as text objects
Image compression	IMAGE_COMPRESS	[enumeration] Default: 0	Determines compression level of the image and how suitable the file could be for printing outputs or post-production in external applications. Possible options are: • 0 - Lossy (JPEG) • 1 - Lossless
Export layers as separate PDF files	SEPARATE_LAYER	[boolean] Default: False	If True, then a separate PDF file will be created per layer per map item in the layout. Additionally, separate PDF files may be created for other complex layout items, resulting in a set of PDF files which contain logical atomic components of the layout.

输出

标注	名称	类型	描述
PDF file	OUTPUT	[file]	PDF file(s) corresponding to the exported print layout

Python 代码

Algorithm ID: `native:printlayouttopdf`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Extract labels

Extracts label information from a rendered map at a given extent and scale.

If a map theme is provided, the rendered map will match the visibility and symbology of that theme. If left blank, all visible layers from the project will be used. Extracted label information include: position (served as point geometries), the associated layer name and feature ID, label text, rotation (in degree, clockwise), multiline alignment, and font details.

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

参数

Basic parameters

标注	名称	类型	描述
Map extent	`EXTENT`	[范围]	Extent of the map to extract the labels from Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Map scale	`SCALE`	[scale]	Extracted labels will be rendered using their properties set at this scale.
Map theme 可选	`MAP_THEME`	[maptheme]	A map theme displaying the layers to extract the labels from. If unset, labels of the currently visible layers are extracted.
Include unplaced labels	`INCLUDE_UNPLAC`	[boolean] Default: True	Specify whether all overlapping labels should be extracted, including the conflicting (thus unplaced) ones.
Extracted labels	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the output vector layer for the extent(s). *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Map resolution (in DPI) 可选	`DPI` Default: 96.0	[numeric: double]	

输出

标注	名称	类型	描述
Extracted labels	`OUTPUT`	[vector: point]	Point vector layer representing the fetched labels. Each feature has attributes identifying its source (layer, feature ID) and the assigned labeling properties (text, font, size, rotation, ...). A default style with labeling and null symbol is also applied to the layer. **Warning** Because some of the generated fields have name with more than 10 characters, using the ESRI shapefile format (`.SHP`) to store the output may lead to unexpected rendering while loading the layer in QGIS.

Python 代码

Algorithm ID: `native:extractlabels`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Print layout map extent to layer

Creates a polygon layer containing the extent of a print layout map item (or items), with attributes specifying the map size (in layout units, i.e. the *reference map* units), scale and rotation.

If the map item parameter is specified, then only the matching map extent will be exported. If it is not specified, all map extents from the layout will be exported.

Optionally, a specific output CRS can be specified. If it is not specified, the original map item CRS will be used.

参数

Basic parameters

标注	名称	类型	描述
Print layout	LAYOUT	[enumeration]	A print layout in the current project
Map item 可选	MAP	[enumeration] Default: *All the map items*	The map item(s) whose information you want to extract. If none is provided then all the map items are processed.
Extent	OUTPUT	[vector: polygon] Default: [Create temporary layer]	Specify the output vector layer for the extent(s). *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Overrride CRS 可选	CRS	[crs] Default: *The layout CRS*	Select the CRS for the layer in which the information will be reported.

输出

标注	名称	类型	描述
Map height	HEIGHT	[numeric: double]	
Extent	OUTPUT	[vector: polygon]	Output polygon vector layer containing extents of all the input layout map item(s)
Map rotation	ROTATION	[numeric: double]	
Map scale	SCALE	[numeric: double]	
Map width	WIDTH	[numeric: double]	

Python 代码

Algorithm ID: native:printlayoutmapextenttolayer

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Set layer style

Applies a provided style to a layer. The style must be defined in a QML file.

No new output are created: the style is immediately assigned to the layer.

参数

标注	名称	类型	描述
Input Layer	INPUT	[layer]	Input layer you want to apply the style to
Style file	STYLE	[file]	Path to the .qml file of the style

输出

标注	名称	类型	描述
	OUTPUT	[same as input]	The input layer with the new style assigned. No new layer is created.

Python 代码

Algorithm ID: native:setlayerstyle

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Topological coloring

Assigns a color index to polygon features in such a way that no adjacent polygons share the same color index, whilst minimizing the number of colors required.

The algorithm allows choice of method to use when assigning colors.

A minimum number of colors can be specified if desired. The color index is saved to a new attribute named **color_id**.

The following example shows the algorithm with four different colors chosen; as you can see each color class has the same amount of features.

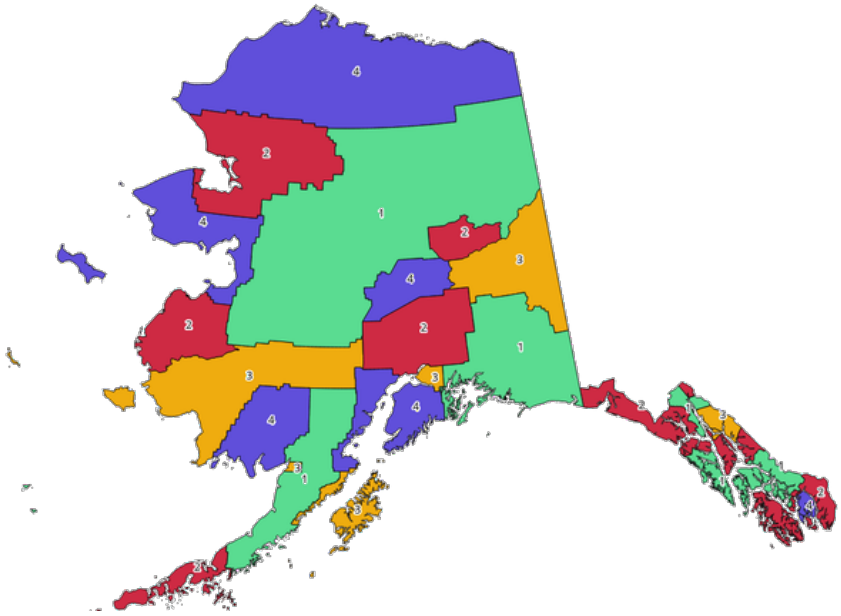


Fig. 24.1: Topological colors example

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: polygon]	The input polygon layer
Minimum number of colors	`MIN_COLORS`	[numeric: integer] Default: 4	The minimum number of colors to assign. Minimum 1, maximum 1000.
Minimum distance between features	`MIN_DISTANCE`	[numeric: double] Default: 0.0	Prevent nearby (but non-touching) features from being assigned equal colors. Minimum 0.0.
Balance color assignment	`BALANCE`	[enumeration] Default: 0	Options are: • 0 --- By feature count Attempts to assign colors so that the count of features assigned to each individual color index is balanced. • 1 --- By assigned area Assigns colors so that the total area of features assigned to each color is balanced. This mode can be useful to help avoid large features resulting in one of the colors appearing more dominant on a colored map. • 2 --- By distance between colors Assigns colors in order to maximize the distance between features of the same color. This mode helps to create a more uniform distribution of colors across a map.

continues on next page

Table 24.36 – continued from previous page

标注	名称	类型	描述
Colored	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Colored	`OUTPUT`	[vector: polygon]	Polygon vector layer with an added `color_id` column

Python 代码

Algorithm ID: `qgis:topologicalcoloring`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Transfer annotations from main layer

Transfers all *annotations* from the main annotation layer in a project to a new annotation layer. Items placement can then be adjusted within the layer stack.

参数

标注	名称	类型	描述
New layer name	`LAYER_NAME`	[string] Default: 'Annotations'	Name of the annotations layer to create

输出

标注	名称	类型	描述
New layer name	`OUTPUT`	[layer]	A layer with items from the main annotation layer

Python 代码

Algorithm ID: `native:transferannotationsfrommain`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.3 Check Geometry

Dangle-end lines

Detects dangle-end lines in line geometries and reports them as errors. A dangle-end line is a line feature that terminates at a vertex connected to only one segment, resulting in an endpoint without a proper connection.

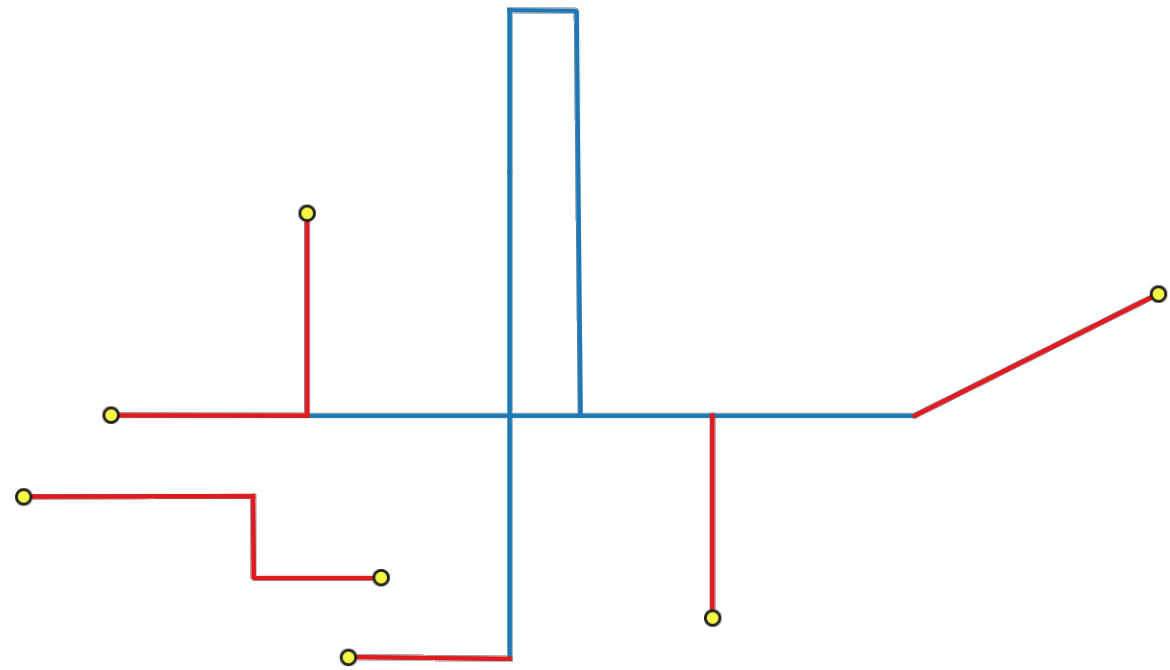


Fig. 24.2: Identifying dangle-end features (in red) and vertices (yellow).

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.

continues on next page

Table 24.40 – continued from previous page

Label	Name	Type	Description
Dangle-end errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Dangle-end features Optional	`OUTPUT`	[vector: line] Default: `[Skip output]`	Line layer containing the identified dangle-end line features. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Dangle-end errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx: the index of the feature's geometry part containing the dangle-end. • gc_ringidx: the index of the feature's geometry ring containing the dangle-end. • gc_vertidx: the vertex index of the dangle-end in the feature's geometry ring. • gc_errorx: the x coordinate of the dangle-end. • gc_errory: the y coordinate of the dangle-end. • gc_error • UNIQUE_ID field: the unique ID of the input feature with the dangle-end.
Dangle-end features	OUTPUT	[vector: line]	Output line layer containing the input dangle-end features. If no dangle-end features are found, the output layer will be empty. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometrydangle

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Degenerate polygons

Checks the polygons with less than 3 points, which are degenerate polygons. Degenerate polygons are errors.

Note

This algorithm detects invalid polygon geometries (degenerate polygons). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to Do not Filter (Better Performance).

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Degenerate polygons errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the centroid of the degenerate polygons. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Degenerate polygons features Optional	`OUTPUT`	[vector: polygon] Default: `[Skip output]`	Specification of the output layer containing the degenerate polygons. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Degenerate polygons errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx: the index of the geometry part of the degenerate polygon. • gc_ringidx: the index of the ring of the degenerate polygon. • gc_vertidx • gc_errorx: the x coordinate of the centroid of the degenerate polygon • gc_errory: the y coordinate of the centroid of the degenerate polygon • gc_error • UNIQUE_ID field: the unique ID of the input feature that is degenerate.
Degenerate polygons features	OUTPUT	[vector: polygon]	Output polygon layer with features containing the degenerate polygons. If no degenerate polygon features are found, the output layer will be empty. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometrydegeneratepolygon

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Duplicated geometries

Checks for duplicated geometries in a vector layer, and reports them as errors.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: geometry]	Layer with the geometries to check
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification
Duplicate geometries errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Duplicate geometries Optional	`OUTPUT`	[vector: same as input] Default: `[Skip output]`	Specification of the output layer containing the geometries that were found to be duplicated. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Duplicate geometries errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx • gc_ringidx • gc_vertidx • gc_errorx: the x coordinate of the centroid of the duplicate geometry. • gc_errory: the y coordinate of the centroid of the duplicate geometry. • gc_error: the indices of the duplicate geometry (all except the highest index). • UNIQUE_ID field: the unique ID of the input feature that is a duplicate.
Duplicate geometries	OUTPUT	[vector: same as input]	Output layer containing the geometries that were found to be duplicated. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometryduplicate

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Duplicated vertices

Checks for duplicated vertices in line or polygon geometries, and reports them as errors.

See also

Delete duplicated vertices, *Remove duplicate vertices*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to check
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification
Duplicated vertices errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Duplicated vertices features Optional	`OUTPUT`	[vector: same as input] Default: `[Skip output]`	Specification of the output layer containing the features with duplicated vertices. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Duplicated vertices errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx: the index of the feature's geometry part containing the duplicate vertex. • gc_ringidx: the index of the feature's geometry ring containing the duplicate vertex. • gc_vertidx: the index of the duplicate vertex in the feature's geometry ring. • gc_errorx: the x coordinate of the duplicate vertex. • gc_errory: the y coordinate of the duplicate vertex. • gc_error • UNIQUE_ID field: the unique ID of the input feature that has duplicate vertices.
Duplicated vertices features	OUTPUT	[vector: same as input]	Output layer containing the features with duplicated vertices. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometryduplicatenodes

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Features inside polygon

Checks the input geometries contained in the polygons from the polygon layers list. A polygon layer can be checked against itself. Any contained features are reported as errors.

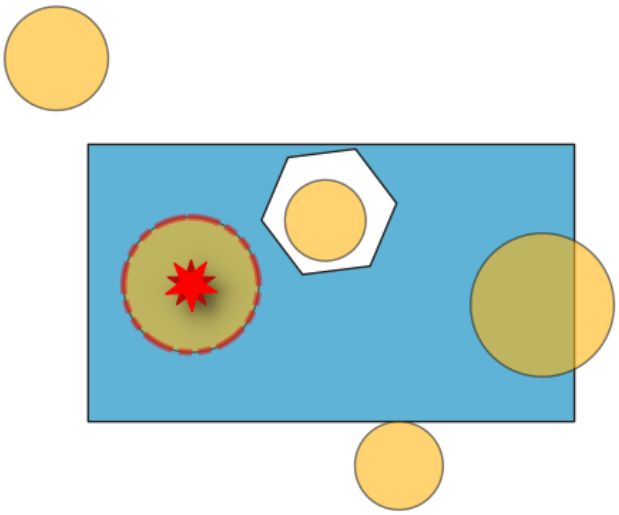


Fig. 24.3: Reporting errors on features inside a polygon.

See also

Delete features

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: geometry]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Polygon layers	`POLYGONS`	[vector: polygon][list]	List of polygon layers to check against.
Errors from contained features	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Point layer representing the contained features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

continues on next page

Table 24.52 – continued from previous page

Label	Name	Type	Description
Contained features Optional	OUTPUT	[vector: same as input] Default: [Skip output]	Specification of the output layer containing features within the checked polygons. *One of*: • Skip Output • Create Temporary Layer (TEMPORARY_OUTPUT) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	TOLERANCE	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Errors from contained features	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx • gc_ringidx • gc_vertidx • gc_errorx: the x coordinate of the centroid of the contained feature. • gc_errory: the y coordinate of the centroid of the contained feature. • gc_error: the layer name and feature ID of a polygon that contains the feature. • UNIQUE_ID field: the unique ID of the input feature that is contained within another.
Contained features	OUTPUT	[vector: same as input]	Output layer containing features within the checked polygons. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: `native:checkgeometrycontained`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Holes

Detects holes in polygon geometries and reports them as errors.

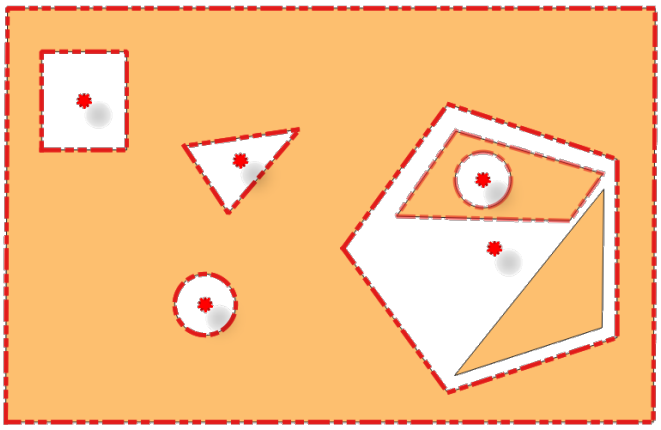


Fig. 24.4: Reporting errors for holes on a polygon feature.

See also

Fill holes

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification

continues on next page

Table 24.55 – continued from previous page

Label	Name	Type	Description
Holes errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Polygons with holes Optional	`OUTPUT`	[vector: polygon] Default: `[Skip output]`	Polygon layer with the features containing holes. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Holes errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx: the index of the feature's geometry part containing the hole. • gc_ringidx: the index of the feature's geometry ring containing the hole. • gc_vertidx • gc_errorx: the x coordinate of the centroid of the hole. • gc_errory: the y coordinate of the centroid of the hole. • gc_error • UNIQUE_ID field: the unique ID of the input feature that has a hole.
Polygon with holes	OUTPUT	[vector: polygon]	Output polygon layer with features containing holes. If no holes are found, the output layer will be empty. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometryhole

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Lines intersecting each other

Checks intersections between line geometries within a layer. Intersections between two different lines are errors.

See also

Line intersections, *Intersection*, *Self-intersections*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Intersection errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the intersections location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Intersecting feature Optional	`OUTPUT`	[vector: line] Default: `[Skip output]`	Specification of the output layer for features intersecting each other. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Intersection errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx: the index of the feature's geometry part where the intersection occurs. • gc_ringidx • gc_vertidx • gc_errorx: the x coordinate of the intersection point. • gc_errory: the y coordinate of the intersection point. • UNIQUE_ID field: the unique ID of an intersecting input feature. • gc_error: the layer name and index of the other intersecting feature.
Intersecting feature	OUTPUT	[vector: line]	Output layer containing, for each identified intersection, the feature (with same UNIQUE_ID value) it belongs to. If no intersections are found, the output layer will be empty. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometrylineintersection

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Lines intersecting other layer

Checks if the input line layer features intersect with the check layer features. An input feature that intersects with a check layer feature is an error.

See also

Line intersections, *Intersection*, *Self-intersections*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Check layer	`CHECK_LAYER`	[vector: line, polygon]	Layer with the geometries to check against.
Line intersecting other layer errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output point layer containing the locations of intersection errors. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Line intersecting other layer features Optional	`OUTPUT`	[line] Default: `[Skip output]`	Line layer with the input features that intersect features in the check layer. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Line intersecting other layer errors	ERRORS	[vector: point]	Output point layer representing the error location and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx: the index of the feature's geometry part where the intersection occurs. • gc_ringidx • gc_vertidx • gc_errorx: the x coordinate of the intersection point. • gc_errory: the y coordinate of the intersection point. • UNIQUE_ID field: the unique ID of the intersecting input feature. • gc_error: the index of the check layer where the intersection occurs.
Line intersecting other layer features	OUTPUT	[vector: line]	Output layer containing, for each identified intersection, the input feature it belongs to. If no intersections are found, the output layer will be empty. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometrylinelayerintersection

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Missing vertices along borders

Checks for missing vertices along polygon borders. To be topologically correct, a vertex at the junction of two polygons must be present on both polygons. Missing vertices are errors.

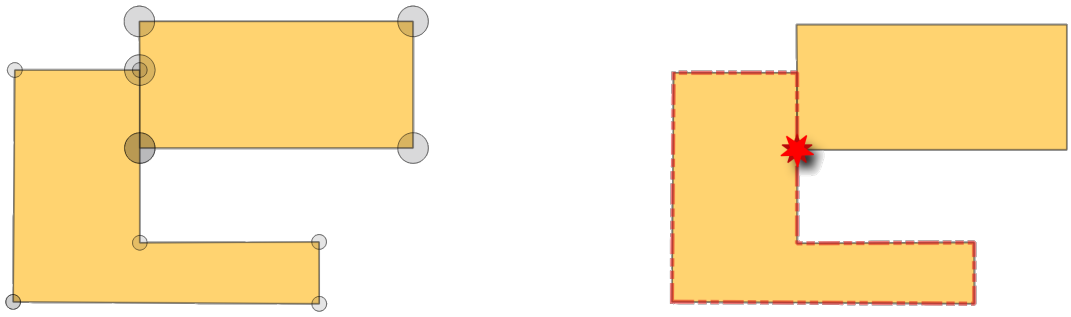


Fig. 24.5: Reporting errors for missing vertices on polygon features.

See also

Add missing vertices along borders

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Missing vertices errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Missing vertices features Optional	`OUTPUT`	[vector: polygon] Default: `[Skip output]`	Polygon layer with the features whose vertices are missing. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	TOLERANCE	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Missing vertices errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx • gc_ringidx • gc_vertidx • gc_errorx: the x coordinate of the missing vertex. • gc_errory: the y coordinate of the missing vertex. • gc_error • UNIQUE_ID field: the unique ID of the input feature that has a missing vertex.
Missing vertices features	OUTPUT	[vector: polygon]	Output polygon layer with features whose vertices are missing. There will be as many (duplicate) features as there are missing vertices in each geometry. If no vertices are missing, the output layer will be empty. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometrymissingvertex

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Overlaps

Calculates overlapping areas in polygon geometries, and reports areas smaller than a given minimum as errors.

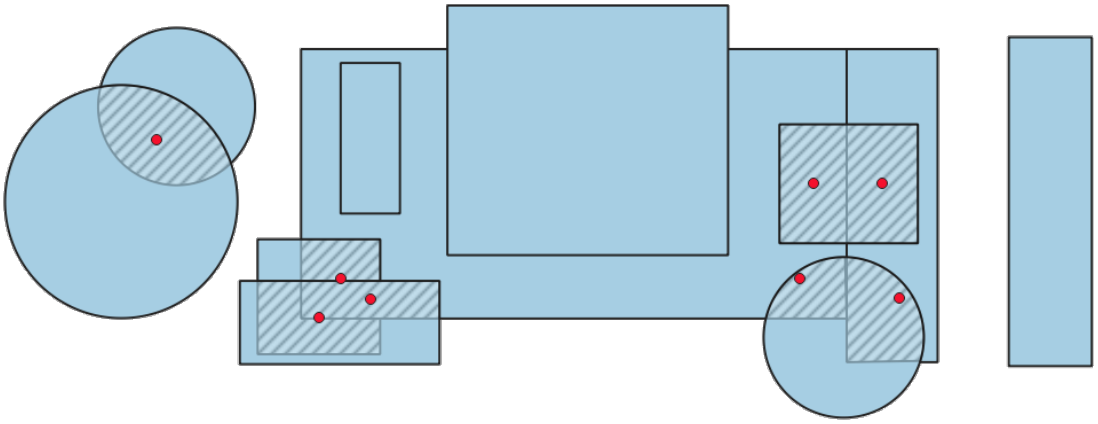


Fig. 24.6: Hashed polygons indicate overlapping areas smaller than the specified minimum.

See also

Delete overlaps, *Validate coverage*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Minimum overlap area	`MIN_OVERLAP_AR`	[numeric: double]	Minimum area of the overlap to be reported as an error. If set to 0, all overlaps are reported.
Overlap errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing centroid points of the overlapping areas. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

continues on next page

Table 24.67 – continued from previous page

Label	Name	Type	Description
Overlap features Optional	`OUTPUT`	[vector: polygon] Default: `[Skip output]`	Specification of the output layer for features containing overlaps. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Overlap errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_errorx`: the x coordinate of the centroid of the overlapping area. • `gc_errory`: the y coordinate of the centroid of the overlapping area. • `gc_error`: the area of the overlapping geometry. • `UNIQUE_ID` field: the unique ID of an overlapped input feature. • `gc_overlap_feature_{unique_id}`: the `UNIQUE_ID` field value for the other overlapping feature.
Overlap features	`OUTPUT`	[vector: polygon]	Output layer containing the overlapping areas. Available fields are the same as in the `ERRORS` output.

Python code

Algorithm ID: `native:checkgeometryoverlap`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Points outside lines

Checks if the points in the input layer are covered by a line in the selected line layers. A point not covered by a line is an error.

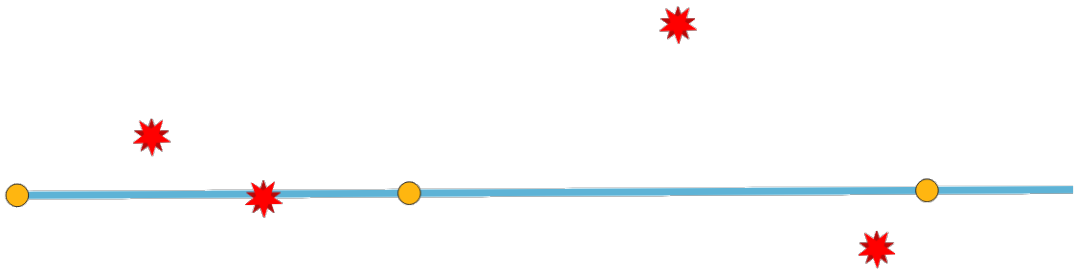


Fig. 24.7: Reporting errors on points not covered by a line.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: point]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Line layers	`LINES`	[vector: line][list]	Layer(s) with the lines to check against.
Points not covered by a line	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing points not covered by a line. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	TOLERANCE	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Points not covered by a line	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx • gc_ringidx • gc_vertidx • gc_errorx: the x coordinate of point outside the lines. • gc_errory: the y coordinate of point outside the lines. • gc_error • UNIQUE_ID field: the unique ID of the input feature that is outside the lines.

Python code

Algorithm ID: native:checkgeometrypointcoveredbyline

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Points outside polygons

Checks if points from the input layer are in polygons from the selected polygon layers. Points that are not fully inside polygons are errors.

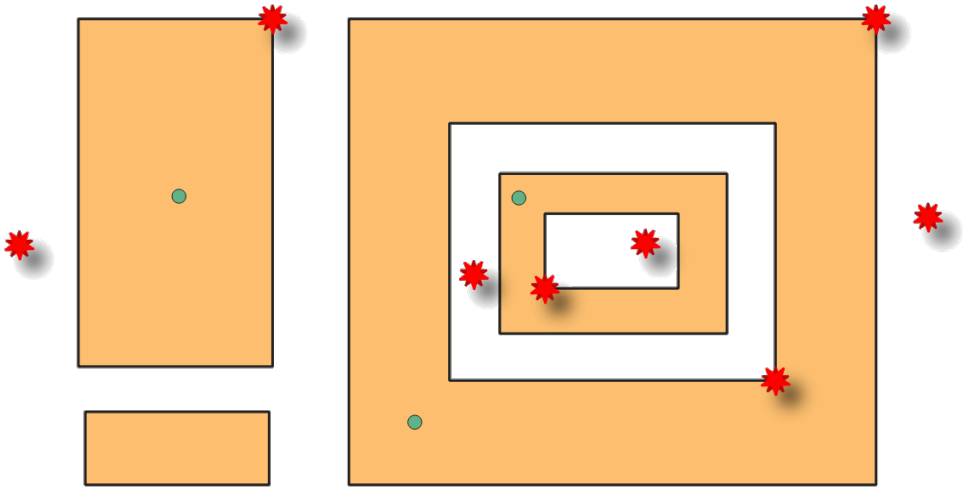


Fig. 24.8: Reporting errors on points outside polygons.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: point]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Polygon layers	`POLYGONS`	[vector: polygon][list]	Layer(s) with the polygons to check against.
Points outside polygons errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing points outside the polygons. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	TOLERANCE	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Points outside polygons errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx • gc_ringidx • gc_vertidx • gc_errorx: the x coordinate of point outside the polygons. • gc_errory: the y coordinate of point outside the polygons. • gc_error • UNIQUE_ID field: the unique ID of the input feature that is outside the polygons.

Python code

Algorithm ID: native:checkgeometrypointinpolygon

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Self-contacts

Checks if the geometry has self contact points (in line or polygon), i.e., a vertex that touches more than two segments of the same ring. Self contacts are errors.

Note

This algorithm detects invalid polygon geometries (self-contacts). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.

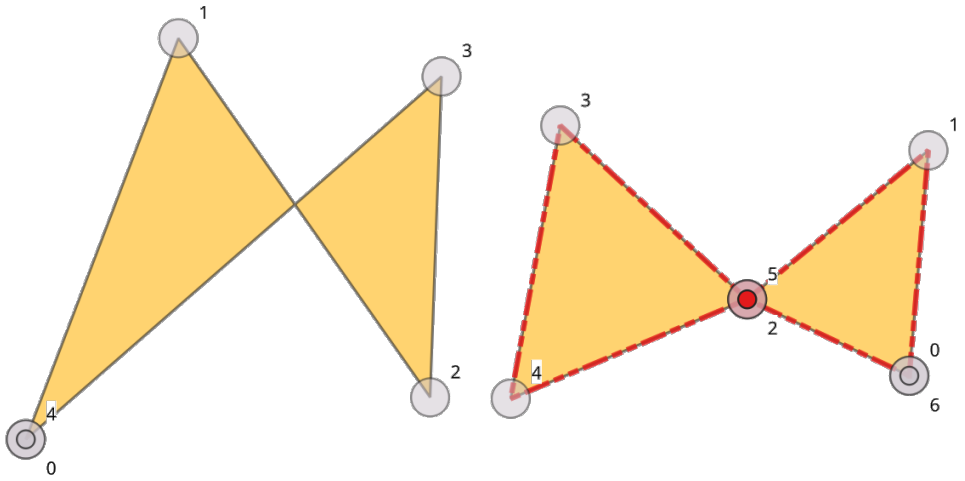


Fig. 24.9: Self-intersection vs self-contact.

See also

Self-intersections

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.

continues on next page

Table 24.76 – continued from previous page

Label	Name	Type	Description
Self contact error points	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Self contact features Optional	`OUTPUT`	[same as input] Default: `[Skip output]`	Line or polygon layer containing self-contact features. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Self contact error points	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx: the index of the feature's geometry part where the self-contact occurs. • gc_ringidx: the index of the ring where the self-contact occurs. • gc_vertidx: the index of the vertex where the self-contact occurs. • gc_errorx: the x coordinate of the self-contact. • gc_errory: the y coordinate of the self-contact. • gc_error • UNIQUE_ID field: the unique ID of the self-contacting input feature.
Self contact features	OUTPUT	[vector: same as input]	Output polygon or line layer with features containing the self-contact features. If no self-contact features are found, the output layer will be empty. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometryselfcontact

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Self-intersections

Detects self-intersections in line or polygon geometries, and reports them as errors. Self-intersections occur when the segments of a geometry cross over each other without having a common vertex.

Note

This algorithm detects invalid polygon geometries (self-intersections). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to Do not Filter (Better Performance).

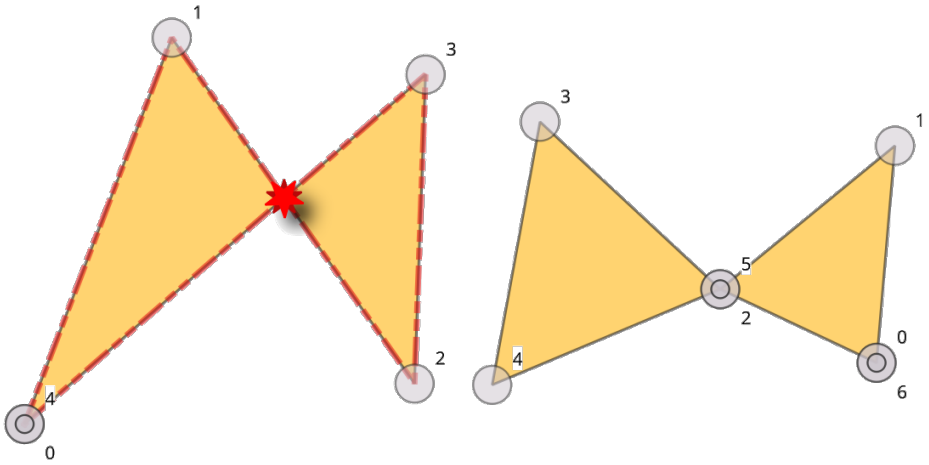


Fig. 24.10: Self-intersection vs self-contact.

See also

Split self-intersecting geometries, *Self-contacts*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to check
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification
Self-intersecting errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Self-intersecting features Optional	`OUTPUT`	[same as input] Default: `[Skip output]`	Line or polygon layer containing self-intersected features. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Self-intersecting errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part where the self-intersection occurs. • `gc_ringidx`: the index of the feature's geometry ring where the self-intersection occurs. • `gc_vertidx` • `gc_errorx`: the x coordinate of the self-intersection. • `gc_errory`: the y coordinate of the self-intersection. • `gc_error` • `gc_segment_1`: the index of the first segment involved in the intersection. • `gc_segment_2`: the index of the second segment involved in the intersection. • `UNIQUE_ID` field: the unique ID of the self-intersecting input feature.
Self-intersecting features	`OUTPUT`	[vector: line, polygon]	Output polygon or line layer with features containing the self-intersecting features. If no self-intersecting features are found, the output layer will be empty. Available fields are the same as in the `ERRORS` output.

Python code

Algorithm ID: `native:checkgeometryselfintersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Sliver polygons

Detects sliver polygons in a polygon layer, i.e., polygons with a thinness greater than a specified value.

The thinness is the ratio between the area of the minimum square containing the polygon and the area of the polygon itself (a square has a thinness value of 1). The thinness value is between 1 and +infinity. A maximum area can be set for limiting the checks to polygons of a lower area. Polygons having a thinness higher than the maximum thinness are errors. To fix sliver polygons, use the *Fix small polygons* algorithm.

Note

This algorithm detects invalid polygon geometries (sliver polygons). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.

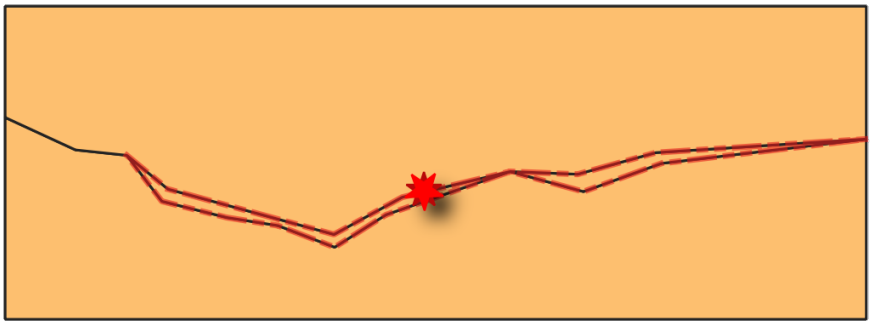


Fig. 24.11: Reporting errors on features with sliver polygons.

See also

Fix small polygons

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Maximum thinness	`MAX_THINNESS`	[numeric: double] Default: 20.0	Maximum thinness value of a polygon to be considered valid. Any polygon with a thinness value above this threshold is reported as error.
Maximum area (map units squared)	`MAX_AREA`	[numeric: double] Default: 0.0	Maximum area of a polygon to be checked (in map units). Any polygon with an area above this threshold is skipped. A value of 0 means all polygons are checked.
Sliver polygon errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the centroid points of sliver polygon features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Sliver polygon features Optional	`OUTPUT`	[vector: polygon] Default: `[Skip output]`	Polygon layer containing sliver polygon features. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Sliver polygon errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx • gc_ringidx • gc_vertidx • gc_errorx: the x coordinate of the centroid of the sliver polygon feature. • gc_errory: the y coordinate of the centroid of the sliver polygon feature. • gc_error: the thinness value of the sliver polygon feature. • UNIQUE_ID field: the unique ID of the input feature that is a sliver polygon.
Sliver polygon features	OUTPUT	[vector: polygon]	Output layer with features containing the sliver polygons. If no sliver polygon features are found, the output layer will be empty. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometrysliverpolygon

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Small angles

Compares the angles within line or polygon geometries to a specified threshold, and reports as error any angle below that value.

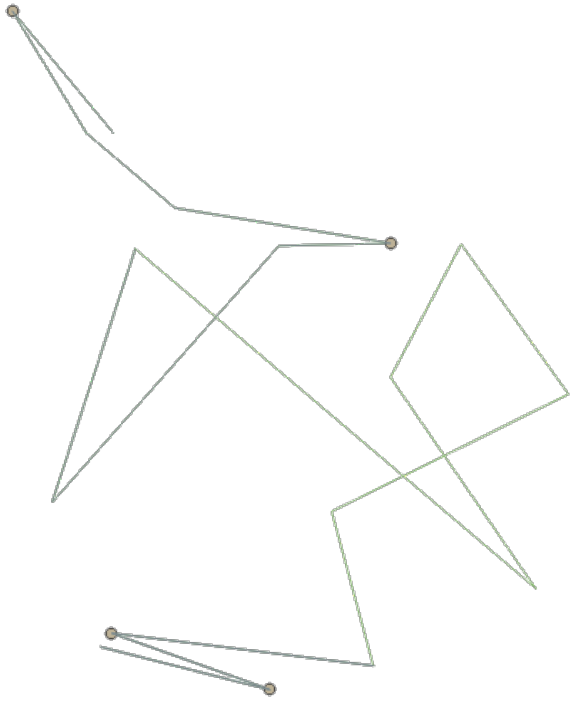


Fig. 24.12: Reporting errors on a line feature for angles lower than 15°.

See also

Delete small angles

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Minimum angle	`MIN_ANGLE`	[numeric: double]	Allowed minimum angle between adjacent segments in a feature's geometry. Any angle below this threshold is reported as error.

continues on next page

Table 24.85 – continued from previous page

Label	Name	Type	Description
Small angle errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Small angle errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part containing the small angle. • `gc_ringidx`: the index of the feature's geometry ring containing the small angle. • `gc_vertidx`: the index of the vertex with the small angle in the feature's geometry ring. • `gc_errorx`: the x coordinate of the vertex with the small angle. • `gc_errory`: the y coordinate of the vertex with the small angle. • `gc_error`: the error angle value. • `UNIQUE_ID` field: the unique ID of the input feature with the small angle.

Python code

Algorithm ID: `native:checkgeometryangle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Small polygons

Detects polygon features whose area is below a specified value as errors.

Note

This algorithm detects invalid polygon geometries (small polygons). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.

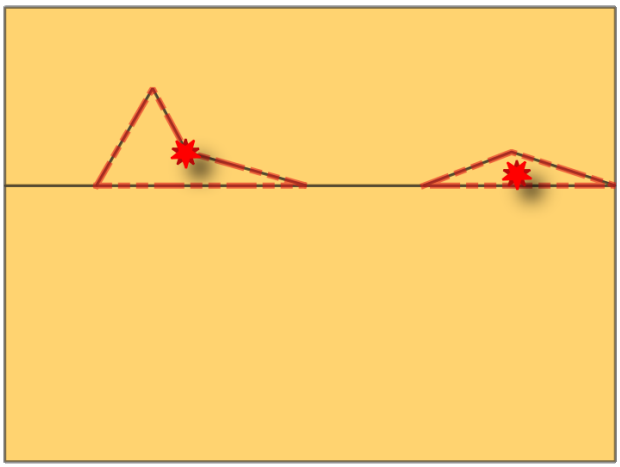


Fig. 24.13: Reporting errors on features with small polygons (below given area threshold).

See also

Fix small polygons

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Area threshold	`AREATHRESHOLD`	[numeric: double] Default: 0	Minimum area of a polygon to be considered valid, in selected units. Any polygon with an area below this threshold is reported as error.
Small polygons errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing centroid points of small polygons. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Small polygons features Optional	`OUTPUT`	[vector: polygon] Default: `[Skip output]`	Specification of the output layer for features containing small polygon features. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Small polygons errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx • gc_ringidx • gc_vertidx • gc_errorx: the x coordinate of the centroid of the small polygon. • gc_errory: the y coordinate of the centroid of the small polygon. • gc_error: the area of the small polygon. • UNIQUE_ID field: the unique ID of the input feature that is a small polygon.
Small polygons features	OUTPUT	[vector: polygon]	Output polygon layer with features containing the small polygon features. If no small polygon features are found, the output layer will be empty. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometryarea

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Small gaps

Checks for gaps between polygons in the input layer. Gaps with an area smaller than the gap threshold are reported as errors. If an allowed gaps layer is provided, the algorithm ignores gaps that fall entirely within polygons from this layer. An optional buffer can be applied to the allowed gaps.

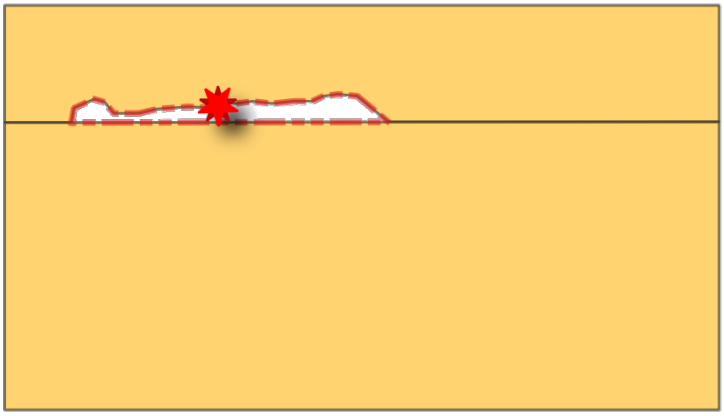


Fig. 24.14: Reporting errors on polygon features for gaps smaller than the specified threshold.

See also

Fill gaps, *Validate coverage*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Gap threshold	`GAP_THRESHOLD`	[numeric: double] Default: 0.0	Maximum area of gaps to be reported as errors, in map units. If set to 0, all the gaps are reported.
Allowed gaps layer Optional	`ALLOWED_GAPS_L`	[vector: polygon]	Optional layer specifying polygons whose areas should be ignored during the gap check. Gaps that fall entirely within these polygons will not be reported as errors.
Allowed gaps buffer Optional	`ALLOWED_GAPS_B`	[numeric: double] Default: 0.0	Buffer distance to apply to the allowed gaps layer, in selected units. Gaps located within this buffered area are ignored. Allows adding a spatial tolerance around allowed gaps to avoid reporting near-boundary gaps as errors.

continues on next page

Table 24.91 – continued from previous page

Label	Name	Type	Description
Neighbors layer	`NEIGHBORS`	[vector: table] Default: `[Create temporary layer]`	Specification of the output table representing the gap ID and the unique ID of its neighbor features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Gap errors Optional	`ERRORS`	[vector: point] Default: `[Skip output]`	Specification of the output layer containing the centroid points of the gaps. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Gap features	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output layer containing the gap geometries. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Neighbors layer	`NEIGHBORS`	[vector: table]	A 1–N relational table, meaning that one gap can be associated with multiple neighboring polygons. The output table contains the following fields: • `gc_errorid`: the ID of the gap. • `UNIQUE_ID` field: the unique ID of an input feature that is a neighbor of the gap.

continues on next page

Table 24.93 – continued from previous page

Label	Name	Type	Description
Gap errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx • gc_ringidx • gc_vertidx • gc_errorx: the x coordinate of the centroid of the gap. • gc_errory: the y coordinate of the centroid of the gap. • gc_error: the area of the gap. • gc_errorid: the ID of the gap.
Gap features	OUTPUT	[vector: polygon]	Output layer containing the gap geometries. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometrygap

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Small segments

Calculates length of individual segments in line or polygon geometries, and reports segments shorter than a minimum length as errors.

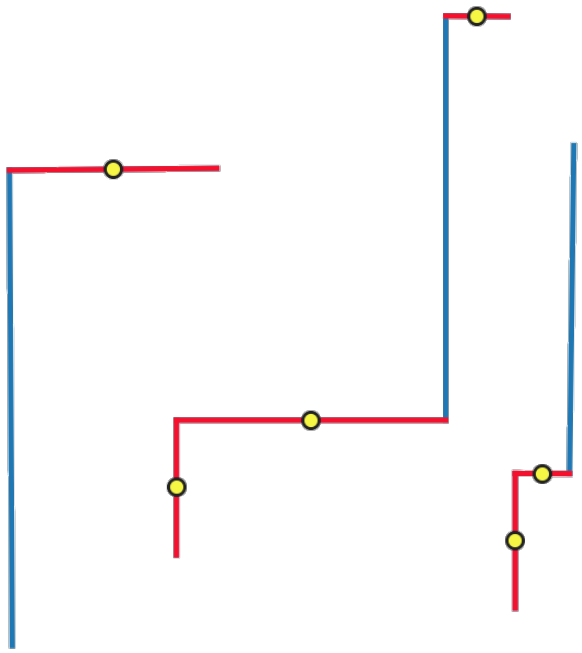


Fig. 24.15: Reporting errors (in red) on single line segments shorter than the given threshold.

See also

Delete features

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Minimum segment length	`MIN_SEGMENT_LE`	[numeric: double] Default: 0.0	Minimum length of segments to be considered valid, in map units.
Short segments errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing centroid of short segments. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

continues on next page

Table 24.94 – continued from previous page

Label	Name	Type	Description
Short segments features Optional	`OUTPUT`	[vector: same as input] Default: `[Skip output]`	Specification of the output layer for features containing short segments. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Short segments errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part containing the small segment. • `gc_ringidx`: the index of the feature's geometry ring containing the small segment. • `gc_vertidx`: the index of the end vertex of the small segment in the feature's geometry ring. • `gc_errorx`: the x coordinate of the centroid of the small segment. • `gc_errory`: the y coordinate of the centroid of the small segment. • `gc_error`: the error segment length. • `UNIQUE_ID` field: the unique ID of the input feature with the small segment.

continues on next page

Table 24.96 – continued from previous page

Label	Name	Type	Description
Short segments features	OUTPUT	[vector: same as input]	Output layer containing, for each identified short segment, the feature it belongs to. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometrysegmentlength

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Strictly multipart

Checks if multipart geometries have more than one part. Multipart geometries with only one part are errors.

See also

Convert to strictly multipart

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	INPUT	[vector: geometry]	Layer with the geometries to check.
Unique feature identifier	UNIQUE_ID	[tablefield: any]	Field storing unique values for feature identification.
One-part geometry errors	ERRORS	[vector: point] Default: [Create temporary layer]	Specification of the output layer containing centroid of the multipart geometries that have only one part. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

continues on next page

Table 24.97 – continued from previous page

Label	Name	Type	Description
One-part geometry features Optional	`OUTPUT`	[vector: same as input] Default: `[Skip output]`	Specification of the output layer for features containing multipart geometries that have only one part. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
One-part geometry errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the multipart geometry that has only one part. • `gc_errory`: the y coordinate of the centroid of the multipart geometry that has only one part. • `gc_error` • `UNIQUE_ID` field: the unique ID of the input feature that is multipart but has only one part.
One-part geometry features	`OUTPUT`	[vector: same as input]	Output layer containing features that are multipart but have only one part. Available fields are the same as in the `ERRORS` output.

Python code

Algorithm ID: `native:checkgeometrymultipart`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

24.1.4 数据库

Export to PostgreSQL

Exports a vector layer to a PostgreSQL database, creating a new relation. If a relation with the same name exists, it can be removed before the new relation is created. Prior to this a connection between QGIS and the PostgreSQL database has to be created (see eg 创建存储的连接).

参数

标注	名称	类型	描述
Layer to import	`INPUT`	[vector: any]	Vector layer to add to the database
Database (connection name)	`DATABASE`	[string]	Name of the database connection (not the database name). Existing connections will be shown in the combobox.
Schema (schema name) 可选	`SCHEMA`	[string] Default: 'public'	Name of the schema to store the data. It can be a new one or already exist.
Table to import to (leave blank to use layer name) 可选	`TABLENAME`	[string] Default: ''	Defines a table name for the imported vector file. If nothing is added, the layer name will be used.
Primary key field 可选	`PRIMARY_KEY`	[tablefield: any]	Sets the primary key field from an existing field in the vector layer. A column with **unique** values can be used as Primary key for the database.
Geometry column	`GEOMETRY_COLUM`	[string] Default: 'geom'	Defines the name of the geometry column in the new PostgreSQL table. Geometry information for the features is stored in this column.
Encoding 可选	`ENCODING`	[string] Default: 'UTF-8'	Defines the encoding of the output layer
Overwrite	`OVERWRITE`	[boolean] Default: True	If the specified table exists, setting this option to `True` will make sure that it is deleted and a new table will be created before the features are added. If this option is `False` and the table exists, the algorithm will throw an exception ("relation already exists").
Create spatial index	`CREATEINDEX`	[boolean] Default: True	Specifies whether to create a spatial index or not

continues on next page

Table 24.100 – continued from previous page

标注	名称	类型	描述
Convert field names to lowercase	LOWERCASE_NAME	[boolean] Default: True	Converts the field names of the input vector layer to lowercase
Drop length constraint on character fields	DROP_STRING_LE	[boolean] Default: False	Should length constraints on character fields be dropped or not
Create single-part geometries instead of multi-part	FORCE_SINGLEPA	[boolean] Default: False	Should the features of the output layer be single-part instead of multi-part. By default the existing geometries information are preserved.

输出

The algorithm has no output.

Python 代码

Algorithm ID: `native:importintopostgis`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Export to SpatiaLite

Exports a vector layer to a SpatiaLite database. Prior to this a connection between QGIS and the SpatiaLite database has to be created (see eg *SpatiaLite 图层*).

参数

标注	名称	类型	描述
Layer to import	INPUT	[vector: any]	Vector layer to add to the database
File database	DATABASE	[vector: any]	The SQLite/SpatiaLite database file to connect to
Table to import to (leave blank to use layer name) 可选	TABLENAME	[string] Default: ''	Defines the table name for the imported vector file. If nothing is specified, the layer name will be used.
Primary key field 可选	PRIMARY_KEY	[tablefield: any]	Use a field in the input vector layer as the primary key
Geometry column	GEOMETRY_COLUM	[string] Default: 'geom'	Defines the name of the geometry column in the new SpatiaLite table. Geometry information for the features is stored in this column.
Encoding 可选	ENCODING	[string] Default: 'UTF-8'	Defines the encoding of the output layer

continues on next page

Table 24.101 – continued from previous page

标注	名称	类型	描述
Overwrite	OVERWRITE	[boolean] Default: True	If the specified table exists, setting this option to True will make sure that it is deleted and a new table will be created before the features of the layer is added. If this option is False and the table exists, the algorithm will throw an exception ("table already exists").
Create spatial index	CREATEINDEX	[boolean] Default: True	Specifies whether to create a spatial index or not
Convert field names to lowercase	LOWERCASE_NAME	[boolean] Default: True	Convert the field names of the input vector layer to lowercase
Drop length constraint on character fields	DROP_STRING_LE	[boolean] Default: False	Should length constraints on character fields be dropped or not
Create single-part geometries instead of multi-part	FORCE_SINGLEPA	[boolean] Default: False	Should the features of the output layer be single-part instead of multi-part. By default the existing geometries information are preserved.

输出

The algorithm has no output.

Python 代码

Algorithm ID: `native:importintospatialite`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Package layers

Adds layers to a GeoPackage.

If the GeoPackage exists and `Overwrite existing GeoPackage` is checked, it will be overwritten (removed and recreated). If the GeoPackage exists and `Overwrite existing GeoPackage` is not checked, the layer will be appended.

参数

标注	名称	类型	描述
Input layers	`LAYERS`	[vector: any] [list]	The (vector) layers to import into the GeoPackage. Raster layers are not supported. If a raster layer is added, a `QgsProcessingException` will be thrown.
Overwrite existing GeoPackage	`OVERWRITE`	[boolean] Default: False	If the specified GeoPackage exists, setting this option to `True` will make sure that it is deleted and a new one will be created before the layers are added. If set to `False`, layers will be appended.
Save layer styles into GeoPackage	`SAVE_STYLES`	[boolean] Default: True	Save the layer styles.
Save only selected features	`SELECTED_FEATU`	[boolean] Default: False	If a layer has a selection, setting this option to `True` will result in only selected features being saved. For layers without a selection all features will be saved.
Export related layers following relations defined in the project	`EXPORT_RELATED`	[boolean] Default: False	If an input layer has *relations* set in the project, setting this option to `True` will result in exporting also its related layer(s). If the layer has features selected, then only their related features will be exported unless the related layer was also an input layer.
Destination GeoPackage	`OUTPUT`	[file] 默认：[保存到临时文件]	Specify where to store the GeoPackage file. One of • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Layers within new package	`OUTPUT_LAYERS`	[string] [list]	The list of layers added to the GeoPackage.

Python 代码

Algorithm ID: `native:package`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

PostgreSQL execute and load SQL

Allows a SQL database query to be performed on a PostgreSQL database connected to QGIS and loads the result. The algorithm **won't** create a new layer: it is designed to run queries on the layer itself.

Example

1. Set all the values of an existing field to a fixed value. The SQL query string will be:

   ```
   UPDATE your_table SET field_to_update=20;
   ```

   In the example above, the values of the field `field_to_update` of the table `your_table` will be all set to `20`.

2. Create a new `area` column and calculate the area of each feature with the `ST_AREA` PostGIS function.

   ```
   -- Create the new column "area" on the table your_table"
   ALTER TABLE your_table ADD COLUMN area double precision;
   -- Update the "area" column and calculate the area of each feature:
   UPDATE your_table SET area=ST_AREA(geom);
   ```

See also

PostgreSQL execute SQL, *Execute SQL*, *SpatiaLite execute SQL*

参数

标注	名称	类型	描述
Database (connection name)	`DATABASE`	[string]	The database connection (not the database name). Existing connections will be shown in the combobox.
SQL query	`SQL`	[string]	Defines the SQL query, for example `'UPDATE my_table SET field=10'`.
Unique ID field name	`ID_FIELD`	[string] Default: id	Sets the primary key field (a column in the result table)
Geometry field name 可选	`GEOMETRY_FIELD`	[string] Default: 'geom'	Name of the geometry column (a column in the result table)

输出

标注	名称	类型	描述
SQL layer	`OUTPUT`	[vector: any]	The resulting vector layer to be loaded into QGIS.

Python 代码

Algorithm ID: `qgis:postgisexecuteandloadsql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

PostgreSQL execute SQL

Allows a SQL database query to be performed on a PostgreSQL database connected to QGIS. The algorithm **won't** create a new layer: it is designed to run queries on the layer itself.

Example

1. Set all the values of an existing field to a fixed value. The SQL query string will be:

   ```
   UPDATE your_table SET field_to_update=20;
   ```

   In the example above, the values of the field `field_to_update` of the table `your_table` will be all set to `20`.

2. Create a new `area` column and calculate the area of each feature with the `ST_AREA` PostGIS function.

   ```
   -- Create the new column "area" on the table your_table"
   ALTER TABLE your_table ADD COLUMN area double precision;
   -- Update the "area" column and calculate the area of each feature:
   UPDATE your_table SET area=ST_AREA(geom);
   ```

See also

PostgreSQL execute and load SQL, *Execute SQL*, *SpatiaLite execute SQL*

参数

标注	名称	类型	描述
Database (connection name)	`DATABASE`	[string]	The database connection (not the database name). Existing connections will be shown in the combobox.
SQL query	`SQL`	[string]	Defines the SQL query, for example `'UPDATE my_table SET field=10'`.

输出

No output is created. The SQL query is executed in place.

Python 代码

Algorithm ID: `native:postgisexecutesql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

SpatiaLite execute SQL

Allows a SQL database query to be performed on a SpatiaLite database. The database is determined by an input layer or file. The algorithm **won't** create a new layer: it is designed to run queries on the layer itself.

See also

PostgreSQL execute SQL, *Execute SQL*

For some SQL query examples see *PostgreSQL Query Examples*.

参数

标注	名称	类型	描述
Database layer (or file)	`DATABASE`	[file]	The SQLite/SpatiaLite database file to connect to
SQL query	`SQL`	[string] Default: ''	Defines the SQL query, for example `'UPDATE my_table SET field=10'`.

输出

No output is created. The SQL query is executed in place.

Python 代码

Algorithm ID: `native:spatialiteexecutesql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

SpatiaLite execute SQL (registered DB)

Allows a SQL database query to be performed on a SpatiaLite database connected to QGIS. The algorithm **won't** create a new layer: it is designed to run queries on the layer itself.

See also

PostgreSQL execute SQL, *Execute SQL*

For some SQL query examples see *PostgreSQL Query Examples*.

参数

标注	名称	类型	描述
Database	`DATABASE`	[enumeration] Default: not set	Select a SQLite/SpatiaLite database connected to the current session
SQL query	`SQL`	[string] Default: ''	Defines the SQL query, for example `'UPDATE my_table SET field=10'`.

输出

No output is created. The SQL query is executed in place.

Python 代码

Algorithm ID: `native:spatialiteexecutesqlregistered`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.5 File tools

Download file via HTTP(S)

Downloads a URL to the file system with an HTTP(S) GET or POST request.

参数

Basic parameters

标注	名称	类型	描述
URL	`URL`	[string]	The URL of the file to download.
File destination	`OUTPUT`	[string] 默认: [保存到临时文件]	Specification of the file destination. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Method	`METHOD`	[enumeration] Default: 0	The HTTP method to use for the request. Options are: • 0 --- GET • 1 --- POST
Data 可选	`DATA`	[string]	The data to add in the body if the request is a POST.

输出

标注	名称	类型	描述
File destination	`OUTPUT`	[string]	The location of the downloaded file

Python 代码

Algorithm ID: `native:filedownloader`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

HTTP(S) POST/GET request

Performs a HTTP(S) POST/GET request and returns the HTTP status code and the reply data. If an error occurs then the error code and the message will be returned. Optionally, the result can be written to a file on the disk. By default the algorithm will warn on errors. Optionally, the algorithm can be set to treat HTTP errors as failures.

参数

标注	名称	类型	描述
URL or file path	`URL`	[string]	The URL or path of the file to open.
Method	`METHOD`	[enumeration] Default: 0	The HTTP method to use for the request. Options are: • 0 --- GET • 1 --- POST
POST Data 可选	`DATA`	[string]	The data to add in the body if the request is a POST.
Authentication 可选	`AUTH_CONFIG`	[authconfig] Default: No authentication	An authentication configuration to pass
Consider HTTP errors as failures	`FAIL_ON_ERROR`	[boolean] Default: False	If set, the algorithm will fail on encountering an HTTP error.
File destination 可选	`OUTPUT`	[string] Default: `[Skip Output]`	The result can be written to a file instead of being returned as a string. Specification of the file destination. *One of*: • Skip Output • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
File destination	`OUTPUT`	[file]	The downloaded file with the returned result
HTTP Status	`STATUS_CODE`	[string]	The HTTP Status
Network error code	`ERROR_CODE`	[string]	The error code when it fails (like as well on 404 HTTP Status etc.)
Network error message	`ERROR_MESSAGE`	[string]	A string containing the error message in case of failure
Result data	`RESULT_DATA`	[string]	A string containing the result data in case of success

Python 代码

Algorithm ID: `native:httprequest`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Open file or URL

Opens files in their default associated application, or URLs in the user's default web browser.

参数

标注	名称	类型	描述
URL or file path	`URL`	[string]	The URL or path of the file to open.

输出

The algorithm has no output.

Python 代码

Algorithm ID: `native:openurl`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.6 Fix Geometry

Add missing vertices along borders

Adds the missing vertices along polygons borders, based on an error layer from the *Missing vertices along borders* algorithm.

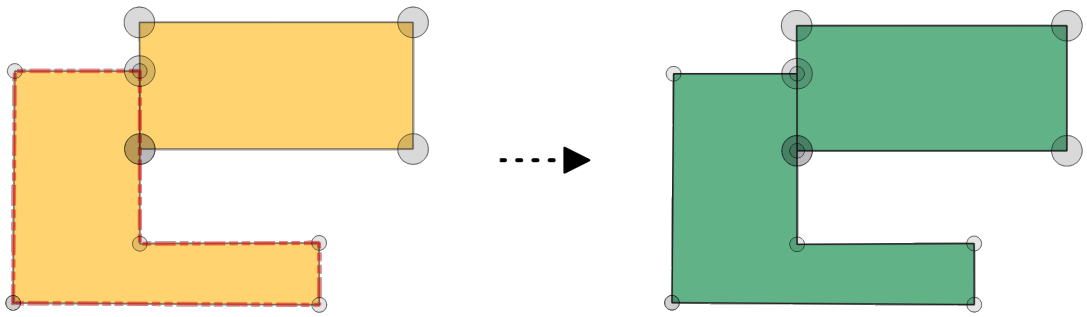


Fig. 24.16: Fixing missing vertices along polygon borders.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to fix, same as input in the check algorithm.
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Field of part index	`PART_IDX`	[tablefield: integer] Default: `gc_partidx`	Field storing the erroneous feature's geometry part number.
Field of ring index	`RING_IDX`	[tablefield: integer] Default: `gc_ringidx`	Field storing the erroneous feature's geometry ring number.
Field of vertex index	`VERTEX_IDX`	[tablefield: integer] Default: `gc_vertidx`	Field storing the erroneous feature's geometry vertex number.
Border vertices fixed layer	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing border vertices	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing border vertices	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the missing vertex. • `gc_errory`: the y coordinate of the missing vertex. • `gc_error` • `report`: a text field describing the fix applied. • `error_fixed`: a boolean field indicating whether the error was fixed. • `UNIQUE_ID` field: the unique ID of the input feature that has a missing vertex.
Border vertices fixed layer	`OUTPUT`	[vector: polygon]	Output layer with the geometry fix applied to the input features. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometrymissingvertex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Convert to strictly multipart

Converts multipart geometries that consists of only one geometry into singlepart geometries, based on an error layer from the *Strictly multipart* algorithm. Layer geometry type will not be changed and will remain multipart.

See also

Promote to multipart

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to fix, same as input in the check algorithm.
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Strictly-multipart layer	`OUTPUT`	[vector: same as input] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing multipart	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the centroids of the fixed geometries. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing multipart	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the multipart geometry that has only one part. • `gc_errory`: the y coordinate of the centroid of the multipart geometry that has only one part. • `gc_error` • `report`: a text field describing the fix applied. • `error_fixed`: a boolean field indicating whether the error was fixed. • `UNIQUE_ID` field: the unique ID of the input feature that is multipart but has only one part.
Strictly-multipart layer	`OUTPUT`	[vector: same as input]	Output layer with the geometry fix applied to the input features. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometrymultipart`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Delete duplicated vertices

Deletes duplicated vertices from the input geometries, based on errors reported by the *Duplicated vertices* algorithm.

See also

Remove duplicate vertices

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to fix, same as input in the check algorithm.
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Field of part index	`PART_IDX`	[tablefield: integer] Default: `gc_partidx`	Field storing the erroneous feature's geometry part number.
Field of ring index	`RING_IDX`	[tablefield: integer] Default: `gc_ringidx`	Field storing the erroneous feature's geometry ring number.
Field of vertex index	`VERTEX_IDX`	[tablefield: integer] Default: `gc_vertidx`	Field storing the erroneous feature's geometry vertex number.
Fixed duplicate vertices layer	`OUTPUT`	[vector: same as input] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing duplicate vertices	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing duplicate vertices	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part containing the duplicate vertex. • `gc_ringidx`: the index of the feature's geometry ring containing the duplicate vertex. • `gc_vertidx`: the index of the duplicate vertex in the feature's geometry ring. • `gc_errorx`: the x coordinate of the duplicate vertex. • `gc_errory`: the y coordinate of the duplicate vertex. • `gc_error` • `report`: a text field describing the fix applied. • `error_fixed`: a boolean field indicating whether the error was fixed. • `UNIQUE_ID` field: the unique ID of the input feature that has duplicate vertices.
Fixed duplicate vertices layer	`OUTPUT`	[vector: same as input]	Output layer with the geometry fix applied to the input features. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometryduplicatenodes`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Delete features

Deletes error features based on an error layer from some check geometry algorithms, such as:

- *Features inside polygon*
- *Degenerate polygons*
- *Small segments*
- *Duplicated geometries*

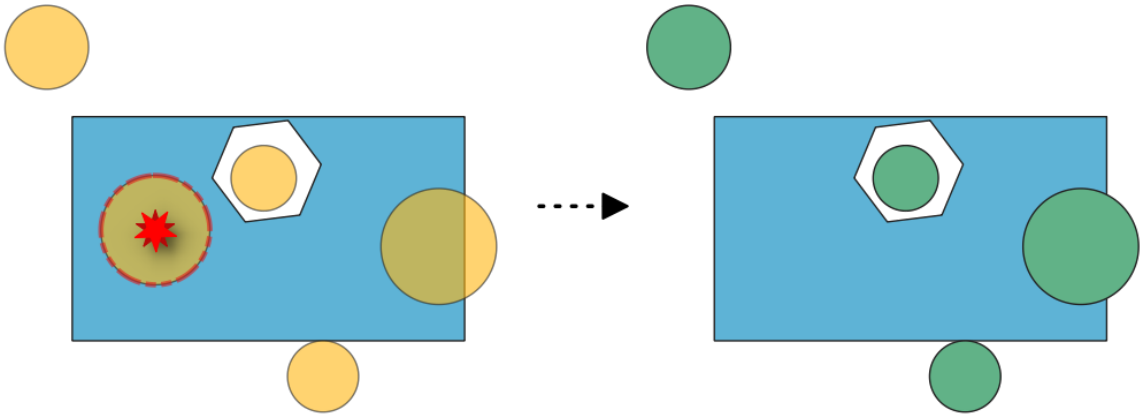


Fig. 24.17: Fixing errors from the "*Features inside polygon*" algorithm check.

See also

Delete duplicate geometries, *Drop geometries*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: geometry]	Layer with the geometries to fix, same as input in the check algorithm.
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Cleaned layer	`OUTPUT`	[vector: same as input] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

continues on next page

Table 24.124 – continued from previous page

Label	Name	Type	Description
Report layer from deleting features	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Outputs

Label	Name	Type	Description
Report layer from deleting features	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the erroneous geometry. • `gc_errory`: the y coordinate of the centroid of the erroneous geometry. • `gc_error`: the index of the feature where the error belongs. • `report`: a text field describing the fix applied. • `error_fixed`: a boolean field indicating whether the error was fixed. • `UNIQUE_ID` field: the unique ID of the input feature with the error.
Cleaned layer	`OUTPUT`	[vector: same as input]	Output layer with features removed based on detected errors. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometrydeletefeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Delete overlaps

Deletes overlapping areas based on an error layer from the *Overlaps* algorithm.

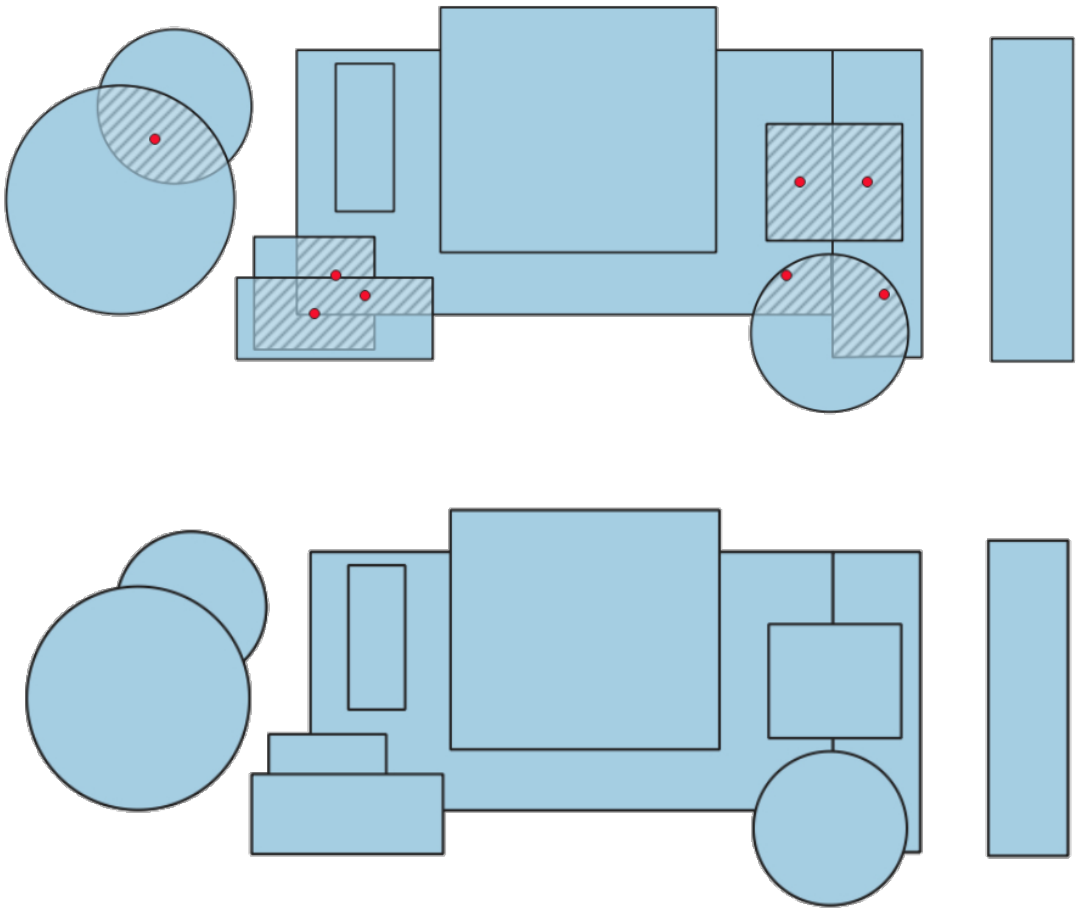


Fig. 24.18: Deleting overlapping areas.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to fix, same as input in the check algorithm.
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Field of overlap feature unique identifier	`OVERLAP_FEATUR`	[tablefield: any]	Field storing unique values for the overlapping feature identification in input layer, as set in the check algorithm.
Field of error value	`ERROR_VALUE_ID`	[tablefield: any]	Field storing the error value for each feature, as set in the check algorithm.
No-overlap layer	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing overlaps	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing overlaps	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_errorx`: the x coordinate of the centroid of the overlapping area. • `gc_errory`: the y coordinate of the centroid of the overlapping area. • `gc_error`: the area of the overlapping geometry. • `UNIQUE_ID` field: the unique ID of the unique ID of an overlapped input feature. • `gc_overlap_feature_{unique_id}`: the `UNIQUE_ID` field value for the other overlapping feature. • `report`: a text field describing the fix applied or justifying the failure. Possible values are: – Remove overlapping area from neighboring polygon with shortest shared edge – Error is obsolete • `error_fixed`: a boolean field indicating whether the error was fixed.
No-overlap layer	`OUTPUT`	[vector: polygon]	Output layer with input features edited. Overlapping areas reported as errors are removed. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometryoverlap`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Delete small angles

Deletes vertices based on an error layer from the *Small angles* algorithm. When deletion of a vertex results in a duplicate vertex (when a spike vertex is deleted), the duplicate vertex is deleted to keep a single vertex and preserve topology.

> **Attention**
>
> This algorithm removes the vertex at the reported small angles, generating new segments that may form a new small angle.

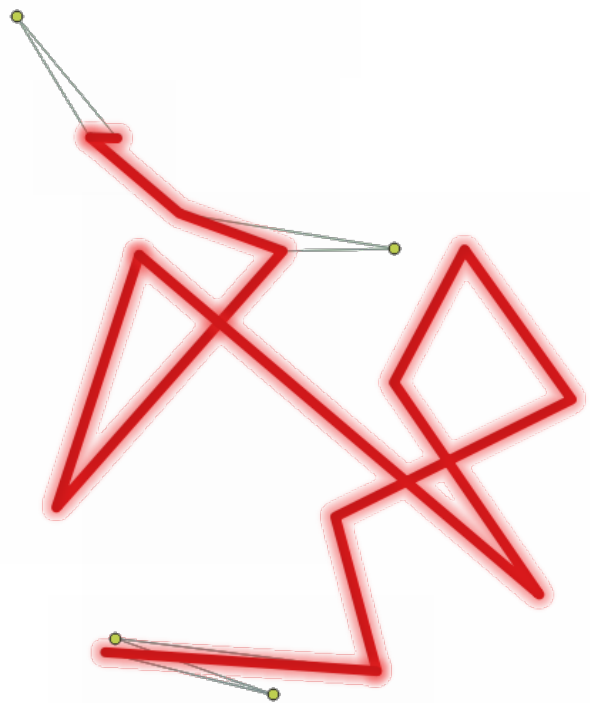


Fig. 24.19: Fixing reported errors on a line feature for angles lower than 15°.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to fix, same as input in the check algorithm
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm
Field of part index	`PART_IDX`	[tablefield: integer] Default: `gc_partidx`	Field storing the erroneous feature's geometry part number.

continues on next page

Table 24.129 – continued from previous page

Label	Name	Type	Description
Field of ring index	`RING_IDX`	[tablefield: integer] Default: `gc_ringidx`	Field storing the erroneous feature's geometry ring number.
Field of vertex index	`VERTEX_IDX`	[tablefield: integer] Default: `gc_vertidx`	Field storing the erroneous feature's geometry vertex number.
Small angle fixed layer	`OUTPUT`	[vector: same as input] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing small angles	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing small angles	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part containing the small angle. • `gc_ringidx`: the index of the feature's geometry ring containing the small angle. • `gc_vertidx`: the index of the vertex with the small angle in the feature's geometry ring. • `gc_errorx`: the x coordinate of the vertex with the small angle. • `gc_errory`: the y coordinate of the vertex with the small angle. • `gc_error`: the error angle value. • `UNIQUE_ID` field: the unique ID of the input feature with the small angle. • `report`: a text field describing the fix applied or justifying the failure. • `error_fixed`: a boolean field indicating whether the error was fixed.
Small angle fixed layer	`OUTPUT`	[same as input]	Output layer with the geometry fix applied to the input features The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometryangle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Fill gaps

Fills the gaps based on a gap and neighbors layer from the *Small gaps* algorithm. Three different fixing methods are available.

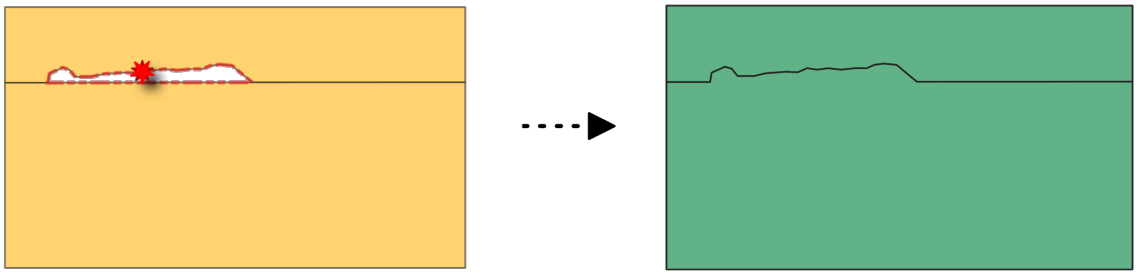


Fig. 24.20: Fixing gap between polygon features using the `Add to largest neighbouring area` method.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to fix, same as input in the check algorithm.
Neighbors layer	`NEIGHBORS`	[vector: table]	Relational table with the unique ID of the gap and its neighbors in input layer, generated by the check algorithm.
Gaps layer	`GAPS`	[vector: polygon]	Polygon layer with the gaps geometries, generated by the check algorithm.
Method	`METHOD`	[enumeration] Default: 0	Method to apply to the gap fixing. Possible values are: • 0 --- `Add to longest shared edge`: adds the gap to the neighboring feature with the longest shared edge. • 1 --- `Create new feature`: creates a new feature for each gap. • 2 --- `Add to largest neighbouring area`: adds the gap to the neighboring feature with the largest area.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any] Default: gc_errorid	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Field of error id	`ERROR_ID_IDX`	[tablefield: any]	Field storing the error ID for each gap, as output by the check algorithm.

continues on next page

Table 24.132 – continued from previous page

Label	Name	Type	Description
Gaps-filled layer	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing gaps	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing gaps	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the gap. • `gc_errory`: the y coordinate of the centroid of the gap. • `gc_error`: the area of the gap. • `gc_errorid`: the ID of the gap. • `report`: a text field mentioning the method used to fix the gap. • `error_fixed`: a boolean field indicating whether the error was fixed.
Gaps-filled layer	`OUTPUT`	[vector: polygon]	Output layer of polygons without gaps. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometrygap`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Fill holes

Deletes holes in polygon geometries based on an error layer from the *Holes* algorithm.

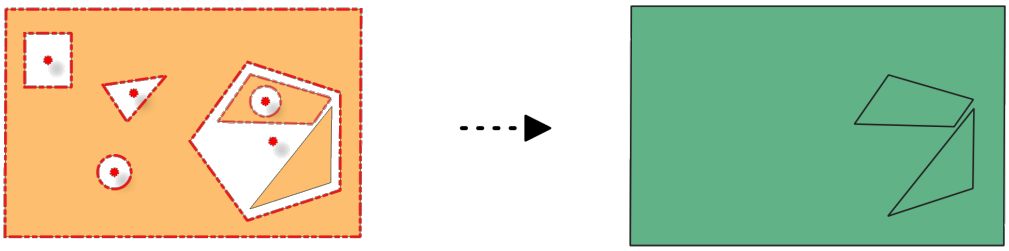


Fig. 24.21: Deleting holes in a polygon feature.

See also

Delete holes

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to fix, same as input in the check algorithm
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm
Field of part index	`PART_IDX`	[tablefield: integer] Default: `gc_partidx`	Field storing the erroneous feature's geometry part number.
Field of ring index	`RING_IDX`	[tablefield: integer] Default: `gc_ringidx`	Field storing the erroneous feature's geometry ring number.
Field of vertex index	`VERTEX_IDX`	[tablefield: integer] Default: `gc_vertidx`	Field storing the erroneous feature's geometry vertex number.
Holes-filled layer	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing holes	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing holes	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part containing the hole. • `gc_ringidx`: the index of the feature's geometry ring containing the hole. • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the hole. • `gc_errory`: the y coordinate of the centroid of the hole. • `gc_error` • `UNIQUE_ID` field: the unique ID of the input feature that has a hole. • `report`: a text field describing the fix applied or justifying the failure. • `error_fixed`: a boolean field indicating whether the error was fixed.
Holes-filled layer	`OUTPUT`	[vector: polygon]	Output layer of polygons without holes. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometryhole`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Fix small polygons

Merges neighboring polygons according to the chosen method, based on an error layer from the *Small polygons* or *Sliver polygons* algorithm.

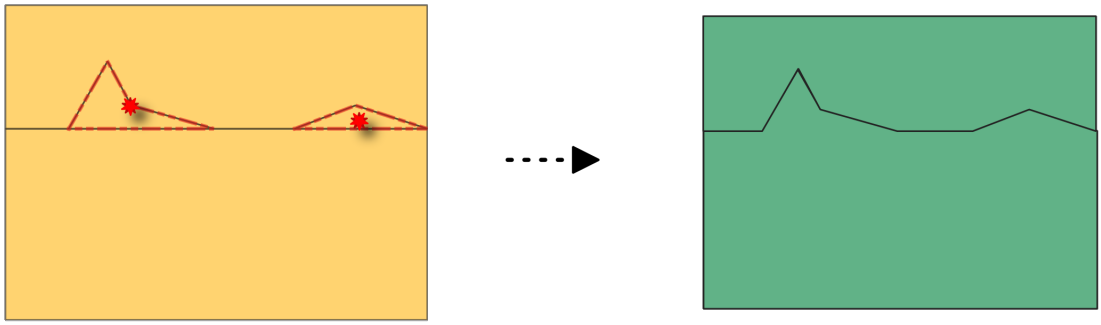


Fig. 24.22: Before and after fixing small polygons (method: merge with neighboring polygon with largest area).

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to fix, same as input in the check algorithm.
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm.
Method	`METHOD`	[enumeration] Default: 0	Method to apply to the small polygon geometries: • 0 --- `Merge with neighboring polygon with longest shared edge`: merges the small polygon geometry with the neighboring polygon that has the longest shared edge. • 1 --- `Merge with neighboring polygon with largest area`: merges the small polygon geometry with the neighboring polygon that has the largest area. • 2 --- `Merge with neighboring polygon with identical attribute value, if any, or leave as is`: merges the small polygon geometry with a neighboring polygon that has the same attribute value, if such a polygon exists.

continues on next page

Table 24.138 – continued from previous page

Label	Name	Type	Description
Field to consider when merging polygons with the identical attribute method Optional	MERGE_ATTRIBUT	[tablefield: any]	Field in the input layer to consider when merging polygons with the identical attribute method.
Field of original feature unique identifier	UNIQUE_ID	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Field of part index	PART_IDX	[tablefield: integer] Default: gc_partidx	Field storing the erroneous feature's geometry part number.
Field of ring index	RING_IDX	[tablefield: integer] Default: gc_ringidx	Field storing the erroneous feature's geometry ring number.
Field of vertex index	VERTEX_IDX	[tablefield: integer] Default: gc_vertidx	Field storing the erroneous feature's geometry vertex number.
Small polygons merged layer	OUTPUT	[vector: polygon] Default: [Create temporary layer]	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from merging small polygons	REPORT	[vector: point] Default: [Create temporary layer]	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	TOLERANCE	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from merging small polygons	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the small polygon. • `gc_errory`: the y coordinate of the centroid of the small polygon. • `gc_error`: the area of the small polygon. • `UNIQUE_ID` field: the unique ID of the input feature that is a small polygon. • `report`: a text field describing the fix applied or justifying the failure. • `error_fixed`: a boolean field indicating whether the error was fixed.
Small polygons merged layer	`OUTPUT`	[vector: polygon]	Output layer of polygons with small polygons merged according to the chosen method. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometryarea`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Split self-intersecting geometries

Splits self-intersecting geometries based on an error layer from the *Self-intersections* algorithm.

Note

This algorithm detects invalid polygon geometries (self-intersections). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.

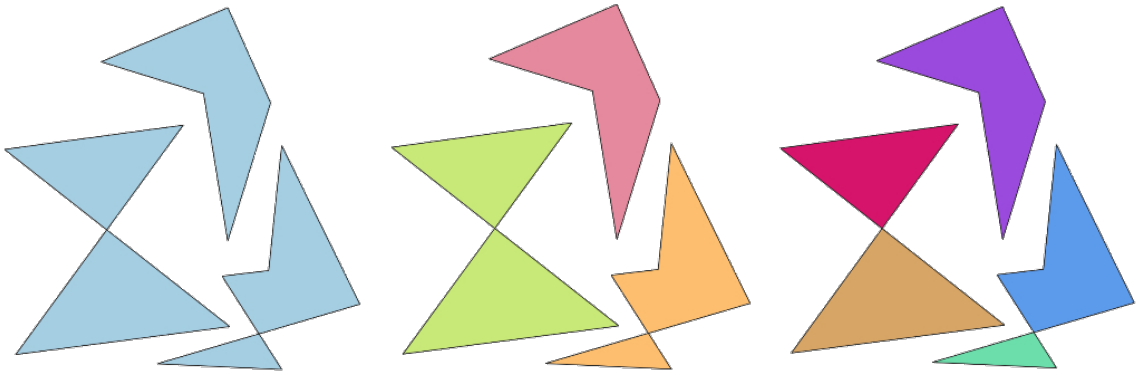


Fig. 24.23: Fixing self-intersection using various methods: as multi-part (middle), and multiple single-part geometries (right).

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to fix, same as input in the check algorithm
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm
Method	`METHOD`	[enumeration]	Method to apply to the self-intersecting geometries: • 0 -- `Split feature into a multi-part feature`: splits the self-intersecting geometry into multiple parts stored in a single multipart feature. • 1 -- `Split feature into multiple single-part features`: splits the self-intersecting geometry into multiple parts, each part stored in a separate feature.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm
Field of part index	`PART_IDX`	[tablefield: integer] Default: `gc_partidx`	Field storing the erroneous feature's geometry part number.
Field of ring index	`RING_IDX`	[tablefield: integer] Default: `gc_ringidx`	Field storing the erroneous feature's geometry ring number.

continues on next page

Table 24.141 – continued from previous page

Label	Name	Type	Description
Field of vertex index	`VERTEX_IDX`	[tablefield: integer] Default: `gc_vertidx`	Field storing the erroneous feature's geometry vertex number.
Field of segment 1	`SEGMENT_1`	[tablefield: integer] Default: `gc_segment_1`	Field storing the number of the first segment involved in the erroneous intersection.
Field of segment 2	`SEGMENT_2`	[tablefield: integer] Default: `gc_segment_2`	Field storing the number of the second segment involved in the erroneous intersection.
Self-intersections fixed layer	`OUTPUT`	[vector: same as input] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing self-intersections	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing self-intersections	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part where the self-intersection occurs. • `gc_ringidx`: the index of the feature's geometry ring where the self-intersection occurs. • `gc_vertidx` • `gc_errorx`: the x coordinate of the self-intersection. • `gc_errory`: the y coordinate of the self-intersection. • `gc_error` • `gc_segment_1`: the index of the first segment involved in the intersection. • `gc_segment_2`: the index of the second segment involved in the intersection. • `UNIQUE_ID` field: the unique ID of the self-intersecting input feature. • `report`: a text field describing the fix applied. • `error_fixed`: a boolean field indicating whether the error was fixed.
Self-intersections fixed layer	`OUTPUT`	[vector: same as input]	Output layer with the geometry fix applied to the input features. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometryselfintersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

24.1.7 GPS

Convert GPS data

Uses the GPSBabel tool to convert a GPS data file from a range of formats to the GPX standard format.

参数

标注	名称	类型	描述
Input file	`INPUT`	[file]	File containing the data to convert
Format	`FORMAT`	[enumeration]	Format of the file to convert, from this list.
Feature type	`FEATURE_TYPE`	[enumeration] Default: 0	The type of data to convert • 0 --- Waypoints • 1 --- Routes • 2 --- Tracks
Output	`OUTPUT`	[vector: any] 默认：[保存到临时文件]	Specification of the output GPX file. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Output layer	`OUTPUT_LAYER`	[vector: any]	Output layer with data in GPX standard format

Python 代码

Algorithm ID: `native:convertgpsdata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Convert GPX feature type

Uses the GPSBabel tool to convert GPX features from one type to another (e.g. converting all waypoint features to a route).

参数

标注	名称	类型	描述
Input file	`INPUT`	[file]	File containing the data to convert
Conversion	`CONVERSION`	[enumeration] Default: 0	The type of conversion to apply • 0 --- Waypoints from a route • 1 --- Waypoints from a track • 2 --- Routes from waypoints • 3 --- Tracks from waypoints
Output	`OUTPUT`	[vector: point or line] 默认: [保存到临时文件]	Specification of the output file. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Output	`OUTPUT`	[vector: any]	Output layer with converted GPX features

Python 代码

Algorithm ID: `native:convertgpxfeaturetype`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Download GPS data from device

Uses the GPSBabel tool to download data from a GPS device into the GPX standard format.

参数

标注	名称	类型	描述
Device	`DEVICE`	[enumeration] Default:*Garmin serial*	The GPS device used to create the data. Must be declared in *GPS Settings* dialog.
Port	`PORT`	[enumeration]	The port the device is connected to. Available ports depend on the OS.

continues on next page

Table 24.148 – continued from previous page

标注	名称	类型	描述
Feature type	FEATURE_TYPE	[enumeration] Default: 0	The type of data to convert • 0 --- Waypoints • 1 --- Routes • 2 --- Tracks
Output	OUTPUT	[vector: any] 默认: [保存到临时文件]	Specification of the output file. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Output layer	OUTPUT_LAYER	[vector: any]	Output layer with data in GPX standard format

Python 代码

Algorithm ID: `native:downloadgpsdata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Upload GPS data to device

Uses the GPSBabel tool to upload data to a GPS device from the GPX standard format.

参数

标注	名称	类型	描述
Input file	INPUT	[file]	.GPX file containing the data to upload
Device	DEVICE	[enumeration] Default:*Garmin serial*	The GPS device you would like to upload the data to. Must be declared in *GPS Settings* dialog.
Port	PORT	[enumeration]	The port the device is connected to. Available ports depend on the OS.
Feature type	FEATURE_TYPE	[enumeration] Default: 0	The type of data to upload • 0 --- Waypoints • 1 --- Routes • 2 --- Tracks

输出

No output is provided. If successful, data are loaded to the device.

Python 代码

Algorithm ID: `native:uploadgpsdata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.8 Interpolation

Heatmap (kernel density estimation)

Creates a density (heatmap) raster of an input point vector layer using kernel density estimation.

The density is calculated based on the number of points in a location, with larger numbers of clustered points resulting in larger values. Heatmaps allow easy identification of *hotspots* and clustering of points.

参数

标注	名称	类型	描述
Point layer	`INPUT`	[vector: point]	Point vector layer to use for the heatmap
Radius	`RADIUS`	[numeric: double] Default: 100.0	Heatmap search radius (or kernel bandwidth) in map units. The radius specifies the distance around a point at which the influence of the point will be felt. Larger values result in greater smoothing, but smaller values may show finer details and variation in point density.
Output raster size	`PIXEL_SIZE`	[numeric: double] Default: 0.1	Pixel size of the output raster layer in layer units. In the GUI, the size can be specified by the number of rows (`Number of rows`) / columns (`Number of columns`) **or** the pixel size(`Pixel Size X` / `Pixel Size Y`). Increasing the number of rows or columns will decrease the cell size and increase the file size of the output raster. The values in `Rows`, `Columns`, `Pixel Size X` and `Pixel Size Y` will be updated simultaneously - doubling the number of rows will double the number of columns, and the cell size will be halved. The extent of the output raster will remain the same (approximately).
Radius from field 可选	`RADIUS_FIELD`	[tablefield: numeric]	Sets the search radius for each feature from an attribute field in the input layer.

continues on next page

Table 24.151 – continued from previous page

标注	名称	类型	描述
Weight from field 可选	`WEIGHT_FIELD`	[tablefield: numeric]	Allows input features to be weighted by an attribute field. This can be used to increase the influence certain features have on the resultant heatmap.
Kernel shape	`KERNEL`	[enumeration] Default: *0*	Controls the rate at which the influence of a point decreases as the distance from the point increases. Different kernels decay at different rates, so a triweight kernel gives features greater weight for distances closer to the point then the Epanechnikov kernel does. Consequently, triweight results in "sharper" hotspots and Epanechnikov results in "smoother" hotspots. There are many shapes available (please see the Wikipedia page for further information): • 0 --- Quartic • 1 --- Triangular • 2 --- Uniform • 3 --- Triweight • 4 --- Epanechnikov
Decay ratio (Triangular kernels only) 可选	`DECAY`	[numeric: double] Default: *0.0*	Can be used with Triangular kernels to further control how heat from a feature decreases with distance from the feature. • A value of 0 (=minimum) indicates that the heat will be concentrated in the center of the given radius and completely extinguished at the edge. • A value of 0.5 indicates that pixels at the edge of the radius will be given half the heat as pixels at the center of the search radius. • A value of 1 means the heat is spread evenly over the whole search radius circle. (This is equivalent to the 'Uniform' kernel.) • A value greater than 1 indicates that the heat is higher towards the edge of the search radius than at the center.
Output value scaling	`OUTPUT_VALUE`	[enumeration] Default: *0*	Allow to change the values of the output heatmap raster. One of: • 0 --- Raw • 1 --- Scaled
Heatmap	`OUTPUT`	[栅格] 默认: [保存到临时文件]	Specify the output raster layer with kernel density values. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Heatmap	`OUTPUT`	[栅格]	Raster layer with kernel density values

Example: Creating a Heatmap

For the following example, we will use the `airports` vector point layer from the QGIS sample dataset (see 下载示例数据). Another excellent QGIS tutorial on making heatmaps can be found at http://qgistutorials.com.

In Fig. 24.24, the airports of Alaska are shown.

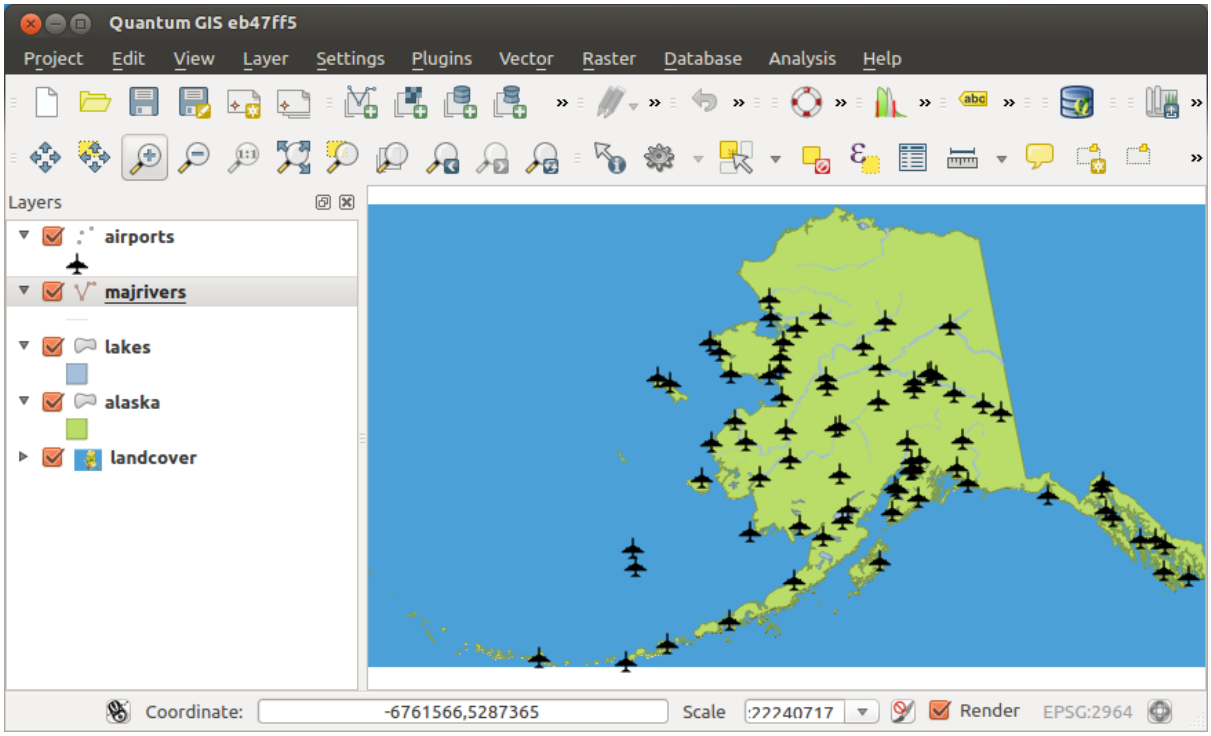


Fig. 24.24: Airports of Alaska

1. Open the *Heatmap (Kernel Density Estimation)* algorithm from the QGIS *Interpolation* group
2. In the *Point layer* field, select `airports` from the list of point layers loaded in the current project.
3. Change the *Radius* to `1000000` meters.
4. Change the *Pixel size X* to `1000`. The *Pixel size Y*, *Rows* and *Columns* will be automatically updated.
5. Click on *Run* to create and load the airports heatmap (see Fig. 24.26).

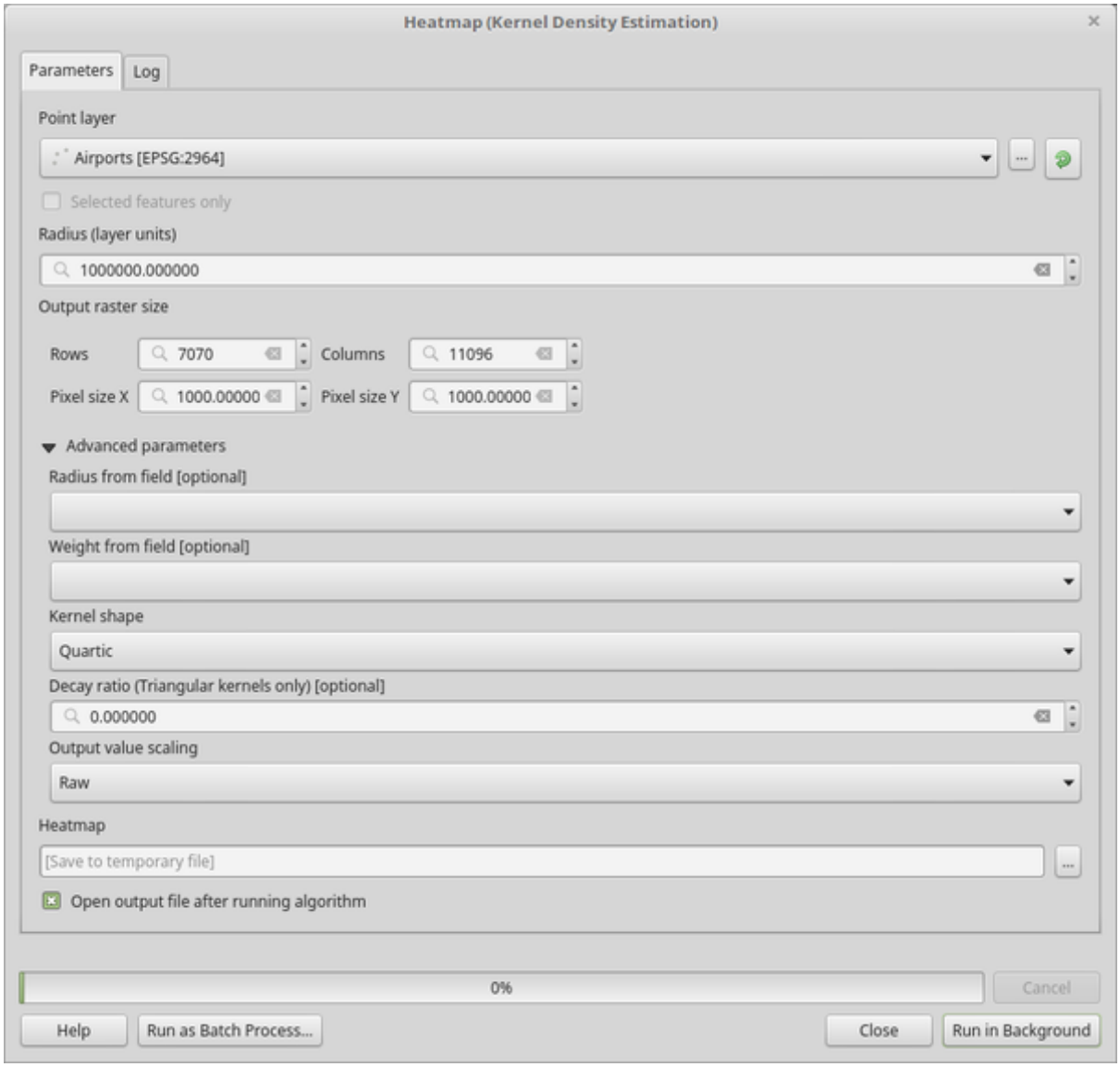


Fig. 24.25: The Heatmap Dialog

QGIS will generate the heatmap and add it to your map window. By default, the heatmap is shaded in greyscale, with lighter areas showing higher concentrations of airports. The heatmap can now be styled in QGIS to improve its appearance.

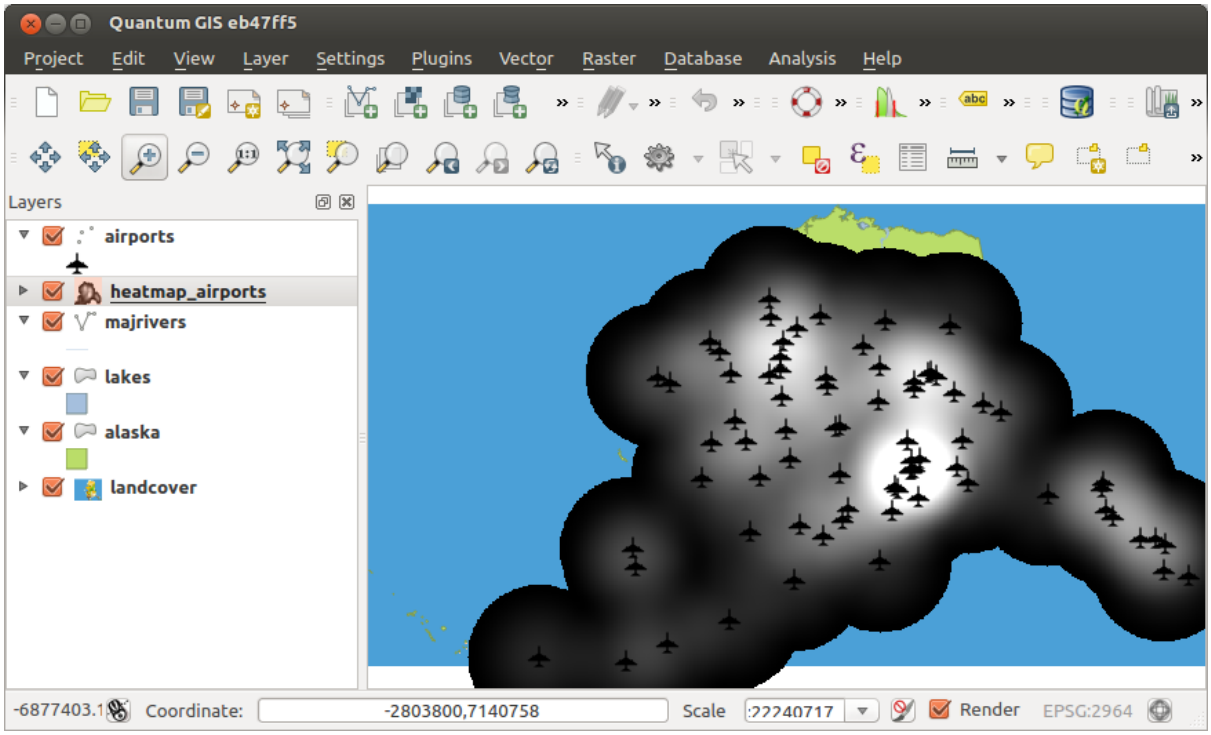


Fig. 24.26: The heatmap after loading looks like a grey surface

1. Open the properties dialog of the `heatmap_airports` layer (select the layer `heatmap_airports`, open the context menu with the right mouse button and select *Properties*).
2. Select the *Symbology* tab.
3. Change the *Render type* to 'Singleband pseudocolor'.
4. Select a suitable *Color ramp* , for instance `YlOrRd`.
5. Click the *Classify* button.
6. Press *OK* to update the layer.

The final result is shown in Fig. 24.27.

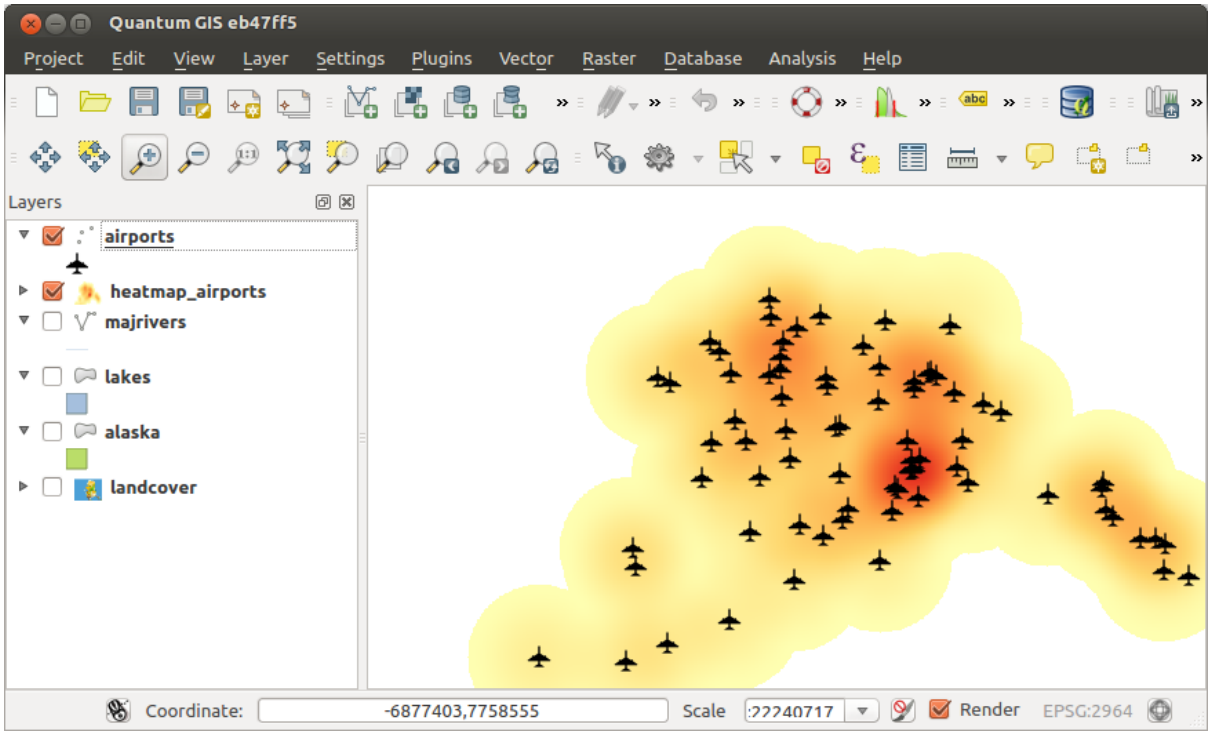


Fig. 24.27: Styled heatmap of airports of Alaska

Python 代码

Algorithm ID: `qgis:heatmapkerneldensityestimation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

IDW Interpolation

Generates an Inverse Distance Weighted (IDW) interpolation of a point vector layer.

Sample points are weighted during interpolation such that the influence of one point relative to another declines with distance from the unknown point you want to create.

The IDW interpolation method also has some disadvantages: the quality of the interpolation result can decrease, if the distribution of sample data points is uneven.

Furthermore, maximum and minimum values in the interpolated surface can only occur at sample data points.

参数

标注	名称	类型	描述
Input layer(s)	`INTERPOLATION_`	[string]	Vector layer(s) and field(s) to use for the interpolation, coded in a string (see the `ParameterInterpolationData` class in InterpolationWidgets for more details). The following GUI elements are provided to compose the interpolation data string: • **Vector layer** [vector: any] • **Interpolation attribute** [tablefield: numeric]: Attribute to use in the interpolation • **Use Z-coordinate for interpolation** [boolean]: Uses the layer's stored Z values (Default: False) For each of the added layer-field combinations, a type can be chosen: • *Points*: Elevation sample locations containing the values to be interpolated (e.g., terrain height, Z). These points are the primary input for calculating the output surface. • *Structured lines*: Lines representing known continuous surface features that guide the terrain shape or structure (e.g., ridges, road centerlines, levees). They describe terrain direction while assuming continuity along the line. • *Break lines*: Lines representing known surface discontinuities or hard boundaries (e.g., cliffs, river banks, excavation edges, lake/pond outlines). They mark where the surface may change sharply. It is required if there are significant changes in elevation along the line. In the string, the layer-field elements are separated by `'::|::'`. The sub-elements of the layer-field elements are separated by `'::~::'`.
Distance coefficient P	`DISTANCE_COEFF`	[numeric: double] Default: 2.0	Sets the distance coefficient for the interpolation. Minimum: 0.0, maximum: 100.0.

continues on next page

Table 24.153 – continued from previous page

标注	名称	类型	描述
Extent (xmin, xmax, ymin, ymax)	EXTENT	[范围]	Extent of the output raster layer. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Output raster size	PIXEL_SIZE	[numeric: double] Default: 0.1	Pixel size of the output raster layer in layer units. In the GUI, the size can be specified by the number of rows (`Number of rows`) / columns (`Number of columns`) **or** the pixel size(`Pixel Size X` / `Pixel Size Y`). Increasing the number of rows or columns will decrease the cell size and increase the file size of the output raster. The values in `Rows`, `Columns`, `Pixel Size X` and `Pixel Size Y` will be updated simultaneously - doubling the number of rows will double the number of columns, and the cell size will be halved. The extent of the output raster will remain the same (approximately).
Interpolated	OUTPUT	[栅格] 默认: [保存到临时文件]	Raster layer of interpolated values. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Interpolated	OUTPUT	[栅格]	Raster layer of interpolated values

Python 代码

Algorithm ID: `qgis:idwinterpolation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Line Density

Calculates for each raster cell, the density measure of linear features within a circular neighbourhood. This measure is obtained by summing all the line segments intersecting the circular neighbourhood and dividing this sum by the area of such neighbourhood. A weighting factor can be applied to the line segments.

> **Note**
>
> This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

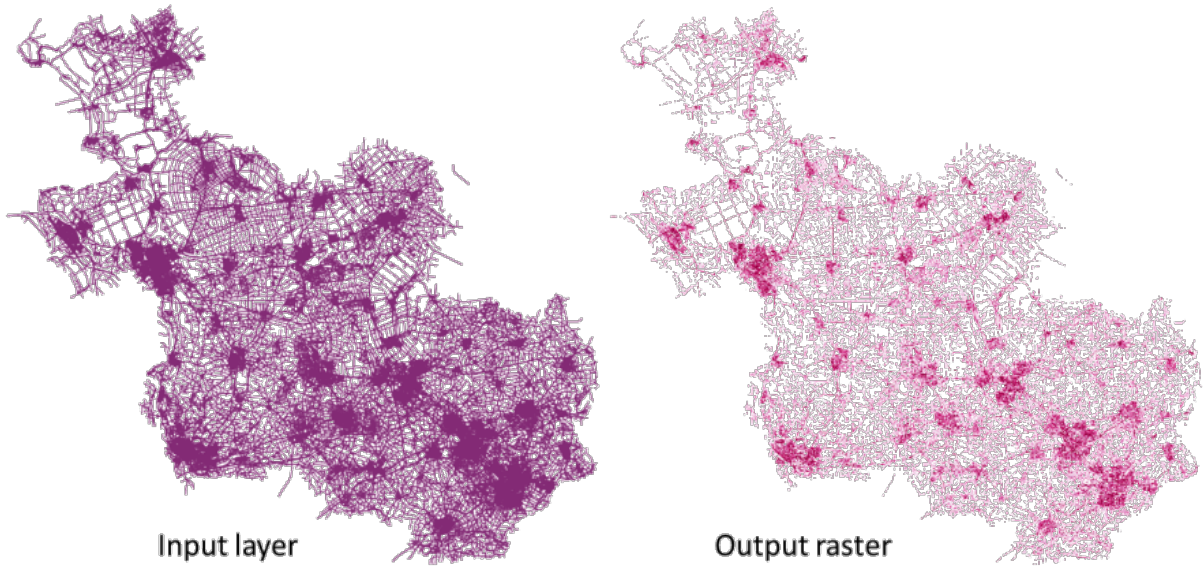


Fig. 24.28: Line density example. Input layer source: Roads Overijssel - The Netherlands (OSM).

参数

Basic parameters

标注	名称	类型	描述
Input line layer	`INPUT`	[vector: line]	Input vector layer containing line features
Weight field	`WEIGHT`	[tablefield: numeric]	Field of the layer containing the weight factor to use during the calculation
Search Radius	`RADIUS`	[numeric: double] Default: 10.0	Radius of the circular neighbourhood. Units can be specified here.

continues on next page

Table 24.155 – continued from previous page

标注	名称	类型	描述
Pixel size	PIXEL_SIZE	[numeric: double] Default: 10.0	Pixel size of the output raster layer in layer units. The raster has square pixels.
Line density raster	OUTPUT	[栅格] 默认: [保存到临时文件]	The output as a raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Line density raster	OUTPUT	[栅格]	The output line density raster layer.

Python 代码

Algorithm ID: `native:linedensity`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

TIN Interpolation

Generates a Triangulated Irregular Network (TIN) interpolation of a point vector layer.

With the TIN method you can create a surface formed by triangles of nearest neighbor points. To do this, circumcircles around selected sample points are created and their intersections are connected to a network of non overlapping and as compact as possible triangles. The resulting surfaces are not smooth.

The algorithm creates both the raster layer of the interpolated values and the vector line layer with the triangulation boundaries.

参数

标注	名称	类型	描述
Input layer(s)	`INTERPOLATION_`	[string]	Vector layer(s) and field(s) to use for the interpolation, coded in a string (see the `ParameterInterpolationData` class in InterpolationWidgets for more details). The following GUI elements are provided to compose the interpolation data string: • **Vector layer** [vector: any] • **Interpolation attribute** [tablefield: numeric]: Attribute to use in the interpolation • **Use Z-coordinate for interpolation** [boolean]: Uses the layer's stored Z values (Default: False) For each of the added layer-field combinations, a type can be chosen: • *Points*: Elevation sample locations containing the values to be interpolated (e.g., terrain height, Z). These points are the primary input for calculating the output surface. • *Structured lines*: Lines representing known continuous surface features that guide the terrain shape or structure (e.g., ridges, road centerlines, levees). They describe terrain direction while assuming continuity along the line. • *Break lines*: Lines representing known surface discontinuities or hard boundaries (e.g., cliffs, river banks, excavation edges, lake/pond outlines). They mark where the surface may change sharply. It is required if there are significant changes in elevation along the line. In the string, the layer-field elements are separated by `'::|::'`. The sub-elements of the layer-field elements are separated by `'::~::'`.
Interpolation method	`METHOD`	[enumeration] Default: 0	Set the interpolation method to be used. One of: • *Linear* • *Clough-Toucher (cubic)*

continues on next page

Table 24.158 – continued from previous page

标注	名称	类型	描述
Extent (xmin, xmax, ymin, ymax)	EXTENT	[范围]	Extent of the output raster layer. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Output raster size	PIXEL_SIZE	[numeric: double] Default: 0.1	Pixel size of the output raster layer in layer units. In the GUI, the size can be specified by the number of rows (`Number of rows`) / columns (`Number of columns`) **or** the pixel size(`Pixel Size X` / `Pixel Size Y`). Increasing the number of rows or columns will decrease the cell size and increase the file size of the output raster. The values in `Rows`, `Columns`, `Pixel Size X` and `Pixel Size Y` will be updated simultaneously - doubling the number of rows will double the number of columns, and the cell size will be halved. The extent of the output raster will remain the same (approximately).
Interpolated	OUTPUT	[栅格] 默认: [保存到临时文件]	The output TIN interpolation as a raster layer. *One of*: • 保存到临时文件 • 保存到文件…
Triangulation	TRIANGULATION	[vector: line] Default: [Skip output]	The output TIN as a vector layer. *One of*: • Skip Output • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Interpolated	`OUTPUT`	[栅格]	The output TIN interpolation as a raster layer
Triangulation	`TRIANGULATION`	[vector: line]	The output TIN as a vector layer.

Python 代码

Algorithm ID: `qgis:tininterpolation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.9 Layer tools

Export layer(s) information

Creates a polygon layer with features corresponding to the extent of selected layer(s).

Additional layer details (CRS, provider name, file path, layer name, subset filter, abstract and attribution) are attached as attributes to each feature.

参数

标注	名称	类型	描述
Input layers	`LAYERS`	[layer] [list]	Input layers to get information on.
Output	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output layer with information. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Output	OUTPUT	[vector: polygon]	Polygon vector layer showing extent of input layers and associated information in attributes.

Python 代码

Algorithm ID: native:exportlayersinformation

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Export to spreadsheet

Exports the attributes of a selection of vector layers into a spreadsheet document or optionally appends them to an existing spreadsheet as additional sheets.

参数

标注	名称	类型	描述
Input layers	LAYERS	[vector: any] [list]	Input vector layers. The output spreadsheet will consist of a sheet, for each layer, that contains the attributes of this layer.
Use field aliases as column headings	USE_ALIAS	[boolean] Default: False	Use the field aliases from the attribute table for the spreadsheet.
Export formatted values instead of raw values	FORMATTED_VALU	[boolean] Default: False	If True, exports the formatted, human readable values (e.g., from a *value map or value relation*) to the spreadsheet.
Overwrite existing spreadsheet	OVERWRITE	[boolean] Default: True	If the specified spreadsheet exists, setting this option to True will overwrite the existing spreadsheet. If this option is False and the spreadsheet exists, the layers will be appended as additional sheets.
Destination spreadsheet	OUTPUT	[file] 默认：[保存到临时文件]	Output spreadsheet with a sheet for every layer. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Destination spreadsheet	`OUTPUT`	[file]	Spreadsheet with a sheet for every layer.
Layers within spreadsheet	`OUTPUT_LAYERS`	[list]	The list of sheets added to the spreadsheet.

Python 代码

Algorithm ID: `native:exporttospreadsheet`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Extract layer extent

Generates a vector layer with the minimum bounding box (rectangle with N-S orientation) that covers all the input features.

The output layer contains a single bounding box for the whole input layer.

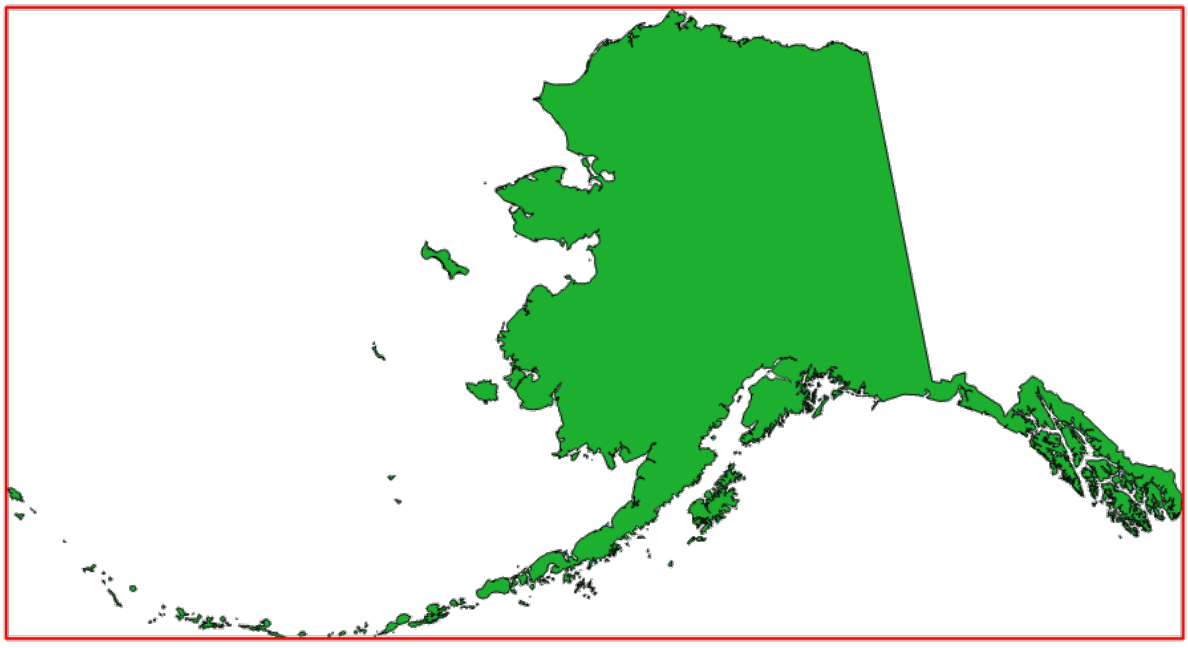


Fig. 24.29: In red the bounding box of the source layer

Default menu: *Vector ► Research Tools*

参数

标注	名称	类型	描述
Layer	`INPUT`	[layer]	Input layer
Extent	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the polygon vector layer for the output extent. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Extent	`OUTPUT`	[vector: polygon]	Output (polygon) vector layer with the extent (minimum bounding box)

Python 代码

Algorithm ID: `native:polygonfromlayerextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.10 网孔

Export contours

Creates contours as a vector layer from a mesh scalar dataset.

参数

标注	名称	类型	描述
Input mesh layer	`INPUT`	[mesh]	The mesh layer to export data from
Dataset groups	`DATASET_GROUPS`	[layer] [list]	The dataset groups
Dataset time	`DATASET_TIME`	[datetime]	The time range to take into account • 0 --- Current canvas time • 1 --- Defined date/time • 2 --- Dataset group time step

continues on next page

Table 24.166 – continued from previous page

标注	名称	类型	描述
Increment between contour levels 可选	INCREMENT	[numeric: double] Default: Not set	Interval between generated levels.
Minimum contour level 可选	MINIMUM	[numeric: double] Default: Not set	Starting level values of contours. No generated levels will be lower than this value.
Maximum contour level 可选	MAXIMUM	[numeric: double] Default: Not set	Maximum values of contours, i.e. no generated levels will be greater than this value.
List of contours level 可选	CONTOUR_LEVEL_	[string] Default: Not set	List of desired levels of contours (separated by commas). If filled, the increment, minimum, and maximum fields will not be considered.
Output coordinate system 可选	CRS_OUTPUT	[crs]	Coordinate Reference System to assign to the output
Exported contour lines	OUTPUT_LINES	[vector: line] Default: [Create temporary layer]	Specify the output line layer representing the contours of the mesh layer. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。
Exported contour polygons	OUTPUT_POLYGON	[vector: polygon] Default: [Create temporary layer]	Specify the output polygon layer representing the contours of the mesh layer. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Exported contour lines	OUTPUT_LINES	[vector: line]	Line layer representing the contours of the mesh layer.
Exported contour polygons	OUTPUT_POLYGON	[vector: polygon]	Polygon layer representing the contours of the mesh layer.

Python 代码

Algorithm ID: `native:meshcontours`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Export cross section dataset values on lines from mesh

Extracts a mesh dataset's values from lines contained in a vector layer.

Each line is discretized with a resolution distance parameter for extraction of values on its vertices.

参数

标注	名称	类型	描述
Input mesh layer	`INPUT`	[mesh]	The mesh layer to export data from
Dataset groups	`DATASET_GROUPS`	[layer] [list]	The dataset groups
Dataset time	`DATASET_TIME`	[datetime]	The time range to take into account • 0 --- Current canvas time • 1 --- Defined date/time • 2 --- Dataset group time step
Lines for data export	`INPUT_LINES`	[vector: line]	Lines where the data will be extracted from the dataset mesh
Line segmentation resolution	`RESOLUTION`	[numeric: double] Default: 10.0	The distance between points on the lines where the data will be extracted from the dataset mesh. Units can be selected.
Digits count for coordinates	`COORDINATES_DI`	[numeric: integer] Default: 2	Number of digits for rounding the coordinate values
Digits count for dataset value	`DATASET_DIGITS`	[numeric: integer] Default: 2	Number of digits for rounding the dataset values
Exported data CSV file	`OUTPUT`	[file] 默认：[保存到临时文件]	Specification of the output file. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Exported data CSV file	`OUTPUT`	[file]	

Python 代码

Algorithm ID: `native:meshexportcrosssection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Export mesh edges

Exports a mesh layer's edges to a line vector layer, with the dataset values on edges as attribute values.

参数

标注	名称	类型	描述
Input mesh layer	`INPUT`	[mesh]	The mesh layer to export data from
Dataset groups	`DATASET_GROUPS`	[layer] [list]	The dataset groups
Dataset time	`DATASET_TIME`	[datetime]	The time range to take into account • 0 --- Current canvas time • 1 --- Defined date/time • 2 --- Dataset group time step
Output coordinate system 可选	`CRS_OUTPUT`	[crs]	Coordinate Reference System to assign to the output
Export vector option	`VECTOR_OPTION`	[enumeration]	Coordinate type of vector value exportation. • 0 --- Cartesian (x,y) • 1 --- Polar (magnitude, degree) • 2 --- Cartesian and polar
Output vector layer	`OUTPUT`	[vector: line] Default: `[Create temporary layer]`	Specification of the output file. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Output vector layer	`OUTPUT`	[vector: line]	Output vector line layer containing the edges of the input mesh layer with associated dataset values

Python 代码

Algorithm ID: `native:exportmeshedges`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Export mesh faces

Exports a mesh layer's faces to a polygon vector layer, with the dataset values on faces as attribute values.

参数

标注	名称	类型	描述
Input mesh layer	`INPUT`	[mesh]	The mesh layer to export data from
Dataset groups	`DATASET_GROUPS`	[layer] [list]	The dataset groups
Dataset time	`DATASET_TIME`	[datetime]	The time range to take into account • 0 --- Current canvas time • 1 --- Defined date/time • 2 --- Dataset group time step
Output coordinate system 可选	`CRS_OUTPUT`	[crs]	Coordinate Reference System to assign to the output
Export vector option	`VECTOR_OPTION`	[enumeration]	Coordinate type of vector value exportation. • 0 --- Cartesian (x,y) • 1 --- Polar (magnitude, degree) • 2 --- Cartesian and polar
Output vector layer	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output file. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Output vector layer	`OUTPUT`	[vector: polygon]	Output vector polygon layer containing the faces of the input mesh layer with associated dataset values

Python 代码

Algorithm ID: `native:exportmeshfaces`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Export mesh on grid

Exports a mesh layer's dataset values to a gridded point vector layer, with the dataset values on this point as attribute values.

For data on volume (3D stacked dataset values), the exported dataset values are averaged on faces using the method defined in *the mesh layer properties* (default is Multi level averaging method). 1D meshes are not supported.

参数

标注	名称	类型	描述
Input mesh layer	`INPUT`	[mesh]	The mesh layer to export data from
Dataset groups	`DATASET_GROUPS`	[layer] [list]	The dataset groups
Dataset time	`DATASET_TIME`	[datetime]	The time range to take into account • 0 --- Current canvas time • 1 --- Defined date/time • 2 --- Dataset group time step
Extent 可选	`EXTENT`	[范围]	Specify the spatial extent on which to process the data. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Grid spacing 可选	`GRID_SPACING`	[numeric: double] Default: 10.0	Spacing between the sample points to use
Output coordinate system 可选	`CRS_OUTPUT`	[crs]	Coordinate Reference System to assign to the output

continues on next page

Table 24.174 – continued from previous page

标注	名称	类型	描述
Export vector option	VECTOR_OPTION	[enumeration]	Coordinate type of vector value exportation. • 0 --- Cartesian (x,y) • 1 --- Polar (magnitude, degree) • 2 --- Cartesian and polar
Output vector layer	OUTPUT	[vector: point] Default: [Create temporary layer]	Specification of the output file. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Output vector layer	OUTPUT	[vector: point]	Output vector point layer with dataset values computed from the overlaid face.

Python 代码

Algorithm ID: native:exportmeshongrid

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Export mesh vertices

Exports a mesh layer's vertices to a point vector layer, with the dataset values on vertices as attribute values.

参数

标注	名称	类型	描述
Input mesh layer	INPUT	[mesh]	The mesh layer to export data from
Dataset groups	DATASET_GROUPS	[layer] [list]	The dataset groups
Dataset time	DATASET_TIME	[datetime]	The time range to take into account • 0 --- Current canvas time • 1 --- Defined date/time • 2 --- Dataset group time step
Output coordinate system 可选	CRS_OUTPUT	[crs]	Coordinate Reference System to assign to the output

continues on next page

Table 24.176 – continued from previous page

标注	名称	类型	描述
Export vector option	VECTOR_OPTION	[enumeration]	Coordinate type of vector value exportation. • 0 --- Cartesian (x,y) • 1 --- Polar (magnitude, degree) • 2 --- Cartesian and polar
Output vector layer	OUTPUT	[vector: point] Default: [Create temporary layer]	Specification of the output file. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Output vector layer	OUTPUT	[vector: point]	Output vector point layer containing the vertices of the input mesh layer with associated dataset values

Python 代码

Algorithm ID: native:exportmeshvertices

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Export time series values from points of a mesh dataset

Extracts a mesh dataset's time series values from points contained in a vector layer.

If the time step is kept to its default value (0 hours), the time step used is the one of the two first datasets of the first selected dataset group.

参数

标注	名称	类型	描述
Input mesh layer	INPUT	[mesh]	The mesh layer to extract data from
Dataset groups	DATASET_GROUPS	[layer] [list]	The dataset groups

continues on next page

Table 24.178 – continued from previous page

标注	名称	类型	描述
Starting time	STARTING_TIME	[datetime]	The start of the time range to take into account • 0 --- Current canvas time • 1 --- Defined date/time • 2 --- Dataset group time step
Finishing time	FINISHING_TIME	[datetime]	The end of the time range to take into account • 0 --- Current canvas time • 1 --- Defined date/time • 2 --- Dataset group time step
Time step (hours) 可选	TIME_STEP	[numeric: double] Default: 0	Time between two consecutive steps to extract. Keep 0 to use time step of the first selected dataset group.
Points for data export	INPUT_POINTS	[vector: point]	Vector layer containing points where the data will be extracted from the dataset mesh
Digits count for coordinates	COORDINATES_DI	[numeric: integer] Number of digits for rounding coordinate values Default: 2	
Digits count for dataset value	DATASET_DIGITS	[numeric: integer] Default: 2	Number of digits for rounding dataset values
Exported data CSV file	OUTPUT	[file] 默认：[保存到临时文件]	Specification of the output file. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Exported data CSV file	OUTPUT	[file]	.CSV file containing the mesh dataset time series values at the overlaying point features

Python 代码

Algorithm ID: native:meshexporttimeseries

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Rasterize mesh dataset

Creates a raster layer from a mesh dataset.

For data on volume (3D stacked dataset values), the exported dataset values are averaged on faces using the method defined in *the mesh layer properties* (default is Multi level averaging method). 1D meshes are not supported.

参数

Basic parameters

标注	名称	类型	描述
Input mesh layer	`INPUT`	[mesh]	The mesh layer to export data from
Dataset groups	`DATASET_GROUPS`	[layer] [list]	The dataset groups
Dataset time	`DATASET_TIME`	[datetime]	The time range to take into account • 0 --- Current canvas time • 1 --- Defined date/time • 2 --- Dataset group time step
Extent 可选	`EXTENT`	[范围]	Specify the spatial extent on which to process the data. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Pixel size	`PIXEL_SIZE`	[numeric: double] 默认值：1.0	Pixel size of the output raster layer.
Output coordinate system 可选	`CRS_OUTPUT`	[crs]	Coordinate Reference System to assign to the output
Output raster layer	`OUTPUT`	[栅格] 默认：[保存到临时文件]	Specification of the output file. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output raster layer	OUTPUT	[栅格]	Output raster layer with dataset values computed from the mesh layer.

Python 代码

Algorithm ID: `native:meshrasterize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Surface to Polygon

Exports a polygon file containing the boundary of a mesh layer. The resulting polygon may contain holes and may be a multi-part polygon.

参数

标注	名称	类型	描述
Input mesh layer	INPUT	[mesh]	The input mesh layer whose boundary will be exported as a polygon
Output coordinate system 可选	OUTPUT_CRS	[crs]	The coordinate reference system (CRS) for the output polygon layer. If not specified, the CRS of the input mesh layer will be used.

continues on next page

Table 24.183 – continued from previous page

标注	名称	类型	描述
Output vector layer	OUTPUT	[vector: polygon] Default: [Create temporary layer]	Specification of the output file. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Output vector layer	OUTPUT	[vector: polygon]	The resulting polygon layer containing the boundary of the mesh layer. The polygon may contain holes and may be multi-part.

Python 代码

Algorithm ID: native:surfacetopolygon

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

TIN mesh creation

Creates a TIN mesh layer from vector layers. The TIN mesh is created using a Delaunay triangulation.

参数

标注	名称	类型	描述
Input layers	SOURCE_DATA	[vector: any] [list]	Vector layers to combine to generate the mesh layer
Vector layer	GUI ONLY	[vector: any] [list]	A selector for the vector layers to combine to generate the mesh layer
Value on vertex	GUI ONLY	[tablefield: any]	A selector of the field to use from the selected layer. Each vertex is assigned the corresponding value of its original feature.
Use Z-coordinate for value on vertex	GUI ONLY	[boolean] Default: False	If checked, the Z value of vector layer points or polygons/lines vertices will be used to assign the Z value of the vertex mesh layer. Only available if the input layers are 3D.

continues on next page

Table 24.185 – continued from previous page

标注	名称	类型	描述
Output format	MESH_FORMAT	[enumeration] Default: 2DM	Output format of the generated layer • 0 --- 2DM • 1 --- SELAFIN • 2 --- PLY • 3 --- Ugrid • 4 --- Mike21
Output coordinate system 可选	CRS_OUTPUT	[crs]	Coordinate Reference System to assign to the output
Output file	OUTPUT_MESH	[mesh] 默认：[保存到临时文件]	Specification of the output file. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Output file	OUTPUT_MESH	[mesh]	Output mesh layer with dataset values computed from the vector layers.

Python 代码

Algorithm ID: native:tinmeshcreation

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.11 Metadata tools

Add history metadata

Adds a new history entry to the layer's metadata.

Parameters

Label	Name	Type	Description
Layer	INPUT	[layer]	The input layer (vector, raster, etc.) to which the history entry will be added.
History entry	HISTORY	[string]	The text to be appended as a new entry in the layer's history metadata. This will be added to any existing history entries.

Outputs

Label	Name	Type	Description
Output	OUTPUT	[same as input]	The resulting layer with the updated history in its **Metadata properties**.

Python code

Algorithm ID: native:addhistorymetadata

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Copy Layer Metadata

Copies metadata from a source layer to a target layer. Any existing metadata in the target layer will be replaced with the metadata from the source layer. This includes all metadata fields, such as history, abstract, and other properties.

Parameters

Label	Name	Type	Description
Source layer	SOURCE	[layer]	The layer from which metadata will be copied.
Target layer	TARGET	[layer]	The layer to which metadata will be pasted. Any existing metadata in this layer will be replaced.
Save metadata as default	DEFAULT	[boolean] Default: False	If checked, the metadata information will be saved with the layer, hence available by default in subsequent projects.

Outputs

Label	Name	Type	Description
Output layer	OUTPUT	[layer]	The target layer with the metadata replaced by the metadata from the source layer. This includes all metadata fields, such as history, abstract, and other properties.

Python code

Algorithm ID: `native:copylayermetadata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Export Layer Metadata

Exports the metadata of a layer to a QMD file.

Parameters

Label	Name	Type	Description
Layer	`INPUT`	[layer]	The layer whose metadata will be exported.
Output	`OUTPUT`	[file] Default: `[Save to temporary file]`	Specification of the output file. *One of*: • Save to a Temporary File • Save to File…

Outputs

Label	Name	Type	Description
Output file	`OUTPUT`	[file]	The `.qmd` file containing the exported metadata. This file can be used to import metadata into another layer.

Python code

Algorithm ID: `native:exportlayermetadata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Set Layer Metadata

Applies metadata to a layer from a `.qmd` file.

Parameters

Label	Name	Type	Description
Layer	`INPUT`	[layer]	The layer to which the metadata will be applied.
Metadata file	`METADATA`	[file]	The `.qmd` file containing the metadata to be applied.
Save metadata as default	`DEFAULT`	[boolean] Default: False	If checked, the metadata information will be saved with the layer, hence available by default in subsequent projects.

Outputs

Label	Name	Type	Description
Output layer	`OUTPUT`	[same as input]	The input layer with the metadata replaced by the metadata from the `.qmd` file.

Python code

Algorithm ID: `native:setlayermetadata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Set Metadata Fields

Sets various metadata fields for a layer.

Parameters

Label	Name	Type	Description
Layer	`INPUT`	[layer]	The layer whose metadata fields will be updated
Identifier Optional	`IDENTIFIER`	[string]	A unique identifier for the layer
Parent identifier Optional	`PARENT_IDENTIF`	[string]	The identifier of the parent resource, if applicable
Title Optional	`TITLE`	[string]	The title of the layer

continues on next page

Table 24.195 – continued from previous page

Label	Name	Type	Description
Type Optional	TYPE	[string]	The type of data stored in the layer
Language Optional	LANGUAGE	[string]	The language of the metadata
Encoding Optional	ENCODING	[string]	The character encoding used in the metadata
Abstract Optional	ABSTRACT	[string]	A brief description or abstract of the layer
Coordinate reference system Optional	CRS	[crs]	The coordinate reference system of the layer
Fees Optional	FEES	[string]	Information about any fees associated with accessing the layer
Ignore empty fields	IGNORE_EMPTY	[boolean] Default: False	If checked, no update will be done to metadata fields that are not filled

Outputs

Label	Name	Type	Description
Output layer	OUTPUT	[same as input]	The input layer with the specified metadata fields updated.

Python code

Algorithm ID: `native:setmetadatafields`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

Update Layer Metadata

Copies all non-empty metadata fields from a source layer to a target layer. Leaves empty input fields unchanged in the target.

Parameters

Label	Name	Type	Description
Source layer	SOURCE	[layer]	The layer from which non-empty metadata fields will be copied
Target layer	TARGET	[layer]	The layer to which non-empty metadata fields will be pasted. Empty fields in the source layer will not overwrite existing metadata in the target layer.

Outputs

Label	Name	Type	Description
Target layer	`OUTPUT`	[layer]	The target layer with updated metadata.

Python code

Algorithm ID: `native:updatelayermetadata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See 从控制台使用处理算法 for details on how to run processing algorithms from the Python console.

24.1.12 Modeler tools

> **Warning**
>
> These tools are only available in the model designer. They are not available in the Processing Toolbox.

Calculate expression

It calculates the result of a QGIS expression and eliminates the need to use the same expression multiple times throughout a model when the same result is required more than once. Additionally, it enables use cases that would otherwise not be possible. For instance, you can generate a timestamp value once and use it multiple times within the model, if the timestamp were recalculated every time, the values would vary during the model's runtime.

参数

标注	名称	类型	描述
Input	`INPUT`	[expression]	Expression to calculate

输出

标注	名称	类型	描述
Value	`OUTPUT`	[Result Value]	Calculated result value, the data type of the output will vary based on the specific expression used in the algorithm.

Python 代码

Algorithm ID: `native:calculateexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Conditional branch

Adds a conditional branch into a model, allowing parts of the model to be executed based on the result of an expression evaluation. Mostly by using tool dependencies to control the flow of a model.

参数

标注	名称	类型	描述
字段	`BRANCH`	[string]	Name of the condition
字段	`CONDITION`	[expression]	Expression to evaluate

输出

None.

Python 代码

Algorithm ID: `native:condition`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Create directory

Creates a new directory on a file system. Directories will be created recursively, creating all required parent directories in order to construct the full specified directory path. No errors will be raised if the directory already exists.

参数

标注	名称	类型	描述
Directory path	`PATH`	[string]	Folder path to create

输出

标注	名称	类型	描述
Output	`OUTPUT`	[folder]	Created folder

Python 代码

Algorithm ID: `native:createdirectory`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Feature filter

Filters features from the input layer and redirects them to one or several outputs. If you do not know about any attribute names that are common to all possible input layers, filtering is only possible on the feature geometry and general record mechanisms, such as `$id` and `uuid`.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	The input layer.
Outputs and filters (one or more)	`OUTPUT_<name of the filter>`	[same as input]	The output layers with filters (as many as there are filters).

输出

标注	名称	类型	描述
Output (one or more)	`native:filter_ of filter>`	[same as input]	The output layers with filtered features (as many as there are filters).

Python 代码

Algorithm ID: `native:filter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Filter layers by type

Filters layers by their type. Incoming layers will be directed to different outputs based on whether they are a vector or raster layer.

参数

标注	名称	类型	描述
输入图层	INPUT	[layer]	Generic Map Layer

输出

标注	名称	类型	描述
Vector features 可选	VECTOR	[vector]	A Vector Layer of the input, if compatible
Raster layer 可选	RASTER	[栅格]	A Raster Layer of the input, if compatible

Python 代码

Algorithm ID: `native:filterlayersbytype`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Load layer into project

Loads a layer to the current project.

参数

标注	名称	类型	描述
Layer	INPUT	[layer]	Layer to load in the legend
Loaded layer name	NAME	[string]	Name of the loaded layer

输出

标注	名称	类型	描述
Layer	OUTPUT	[same as input]	The (renamed) loaded layer

Python 代码

Algorithm ID: native:loadlayer

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Raise exception

Raises an exception and cancels a model's execution. The exception message can be customized, and optionally an expression based condition can be specified. If an expression condition is used, then the exception will only be raised if the expression result is true. A false result indicates that no exception will be raised, and the model execution can continue uninterrupted.

参数

标注	名称	类型	描述
Message	MESSAGE	[string]	Message to display
Condition 可选	CONDITION	[expression]	Expression to evaluate if true

输出

A message in the log panel.

Python 代码

Algorithm ID: native:raiseexception

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Raise message

Raises an information message in the log. The message can be customized, and optionally an expression based condition can be specified. If an expression condition is used, then the message will only be logged if the expression result is true. A false result indicates that no message will be logged.

参数

标注	名称	类型	描述
Information message	MESSAGE	[string]	Message to display
Condition 可选	CONDITION	[expression]	Expression to evaluate if true

输出

A message in the log panel.

Python 代码

Algorithm ID: `native:raisemessage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Raise warning

Raises a warning message in the log. The warning message can be customized, and optionally an expression based condition can be specified. If an expression condition is used, then the warning will only be logged if the expression result is true. A false result indicates that no warning will be logged.

参数

标注	名称	类型	描述
Message	MESSAGE	[string]	Message to display
Condition 可选	CONDITION	[expression]	Expression to evaluate if true

输出

A message in the log panel.

Python 代码

Algorithm ID: `native:raisewarning`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Rename layer

Renames a layer.

参数

标注	名称	类型	描述
Layer	`INPUT`	[layer]	Layer to rename
New name	`NAME`	[string]	The new name of the layer

输出

标注	名称	类型	描述
Layer	`OUTPUT`	[same as input]	The (renamed) output layer

Python 代码

Algorithm ID: `native:renamelayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Save log to file

Saves the model's execution log to a file. Optionally, the log can be saved in a HTML formatted version.

参数

标注	名称	类型	描述
Use HTML	`USE_HTML`	[Boolean] Default: False	Use HTML formatting

输出

标注	名称	类型	描述
File	`OUTPUT`	[string]	Destination of the log

Python 代码

Algorithm ID: `native:savelog`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Set project variable

Sets an expression variable for the current project.

参数

标注	名称	类型	描述
Variable name	`NAME`	[string]	Name of the variable
Variable value	`VALUE`	[string]	Value to be stored

输出

None.

Python 代码

Algorithm ID: `native:setprojectvariable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

String concatenation

Concatenates two strings into a single one in the Processing Modeler.

参数

标注	名称	类型	描述
Input 1	`INPUT_1`	[string]	First string
Input 2	`INPUT_2`	[string]	Second string

输出

标注	名称	类型	描述
Concatenation	`CONCATENATION`	[string]	The concatenated string

Python 代码

Algorithm ID: `native:stringconcatenation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Variable distance buffer

Warning

This algorithm is deprecated and can be removed anytime. Prefer using *Buffer* algorithm instead.

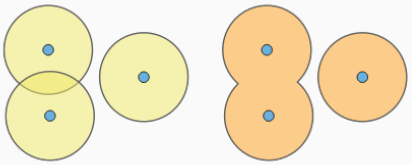
Computes a buffer area for all the features in an input layer.

The size of the buffer for a given feature is defined by an attribute, so it allows different features to have different buffer sizes.

See also

Buffer

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	输入矢量图层
Distance field	`DISTANCE`	[tablefield: numeric]	Attribute for the distance radius of the buffer
Segments	`SEGMENTS`	[numeric: integer] Default: *5*	Controls the number of line segments to use to approximate a quarter circle when creating rounded offsets.
Dissolve result	`DISSOLVE`	[boolean] Default: *False*	Choose to dissolve the final buffer, resulting in a single feature covering all input features.  Fig. 24.30: Normal and dissolved buffer
End cap style	`END_CAP_STYLE`	[enumeration] Default: Round	Controls how line endings are handled in the buffer. Fig. 24.31: Round, flat and square cap styles
Join style	`JOIN_STYLE`	[enumeration] Default: Round	Specifies whether round, miter or beveled joins should be used when offsetting corners in a line.
Miter limit	`MITER_LIMIT`	[numeric: double] Default: 2.0	Only applicable for mitered join styles, and controls the maximum distance from the offset curve to use when creating a mitered join.

输出

标注	名称	类型	描述
Buffer	`OUTPUT`	[vector: polygon]	Buffer polygon vector layer.

Python 代码

Algorithm ID: `qgis:variabledistancebuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.13 Network analysis

Service area (from layer)

Returns all the edges or parts of edges of a network that can be reached within a distance or a time, starting from a point layer. This allows evaluation of accessibility within a network, e.g., what are the places I can navigate to on a road network without spending cost greater than a given value (the cost can be distance or time).

> **Note**
>
> This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

参数

Basic parameters

标注	名称	类型	描述
Vector layer representing network	`INPUT`	[vector: line]	Line vector layer representing the network to be covered
Vector layer with start points	`START_POINTS`	[vector: point]	Point vector layer whose features are used as start points to generate the service areas
Path type to calculate	`STRATEGY`	[enumeration] Default: 0	The type of path to calculate. One of: • 0 --- Shortest • 1 --- Fastest
Travel cost (distance for "Shortest", time for "Fastest")	`TRAVEL_COST`	[numeric: double] Default: 0.0	The value is estimated as a distance (in the network layer units) when looking for the *Shortest* path and as time (in hours) for the *Fastest* path.

continues on next page

Table 24.222 – continued from previous page

标注	名称	类型	描述
Service area (lines)	`OUTPUT_LINES`	[vector: line] Default: `[Create temporary layer]`	Specify the output line layer for the service area. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。
Service area (boundary nodes)	`OUTPUT`	[vector: point] Default: `[Skip output]`	Specify the output point layer for the service area boundary nodes. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。
Non-routable features	`OUTPUT_NON_ROU`	[vector: point] Default: `[Skip output]`	Specify the output which will be used to store any input features which could not be routed (e.g., those which are too far from the network layer). *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Direction field 可选	`DIRECTION_FIEL`	[tablefield: any]	The field used to specify directions for the network edges. The values used in this field are specified with the three parameters `Value for forward direction`, `Value for backward direction` and `Value for both directions`. Forward and reverse directions correspond to a one-way edge, ”both directions” indicates a two-way edge. If a feature does not have a value in this field, or no field is set then the default direction setting (provided with the `Default direction` parameter) is used.
Value for forward direction 可选	`VALUE_FORWARD`	[string] Default: ” (empty string)	Value set in the direction field to identify edges with a forward direction

continues on next page

Table 24.223 – continued from previous page

标注	名称	类型	描述
Value for backward direction 可选	VALUE_BACKWARD	[string] Default: ” (empty string)	Value set in the direction field to identify edges with a backward direction
Value for both directions 可选	VALUE_BOTH	[string] Default: ” (empty string)	Value set in the direction field to identify bidirectional edges
Default direction	DEFAULT_DIRECT	[enumeration] Default: 2	If a feature has no value set in the direction field or if no direction field is set, then this direction value is used. One of: • 0 --- Forward direction • 1 --- Backward direction • 2 --- Both directions
Speed field 可选	SPEED_FIELD	[tablefield: numeric]	Field providing the speed value (in km/h) for the edges of the network when looking for the fastest path. If a feature does not have a value in this field, or no field is set then the default speed value (provided with the Default speed parameter) is used.
Default speed (km/h)	DEFAULT_SPEED	[numeric: double] Default: 50.0	Value to use to calculate the travel time if no speed value is provided for an edge in the specified field
Topology tolerance	TOLERANCE	[numeric: double] Default: 0.0	Two lines with nodes closer than the specified tolerance are considered connected
Include upper/lower bound points	INCLUDE_BOUNDS	[boolean] Default: False	Creates a point layer output with two points for each edge at the boundaries of the service area. One point is the start of that edge, the other is the end.
Maximum point distance from network 可选	POINT_TOLERANC	[numeric: double] Default: Not set	Specifies an optional limit on the distance from the points to the network layer. If a point is further from the network than this distance it will be treated as non-routable. If not set, endpoints will be snapped to the nearest point on the network layer, regardless of how far away from the network they actually are.

输出

标注	名称	类型	描述
Service area (boundary nodes)	OUTPUT	[vector: point]	The output point layer with the service area boundary nodes.
Service area (lines)	OUTPUT_LINES	[vector: line]	Line layer representing the parts of the network that can be serviced by the start points, for the given cost.

continues on next page

Table 24.224 – continued from previous page

标注	名称	类型	描述
Non routable features	OUTPUT_NON_ROU	[vector: point]	An optional output which will be used to store any input features which could not be routed (e.g., those which are too far from the network layer).

Python 代码

Algorithm ID: `native:serviceareafromlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Service area (from point)

Returns all the edges or parts of edges of a network that can be reached within a given distance or time, starting from a point feature. This allows the evaluation of accessibility within a network, e.g., what are the places I can navigate to on a road network without spending a cost greater than a given value (the cost can be distance or time).

Note

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

参数

Basic parameters

标注	名称	类型	描述
Vector layer representing the network	INPUT	[vector: line]	Line vector layer representing the network to be covered
Path type to calculate	STRATEGY	[enumeration] Default: 0	The type of path to calculate. One of: • 0 --- Shortest • 1 --- Fastest
Start point	START_POINT	[coordinates]	Coordinate of the point to calculate the service area around. Press the ... button next to the option and click on the canvas to fill the parameter with the clicked point coordinate.
Travel cost (distance for "Shortest", time for "Fastest")	TRAVEL_COST	[numeric: double] Default: 0.0	The value is estimated as a distance (in the network layer units) when looking for the *Shortest* path and as time (in hours) for the *Fastest* path.

continues on next page

Table 24.225 – continued from previous page

标注	名称	类型	描述
Service area (lines) 可选	`OUTPUT_LINES`	[vector: line] Default: `[Create temporary layer]`	Specify the output line layer for the service area. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。
Service area (boundary nodes) 可选	`OUTPUT`	[vector: point] Default: `[Skip output]`	Specify the output point layer for the service area boundary nodes. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Direction field 可选	`DIRECTION_FIEL`	[tablefield: any]	The field used to specify directions for the network edges. The values used in this field are specified with the three parameters `Value for forward direction`, `Value for backward direction` and `Value for both directions`. Forward and reverse directions correspond to a one-way edge, ”both directions” indicates a two-way edge. If a feature does not have a value in this field, or no field is set then the default direction setting (provided with the `Default direction` parameter) is used.
Value for forward direction 可选	`VALUE_FORWARD`	[string] Default: ” (empty string)	Value set in the direction field to identify edges with a forward direction
Value for backward direction 可选	`VALUE_BACKWARD`	[string] Default: ” (empty string)	Value set in the direction field to identify edges with a backward direction
Value for both directions 可选	`VALUE_BOTH`	[string] Default: ” (empty string)	Value set in the direction field to identify bidirectional edges
Default direction	`DEFAULT_DIRECT`	[enumeration] Default: 2	If a feature has no value set in the direction field or if no direction field is set, then this direction value is used. One of: • 0 --- Forward direction • 1 --- Backward direction • 2 --- Both directions

continues on next page

Table 24.226 – continued from previous page

标注	名称	类型	描述
Speed field 可选	SPEED_FIELD	[tablefield: numeric]	Field providing the speed value (in km/h) for the edges of the network when looking for the fastest path. If a feature does not have a value in this field, or no field is set then the default speed value (provided with the Default speed parameter) is used.
Default speed (km/h)	DEFAULT_SPEED	[numeric: double] Default: 50.0	Value to use to calculate the travel time if no speed value is provided for an edge in the specified field
Topology tolerance	TOLERANCE	[numeric: double] Default: 0.0	Two lines with nodes closer than the specified tolerance are considered connected
Maximum point distance from network 可选	POINT_TOLERANC	[numeric: double] Default: 0.0	Specifies an optional limit on the distance from the start point to the network layer. If the point is further from the network than this distance an error will be raised. If not set, the point will be snapped to the nearest point on the network layer, regardless of how far away from the network it actually is.
Include upper/lower bound points	INCLUDE_BOUNDS	[boolean] Default: False	Creates a point layer output with two points for each edge at the boundaries of the service area. One point is the start of that edge, the other is the end.

输出

标注	名称	类型	描述
Service area (boundary nodes)	OUTPUT	[vector: point]	The output point layer with the service area boundary nodes.
Service area (lines)	OUTPUT_LINES	[vector: line]	Line layer representing the parts of the network that can be serviced by the start point, for the given cost.

Python 代码

Algorithm ID: native:serviceareafrompoint

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Shortest path (layer to point)

Computes the optimal (shortest or fastest) routes from multiple start points defined by a vector layer and a given end point.

Note

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

参数

Basic parameters

标注	名称	类型	描述
Vector layer representing network	`INPUT`	[vector: line]	Line vector layer representing the network to be covered
Path type to calculate	`STRATEGY`	[enumeration] Default: 0	The type of path to calculate. One of: • 0 --- Shortest • 1 --- Fastest
Vector layer with start points	`START_POINTS`	[vector: point]	Point vector layer whose features are used as start points of the routes
End point (x, y)	`END_POINT`	[coordinates]	Point feature representing the end point of the routes Press the ... button next to the option and click on the canvas to fill the parameter with the clicked point coordinate.
Shortest path	`OUTPUT`	[vector: line] Default: `[Create temporary layer]`	Specify the output line layer for the shortest paths. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。
Non-routable features 可选	`OUTPUT_NON_ROU`	[vector: point] Default: `[Skip output]`	Specify the output which will be used to store any input features which could not be routed (e.g., those which are too far from the network layer). *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Direction field 可选	DIRECTION_FIEL	[tablefield: any]	The field used to specify directions for the network edges. The values used in this field are specified with the three parameters `Value for forward direction`, `Value for backward direction` and `Value for both directions`. Forward and reverse directions correspond to a one-way edge, "both directions" indicates a two-way edge. If a feature does not have a value in this field, or no field is set then the default direction setting (provided with the `Default direction` parameter) is used.
Value for forward direction 可选	VALUE_FORWARD	[string] Default: '' (empty string)	Value set in the direction field to identify edges with a forward direction
Value for backward direction 可选	VALUE_BACKWARD	[string] Default: '' (empty string)	Value set in the direction field to identify edges with a backward direction
Value for both directions 可选	VALUE_BOTH	[string] Default: '' (empty string)	Value set in the direction field to identify bidirectional edges
Default direction	DEFAULT_DIRECT	[enumeration] Default: 2	If a feature has no value set in the direction field or if no direction field is set, then this direction value is used. One of: • 0 --- Forward direction • 1 --- Backward direction • 2 --- Both directions
Speed field 可选	SPEED_FIELD	[tablefield: numeric]	Field providing the speed value (in `km/h`) for the edges of the network when looking for the fastest path. If a feature does not have a value in this field, or no field is set then the default speed value (provided with the `Default speed` parameter) is used.
Default speed (km/h)	DEFAULT_SPEED	[numeric: double] Default: 50.0	Value to use to calculate the travel time if no speed value is provided for an edge in the specified field
Topology tolerance	TOLERANCE	[numeric: double] Default: 0.0	Two lines with nodes closer than the specified tolerance are considered connected

Maximum point distance from network 可选	POINT_TOLERANC	[numeric: double] Default: Not set	Specifies an optional limit on the distance from the start and end points to the network layer. If a start feature is further from the network than this distance it will be treated as non-routable. If the end point is further from the network than this distance an error will be raised. If not set, points will be snapped to the nearest point on the network layer, regardless of how far away from the network they actually are.

输出

标注	名称	类型	描述
Shortest path	OUTPUT	[vector: line]	Line layer of the shortest or fastest path from each of the start points to the end point
Non routable features	OUTPUT_NON_ROU	[vector: point]	An optional output layer which will be used to store any input features which could not be routed (e.g., those which are too far from the network layer).

Python 代码

Algorithm ID: `native:shortestpathlayertopoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Shortest path (point to layer)

Computes the optimal (shortest or fastest) routes between a given start point and multiple end points defined by a point vector layer.

Note

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

参数

Basic parameters

标注	名称	类型	描述
Vector layer representing network	`INPUT`	[vector: line]	Line vector layer representing the network to be covered
Path type to calculate	`STRATEGY`	[enumeration] Default: 0	The type of path to calculate. One of: • 0 --- Shortest • 1 --- Fastest
Start point	`START_POINT`	[coordinates]	Point feature representing the start point of the routes Press the ... button next to the option and click on the canvas to fill the parameter with the clicked point coordinate.
Vector layer with end points	`END_POINTS`	[vector: point]	Point vector layer whose features are used as end points of the routes
Shortest path	`OUTPUT`	[vector: line] Default: `[Create temporary layer]`	Specify the output line layer for the shortest paths. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。
Non-routable features	`OUTPUT_NON_ROU` 可选	[vector: point] Default: `[Skip output]`	Specify the output which will be used to store any input features which could not be routed (e.g., those which are too far from the network layer). *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Direction field 可选	DIRECTION_FIEL	[tablefield: any]	The field used to specify directions for the network edges. The values used in this field are specified with the three parameters `Value for forward direction`, `Value for backward direction` and `Value for both directions`. Forward and reverse directions correspond to a one-way edge, "both directions" indicates a two-way edge. If a feature does not have a value in this field, or no field is set then the default direction setting (provided with the `Default direction` parameter) is used.
Value for forward direction 可选	VALUE_FORWARD	[string] Default: '' (empty string)	Value set in the direction field to identify edges with a forward direction
Value for backward direction 可选	VALUE_BACKWARD	[string] Default: '' (empty string)	Value set in the direction field to identify edges with a backward direction
Value for both directions 可选	VALUE_BOTH	[string] Default: '' (empty string)	Value set in the direction field to identify bidirectional edges
Default direction	DEFAULT_DIRECT	[enumeration] Default: 2	If a feature has no value set in the direction field or if no direction field is set, then this direction value is used. One of: • 0 --- Forward direction • 1 --- Backward direction • 2 --- Both directions
Speed field 可选	SPEED_FIELD	[tablefield: numeric]	Field providing the speed value (in `km/h`) for the edges of the network when looking for the fastest path. If a feature does not have a value in this field, or no field is set then the default speed value (provided with the `Default speed` parameter) is used.
Default speed (km/h)	DEFAULT_SPEED	[numeric: double] Default: 50.0	Value to use to calculate the travel time if no speed value is provided for an edge in the specified field
Topology tolerance	TOLERANCE	[numeric: double] Default: 0.0	Two lines with nodes closer than the specified tolerance are considered connected

Maximum point distance from network 可选	POINT_TOLERANC	[numeric: double] Default: Not set	Specifies an optional limit on the distance from the start and end points to the network layer. If the start point is further from the network than this distance an error will be raised. If an end feature is further from the network than this distance it will be treated as non-routable. If not set, points will be snapped to the nearest point on the network layer, regardless of how far away from the network they actually are.

输出

标注	名称	类型	描述
Shortest path	`OUTPUT`	[vector: line]	Line layer of the shortest or fastest path from each of the start points to the end point
Non routable features	`OUTPUT_NON_ROU`	[vector: point]	An optional output layer which will be used to store any input features which could not be routed (e.g., those which are too far from the network layer).

Python 代码

Algorithm ID: `native:shortestpathpointtolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Shortest path (point to point)

Computes the optimal (shortest or fastest) route between a given start point and a given end point.

> **Note**
>
> This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

参数

Basic parameters

标注	名称	类型	描述
Vector layer representing network	`INPUT`	[vector: line]	Line vector layer representing the network to be covered
Path type to calculate	`STRATEGY`	[enumeration] Default: 0	The type of path to calculate. One of: • 0 --- Shortest • 1 --- Fastest
Start point (x, y)	`START_POINT`	[coordinates]	Point feature representing the start point of the routes. Press the ... button next to the option and click on the canvas to fill the parameter with the clicked point coordinate.
End point (x, y)	`END_POINT`	[coordinates]	Point feature representing the end point of the routes. Press the ... button next to the option and click on the canvas to fill the parameter with the clicked point coordinate.

continues on next page

Table 24.234 – continued from previous page

标注	名称	类型	描述
Shortest path	OUTPUT	[vector: line]	Specify the output line layer for the shortest paths. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Direction field 可选	DIRECTION_FIEL	[tablefield: any]	The field used to specify directions for the network edges. The values used in this field are specified with the three parameters Value for forward direction, Value for backward direction and Value for both directions. Forward and reverse directions correspond to a one-way edge, "both directions" indicates a two-way edge. If a feature does not have a value in this field, or no field is set then the default direction setting (provided with the Default direction parameter) is used.
Value for forward direction 可选	VALUE_FORWARD	[string] Default: '' (empty string)	Value set in the direction field to identify edges with a forward direction
Value for backward direction 可选	VALUE_BACKWARD	[string] Default: '' (empty string)	Value set in the direction field to identify edges with a backward direction
Value for both directions 可选	VALUE_BOTH	[string] Default: '' (empty string)	Value set in the direction field to identify bidirectional edges
Default direction	DEFAULT_DIRECT	[enumeration] Default: 2	If a feature has no value set in the direction field or if no direction field is set, then this direction value is used. One of: • 0 --- Forward direction • 1 --- Backward direction • 2 --- Both directions
Speed field 可选	SPEED_FIELD	[tablefield: numeric]	Field providing the speed value (in km/h) for the edges of the network when looking for the fastest path. If a feature does not have a value in this field, or no field is set then the default speed value (provided with the Default speed parameter) is used.
Default speed (km/h)	DEFAULT_SPEED	[numeric: double] Default: 50.0	Value to use to calculate the travel time if no speed value is provided for an edge in the specified field

continues on next page

Table 24.235 – continued from previous page

标注	名称	类型	描述
Topology tolerance	TOLERANCE	[numeric: double] Default: 0.0	Two lines with nodes closer than the specified tolerance are considered connected
Maximum point distance from network 可选	POINT_TOLERANC	[numeric: double] Default: Not set	Specifies an optional limit on the distance from the start and end points to the network layer. If either point is further from the network than this distance an error will be raised. If not set, points will be snapped to the nearest point on the network layer, regardless of how far away from the network they actually are.

输出

标注	名称	类型	描述
Shortest path	OUTPUT	[vector: line]	Line layer of the shortest or fastest path from each of the start point to the end point

Python 代码

Algorithm ID: `native:shortestpathpointtopoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.14 Plots

Bar plot

Creates a bar plot from a category and a layer field.

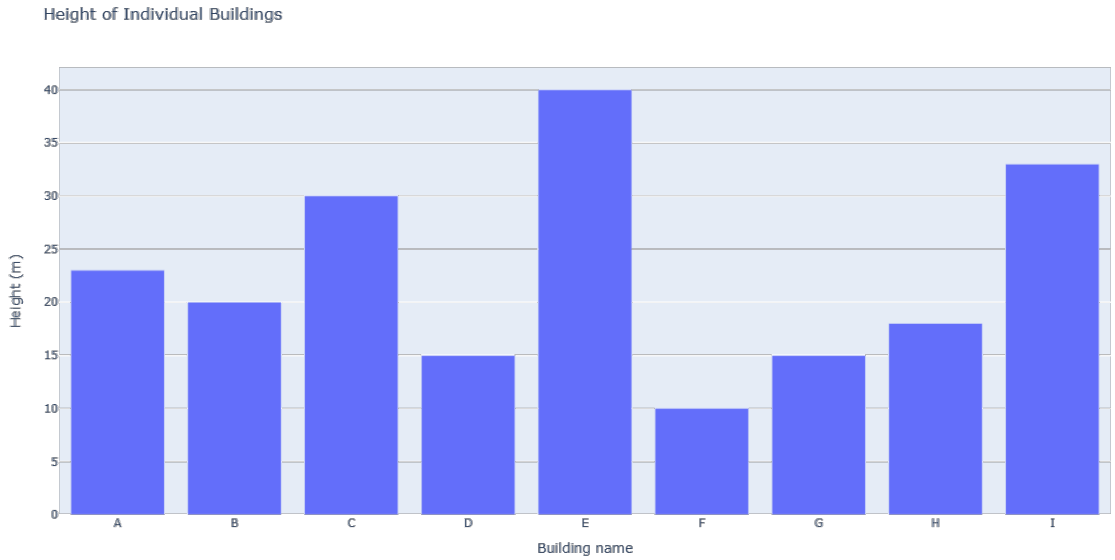


Fig. 24.32: Bar plot example

参数

标注	名称	类型	描述
输入图层	INPUT	[vector: geometry]	输入矢量图层
Category field name	NAME_FIELD	[tablefield: any]	Categorical field to use for grouping the bars (X axis)
Value field	VALUE_FIELD	[tablefield: numeric]	Value to use for the plot (Y axis).
Title Optional	TITLE	[string] Default: ""	Title of the plot
X-axis Title Optional	XAXIS_TITLE	[string] Default: ""	If empty, the name of the category field is used. With a single space, the axis title is hidden.
Y-axis Title Optional	YAXIS_TITLE	[string] Default: ""	If empty, the name of the value field is used. With a single space, the axis title is hidden.
Bar plot	OUTPUT	[html] 默认: [保存到临时文件]	Specify the HTML file for the plot. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Bar plot	OUTPUT	[html]	HTML file with the plot. Available in the *Processing* ► *Result Viewer*.

Python 代码

Algorithm ID: `qgis:barplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Box plot

Creates a box plot from a category field and a numerical layer field.

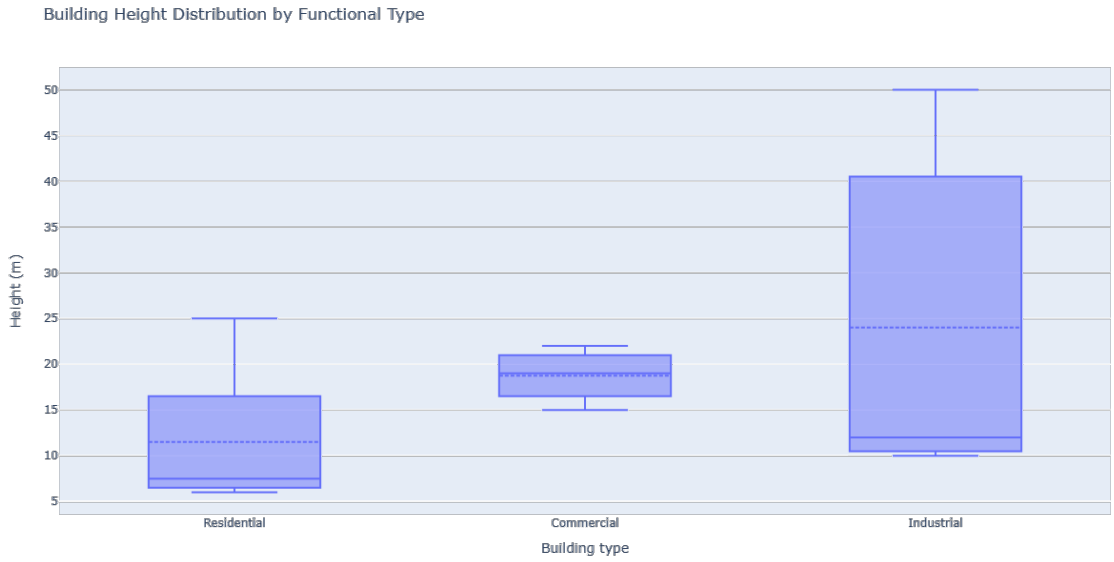


Fig. 24.33: Example of box plot

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Category name field	`NAME_FIELD`	[tablefield: any]	Categorical field to use for grouping the boxes (X axis)
Value field	`VALUE_FIELD`	[tablefield: numeric]	Value to use for the plot (Y axis).
Additional statistic lines	`MSD`	[enumeration] Default: 0	Additional statistics information to add to the plot. One of: • 0 --- Show Mean • 1 --- Show Standard Deviation • 2 --- Don't show mean and standard deviation
Title Optional	`TITLE`	[string] Default: ""	Title of the plot

continues on next page

Table 24.239 – continued from previous page

标注	名称	类型	描述
X-axis Title Optional	XAXIS_TITLE	[string] Default: ""	If empty, the name of the category field is used. With a single space, the axis title is not shown.
Y-axis Title Optional	YAXIS_TITLE	[string] Default: ""	If empty, the name of the value field is used. With a single space, the axis title is not shown.
Box plot	OUTPUT	[html] 默认: [保存到临时文件]	Specify the HTML file for the plot. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Box plot	OUTPUT	[html]	HTML file with the plot. Available in the *Processing* ► *Result Viewer*.

Python 代码

Algorithm ID: `qgis:boxplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Generate elevation profile image

Creates an elevation profile image from a list of map layers and an optional terrain layer.

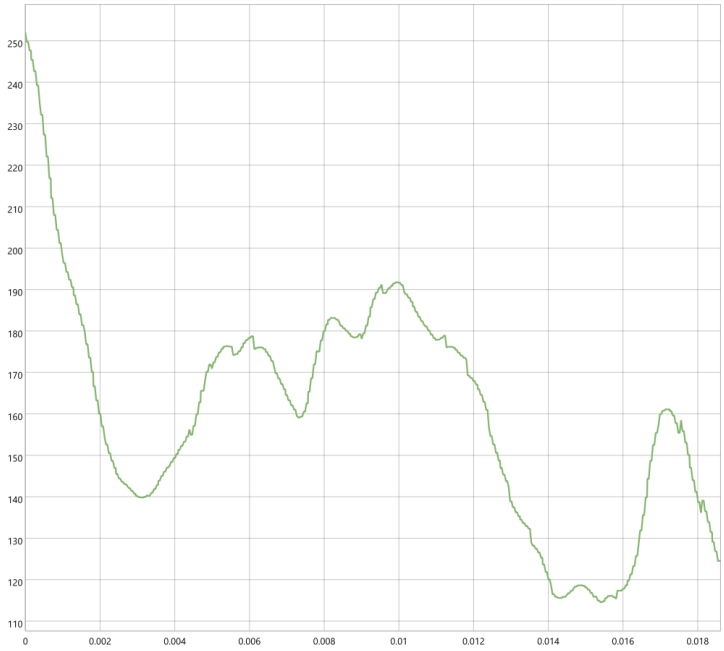


Fig. 24.34: Example of elevation profile

参数

Basic parameters

标注	名称	类型	描述
Profile curve	`CURVE`	[geometry: line]	The LineString geometry (WKT or GeoJSON) representing the profile curve along which the elevation profile will be generated
Map layers	`MAP_LAYERS`	[layer] [list]	The list of map layers to include in the elevation profile
Chart width (in pixels)	`WIDTH`	[numeric: integer] Default: 400 Minimum value: 0	The width of the output chart in pixels.
Chart height (in pixels)	`HEIGHT`	[numeric: integer] Default: 300 Minimum value: 0	The height of the output chart in pixels.
Terrain layer Optional	`TERRAIN_LAYER`	[栅格]	A terrain layer (e.g., DEM) to use for elevation data. If not provided, elevation data will be derived from the map layers.

Advanced parameters

标注	名称	类型	描述
Chart minimum distance (X axis) Optional	MINIMUM_DISTAN	[numeric: double]	The minimum distance (X axis) to display on the chart. If not specified, the chart will auto-scale.
Chart maximum distance (X axis) Optional	MAXIMUM_DISTAN	[numeric: double]	The maximum distance (X axis) to display on the chart. If not specified, the chart will auto-scale.
Chart minimum elevation (Y axis) Optional	MINIMUM_ELEVAT	[numeric: double]	The minimum elevation (Y axis) to display on the chart. If not specified, the chart will auto-scale.
Chart maximum elevation (Y axis) Optional	MAXIMUM_ELEVAT	[numeric: double]	The maximum elevation (Y axis) to display on the chart. If not specified, the chart will auto-scale.
Chart text color Optional	TEXT_COLOR	[color]	The color of the text in the chart (e.g., axis labels, titles).
Chart background color Optional	BACKGROUND_COL	[color]	The background color of the chart.
Chart border color Optional	BORDER_COLOR	[color]	The color of the chart border.
Profile tolerance	TOLERANCE	[numeric: double] Default: 5.0 Minimum value: 0	Defines how far a feature (vector point, line, polygon, or point cloud) can be from the profile line to be included in the results. It uses map units and does not affect other layer types.
Chart DPI	DPI	[numeric: integer] Default: 96 Minimum value: 0	The resolution of the output image in dots per inch (DPI).
Output image	OUTPUT	[file] 默认: [保存到临时文件]	Specify the image file for the plot. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Output image	OUTPUT	[file]	The generated elevation profile image in the specified format.

Python 代码

Algorithm ID: `native:generateelevationprofileimage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Mean and standard deviation plot

Creates a box plot with mean and standard deviation values.

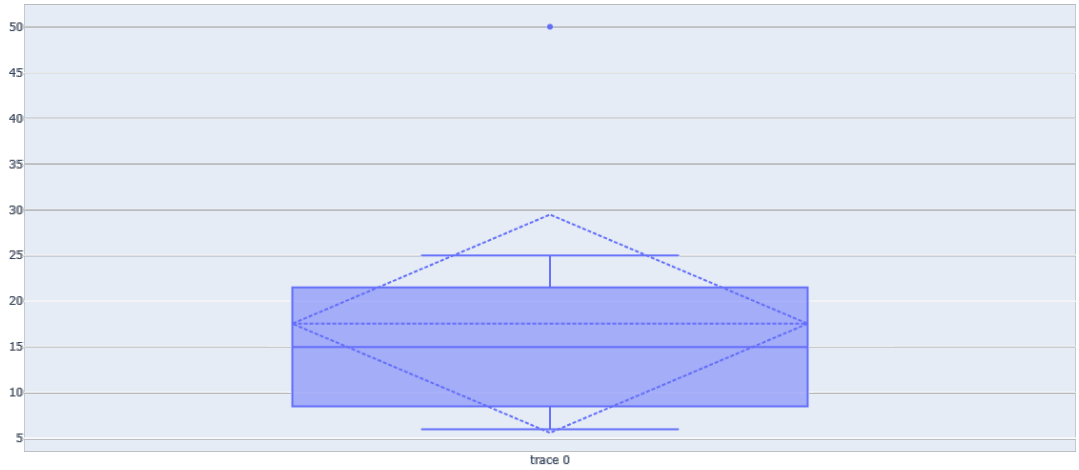


Fig. 24.35: Example of mean and standard deviation plot

参数

标注	名称	类型	描述
Input table	`INPUT`	[vector: geometry]	输入矢量图层
Category name field	`NAME_FIELD`	[tablefield: any]	Categorical field to use for grouping the boxes (X axis)
Value field	`VALUE_FIELD`	[tablefield: numeric]	Value to use for the plot (Y axis).
Plot	`OUTPUT`	[html] 默认：[保存到临时文件]	Specify the HTML file for the plot. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Plot	`OUTPUT`	[html]	HTML file with the plot. Available in the *Processing* ► *Result Viewer*.

Python 代码

Algorithm ID: `qgis:meanandstandarddeviationplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Polar plot

Generates a polar plot based on the value of an input vector layer.

Two fields must be entered as parameters: one that defines the category each feature (to group features) and another one with the variable to plot (this has to be a numeric one).

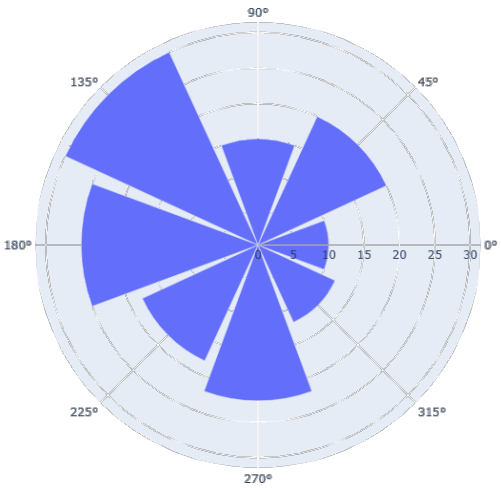


Fig. 24.36: Example of polar plot

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Category field name	`NAME_FIELD`	[tablefield: any]	Categorical field to use for grouping the features (X axis)
Value field	`VALUE_FIELD`	[tablefield: numeric]	Value to use for the plot (Y axis).
Polar plot	`OUTPUT`	[html] 默认：[保存到临时文件]	Specify the HTML file for the plot. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Polar plot	`OUTPUT`	[html]	HTML file with the plot. Available in the *Processing* ► *Result Viewer*.

Python 代码

Algorithm ID: `qgis:polarplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Raster layer histogram

Generates a histogram with the values of a raster layer.

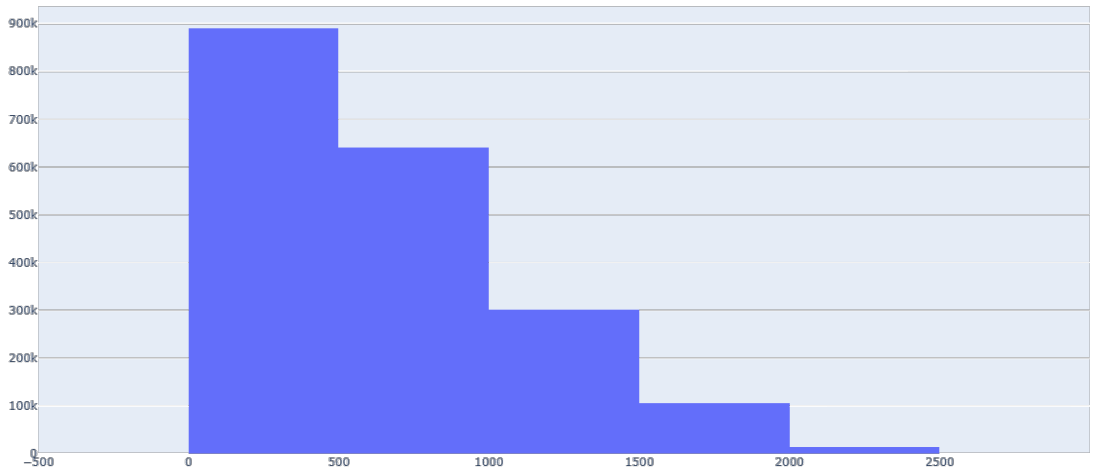


Fig. 24.37: Raster layer histogram example

参数

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	Input raster layer
Band number	`BAND`	[raster band]	Raster band to use for the histogram
number of bins	`BINS`	[numeric: integer] Default: 10	The number of bins to use in the histogram (X axis). Minimum 2.

continues on next page

Table 24.248 – continued from previous page

标注	名称	类型	描述
Histogram	OUTPUT	[html] 默认：[保存到临时文件]	Specify the HTML file for the plot. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Histogram	OUTPUT	[html]	HTML file with the plot. Available in the *Processing* ► *Result Viewer*.

Python 代码

Algorithm ID: `qgis:rasterlayerhistogram`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Vector layer histogram

Generates a histogram with the values of the attribute of a vector layer.

The attribute to use for computing the histogram must be numeric.

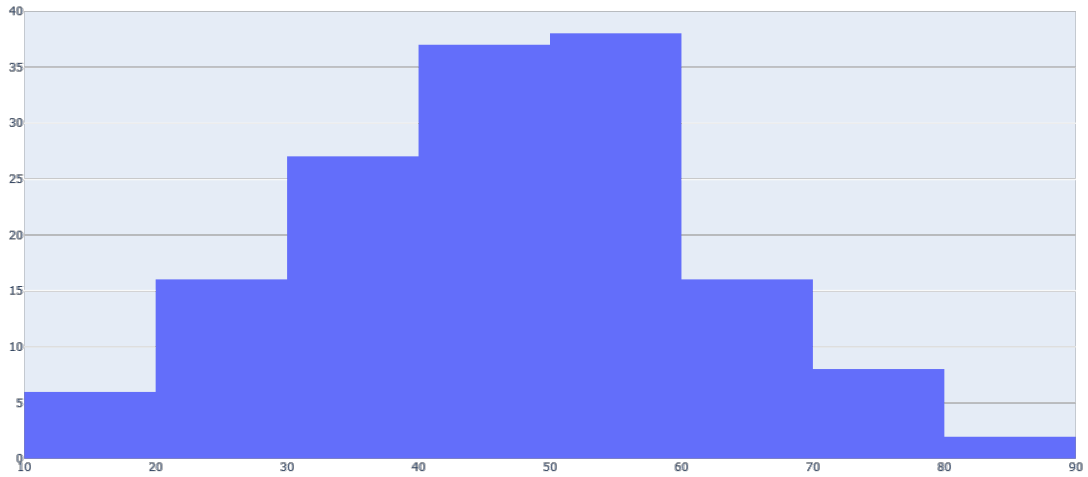


Fig. 24.38: Example of vector layer histogram

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	输入矢量图层
Attribute	`FIELD`	[tablefield: numeric]	Value to use for the plot (Y axis).
number of bins	`BINS`	[numeric: integer] Default: 10	The number of bins to use in the histogram (X axis). Minimum 2.
Histogram	`OUTPUT`	[html] 默认: [保存到临时文件]	Specify the HTML file for the plot. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Histogram	`OUTPUT`	[html]	HTML file with the plot. Available in the *Processing* ► *Result Viewer*.

Python 代码

Algorithm ID: `qgis:vectorlayerhistogram`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Vector layer scatterplot

Creates a simple `X` - `Y` scatter plot for a vector layer.

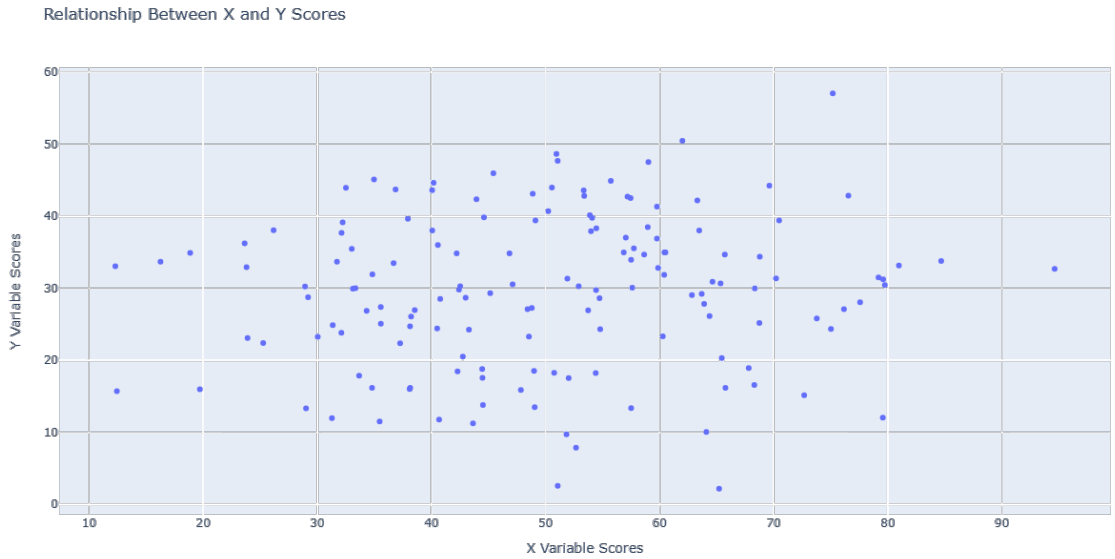


Fig. 24.39: Example of vector layer scatterplot

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
X attribute	`XFIELD`	[tablefield: numeric]	Field to use for the X axis
Y attribute	`YFIELD`	[tablefield: numeric]	Field to use for the Y axis
Hover text Optional	`HOVERTEXT`	[expression] Default: ""	Text to be shown when hovering with the mouse over data points. It can be picked from a field or based on an expression.
Title Optional	`TITLE`	[string] Default: ""	Title of the plot
X-axis Title Optional	`XAXIS_TITLE`	[string] Default: ""	If empty, the field name of the x attribute is used. With a single space, the axis title is not shown.
Y-axis Title Optional	`YAXIS_TITLE`	[string] Default: ""	If empty, the field name of the y attribute is used. With a single space, the axis title is not shown.
Use logarithmic scale for x-axis	`XAXIS_LOG`	[boolean] Default: False	When enabled, uses logarithmic scale for the x-axis
Use logarithmic scale for y-axis	`YAXIS_LOG`	[boolean] Default: False	When enabled, uses logarithmic scale for the y-axis
Scatterplot	`OUTPUT`	[html] 默认: [保存到临时文件]	Specify the HTML file for the plot. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Scatterplot	`OUTPUT`	[html]	HTML file with the plot. Available in the *Processing* ► *Result Viewer*.

Python 代码

Algorithm ID: `qgis:vectorlayerscatterplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Vector layer scatterplot 3D

Creates a 3D scatter plot for a vector layer.

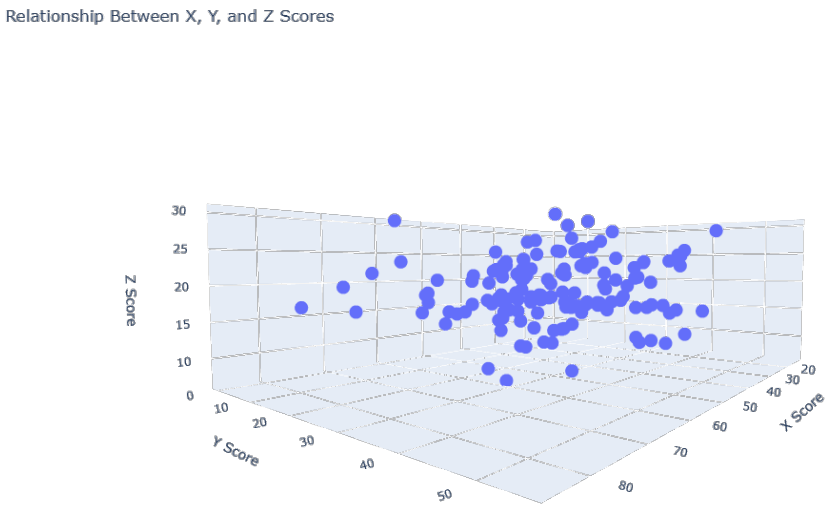


Fig. 24.40: Example of 3D scatter plot

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
X attribute	`XFIELD`	[tablefield: numeric]	Field to use for the X axis
Y attribute	`YFIELD`	[tablefield: numeric]	Field to use for the Y axis

continues on next page

Table 24.254 – continued from previous page

标注	名称	类型	描述
Z attribute	`ZFIELD`	[tablefield: numeric]	Field to use for the Z axis
Title Optional	`TITLE`	[string] Default: ""	Title of the plot
X-axis Title Optional	`XAXIS_TITLE`	[string] Default: ""	If empty, the field name of the X attribute is used.
Y-axis Title Optional	`YAXIS_TITLE`	[string] Default: ""	If empty, the field name of the Y attribute is used.
Z-axis Title Optional	`ZAXIS_TITLE`	[string] Default: ""	If empty, the field name of the Z attribute is used.
Scatterplot 3D	`OUTPUT`	[html] 默认：[保存到临时文件]	Specify the HTML file for the plot. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Scatterplot 3D	`OUTPUT`	[html]	HTML file with the plot. Available in the *Processing* ► *Result Viewer*.

Python 代码

Algorithm ID: `qgis:scatter3dplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.15 Point Cloud Conversion

> **Attention**
>
> Running these algorithms requires QGIS installed with PDAL >= 2.5.0 (see *Help* ► *About* menu).

Convert format

Converts a point cloud to a different file format, e.g. creates a compressed `.LAZ`.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[point cloud]	Input point cloud layer to convert
Converted	`OUTPUT`	[point cloud] 默认: [保存到临时文件]	Specify the point cloud file to use as output. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Converted	`OUTPUT`	[point cloud]	Output point cloud layer in a modified file format. Currently supported formats are `.LAS`, `.LAZ`, `.COPC.LAZ` and `.VPC`.

Python 代码

Algorithm ID: `pdal:convertformat`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Export to raster

Exports point cloud data to a 2D raster grid having cell size of given resolution, writing values from the specified attribute.

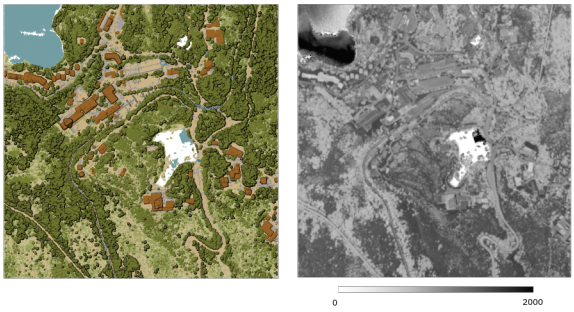


Fig. 24.41: Raster output using Intensity attribute of points

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[point cloud]	Input point cloud layer to export
Attribute	`ATTRIBUTE`	[field] [enumeration]	A Field of the point cloud layer to extract the values from
Resolution of the density raster	`RESOLUTION`	[numeric: double] 默认值：1.0	Cell size of the output raster
Tile size for parallel runs	`TILE_SIZE`	[numeric: integer] Default: 1000	
Exported	`OUTPUT`	[栅格] 默认：[保存到临时文件]	Specify the raster file to export the data to. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Filter expression 可选	`FILTER_EXPRESS`	[expression]	A *PDAL expression* for selecting a subset of features in the point cloud data
Cropping extent 可选	`FILTER_EXTENT`	[范围]	A map extent for selecting a subset of features in the point cloud data Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
X origin of a tile for parallel runs 可选	`ORIGIN_X`	[numeric: double]	
Y origin of a tile for parallel runs 可选	`ORIGIN_Y`	[numeric: double]	

输出

标注	名称	类型	描述
Exported	`OUTPUT`	[栅格]	Output raster layer features of the point cloud layer are exported to. Currently supported format is `.TIF`.

Python 代码

Algorithm ID: `pdal:exportraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Export raster (using triangulation)

Exports point cloud data to a 2D raster grid using a triangulation of points and then interpolating cell values from triangles.

> **Note**
>
> Using this algorithm can be slower if you are dealing with a large dataset. If your point cloud is dense, you can export your ground points as a raster using the *Export to raster* algorithm.

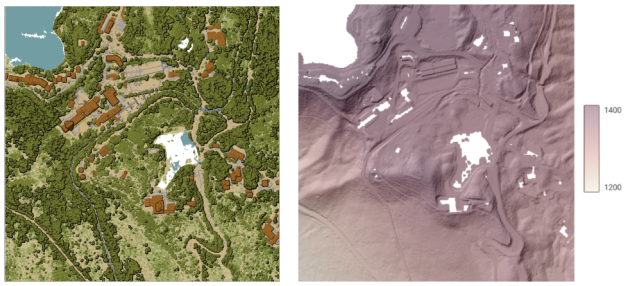


Fig. 24.42: Terrain raster output generated by point cloud triangulation

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[point cloud]	Input point cloud layer to export
Resolution of the density raster	`RESOLUTION`	[numeric: double] 默认值：1.0	Cell size of the output raster
Tile size for parallel runs	`TILE_SIZE`	[numeric: integer] Default: 1000	

continues on next page

Table 24.261 – continued from previous page

标注	名称	类型	描述
Exported	OUTPUT	[栅格] 默认: [保存到临时文件]	Specify the raster file to export the data to. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Filter expression 可选	FILTER_EXPRESS	[expression]	A *PDAL expression* for selecting a subset of features in the point cloud data
Cropping extent 可选	FILTER_EXTENT	[范围]	A map extent for selecting a subset of features in the point cloud data Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
X origin of a tile for parallel runs 可选	ORIGIN_X	[numeric: double]	
Y origin of a tile for parallel runs 可选	ORIGIN_Y	[numeric: double]	

输出

标注	名称	类型	描述
Exported (using triangulation)	OUTPUT	[栅格]	Output raster layer features of the point cloud layer are exported to. Currently supported format is `.TIF`.

Python 代码

Algorithm ID: `pdal:exportrastertin`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Export to vector

Exports point cloud data to a vector layer with 3D points (a GeoPackage), optionally with extra attributes.

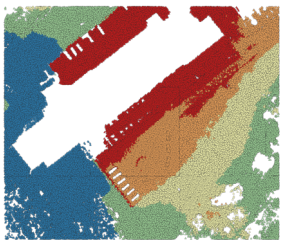


Fig. 24.43: Exporting point cloud (ground points) to a vector layer styled based on the elevation

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[point cloud]	Input point cloud layer to export
Attribute 可选	`ATTRIBUTE`	[field] [list]	One or more fields of the point cloud layer to export with the points.
Exported	`OUTPUT`	[vector] 默认：[保存到临时文件]	Specify the vector file to export the data to. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Filter expression 可选	`FILTER_EXPRESS`	[expression]	A *PDAL expression* for selecting a subset of features in the point cloud data

continues on next page

Table 24.265 – continued from previous page

标注	名称	类型	描述
Cropping extent 可选	`FILTER_EXTENT`	[范围]	A map extent for selecting a subset of features in the point cloud data Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`

输出

标注	名称	类型	描述
Exported	`OUTPUT`	[vector]	Output vector layer features of the point cloud layer are exported to. Currently supported format is `.GPKG`.

Python 代码

Algorithm ID: `pdal:exportvector`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.16 Point Cloud Data Management

> **Attention**
>
> Running these algorithms requires QGIS installed with PDAL >= 2.5.0 (see *Help* ► *About* menu).

Assign projection

Assigns a Coordinate Reference System to a point cloud layer, if it is missing or wrong. A new layer is created.

See also

Reproject

参数

标注	名称	类型	描述
输入图层	`INPUT`	[point cloud]	Input point cloud layer to assign a CRS to
Desired CRS	`CRS`	[crs]	The CRS to apply to the layer
Output layer	`OUTPUT`	[point cloud] 默认：[保存到临时文件]	Specify the point cloud file to use as output. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Converted	`OUTPUT`	[point cloud]	Output point cloud layer with a new CRS. Currently supported formats are `.LAS`, `.LAZ`, `.COPC.LAZ` and `.VPC`.

Python 代码

Algorithm ID: `pdal:assignprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Build virtual point cloud (VPC)

Creates a *virtual point cloud (VPC)* from input point cloud data.

If you leave the optional parameters unchecked, the VPC file will be built very quickly as the algorithm will only read metadata of input files. With any of the optional parameters set, the algorithm will read all points which can take some time.

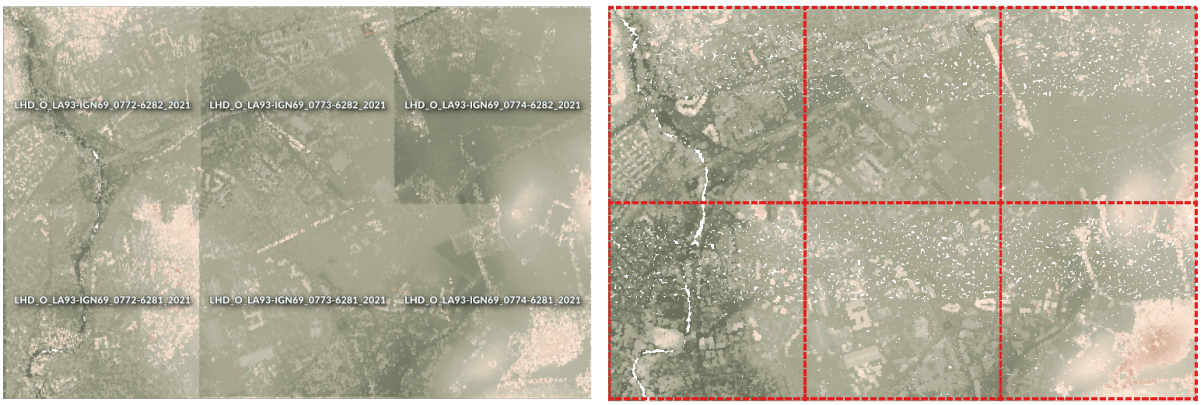


Fig. 24.44: Generating Virtual Point Cloud with overview from a set of point cloud tiles

See also

Merge

参数

标注	名称	类型	描述
Input layers	`LAYERS`	[point cloud] [list]	Input point cloud layers to combine inside a virtual point cloud layer
Calculate boundary polygon	`BOUNDARY`	[boolean] Default: False	Set to True to show the exact boundaries of data (rather than just rectangular extent)
Calculate statistics	`STATISTICS`	[boolean] Default: False	Set to True to understand ranges of values of the various attributes
Build overview point cloud	`OVERVIEW`	[boolean] Default: False	Generates a single "thinned" point cloud of all the input data (using only every 1000th point from original data). The overview point cloud will be created next to the VPC file - for example, for `mydata.vpc`, the overview point cloud would be named `mydata-overview.copc.laz`.
Virtual point cloud	`OUTPUT`	[point cloud] 默认：[保存到临时文件]	Specify the point cloud file to build the data into. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Virtual point cloud	OUTPUT	[栅格]	Output point cloud layer combining all the input data, as a virtual file.

Python 代码

Algorithm ID: `pdal:virtualpointcloud`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Clip

Clips a point cloud layer by a polygon layer so that the resulting point cloud contains only points within the polygons.

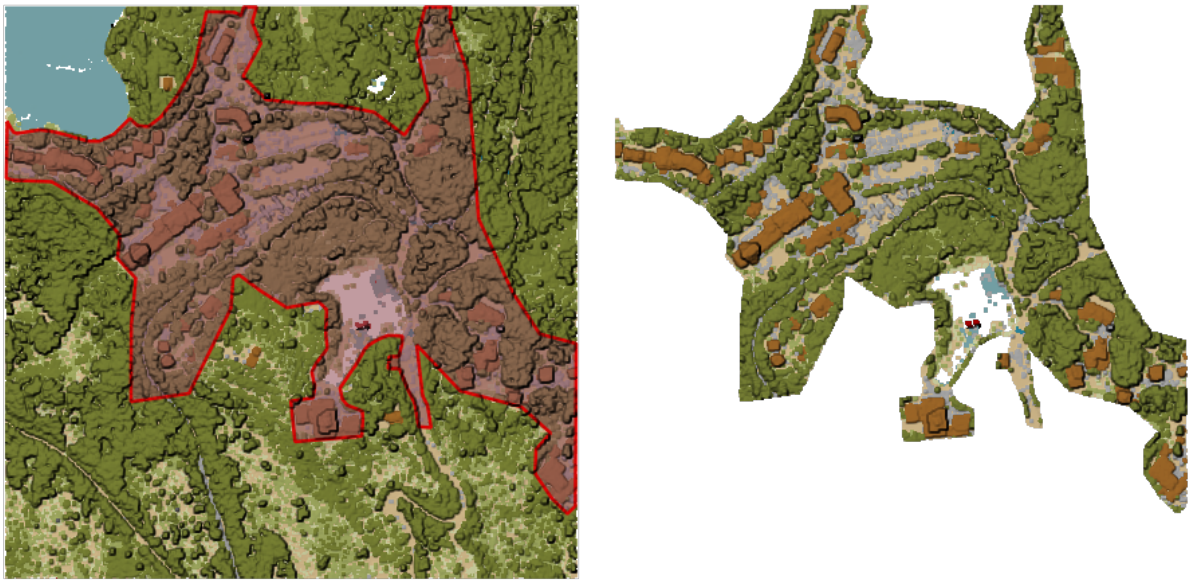


Fig. 24.45: Clipping an input point cloud layer with a polygon coverage

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[point cloud]	Input point cloud layer to clip
Clipping polygons	OVERLAY	[vector: polygon]	Polygon vector layer to use as coverage for clipping the points

continues on next page

Table 24.271 – continued from previous page

标注	名称	类型	描述
Clipped	`OUTPUT`	[point cloud] 默认：[保存到临时文件]	Specify the point cloud file to export the clipped points to. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Filter expression 可选	`FILTER_EXPRESS`	[expression]	A *PDAL expression* for selecting a subset of features in the point cloud data
Cropping extent 可选	`FILTER_EXTENT`	[范围]	A map extent for selecting a subset of features in the point cloud data Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`

输出

标注	名称	类型	描述
Clipped	`OUTPUT`	[point cloud]	Output point cloud whose features are the points within the coverage polygon layer.

Python 代码

Algorithm ID: `pdal:clip`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Create COPC

Creates the index for all the input point cloud files in a batch mode.

参数

标注	名称	类型	描述
Input layers	`LAYERS`	[point cloud] [list]	Input point cloud layers to create an index for
Output directory 可选	`OUTPUT`	[folder] Default: `[Skip output]`	Specify the folder to create the new files in. *One of*: • Skip Output • Save to a Temporary Directory • Save to Directory

输出

标注	名称	类型	描述
Output directory	`OUTPUT`	[folder]	Output folder containing point cloud layers with accompanying COPC index files.

Python 代码

Algorithm ID: `pdal:createcopc`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Information

Outputs basic metadata from an input point cloud file.

Example of output information:

```
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
```

(continues on next page)

(continued from previous page)

```
- 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
```

参数

标注	名称	类型	描述
输入图层	`INPUT`	[point cloud]	Input point cloud layer to extract metadata information from
Layer information	`OUTPUT`	[file] 默认：[保存到临时文件]	Specify the file to store the metadata information. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Layer information	`OUTPUT`	[html]	`HTML` file to store the metadata information.

Python 代码

Algorithm ID: `pdal:info`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Merge

Merges multiple point cloud files into a single one.

See also

Build virtual point cloud (VPC)

参数

Basic parameters

标注	名称	类型	描述
Input layers	`LAYERS`	[point cloud] [list]	Input point cloud layers to merge into a single one
Merged	`OUTPUT`	[point cloud] 默认：[保存到临时文件]	Specify the output point cloud merging input files. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Filter expression 可选	`FILTER_EXPRESS`	[expression]	A *PDAL expression* for selecting a subset of features in the point cloud data
Cropping extent 可选	`FILTER_EXTENT`	[范围]	A map extent for selecting a subset of features in the point cloud data Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`

输出

标注	名称	类型	描述
Merged	`OUTPUT`	[point cloud]	Output point cloud layer merging all the input files.

Python 代码

Algorithm ID: `pdal:merge`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Reproject

Reprojects a point cloud to a different Coordinate Reference System (CRS).

See also

Assign projection

参数

标注	名称	类型	描述
输入图层	`INPUT`	[point cloud]	Input point cloud layer to reproject to a different CRS
Target CRS	`CRS`	[crs]	The CRS to apply to the layer
Reprojected	`OUTPUT`	[point cloud] 默认: [保存到临时文件]	Specify the reprojected point cloud file. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Coordinate operation 可选	`OPERATION`	[datum]	The *datum transformation* to use to reproject the data between the origin and target systems.

输出

标注	名称	类型	描述
REPROJECTED	`OUTPUT`	[point cloud]	Output point cloud layer in the target CRS.

Python 代码

Algorithm ID: `pdal:reproject`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Thin (by sampling radius)

Creates a thinned version of the point cloud by performing sampling by distance point (reduces the number of points within a certain radius).

See also

Thin (by skipping points)

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[point cloud]	Input point cloud layer to create a thinned version from
Sampling radius (in map units)	`SAMPLING_RADIU`	[numeric: double] 默认值：1.0	Distance within which points are sampled to a unique point
Thinned (by radius)	`OUTPUT`	[point cloud] 默认：[保存到临时文件]	Specify the output point cloud with reduced points. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Filter expression 可选	`FILTER_EXPRESS`	[expression]	A *PDAL expression* for selecting a subset of features in the point cloud data

continues on next page

Table 24.285 – continued from previous page

标注	名称	类型	描述
Cropping extent 可选	`FILTER_EXTENT`	[范围]	A map extent for selecting a subset of features in the point cloud data Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`

输出

标注	名称	类型	描述
Thinned (by radius)	`OUTPUT`	[point cloud]	Output point cloud layer with reduced points.

Python 代码

Algorithm ID: `pdal:thinbyradius`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Thin (by skipping points)

Creates a thinned version of the point cloud by keeping only every N-th point (reduces the number of points by skipping nearby points).

See also

Thin (by sampling radius)

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[point cloud]	Input point cloud layer to create a thinned version from
Number of points to skip	`POINTS_NUMBER`	[numeric: integer] Default: 1	Keep only every N-th point in the input layer
Thinned (by decimation)	`OUTPUT`	[point cloud] 默认: [保存到临时文件]	Specify the output point cloud with reduced points. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Filter expression 可选	`FILTER_EXPRESS`	[expression]	A *PDAL expression* for selecting a subset of features in the point cloud data
Cropping extent 可选	`FILTER_EXTENT`	[范围]	A map extent for selecting a subset of features in the point cloud data Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`

输出

标注	名称	类型	描述
Thinned (by decimation)	`OUTPUT`	[point cloud]	Output point cloud layer with reduced points.

Python 代码

Algorithm ID: `pdal:thinbydecimate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Tile

Creates tiles from input point cloud files, recommended for best performance (in display or analysis) with such datasets in QGIS.

See also

Build virtual point cloud (VPC), *Create COPC*

参数

Basic parameters

标注	名称	类型	描述
Input layers	`LAYERS`	[point cloud] [list]	Input point cloud layers to create tiles from
Tile length	`LENGTH`	[numeric: double] Default: 1000.0	Size of the edge of each generated tile
Output directory	`OUTPUT`	[folder] Default: `[Save to temporary folder]`	Specify the folder to store the generated tiles. *One of*: • Save to a Temporary Directory • Save to Directory

Advanced parameters

标注	名称	类型	描述
Assign CRS 可选	`CRS`	[crs]	The CRS to apply to the layer

输出

标注	名称	类型	描述
Output directory	`OUTPUT`	[folder]	Output folder containing the tiles generated from input files.

Python 代码

Algorithm ID: `pdal:tile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.17 Point Cloud Extraction

Attention

Running these algorithms requires QGIS installed with PDAL >= 2.5.0 (see *Help* ► *About* menu).

Boundary

Exports a polygon file containing point cloud layer boundary. It may contain holes and it may be a multi-part polygon.

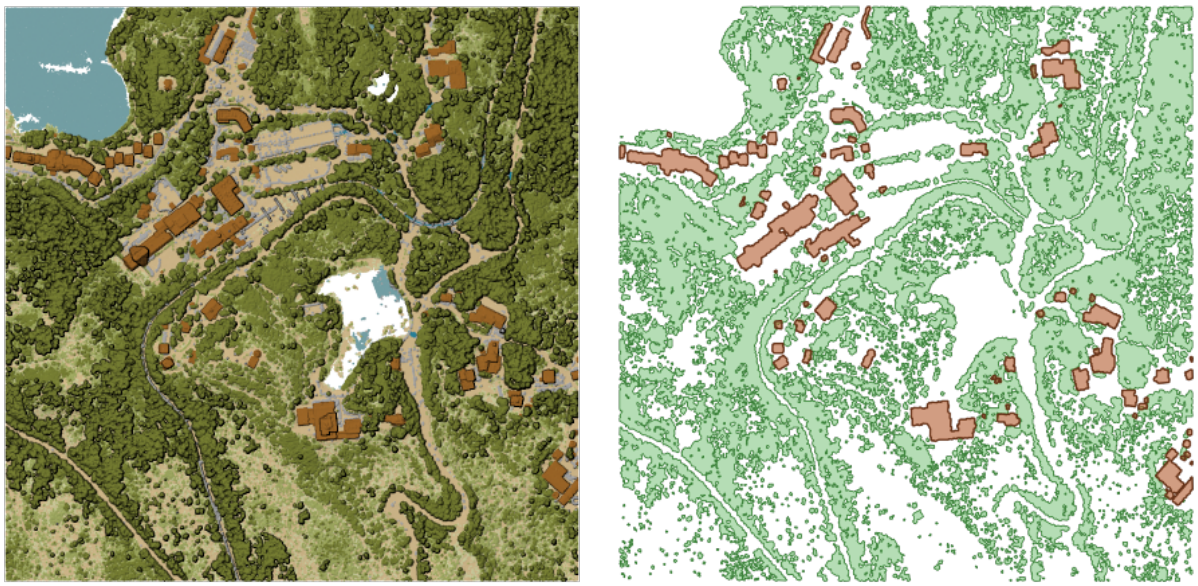


Fig. 24.46: Extracting high vegetation and building polygons from an input point cloud layer

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[point cloud]	Input point cloud layer to calculate boundary for
Resolution of cells used to calculate boundary 可选	`RESOLUTION`	[numeric: double]	Resolution of cells used to calculate boundary
Minimal number of points in a cell to consider cell occupied 可选	`THRESHOLD`	[numeric: integer]	Minimal number of points in a cell to consider cell occupied
Boundary	`OUTPUT`	[vector: polygon] 默认: [保存到临时文件]	Specify the point cloud file to use as output. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Filter expression 可选	`FILTER_EXPRESS`	[expression]	A *PDAL expression* for selecting a subset of features in the point cloud data
Cropping extent 可选	`FILTER_EXTENT`	[范围]	A map extent for selecting a subset of features in the point cloud data Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`

输出

标注	名称	类型	描述
Boundary	`OUTPUT`	[vector: polygon]	Output polygon vector layer with the point cloud boundary. Currently supported format is `.GPKG`.

Python 代码

Algorithm ID: `pdal:boundary`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Density

Exports a raster file based on the number of points within each raster cell - useful for quality checking of point cloud datasets.

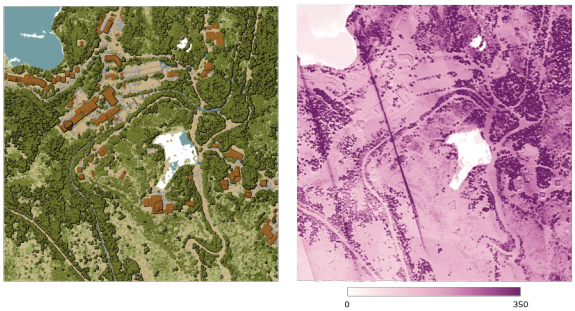


Fig. 24.47: Point density (number of points per 2x2 m) as a raster

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[point cloud]	Input point cloud layer to export
Resolution of the density raster	`RESOLUTION`	[numeric: double] 默认值：1.0	Cell size of the output raster
Tile size for parallel runs	`TILE_SIZE`	[numeric: integer] Default: 1000	Size of the tiles to split the data into for parallel runs
Density	`OUTPUT`	[栅格] 默认：[保存到临时文件]	Specify the raster file to export the data to. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Filter expression 可选	`FILTER_EXPRESS`	[expression]	A *PDAL expression* for selecting a subset of features in the point cloud data
Cropping extent 可选	`FILTER_EXTENT`	[范围]	A map extent for selecting a subset of features in the point cloud data Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
X origin of a tile for parallel runs 可选	`ORIGIN_X`	[numeric: double]	
Y origin of a tile for parallel runs 可选	`ORIGIN_Y`	[numeric: double]	

输出

标注	名称	类型	描述
Density	`OUTPUT`	[栅格]	Output raster layer with number of points within each cell. Currently supported format is `.TIF`.

Python 代码

Algorithm ID: `pdal:density`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Filter

Extracts point from the input point cloud which match PDAL expression and/or are inside of a cropping rectangle.

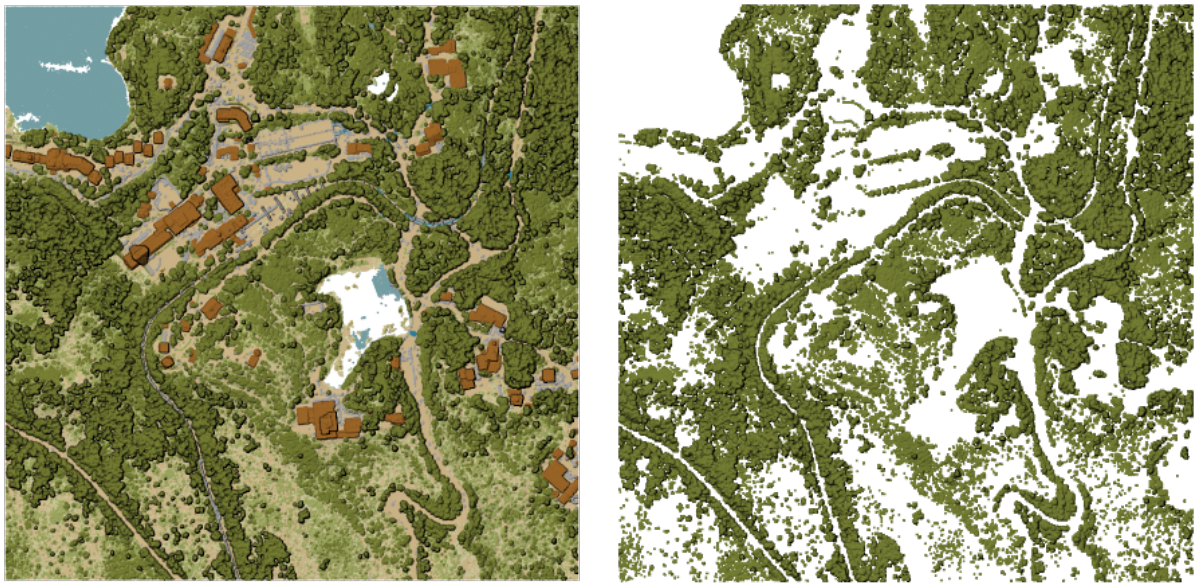


Fig. 24.48: Filtering of high vegetation class from an input point cloud layer

参数

标注	名称	类型	描述
输入图层	`INPUT`	[point cloud]	Input point cloud layer to export
Filter expression 可选	`FILTER_EXPRESS`	[expression]	A *PDAL expression* for selecting a subset of features in the point cloud data
Cropping extent 可选	`FILTER_EXTENT`	[范围]	A map extent for selecting a subset of features in the point cloud data Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`

continues on next page

Table 24.299 – continued from previous page

标注	名称	类型	描述
Filtered	`OUTPUT`	[point cloud] 默认：[保存到临时文件]	Specify the point cloud file to export the data to. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Filtered	`OUTPUT`	[point cloud]	Output point cloud layer with the filtered features of the input point cloud layer. Currently supported formats are `.LAS`, `.LAZ`, `.COPC.LAZ` and `.VPC`.

Python 代码

Algorithm ID: `pdal:filter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.18 栅格分析

Cell stack percent rank from value

Calculates the cell-wise percentrank value of a stack of rasters based on a single input value and writes them to an output raster.

At each cell location, the specified value is ranked among the respective values in the stack of all overlaid and sorted cell values from the input rasters. For values outside of the stack value distribution, the algorithm returns NoData because the value cannot be ranked among the cell values.

There are two methods for percentile calculation:

- Inclusive linear interpolation (PERCENTRANK.INC)
- Exclusive linear interpolation (PERCENTRANK.EXC)

The linear interpolation methods return the unique percent rank for different values. Both interpolation methods follow their counterpart methods implemented by LibreOffice or Microsoft Excel.

The output raster's extent and resolution is defined by a reference raster. Input raster layers that do not match the cell size of the reference raster layer will be resampled using nearest neighbor resampling. NoData values in any of the input layers will result in a NoData cell output if the "Ignore NoData values" parameter is not set. The output raster data type will always be `Float32`.

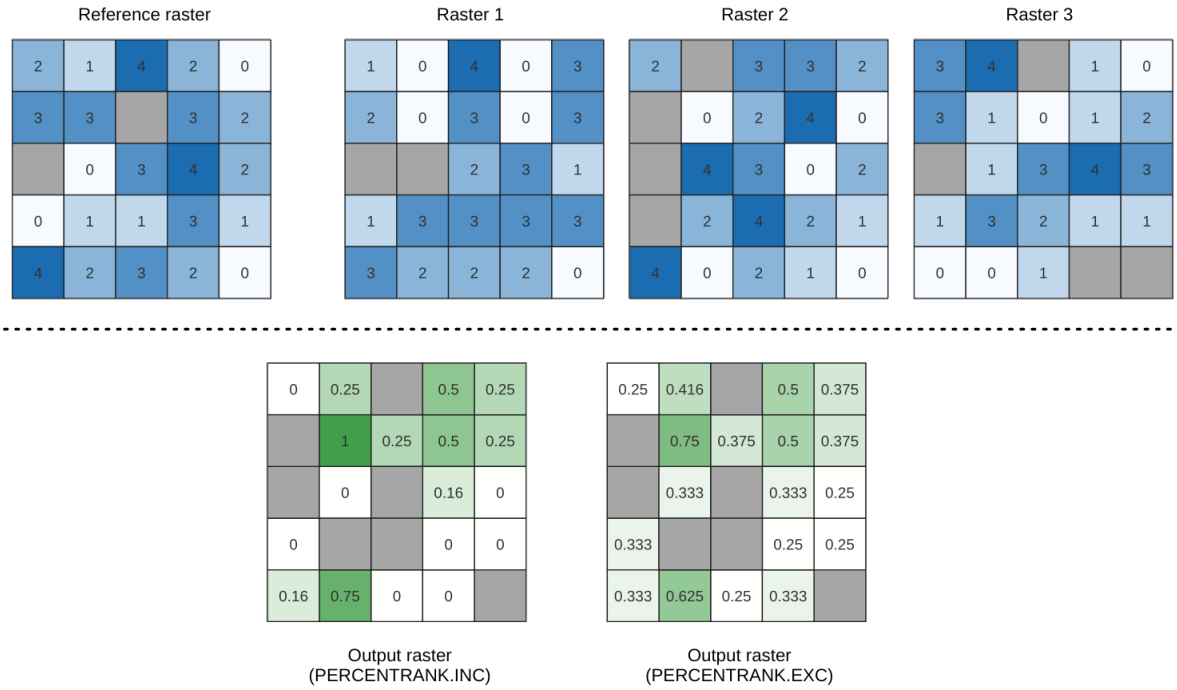


Fig. 24.49: Percent ranking Value = 1. `NoData` cells (grey) are ignored.

See also

Cell stack percentile, *Cell stack percentrank from raster layer*

参数

Basic parameters

标注	名称	类型	描述
Input layers	`INPUT`	[raster] [list]	Raster layers to evaluate. If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters
Method	`METHOD`	[enumeration] Default: 0	Method for percentile calculation: • 0 --- Inclusive linear interpolation (PERCENTRANK.INC) • 1 --- Exclusive linear interpolation (PERCENTRANK.EXC)
Value	`VALUE`	[numeric: double] Default: 10.0	Value to rank among the respective values in the stack of all overlaid and sorted cell values from the input rasters
Ignore NoData values	`IGNORE_NODATA`	[boolean] Default: True	If unchecked, any NoData cells in the input layers will result in a NoData cell in the output raster
Reference layer	`REFERENCE_LAYE`	[栅格]	The reference layer for the output layer creation (extent, CRS, pixel dimensions)

continues on next page

Table 24.301 – continued from previous page

标注	名称	类型	描述
Output layer	OUTPUT	[same as input] 默认：[保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output NoData value	OUTPUT_NODATA_	[numeric: double] Default: -9999.0	Value to use for NoData in the output layer
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output layer	OUTPUT	[栅格]	Output raster layer containing the result
CRS authority identifier	CRS_AUTHID	[string]	The coordinate reference system of the output raster layer
Extent	EXTENT	[string]	The spatial extent of the output raster layer
Width in pixels	WIDTH_IN_PIXEL	[numeric: integer]	The number of columns in the output raster layer
Height in pixels	HEIGHT_IN_PIXE	[numeric: integer]	The number of rows in the output raster layer
Total pixel count	TOTAL_PIXEL_CO	[numeric: integer]	The count of pixels in the output raster layer

Python 代码

Algorithm ID: `native:cellstackpercentrankfromvalue`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

Cell stack percentile

Calculates the cell-wise percentile value of a stack of rasters and writes the results to an output raster. The percentile to return is determined by the percentile input value (ranges between 0 and 1). At each cell location, the specified percentile is obtained using the respective value from the stack of all overlaid and sorted cell values of the input rasters.

There are three methods for percentile calculation:

- Nearest rank: returns the value that is nearest to the specified percentile
- Inclusive linear interpolation (PERCENTRANK.INC)
- Exclusive linear interpolation (PERCENTRANK.EXC)

The linear interpolation methods return the unique values for different percentiles. Both interpolation methods follow their counterpart methods implemented by LibreOffice or Microsoft Excel.

The output raster's extent and resolution is defined by a reference raster. Input raster layers that do not match the cell size of the reference raster layer will be resampled using nearest neighbor resampling. NoData values in any of the input layers will result in a NoData cell output if the "Ignore NoData values" parameter is not set. The output raster data type will always be `Float32`.

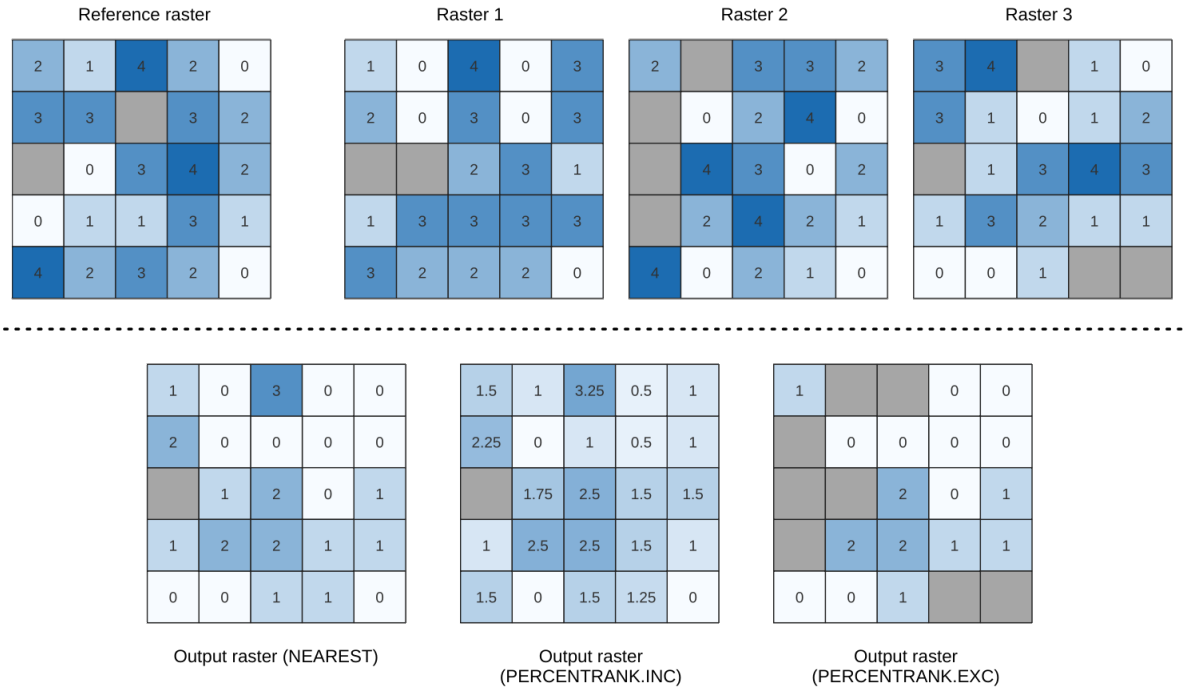


Fig. 24.50: Percentile = 0.25. `NoData` cells (grey) are ignored.

See also

Cell stack percentile, *Cell stack percentrank from raster layer*

参数

Basic parameters

标注	名称	类型	描述
Input layers	INPUT	[raster] [list]	Raster layers to evaluate. If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters
Method	METHOD	[enumeration] Default: 0	Method for percentile calculation: • 0 --- Nearest rank: returns the value that is nearest to the specified percentile • 1 --- Inclusive linear interpolation (PERCENTILE.INC) • 2 --- Exclusive linear interpolation (PERCENTILE.EXC)
Percentile	VALUE	[numeric: double] Default: 0.25	Value to rank among the respective values in the stack of all overlaid and sorted cell values from the input rasters. Between 0 and 1.
Ignore NoData values	IGNORE_NODATA	[boolean] Default: True	If unchecked, any NoData cells in the input layers will result in a NoData cell in the output raster
Reference layer	REFERENCE_LAYE	[栅格]	The reference layer for the output layer creation (extent, CRS, pixel dimensions)
Output layer	OUTPUT	[same as input] 默认: [保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output NoData value	OUTPUT_NODATA_	[numeric: double] Default: -9999.0	Value to use for NoData in the output layer
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output layer	`OUTPUT`	[栅格]	Output raster layer containing the result
CRS authority identifier	`CRS_AUTHID`	[string]	The coordinate reference system of the output raster layer
Extent	`EXTENT`	[string]	The spatial extent of the output raster layer
Width in pixels	`WIDTH_IN_PIXEL`	[numeric: integer]	The number of columns in the output raster layer
Height in pixels	`HEIGHT_IN_PIXE`	[numeric: integer]	The number of rows in the output raster layer
Total pixel count	`TOTAL_PIXEL_CO`	[numeric: integer]	The count of pixels in the output raster layer

Python 代码

Algorithm ID: `native:cellstackpercentile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Cell stack percentrank from raster layer

Calculates the cell-wise percentrank value of a stack of rasters based on an input value raster and writes them to an output raster.

At each cell location, the current value of the value raster is ranked among the respective values in the stack of all overlaid and sorted cell values of the input rasters. For values outside of the the stack value distribution, the algorithm returns NoData because the value cannot be ranked among the cell values.

There are two methods for percentile calculation:

- Inclusive linear interpolation (PERCENTRANK.INC)
- Exclusive linear interpolation (PERCENTRANK.EXC)

The linear interpolation methods return the unique values for different percentiles. Both interpolation methods follow their counterpart methods implemented by LibreOffice or Microsoft Excel.

The output raster's extent and resolution is defined by a reference raster. Input raster layers that do not match the cell size of the reference raster layer will be resampled using nearest neighbor resampling. NoData values in any of the input layers will result in a NoData cell output if the "Ignore NoData values" parameter is not set. The output raster data type will always be `Float32`.

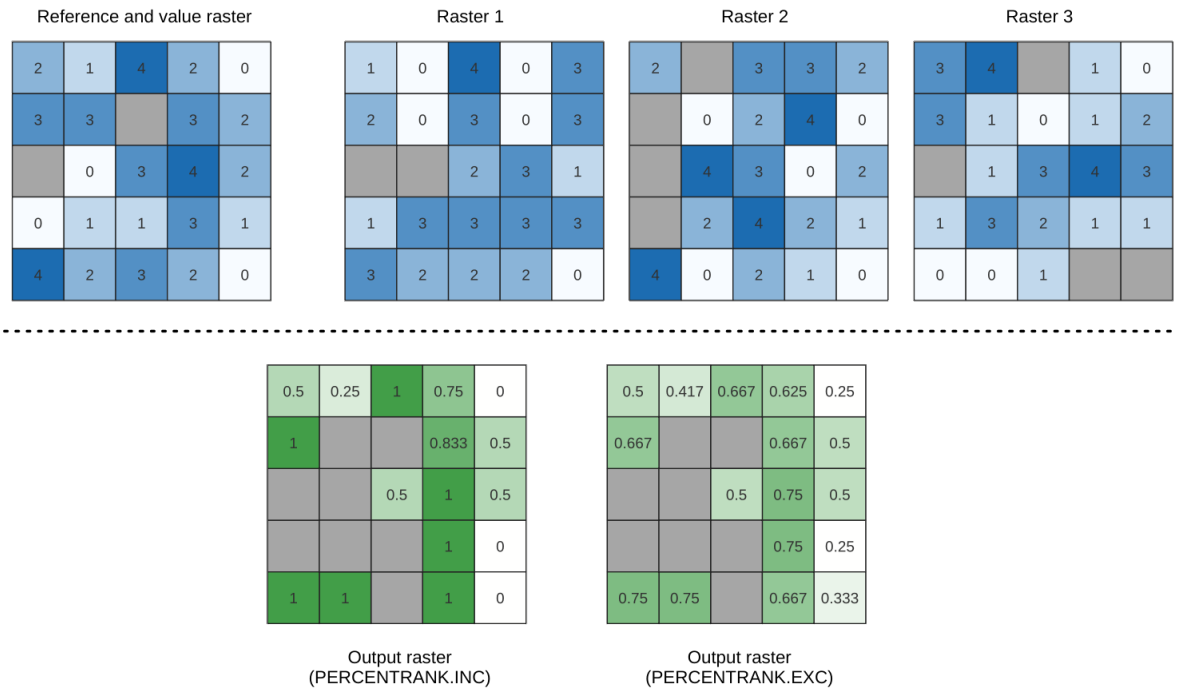


Fig. 24.51: Ranking the value raster layer cells. `NoData` cells (grey) are ignored.

See also

Cell stack percentile, *Cell stack percent rank from value*

参数

Basic parameters

标注	名称	类型	描述
Input layers	`INPUT`	[raster] [list]	Raster layers to evaluate. If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters
Value raster layer	`INPUT_VALUE_RA`	[栅格]	The layer to rank the values among the stack of all overlaid layers
Value raster band	`VALUE_RASTER_B`	[numeric: integer] Default: 1	Band of the "value raster layer" to compare to
Method	`METHOD`	[enumeration] Default: 0	Method for percentile calculation: • 0 --- Inclusive linear interpolation (PERCENTRANK.INC) • 1 --- Exclusive linear interpolation (PERCENTRANK.EXC)
Ignore NoData values	`IGNORE_NODATA`	[boolean] Default: True	If unchecked, any NoData cells in the input layers will result in a NoData cell in the output raster
Reference layer	`REFERENCE_LAYE`	[栅格]	The reference layer for the output layer creation (extent, CRS, pixel dimensions)

continues on next page

Table 24.307 – continued from previous page

标注	名称	类型	描述
Output layer	OUTPUT	[same as input] 默认：[保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output NoData value	OUTPUT_NODATA_	[numeric: double] Default: -9999.0	Value to use for NoData in the output layer
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output layer	OUTPUT	[栅格]	Output raster layer containing the result
CRS authority identifier	CRS_AUTHID	[string]	The coordinate reference system of the output raster layer
Extent	EXTENT	[string]	The spatial extent of the output raster layer
Width in pixels	WIDTH_IN_PIXEL	[numeric: integer]	The number of columns in the output raster layer
Height in pixels	HEIGHT_IN_PIXE	[numeric: integer]	The number of rows in the output raster layer
Total pixel count	TOTAL_PIXEL_CO	[numeric: integer]	The count of pixels in the output raster layer

Python 代码

Algorithm ID: `native:cellstackpercentrankfromrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

Cell statistics

Computes per-cell statistics based on input raster layers and for each cell writes the resulting statistics to an output raster. At each cell location, the output value is defined as a function of all overlaid cell values of the input rasters.

By default, a NoData cell in ANY of the input layers will result in a NoData cell in the output raster. If the *Ignore NoData values* option is checked, then NoData inputs will be ignored in the statistic calculation. This may result in NoData output for locations where all cells are NoData.

The *Reference layer* parameter specifies an existing raster layer to use as a reference when creating the output raster. The output raster will have the same extent, CRS, and pixel dimensions as this layer.

Calculation details: Input raster layers that do not match the cell size of the reference raster layer will be resampled using `nearest neighbor resampling`. The output raster data type will be set to the most complex data type present in the input datasets except when using the functions `Mean`, `Standard deviation` and `Variance` (data type is always `Float32` or `Float64` depending on input float type) or `Count` and `Variety` (data type is always `Int32`).

- `Count`: The count statistic will always result in the number of cells without NoData values at the current cell location.
- `Median`: If the number of input layers is even, the median will be calculated as the arithmetic mean of the two middle values of the ordered cell input values.
- `Minority/Majority`: If no unique minority or majority could be found, the result is NoData, except all input cell values are equal.

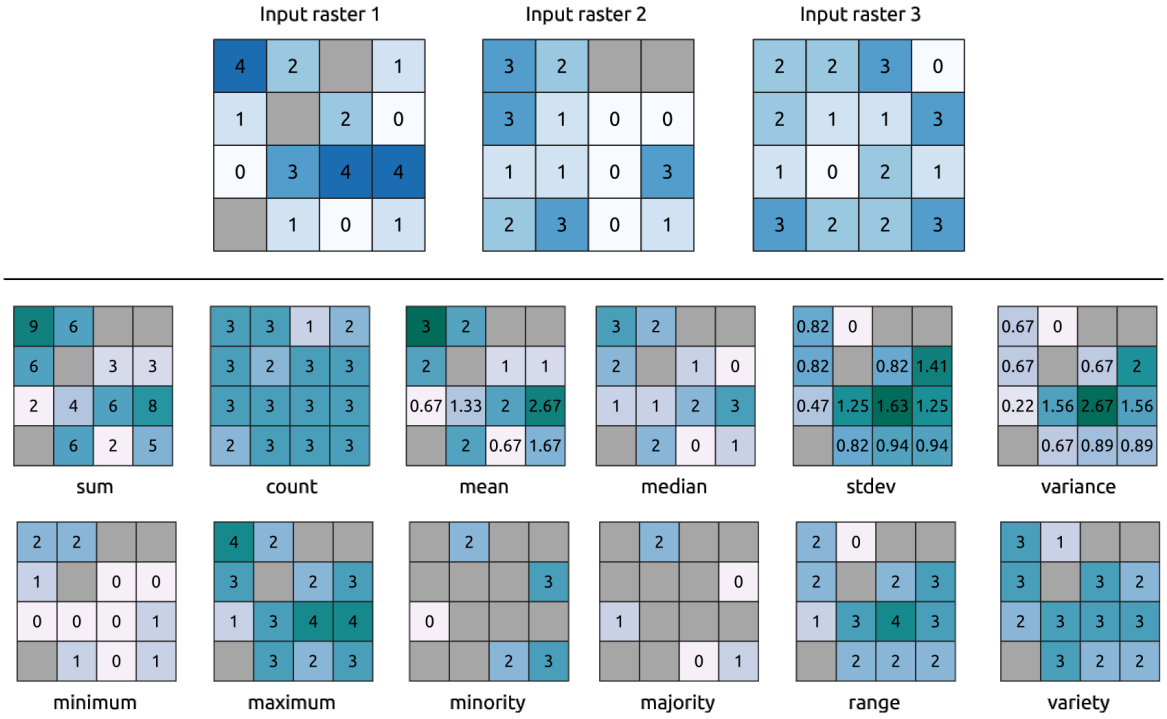


Fig. 24.52: Example with all the statistic functions. `NoData` cells (grey) are taken into account.

参数

Basic parameters

标注	名称	类型	描述
Input layers	INPUT	[raster] [list]	Input raster layers
Statistic	STATISTIC	[enumeration] Default: 0	Available statistics. Options: • 0 --- Sum • 1 --- Count • 2 --- Mean • 3 --- Median • 4 --- Standard deviation • 5 --- Variance • 6 --- Minimum • 7 --- Maximum • 8 --- Minority (least common value) • 9 --- Majority (most common value) • 10 --- Range (max - min) • 11 --- Variety (unique value count)
Ignore NoData values	IGNORE_NODATA	[boolean] Default: True	Calculate statistics also for all cells stacks, ignoring NoData occurrence.
Reference layer	REF_LAYER	[栅格]	The reference layer to create the output layer from (extent, CRS, pixel dimensions)
Output layer	OUTPUT	[same as input] 默认: [保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output NoData value 可选	OUTPUT_NO_DATA	[numeric: double] Default: -9999.0	Value to use for NoData in the output layer
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
CRS authority identifier	CRS_AUTHID	[crs]	The coordinate reference system of the output raster layer
Extent	EXTENT	[string]	The spatial extent of the output raster layer
Height in pixels	HEIGHT_IN_PIXE	[numeric: integer]	The number of rows in the output raster layer
Output raster	OUTPUT	[栅格]	Output raster layer containing the result
Total pixel count	TOTAL_PIXEL_CO	[numeric: integer]	The count of pixels in the output raster layer
Width in pixels	WIDTH_IN_PIXEL	[numeric: integer]	The number of columns in the output raster layer

Python 代码

Algorithm ID: native:cellstatistics

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Equal to frequency

Evaluates on a cell-by-cell basis the frequency (number of times) the values of an input stack of rasters are equal to the value of a value layer. The output raster extent and resolution are defined by the input raster layer and is always of Int32 type.

If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters - use GDAL to use other bands in the analysis. The output NoData value can be set manually.

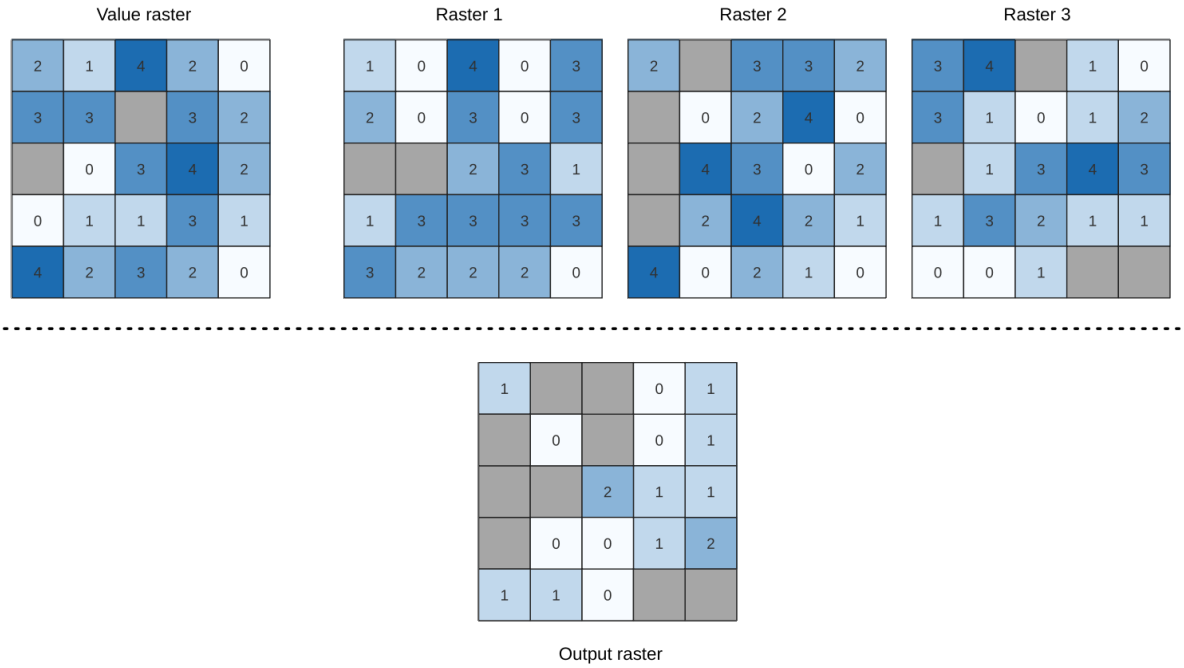


Fig. 24.53: For each cell in the output raster, the value represents the number of times that the corresponding cells in the list of rasters are the same as the value raster. `NoData` cells (grey) are taken into account.

See also

Greater than frequency, *Less than frequency*

参数

Basic parameters

标注	名称	类型	描述
Input value raster	`INPUT_VALUE_RA`	[栅格]	The input value layer serves as reference layer for the sample layers
Value raster band	`INPUT_VALUE_RA`	[raster band] Default: The first band of the raster layer	Select the band you want to use as sample
Input raster layers	`INPUT_RASTERS`	[raster] [list]	Raster layers to evaluate. If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters
Ignore NoData values	`IGNORE_NODATA`	[boolean] Default: False	If unchecked, any NoData cells in the value raster or the data layer stack will result in a NoData cell in the output raster
Output layer	`OUTPUT`	[same as input] 默认: [保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output NoData value 可选	OUTPUT_NO_DATA	[numeric: double] Default: -9999.0	Value to use for NoData in the output layer
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output layer	OUTPUT	[栅格]	Output raster layer containing the result
CRS authority identifier	CRS_AUTHID	[string]	The coordinate reference system of the output raster layer
Extent	EXTENT	[string]	The spatial extent of the output raster layer
Count of cells with equal value occurrences	FOUND_LOCATION	[numeric: integer]	
Height in pixels	HEIGHT_IN_PIXE	[numeric: integer]	The number of rows in the output raster layer
Total pixel count	TOTAL_PIXEL_CO	[numeric: integer]	The count of pixels in the output raster layer
Mean frequency at valid cell locations	MEAN_FREQUENCY	[numeric: double]	
Count of value occurrences	OCCURRENCE_COU	[numeric: integer]	
Width in pixels	WIDTH_IN_PIXEL	[numeric: integer]	The number of columns in the output raster layer

Python 代码

Algorithm ID: `native:equaltofrequency`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Fuzzify raster (gaussian membership)

Transforms an input raster to a fuzzified raster by assigning a membership value to each pixel, using a Gaussian membership function. Membership values range from 0 to 1. In the fuzzified raster, a value of 0 implies no membership of the defined fuzzy set, whereas a value of 1 means full membership. The gaussian membership function is defined as $\mu(x) = e^{-f1*(x-f2)^2}$, where *f1* is the spread and *f2* the midpoint.

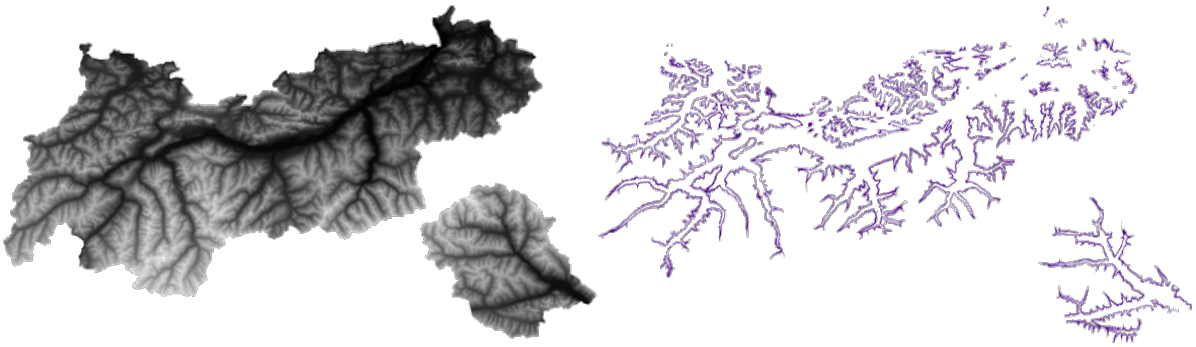


Fig. 24.54: Fuzzify raster example. Input raster source: Land Tirol - data.tirol.gv.at.

See also

Fuzzify raster (large membership), *Fuzzify raster (linear membership)*, *Fuzzify raster (near membership)*, *Fuzzify raster (power membership)*, *Fuzzify raster (small membership)*

参数

Basic parameters

标注	名称	类型	描述
Input Raster	`INPUT`	[栅格]	Input raster layer
Band Number	`BAND`	[raster band] Default: The first band of the raster layer	If the raster is multiband, choose the band that you want to fuzzify.
Function midpoint	`FUZZYMIDPOINT`	[numeric: double] Default: 10.0	Midpoint of the gaussian function
Function spread	`FUZZYSPREAD`	[numeric: double] Default: 0.01	Spread of the gaussian function
Fuzzified raster	`OUTPUT`	[same as input] 默认：[保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Fuzzified raster	OUTPUT	[same as input]	Output raster layer containing the result
CRS authority identifier	CRS_AUTHID	[crs]	The coordinate reference system of the output raster layer
Extent	EXTENT	[string]	The spatial extent of the output raster layer
Width in pixels	WIDTH_IN_PIXEL	[numeric: integer]	The number of columns in the output raster layer
Height in pixels	HEIGHT_IN_PIXE	[numeric: integer]	The number of rows in the output raster layer
Total pixel count	TOTAL_PIXEL_CO	[numeric: integer]	The count of pixels in the output raster layer

Python 代码

Algorithm ID: `native:fuzzifyrastergaussianmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Fuzzify raster (large membership)

Transforms an input raster to a fuzzified raster by assigning a membership value to each pixel, using a Large membership function. Membership values range from 0 to 1. In the fuzzified raster, a value of 0 implies no membership of the defined fuzzy set, whereas a value of 1 means full membership. The large membership function is defined as $\mu(x) = \frac{1}{1+\left(\frac{x}{f2}\right)^{-f1}}$, where *f1* is the spread and *f2* the midpoint.

See also

Fuzzify raster (gaussian membership), *Fuzzify raster (linear membership)*, *Fuzzify raster (near membership)*, *Fuzzify raster (power membership)*, *Fuzzify raster (small membership)*

参数

Basic parameters

标注	名称	类型	描述
Input Raster	INPUT	[栅格]	Input raster layer
Band Number	BAND	[raster band] Default: The first band of the raster layer	If the raster is multiband, choose the band that you want to fuzzify.
Function midpoint	FUZZYMIDPOINT	[numeric: double] Default: 50.0	Midpoint of the large function
Function spread	FUZZYSPREAD	[numeric: double] Default: 5.0	Spread of the large function
Fuzzified raster	OUTPUT	[same as input] 默认: [保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Fuzzified raster	OUTPUT	[same as input]	Output raster layer containing the result
CRS authority identifier	CRS_AUTHID	[crs]	The coordinate reference system of the output raster layer
Extent	EXTENT	[string]	The spatial extent of the output raster layer
Width in pixels	WIDTH_IN_PIXEL	[numeric: integer]	The number of columns in the output raster layer
Height in pixels	HEIGHT_IN_PIXE	[numeric: integer]	The number of rows in the output raster layer
Total pixel count	TOTAL_PIXEL_CO	[numeric: integer]	The count of pixels in the output raster layer

Python 代码

Algorithm ID: `native:fuzzifyrasterlargemembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Fuzzify raster (linear membership)

Transforms an input raster to a fuzzified raster by assigning a membership value to each pixel, using a Linear membership function. Membership values range from 0 to 1. In the fuzzified raster, a value of 0 implies no membership of the defined fuzzy set, whereas a value of 1 means full membership. The linear function is defined as $\mu(X)\begin{cases}0 & x \leq a \\ \frac{x-a}{b-a} & a < x < b \\ 1 & x \geq b\end{cases}$, where a is the low bound and b the high bound. This equation assigns membership values using a linear transformation for pixel values between the low and high bounds. Pixels values smaller than the low bound are given 0 membership whereas pixel values greater than the high bound are given 1 membership.

See also

Fuzzify raster (gaussian membership), *Fuzzify raster (large membership)*, *Fuzzify raster (near membership)*, *Fuzzify raster (power membership)*, *Fuzzify raster (small membership)*

参数

Basic parameters

标注	名称	类型	描述
Input Raster	`INPUT`	[栅格]	Input raster layer
Band Number	`BAND`	[raster band] Default: The first band of the raster layer	If the raster is multiband, choose the band that you want to fuzzify.
Low fuzzy membership bound	`FUZZYLOWBOUND`	[numeric: double] Default: 0.0	Low bound of the linear function
High fuzzy membership bound	`FUZZYHIGHBOUND`	[numeric: double] 默认值：1.0	High bound of the linear function
Fuzzified raster	`OUTPUT`	[same as input] 默认：[保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Fuzzified raster	OUTPUT	[same as input]	Output raster layer containing the result
CRS authority identifier	CRS_AUTHID	[crs]	The coordinate reference system of the output raster layer
Extent	EXTENT	[string]	The spatial extent of the output raster layer
Width in pixels	WIDTH_IN_PIXEL	[numeric: integer]	The number of columns in the output raster layer
Height in pixels	HEIGHT_IN_PIXE	[numeric: integer]	The number of rows in the output raster layer
Total pixel count	TOTAL_PIXEL_CO	[numeric: integer]	The count of pixels in the output raster layer

Python 代码

Algorithm ID: `native:fuzzifyrasterlinearmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Fuzzify raster (near membership)

Transforms an input raster to a fuzzified raster by assigning a membership value to each pixel, using a Near membership function. Membership values range from 0 to 1. In the fuzzified raster, a value of 0 implies no membership of the defined fuzzy set, whereas a value of 1 means full membership. The near membership function is defined as $\mu(x) = \frac{1}{1 + f1 * (x - f2)^2}$, where *f1* is the spread and *f2* the midpoint.

See also

Fuzzify raster (gaussian membership), *Fuzzify raster (large membership)*, *Fuzzify raster (linear membership)*, *Fuzzify raster (power membership)*, *Fuzzify raster (small membership)*

参数

Basic parameters

标注	名称	类型	描述
Input Raster	INPUT	[栅格]	Input raster layer
Band Number	BAND	[raster band] Default: The first band of the raster layer	If the raster is multiband, choose the band that you want to fuzzify.
Function midpoint	FUZZYMIDPOINT	[numeric: double] Default: 50.0	Midpoint of the near function
Function spread	FUZZYSPREAD	[numeric: double] Default: 0.01	Spread of the near function
Fuzzified raster	OUTPUT	[same as input] 默认: [保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Fuzzified raster	OUTPUT	[same as input]	Output raster layer containing the result
CRS authority identifier	CRS_AUTHID	[crs]	The coordinate reference system of the output raster layer
Extent	EXTENT	[string]	The spatial extent of the output raster layer
Width in pixels	WIDTH_IN_PIXEL	[numeric: integer]	The number of columns in the output raster layer
Height in pixels	HEIGHT_IN_PIXE	[numeric: integer]	The number of rows in the output raster layer
Total pixel count	TOTAL_PIXEL_CO	[numeric: integer]	The count of pixels in the output raster layer

Python 代码

Algorithm ID: `native:fuzzifyrasternearmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Fuzzify raster (power membership)

Transforms an input raster to a fuzzified raster by assigning a membership value to each pixel, using a Power membership function. Membership values range from 0 to 1. In the fuzzified raster, a value of 0 implies no membership of the defined fuzzy set, whereas a value of 1 means full membership. The power function is defined as $\mu(x)\begin{cases}0 & x \le a\\ \left(\frac{x-a}{b-a}\right)^{f1} & a < x < b\\ 1 & x \ge b\end{cases}$, where a is the low bound, b is the high bound, and $f1$ the exponent. This equation assigns membership values using the power transformation for pixel values between the low and high bounds. Pixels values smaller than the low bound are given 0 membership whereas pixel values greater than the high bound are given 1 membership.

See also

Fuzzify raster (gaussian membership), *Fuzzify raster (large membership)*, *Fuzzify raster (linear membership)*, *Fuzzify raster (near membership)*, *Fuzzify raster (small membership)*

参数

Basic parameters

标注	名称	类型	描述
Input Raster	`INPUT`	[栅格]	Input raster layer
Band Number	`BAND`	[raster band] Default: The first band of the raster layer	If the raster is multiband, choose the band that you want to fuzzify.
Low fuzzy membership bound	`FUZZYLOWBOUND`	[numeric: double] Default: 0.0	Low bound of the power function
High fuzzy membership bound	`FUZZYHIGHBOUND`	[numeric: double] 默认值：1.0	High bound of the power function
High fuzzy membership bound	`FUZZYEXPONENT`	[numeric: double] Default: 2.0	Exponent of the power function
Fuzzified raster	`OUTPUT`	[same as input] 默认：[保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Fuzzified raster	OUTPUT	[same as input]	Output raster layer containing the result
CRS authority identifier	CRS_AUTHID	[crs]	The coordinate reference system of the output raster layer
Extent	EXTENT	[string]	The spatial extent of the output raster layer
Width in pixels	WIDTH_IN_PIXEL	[numeric: integer]	The number of columns in the output raster layer
Height in pixels	HEIGHT_IN_PIXE	[numeric: integer]	The number of rows in the output raster layer
Total pixel count	TOTAL_PIXEL_CO	[numeric: integer]	The count of pixels in the output raster layer

Python 代码

Algorithm ID: `native:fuzzifyrasterpowermembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Fuzzify raster (small membership)

Transforms an input raster to a fuzzified raster by assigning a membership value to each pixel, using a Small membership function. Membership values range from 0 to 1. In the fuzzified raster, a value of 0 implies no membership of the defined fuzzy set, whereas a value of 1 means full membership. The small membership function is defined as $\mu(x) = \frac{1}{1+\left(\frac{x}{f2}\right)^{f1}}$, where *f1* is the spread and *f2* the midpoint.

See also

Fuzzify raster (gaussian membership), *Fuzzify raster (large membership)* *Fuzzify raster (linear membership)*, *Fuzzify raster (near membership)*, *Fuzzify raster (power membership)*

参数

Basic parameters

标注	名称	类型	描述
Input Raster	INPUT	[栅格]	Input raster layer
Band Number	BAND	[raster band] Default: The first band of the raster layer	If the raster is multiband, choose the band that you want to fuzzify.
Function midpoint	FUZZYMIDPOINT	[numeric: double] Default: 50.0	Midpoint of the small function
Function spread	FUZZYSPREAD	[numeric: double] Default: 5.0	Spread of the small function
Fuzzified raster	OUTPUT	[same as input] 默认: [保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Fuzzified raster	OUTPUT	[same as input]	Output raster layer containing the result
CRS authority identifier	CRS_AUTHID	[crs]	The coordinate reference system of the output raster layer
Extent	EXTENT	[string]	The spatial extent of the output raster layer
Width in pixels	WIDTH_IN_PIXEL	[numeric: integer]	The number of columns in the output raster layer
Height in pixels	HEIGHT_IN_PIXE	[numeric: integer]	The number of rows in the output raster layer
Total pixel count	TOTAL_PIXEL_CO	[numeric: integer]	The count of pixels in the output raster layer

Python 代码

Algorithm ID: `native:fuzzifyrastersmallmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Greater than frequency

Evaluates on a cell-by-cell basis the frequency (number of times) the values of an input stack of rasters are equal to the value of a value raster. The output raster extent and resolution is defined by the input raster layer and is always of `Int32` type.

If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters - use GDAL to use other bands in the analysis. The output NoData value can be set manually.

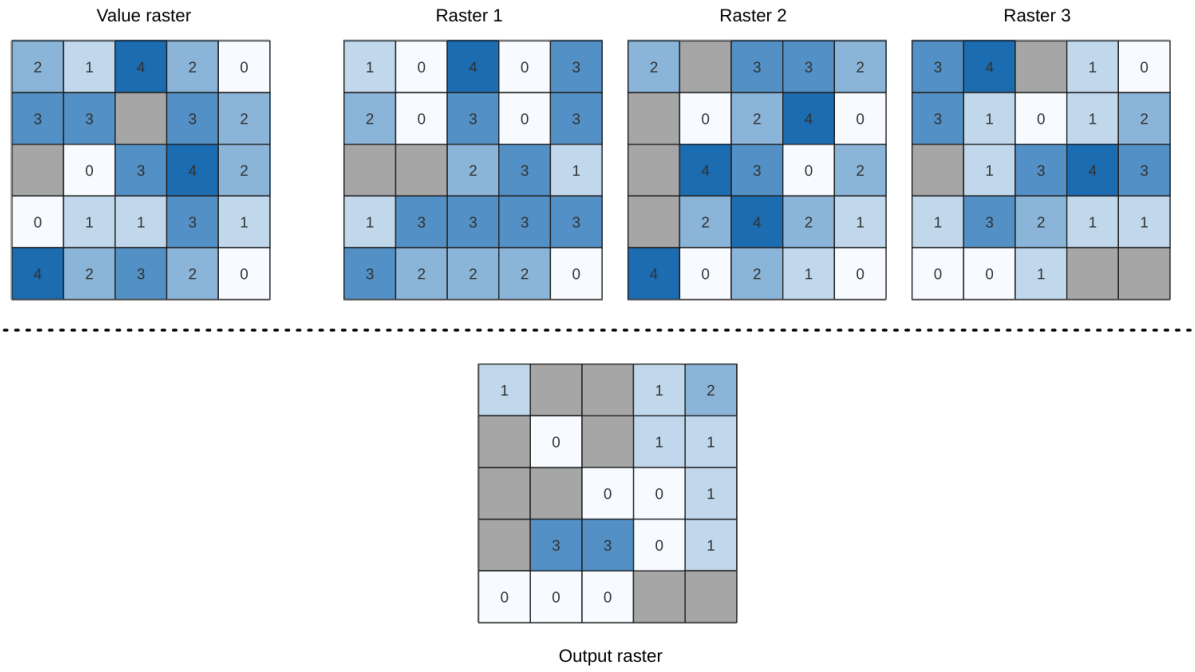


Fig. 24.55: For each cell in the output raster, the value represents the number of times that the corresponding cells in the list of rasters are greater than the value raster. `NoData` cells (grey) are taken into account.

See also

Equal to frequency, *Less than frequency*

参数

Basic parameters

标注	名称	类型	描述
Input value raster	INPUT_VALUE_RA	[栅格]	The input value layer serves as reference layer for the sample layers
Value raster band	INPUT_VALUE_RA	[raster band] Default: The first band of the raster layer	Select the band you want to use as sample
Input raster layers	INPUT_RASTERS	[raster] [list]	Raster layers to evaluate. If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters
Ignore NoData values	IGNORE_NODATA	[boolean] Default: False	If unchecked, any NoData cells in the value raster or the data layer stack will result in a NoData cell in the output raster
Output layer	OUTPUT	[same as input] 默认：[保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output NoData value 可选	OUTPUT_NO_DATA	[numeric: double] Default: -9999.0	Value to use for NoData in the output layer
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output layer	OUTPUT	[栅格]	Output raster layer containing the result
CRS authority identifier	CRS_AUTHID	[string]	The coordinate reference system of the output raster layer
Extent	EXTENT	[string]	The spatial extent of the output raster layer
Count of cells with equal value occurrences	FOUND_LOCATION	[numeric: integer]	

continues on next page

Table 24.336 – continued from previous page

标注	名称	类型	描述
Height in pixels	HEIGHT_IN_PIXE	[numeric: integer]	The number of rows in the output raster layer
Total pixel count	TOTAL_PIXEL_CO	[numeric: integer]	The count of pixels in the output raster layer
Mean frequency at valid cell locations	MEAN_FREQUENCY	[numeric: double]	
Count of value occurrences	OCCURRENCE_COU	[numeric: integer]	
Width in pixels	WIDTH_IN_PIXEL	[numeric: integer]	The number of columns in the output raster layer

Python 代码

Algorithm ID: `native:greaterthanfrequency`

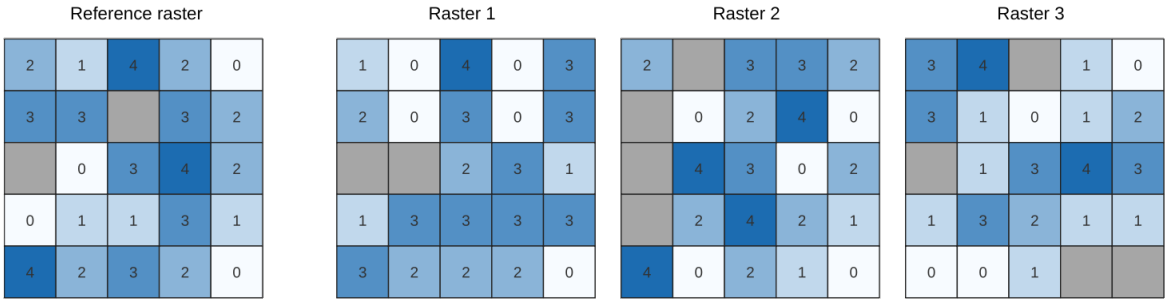
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

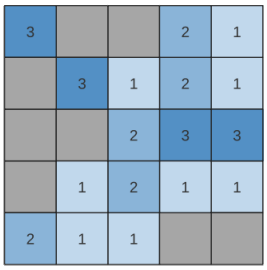
当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Highest position in raster stack

Evaluates on a cell-by-cell basis the position of the raster with the highest value in a stack of rasters. Position counts start with 1 and range to the total number of input rasters. The order of the input rasters is relevant for the algorithm. If multiple rasters feature the highest value, the first raster will be used for the position value.

If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters - use GDAL to use other bands in the analysis. Any NoData cells in the raster layer stack will result in a NoData cell in the output raster unless the "ignore NoData" parameter is checked. The output NoData value can be set manually. The output rasters extent and resolution is defined by a reference raster layer and is always of `Int32` type.





Output raster

See also

Lowest position in raster stack

参数

Basic parameters

标注	名称	类型	描述
Input raster layers	INPUT_RASTERS	[raster] [list]	List of raster layers to compare with
Reference layer	REFERENCE_LAYE	[栅格]	The reference layer for the output layer creation (extent, CRS, pixel dimensions)
Ignore NoData values	IGNORE_NODATA	[boolean] Default: False	If unchecked, any NoData cells in the data layer stack will result in a NoData cell in the output raster
Output layer	OUTPUT	[栅格] 默认: [保存到临时文件]	Specification of the output raster containing the result. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output NoData value	OUTPUT_NODATA_	[numeric: double] Default: -9999.0	Value to use for NoData in the output layer
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output layer	OUTPUT	[栅格]	Output raster layer containing the result
CRS authority identifier	CRS_AUTHID	[string]	The coordinate reference system of the output raster layer
Extent	EXTENT	[string]	The spatial extent of the output raster layer
Width in pixels	WIDTH_IN_PIXEL	[numeric: integer]	The number of columns in the output raster layer
Height in pixels	HEIGHT_IN_PIXE	[numeric: integer]	The number of rows in the output raster layer
Total pixel count	TOTAL_PIXEL_CO	[numeric: integer]	The count of pixels in the output raster layer

Python 代码

Algorithm ID: `native:highestpositioninrasterstack`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Less than frequency

Evaluates on a cell-by-cell basis the frequency (number of times) the values of an input stack of rasters are less than the value of a value raster. The output raster extent and resolution is defined by the input raster layer and is always of `Int32` type.

If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters - use GDAL to use other bands in the analysis. The output NoData value can be set manually.

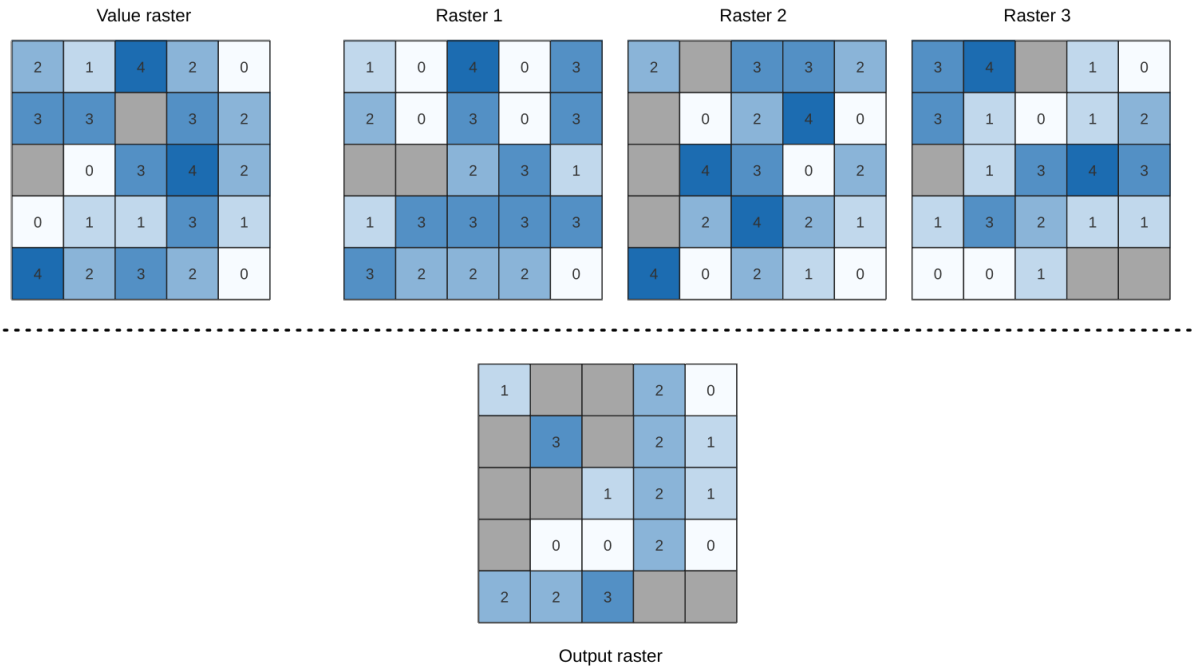


Fig. 24.56: For each cell in the output raster, the value represents the number of times that the corresponding cells in the list of rasters are less than the value raster. `NoData` cells (grey) are taken into account.

See also

Equal to frequency, *Greater than frequency*

参数

Basic parameters

标注	名称	类型	描述
Input value raster	INPUT_VALUE_RA	[栅格]	The input value layer serves as reference layer for the sample layers
Value raster band	INPUT_VALUE_RA	[raster band] Default: The first band of the raster layer	Select the band you want to use as sample
Input raster layers	INPUT_RASTERS	[raster] [list]	Raster layers to evaluate. If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters
Ignore NoData values	IGNORE_NODATA	[boolean] Default: False	If unchecked, any NoData cells in the value raster or the data layer stack will result in a NoData cell in the output raster
Output layer	OUTPUT	[same as input] 默认：[保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output NoData value 可选	OUTPUT_NO_DATA	[numeric: double] Default: -9999.0	Value to use for NoData in the output layer
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output layer	OUTPUT	[栅格]	Output raster layer containing the result
CRS authority identifier	CRS_AUTHID	[string]	The coordinate reference system of the output raster layer
Extent	EXTENT	[string]	The spatial extent of the output raster layer
Count of cells with equal value occurrences	FOUND_LOCATION	[numeric: integer]	

continues on next page

Table 24.342 – continued from previous page

标注	名称	类型	描述
Height in pixels	HEIGHT_IN_PIXE	[numeric: integer]	The number of rows in the output raster layer
Total pixel count	TOTAL_PIXEL_CO	[numeric: integer]	The count of pixels in the output raster layer
Mean frequency at valid cell locations	MEAN_FREQUENCY	[numeric: double]	
Count of value occurrences	OCCURRENCE_COU	[numeric: integer]	
Width in pixels	WIDTH_IN_PIXEL	[numeric: integer]	The number of columns in the output raster layer

Python 代码

Algorithm ID: `native:lessthanfrequency`

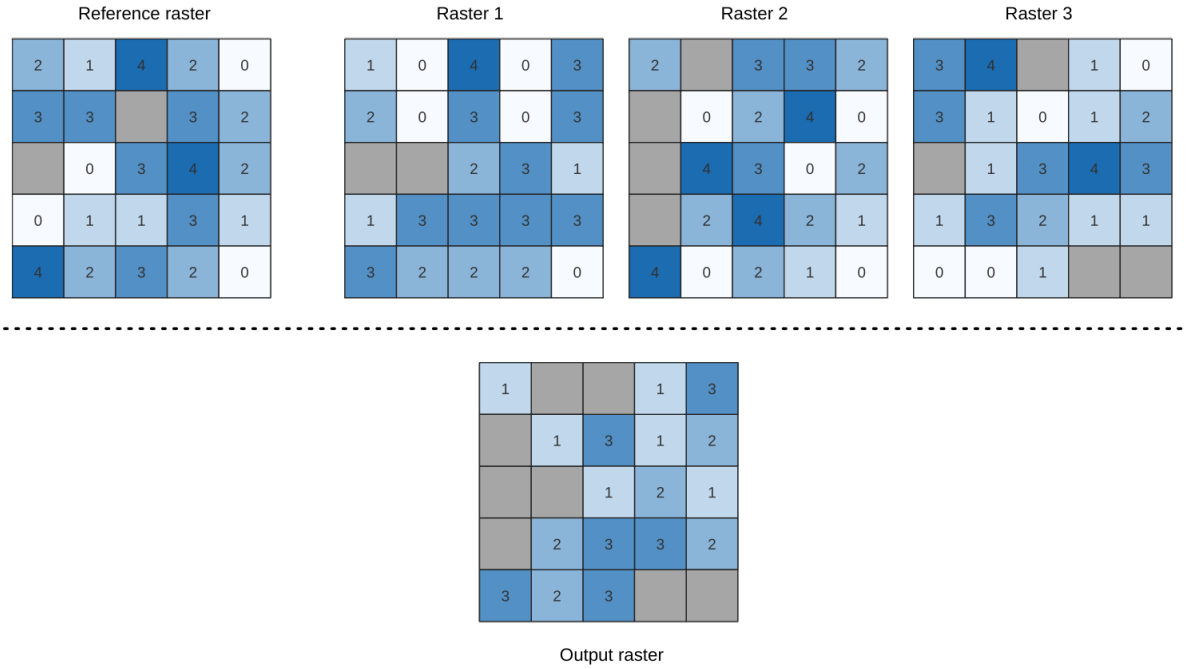
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Lowest position in raster stack

Evaluates on a cell-by-cell basis the position of the raster with the lowest value in a stack of rasters. Position counts start with 1 and range to the total number of input rasters. The order of the input rasters is relevant for the algorithm. If multiple rasters feature the lowest value, the first raster will be used for the position value.

If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters - use GDAL to use other bands in the analysis. Any NoData cells in the raster layer stack will result in a NoData cell in the output raster unless the "ignore NoData" parameter is checked. The output NoData value can be set manually. The output rasters extent and resolution is defined by a reference raster layer and is always of `Int32` type.



See also

Highest position in raster stack

参数

Basic parameters

标注	名称	类型	描述
Input raster layers	INPUT_RASTERS	[raster] [list]	List of raster layers to compare with
Reference layer	REFERENCE_LAYE	[栅格]	The reference layer for the output layer creation (extent, CRS, pixel dimensions)
Ignore NoData values	IGNORE_NODATA	[boolean] Default: False	If unchecked, any NoData cells in the data layer stack will result in a NoData cell in the output raster
Output layer	OUTPUT	[栅格] 默认: [保存到临时文件]	Specification of the output raster containing the result. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output NoData value	OUTPUT_NODATA_	[numeric: double] Default: -9999.0	Value to use for NoData in the output layer
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output layer	OUTPUT	[栅格]	Output raster layer containing the result
CRS authority identifier	CRS_AUTHID	[string]	The coordinate reference system of the output raster layer
Extent	EXTENT	[string]	The spatial extent of the output raster layer
Width in pixels	WIDTH_IN_PIXEL	[numeric: integer]	The number of columns in the output raster layer
Height in pixels	HEIGHT_IN_PIXE	[numeric: integer]	The number of rows in the output raster layer
Total pixel count	TOTAL_PIXEL_CO	[numeric: integer]	The count of pixels in the output raster layer

Python 代码

Algorithm ID: `native:lowestpositioninrasterstack`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Raster boolean AND

Calculates the boolean `AND` for a set of input rasters. If all of the input rasters have a non-zero value for a pixel, that pixel will be set to `1` in the output raster. If any of the input rasters have `0` values for the pixel it will be set to `0` in the output raster.

The reference layer parameter specifies an existing raster layer to use as a reference when creating the output raster. The output raster will have the same extent, CRS, and pixel dimensions as this layer.

By default, a NoData pixel in ANY of the input layers will result in a NoData pixel in the output raster. If the *Treat NoData values as false* option is checked, then NoData inputs will be treated the same as a `0` input value.

See also

Raster boolean OR

参数

Basic parameters

标注	名称	类型	描述
Input layers	`INPUT`	[raster] [list]	List of input raster layers
Reference layer	`REF_LAYER`	[栅格]	The reference layer to create the output layer from (extent, CRS, pixel dimensions)
Treat NoData values as false	`NODATA_AS_FALS`	[boolean] Default: False	Treat NoData values in the input files as 0 when performing the operation
Output layer	`OUTPUT`	[栅格] 默认：[保存到临时文件]	Specification of the output raster containing the result. *One of*: • 保存到临时文件 • 保存到文件…

continues on next page

Table 24.347 – continued from previous page

标注	名称	类型	描述

Advanced parameters

标注	名称	类型	描述
Output NoData value	NO_DATA	[numeric: double] Default: -9999.0	Value to use for NoData in the output layer
Output data type	DATA_TYPE	[enumeration] Default: 5	Output raster data type. Options: • 0 --- Byte (Eight bit unsigned integer (quint8)) • 1 --- Int16 (Sixteen bit signed integer (qint16)) • 2 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 3 --- Int32 (Thirty two bit signed integer (qint32)) • 4 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 5 --- Float32 (Thirty two bit floating point (float)) • 6 --- Float64 (Sixty four bit floating point (double)) • 7 --- CInt16 (Complex Int16) • 8 --- CInt32 (Complex Int32) • 9 --- CFloat32 (Complex Float32) • 10 --- CFloat64 (Complex Float64) • 11 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Extent	EXTENT	[string]	The spatial extent of the output raster layer
CRS authority identifier	CRS_AUTHID	[crs]	The coordinate reference system of the output raster layer
Width in pixels	WIDTH_IN_PIXEL	[numeric: integer]	The number of columns in the output raster layer
Height in pixels	HEIGHT_IN_PIXE	[numeric: integer]	The number of rows in the output raster layer
Total pixel count	TOTAL_PIXEL_CO	[numeric: integer]	The count of pixels in the output raster layer

continues on next page

Table 24.348 – continued from previous page

标注	名称	类型	描述
NoData pixel count	NODATA_PIXEL_C	[numeric: integer]	The count of NoData pixels in the output raster layer
True pixel count	TRUE_PIXEL_COU	[numeric: integer]	The count of True pixels (value = 1) in the output raster layer
False pixel count	FALSE_PIXEL_CO	[numeric: integer]	The count of False pixels (value = 0) in the output raster layer
Output layer	OUTPUT	[栅格]	Output raster layer containing the result

Python 代码

Algorithm ID: `native:rasterbooleanand`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Raster boolean OR

Calculates the boolean OR for a set of input rasters. If all of the input rasters have a zero value for a pixel, that pixel will be set to 0 in the output raster. If any of the input rasters have 1 values for the pixel it will be set to 1 in the output raster.

The reference layer parameter specifies an existing raster layer to use as a reference when creating the output raster. The output raster will have the same extent, CRS, and pixel dimensions as this layer.

By default, a NoData pixel in ANY of the input layers will result in a NoData pixel in the output raster. If the *Treat NoData values as false* option is checked, then NoData inputs will be treated the same as a 0 input value.

See also

Raster boolean AND

参数

Basic parameters

标注	名称	类型	描述
Input layers	INPUT	[raster] [list]	List of input raster layers
Reference layer	REF_LAYER	[栅格]	The reference layer to create the output layer from (extent, CRS, pixel dimensions)
Treat NoData values as false	NODATA_AS_FALS	[boolean] Default: False	Treat NoData values in the input files as 0 when performing the operation
Output layer	OUTPUT	[栅格] 默认：[保存到临时文件]	Specification of the output raster containing the result. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output NoData value	NO_DATA	[numeric: double] Default: -9999.0	Value to use for NoData in the output layer
Output data type	DATA_TYPE	[enumeration] Default: 5	Output raster data type. Options: • 0 --- Byte (Eight bit unsigned integer (quint8)) • 1 --- Int16 (Sixteen bit signed integer (qint16)) • 2 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 3 --- Int32 (Thirty two bit signed integer (qint32)) • 4 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 5 --- Float32 (Thirty two bit floating point (float)) • 6 --- Float64 (Sixty four bit floating point (double)) • 7 --- CInt16 (Complex Int16) • 8 --- CInt32 (Complex Int32) • 9 --- CFloat32 (Complex Float32) • 10 --- CFloat64 (Complex Float64) • 11 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Extent	EXTENT	[string]	The spatial extent of the output raster layer
CRS authority identifier	CRS_AUTHID	[crs]	The coordinate reference system of the output raster layer
Width in pixels	WIDTH_IN_PIXEL	[numeric: integer]	The number of columns in the output raster layer
Height in pixels	HEIGHT_IN_PIXE	[numeric: integer]	The number of rows in the output raster layer
Total pixel count	TOTAL_PIXEL_CO	[numeric: integer]	The count of pixels in the output raster layer
NoData pixel count	NODATA_PIXEL_C	[numeric: integer]	The count of NoData pixels in the output raster layer

continues on next page

Table 24.351 – continued from previous page

标注	名称	类型	描述
True pixel count	TRUE_PIXEL_COU	[numeric: integer]	The count of True pixels (value = 1) in the output raster layer
False pixel count	FALSE_PIXEL_CO	[numeric: integer]	The count of False pixels (value = 0) in the output raster layer
Output layer	OUTPUT	[栅格]	Output raster layer containing the result

Python 代码

Algorithm ID: `native:rasterbooleanor`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Raster calculator

Performs algebraic operations using raster layers.

The resulting layer will have its values computed according to an expression. The expression can contain numerical values, operators and references to any of the layers in the current project.

See also

Raster calculator (virtual), *Raster calculator*, *Raster Calculator*

参数

Basic parameters

标注	名称	类型	描述
Input layers	INPUT	[raster] [list]	List of input raster layers
Expression	EXPRESSION	[expression]	*Raster-based expression* that will be used to calculate the output raster layer.

continues on next page

Table 24.352 – continued from previous page

标注	名称	类型	描述
Output extent 可选	EXTENT	[范围]	Specify the spatial extent of the output raster layer. If the extent is not specified, the minimum extent that covers all the selected reference layers will be used. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Output cell size (leave empty to set automatically) 可选	CELLSIZE	[numeric: double]	Cell size of the output raster layer. If the cell size is not specified, the minimum cell size of the selected reference layer(s) will be used. The cell size will be the same for the X and Y axes.
Output CRS 可选	CRS	[crs]	CRS of the output raster layer. If the output CRS is not specified, the CRS of the first reference layer will be used.
Calculated	OUTPUT	[栅格] 默认: [保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Creation options 可选	CREATION_OPTIO	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Calculated	OUTPUT	[栅格]	Output raster file with the calculated values.

Python 代码

Algorithm ID: `native:rastercalc`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Raster calculator (virtual)

Performs algebraic operations using raster layers and generates in-memory result.

The resulting layer will have its values computed according to an expression. The expression can contain numerical values, operators and references to any of the layers in the current project.

A virtual raster layer is a raster layer defined by its URI and whose pixels are calculated on-the-fly. It's not a new file on disk; the virtual layer is still connected to the rasters used in the calculation meaning that deleting or moving these rasters would break it. A *Layer name* can be provided, otherwise the calculation expression is used as such. Removing the virtual layer from the project deletes it, and it can be made persistent in file using the layer *Export* ► *Save as...* contextual menu.

See also

Raster calculator, *Raster calculator*, *Raster Calculator*

continues on next page

Table 24.355 – continued from previous page

标注	名称	类型	描述

参数

标注	名称	类型	描述
Input layers	`INPUT`	[raster] [list]	List of input raster layers
Expression	`EXPRESSION`	[expression]	*Raster-based expression* that will be used to calculate the output raster layer.
Output extent 可选	`EXTENT`	[范围]	Specify the spatial extent of the output raster layer. If the extent is not specified, the minimum extent that covers all the selected reference layers will be used. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Output cell size (leave empty to set automatically) 可选	`CELLSIZE`	[numeric: double]	Cell size of the output raster layer. If the cell size is not specified, the minimum cell size of the selected reference layer(s) will be used. The cell size will be the same for the X and Y axes.
Output CRS 可选	`CRS`	[crs]	CRS of the output raster layer. If the output CRS is not specified, the CRS of the first reference layer will be used.
Output layer name 可选	`LAYER_NAME`	[string]	The name to assign to the generated layer. If not set, the text of the calculation expression is used.

输出

标注	名称	类型	描述
Calculated	`OUTPUT`	[栅格]	Output virtual raster layer with the calculated values.

Python 代码

Algorithm ID: `native:virtualrastercalc`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Raster layer properties

Returns basic properties of the given raster layer, including the extent, size in pixels and dimensions of pixels (in map units), number of bands, and NoData value.

This algorithm is intended for use as a means of extracting these useful properties to use as the input values to other algorithms in a model - e.g. to allow to pass an existing raster's pixel sizes over to a GDAL raster algorithm.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	Input raster layer
Band number 可选	`BAND`	[raster band] Default: Not set	Whether to also return properties of a specific band. If a band is specified, the noData value for the selected band is also returned.

输出

标注	名称	类型	描述
Number of bands in raster	`BAND_COUNT`	[numeric: integer]	The number of bands in the raster
CRS authority identifier	`CRS_AUTHID`	[string]	The coordinate reference system of the output raster layer
Extent	`EXTENT`	[string]	The raster layer extent in the CRS
Band has a NoData value set	`HAS_NODATA_VAL`	[Boolean]	Indicates whether the raster layer has a value set for NoData pixels in the selected band
Height in pixels	`HEIGHT_IN_PIXE`	[numeric: integer]	The number of columns in the raster layer
Band NoData value	`NODATA_VALUE`	[numeric: double]	The value (if set) of the NoData pixels in the selected band
Pixel size (height) in map units	`PIXEL_HEIGHT`	[numeric: integer]	Vertical size in map units of the pixel
Pixel size (width) in map units	`PIXEL_WIDTH`	[numeric: integer]	Horizontal size in map units of the pixel
Width in pixels	`WIDTH_IN_PIXEL`	[numeric: integer]	The number of rows in the raster layer
Maximum x-coordinate	`X_MAX`	[numeric: double]	
Minimum x-coordinate	`X_MIN`	[numeric: double]	
Maximum y-coordinate	`Y_MAX`	[numeric: double]	

continues on next page

Table 24.358 – continued from previous page

标注	名称	类型	描述
Minimum y-coordinate	Y_MIN	[numeric: double]	

Python 代码

Algorithm ID: native:rasterlayerproperties

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Raster layer statistics

Calculates basic statistics from the values in a given band of the raster layer. The output is loaded in the *Processing* ► *Results viewer* menu.

参数

标注	名称	类型	描述
输入图层	INPUT	[栅格]	Input raster layer
Band number	BAND	[raster band] Default: The first band of the input layer	If the raster is multiband, choose the band you want to get statistics for.
Statistics	OUTPUT_HTML_FI	[html] 默认: [保存到临时文件]	Specification of the output file: • Skip Output • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Maximum value	MAX	[numeric: double]	
Mean value	MEAN	[numeric: double]	
Minimum value	MIN	[numeric: double]	

continues on next page

Table 24.360 – continued from previous page

标注	名称	类型	描述
Statistics	OUTPUT_HTML_FI	[html]	The output file contains the following information: • Analyzed file: path of the raster layer • Minimum value: minimum value of the raster • Maximum value: maximum value of the raster • Range: difference between the maximum and minimum values • Sum: total sum of the values • Mean value: mean of the values • Standard deviation: standard deviation of the values • Sum of the squares: sum of the squared differences of each observation from the overall mean
Range	RANGE	[numeric: double]	
Standard deviation	STD_DEV	[numeric: double]	
Sum	SUM	[numeric: double]	
Sum of the squares	SUM_OF_SQUARES	[numeric: double]	

Python 代码

Algorithm ID: `native:rasterlayerstatistics`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Raster layer unique values report

Returns the count and area of each unique value in a given raster layer. The calculation of the area is done in the area unit of the layer's CRS.

参数

标注	名称	类型	描述
输入图层	INPUT	[栅格]	Input raster layer
Band number	BAND	[raster band] Default: The first band of the input layer	If the raster is multiband, choose the band you want to get statistics for.

continues on next page

Table 24.361 – continued from previous page

标注	名称	类型	描述
Unique values report	OUTPUT_HTML_FI	[file] 默认: [保存到临时文件]	Specification of the output file: • Skip Output • 保存到临时文件 • 保存到文件…
Unique values table	OUTPUT_TABLE	[vector: table] Default: [Skip output]	Specification of the table for unique values: • Skip Output • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
CRS authority identifier	CRS_AUTHID	[string]	The coordinate reference system of the output raster layer
Extent	EXTENT	[string]	The spatial extent of the output raster layer
Height in pixels	HEIGHT_IN_PIXE	[numeric: integer]	The number of rows in the output raster layer
NoData pixel count	NODATA_PIXEL_C	[numeric: integer]	The number of NoData pixels in the output raster layer
Total pixel count	TOTAL_PIXEL_CO	[numeric: integer]	The count of pixels in the output raster layer
Unique values report	OUTPUT_HTML_FI	[html]	The output HTML file contains the following information: • Analyzed file: the path of the raster layer • Extent: xmin, ymin, xmax, ymax coordinates of the extent • Projection: projection of the layer • Width in pixels: number of columns and pixel width size • Height in pixels: number of rows and pixel width size • Total pixel count: count of all the pixels • NoData pixel count: count of pixels with NoData value
Unique values table	OUTPUT_TABLE	[vector: table]	A table with three columns: • *value*: pixel value • *count*: count of pixels with this value • *m2* or *deg2* or *ft2* or ... : total area of pixels with this value. The column name depends on the area unit of the layer's CRS and the calculation is done in that unit.

continues on next page

Table 24.362 – continued from previous page

标注	名称	类型	描述
Width in pixels	WIDTH_IN_PIXEL	[numeric: integer]	The number of columns in the output raster layer

Python 代码

Algorithm ID: native:rasterlayeruniquevaluesreport

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Raster layer zonal statistics

Calculates statistics for a raster layer's values, categorized by zones defined in another raster layer.

See also

Zonal statistics

参数

Basic parameters

标注	名称	类型	描述
Input Layer	INPUT	[栅格]	Input raster layer
Band number	BAND	[raster band] Default: The first band of the raster layer	If the raster is multiband choose the band for which you want to calculate the statistics.
Zones layer	ZONES	[栅格]	Raster layer defining zones. Zones are given by contiguous pixels having the same pixel value.
Zones band number	ZONES_BAND	[raster band] Default: The first band of the raster layer	If the raster is multiband, choose the band that defines the zones
Statistics	OUTPUT_TABLE	[vector: table] Default: [Create temporary layer]	Specification of the output report. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Reference layer 可选	REF_LAYER	[enumeration] Default: 0	Raster layer used to calculate the centroids that will be used as reference when determining the zones in the output layer. One of: • 0 --- Input layer: zones are determined by sampling the zone raster layer value at the centroid of each pixel from the source raster layer • 1 --- Zones layer: the input raster layer will be sampled at the centroid of each pixel from the zones raster layer

输出

标注	名称	类型	描述
CRS authority identifier	CRS_AUTHID	[string]	The coordinate reference system of the output raster layer
Extent	EXTENT	[string]	The spatial extent of the output raster layer
Height in pixels	HEIGHT_IN_PIXE	[numeric: integer]	The number of rows in the output raster layer
NoData pixel count	NODATA_PIXEL_C	[numeric: integer]	The number of NoData pixels in the output raster layer
Statistics	OUTPUT_TABLE	[vector: table]	The output layer contains the following information **for each zone**: • Area: the area in square raster units in the zone; • Sum: the total sum of the pixel values in the zone; • Count: the number of pixels in the zone; • Min: the minimum pixel value in the zone; • Max: the maximum pixel value in the zone; • Mean: the mean of the pixel values in the zone;
Total pixel count	TOTAL_PIXEL_CO	[numeric: integer]	The count of pixels in the output raster layer
Width in pixels	WIDTH_IN_PIXEL	[numeric: integer]	The number of columns in the output raster layer

Python 代码

Algorithm ID: `native:rasterlayerzonalstats`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Raster minimum/maximum

Extracts the minimum and maximum pixel values from a specified band of a raster layer. If multiple pixels share the same minimum or maximum value, only one of them will be included in the output.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	Raster layer from which to extract the minimum and maximum pixel values.
Band number	`BAND`	[raster band] Default: 1	The band of the raster to analyze. If the raster is multiband, specify the band number (starting from 1).
Extract	`EXTREMA`	[enumeration] Default: 0 (Minimum and Maximum)	Choose which extrema to extract: • 0: Minimum and Maximum • 1: Minimum only • 2: Maximum only
Output layer	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Output layer	`OUTPUT`	[vector: point]	Vector layer with point features at the locations of the minimum and/or maximum pixel values.

Python 代码

Algorithm ID: `native:rasterminmax`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Raster rank

`NEW in 3.44`

Performs a cell-by-cell analysis in which output values match the rank of a sorted list of overlapping cell values from input layers. The output raster will be multi-band if multiple ranks are provided. If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters.

See also

Cell stack percent rank from value, *Cell stack percentile*, *Cell stack percentrank from raster layer*, *Cell statistics*, *Raster calculator*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT_RASTERS`	[raster] [list]	Input raster layer(s) to calculate the rank from.
Rank	`RANKS`	[numeric: integer] [list] Default: [1]	List of ranks to calculate for each pixel in the sorted list of cells values from the input rasters. Cells whose values match the given rank(s) are kept in the output layer. The ranks are 1-based, meaning that the first rank is 1. Negative values are also supported.
NoData value handling	`NODATA_HANDLIN`	[enumeration] Default: 0	Choose how to handle NoData values in the input raster: • 0 --- Exclude NoData from values lists: will not include NoData values in the list of pixel values when calculating ranks. • 1 --- Presence of NoData in a values list results in NoData output cells: the output raster will have NoData values in the corresponding locations.
Ranked	`OUTPUT`	[栅格] 默认: [保存到临时文件]	Specification of the output raster containing the rank values. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output extent 可选	`EXTENT`	[范围]	Specify the spatial extent of the output raster layer. If the extent is not specified, the minimum extent that covers all the selected reference layers will be used. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Output cell size 可选	`CELL_SIZE`	[numeric: double]	Cell size of the output raster layer. If the cell size is not specified, the minimum cell size of the selected reference layer(s) will be used. The cell size will be the same for the X and Y axes.
Output CRS 可选	`CRS`	[crs]	CRS of the output raster layer. If the output CRS is not specified, the CRS of the first reference layer will be used.

输出

标注	名称	类型	描述
Ranked	`OUTPUT`	[栅格]	Output raster layer containing the result.

Python 代码

Algorithm ID: `native:rasterrank`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Raster surface volume

Calculates the volume under a raster surface relative to a given base level. This is mainly useful for Digital Elevation Models (DEM).

参数

标注	名称	类型	描述
INPUT layer	`INPUT`	[栅格]	Input raster, representing a surface
Band number	`BAND`	[raster band] Default: The first band of the raster layer	If the raster is multiband, choose the band that shall define the surface.
Base level	`LEVEL`	[numeric: double] Default: 0.0	Define a base or reference value. This base is used in the volume calculation according to the `Method` parameter (see below).
Method	`METHOD`	[enumeration] Default: 0	Define the method for the volume calculation given by the difference between the raster pixel value and the `Base level`. Options: • 0 --- Count Only Above Base Level: only pixels above the base level will add to the volume. • 1 --- Count Only Below Base Level: only pixels below the base level will add to the volume. • 2 --- Subtract Volumes Below Base level: pixels above the base level will add to the volume, pixels below the base level will subtract from the volume. • 3 --- Add Volumes Below Base level: Add the volume regardless whether the pixel is above or below the base level. This is equivalent to sum the absolute values of the difference between the pixel value and the base level.
Surface volume report	`OUTPUT_HTML_FI`	[html] 默认: [保存到临时文件]	Specification of the output HTML report. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

continues on next page

Table 24.371 – continued from previous page

标注	名称	类型	描述
Surface volume table	OUTPUT_TABLE	[vector: table] Default: [Skip output]	Specification of the output table. *One of*: • Skip Output • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Volume	VOLUME	[numeric: double]	The calculated volume
Area	AREA	[numeric: double]	The area in square map units
Pixel_count	PIXEL_COUNT	[numeric: integer]	The total number of pixels that have been analyzed
Surface volume report	OUTPUT_HTML_FI	[html]	The output report (containing volume, area and pixel count) in HTML format
Surface volume table	OUTPUT_TABLE	[vector: table]	The output table (containing volume, area and pixel count)

Python 代码

Algorithm ID: native:rastersurfacevolume

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Reclassify by layer

Reclassifies a raster band by assigning new class values based on the ranges specified in a vector table.

参数

Basic parameters

标注	名称	类型	描述
Raster layer	INPUT_RASTER	[栅格]	Raster layer to reclassify
Band number	RASTER_BAND	[raster band] Default: The first band of the raster layer	If the raster is multiband, choose the band you want to reclassify.
Layer containing class breaks	INPUT_TABLE	[vector: any]	Vector layer containing the values to use for classification.

continues on next page

Table 24.373 – continued from previous page

标注	名称	类型	描述
Minimum class value field	MIN_FIELD	[tablefield: numeric]	Field with the minimum value of the range for the class. Use `-inf` to capture the lowest possible value.
Maximum class value field	MAX_FIELD	[tablefield: numeric]	Field with the maximum value of the range for the class. Use `inf` to capture the highest possible value.
Output value field	VALUE_FIELD	[tablefield: numeric]	Field with the value that will be assigned to the pixels that fall in the class (between the corresponding min and max values). Use `nan` to set the value of the range to NoData.
Reclassified raster	OUTPUT	[栅格] 默认：[保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output NoData value	NO_DATA	[numeric: double] Default: -9999.0	Value to apply to NoData values.
Range boundaries	RANGE_BOUNDARI	[enumeration] Default: 0	Defines comparison rules for the classification. Options: • 0 --- min < value <= max • 1 --- min <= value < max • 2 --- min <= value <= max • 3 --- min < value < max
Use NoData when no range matches value	NODATA_FOR_MIS	[boolean] Default: False	Applies the NoData value to band values that do not fall in any class. If False, the original value is kept.

continues on next page

Table 24.374 – continued from previous page

标注	名称	类型	描述
Output data type	DATA_TYPE	[enumeration] Default: 5	Defines the format of the output raster file. Options: • 0 --- Byte (Eight bit unsigned integer (quint8)) • 1 --- Int16 (Sixteen bit signed integer (qint16)) • 2 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 3 --- Int32 (Thirty two bit signed integer (qint32)) • 4 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 5 --- Float32 (Thirty two bit floating point (float)) • 6 --- Float64 (Sixty four bit floating point (double)) • 7 --- CInt16 (Complex Int16) • 8 --- CInt32 (Complex Int32) • 9 --- CFloat32 (Complex Float32) • 10 --- CFloat64 (Complex Float64) • 11 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Reclassified raster	OUTPUT	[栅格]	Output raster layer with reclassified band values

Python 代码

Algorithm ID: `native:reclassifybylayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

Reclassify by table

Reclassifies a raster band by assigning new class values based on the ranges specified in a fixed table.

参数

Basic parameters

标注	名称	类型	描述
Raster layer	`INPUT_RASTER`	[栅格]	Raster layer to reclassify
Band number	`RASTER_BAND`	[raster band] Default: 1	Raster band for which you want to recalculate values.
Reclassification table	`TABLE`	[vector: table]	A 3-columns table to fill with the values to set the boundaries of each class (`Minimum` and `Maximum`) and the new `Value` to assign to the band values that fall in the class. The value `-inf` can be used as minimum, `inf` as maximum, and `nan` can be used to set the output value to NoData.
Reclassified raster	`OUTPUT`	[栅格] 默认：[保存到临时文件]	Specification of the output raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output NoData value	`NO_DATA`	[numeric: double] Default: -9999.0	Value to apply to NoData values.
Range boundaries	`RANGE_BOUNDARI`	[enumeration] Default: 0	Defines comparison rules for the classification. Options: • 0 --- min < value <= max • 1 --- min <= value < max • 2 --- min <= value <= max • 3 --- min < value < max
Use NoData when no range matches value	`NODATA_FOR_MIS`	[boolean] Default: False	Applies the NoData value to band values that do not fall in any class. If False, the original value is kept.

continues on next page

Table 24.377 – continued from previous page

标注	名称	类型	描述
Output data type	DATA_TYPE	[enumeration] Default: 5	Defines the format of the output raster file. Options: • 0 --- Byte (Eight bit unsigned integer (quint8)) • 1 --- Int16 (Sixteen bit signed integer (qint16)) • 2 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 3 --- Int32 (Thirty two bit signed integer (qint32)) • 4 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 5 --- Float32 (Thirty two bit floating point (float)) • 6 --- Float64 (Sixty four bit floating point (double)) • 7 --- CInt16 (Complex Int16) • 8 --- CInt32 (Complex Int32) • 9 --- CFloat32 (Complex Float32) • 10 --- CFloat64 (Complex Float64) • 11 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Reclassified raster	OUTPUT	[栅格]	Output raster layer with reclassified band values

Python 代码

Algorithm ID: `native:reclassifybytable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

Rescale raster

Rescales raster layer to a new value range, while preserving the shape (distribution) of the raster's histogram (pixel values). Input values are mapped using a linear interpolation from the source raster's minimum and maximum pixel values to the destination minimum and miximum pixel range.

By default the algorithm preserves the original NoData value, but there is an option to override it.

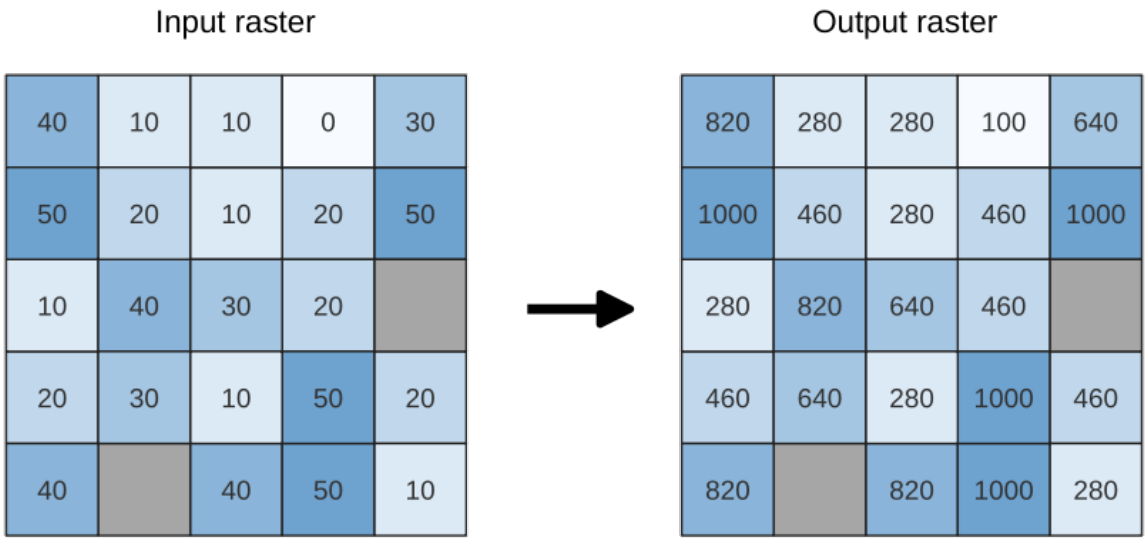


Fig. 24.57: Rescaling values of a raster layer from [0 - 50] to [100 - 1000]

参数

Basic parameters

标注	名称	类型	描述
Input Raster	`INPUT`	[栅格]	Raster layer to use for rescaling
Band number	`BAND`	[raster band] Default: The first band of the input layer	If the raster is multiband, choose a band.
New minimum value	`MINIMUM`	[numeric: double] Default value: 0.0	Minimum pixel value to use in the rescaled layer
New maximum value	`MAXIMUM`	[numeric: double] Default value: 255.0	Maximum pixel value to use in the rescaled layer
New NoData value 可选	`NODATA`	[numeric: double] Default value: Not set	Value to assign to the NoData pixels. If unset, original NoData values are preserved.
Rescaled	`OUTPUT`	[栅格] 默认: [保存到临时文件]	Specification of the output raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Rescaled	OUTPUT	[栅格]	Output raster layer with rescaled band values

Python 代码

Algorithm ID: `native:rescaleraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Round raster

Rounds the cell values of a raster dataset according to the specified number of decimals.

Alternatively, a negative number of decimal places may be used to round values to powers of a base n. For example, with a Base value n of 10 and Decimal places of -1, the algorithm rounds cell values to multiples of 10, -2 rounds to multiples of 100, and so on. Arbitrary base values may be chosen, the algorithm applies the same multiplicative principle. Rounding cell values to multiples of a base n may be used to generalize raster layers.

The algorithm preserves the data type of the input raster. Therefore byte/integer rasters can only be rounded to multiples of a base n, otherwise a warning is raised and the raster gets copied as byte/integer raster.

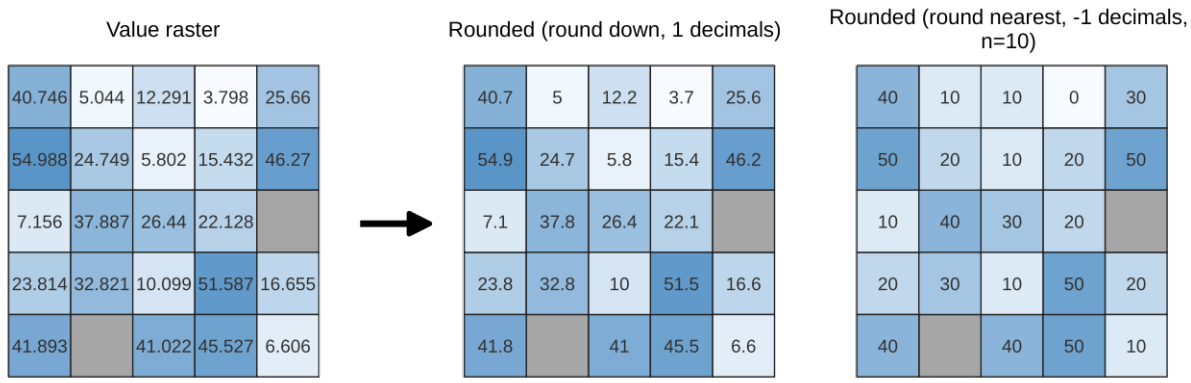


Fig. 24.58: Rounding values of a raster

参数

Basic parameters

标注	名称	类型	描述
输入栅格	`INPUT`	[栅格]	The raster to process.
Band number	`BAND`	[raster band] Default: 1	The band of the raster
Rounding direction	`ROUNDING_DIREC`	[list] Default: 1	How to choose the target rounded value. Options are: • 0 --- Round up • 1 --- Round to nearest • 2 --- Round down
Number of decimals places	`DECIMAL_PLACES`	[numeric: integer] Default: 2	Number of decimals places to round to. Use negative values to round cell values to a multiple of a base n
Output raster	`OUTPUT`	[栅格] 默认：[保存到临时文件]	Specification of the output file. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Base n for rounding to multiples of n	`BASE_N`	[numeric: integer] Default: 10	When the `DECIMAL_PLACES` parameter is negative, raster values are rounded to multiples of the base n value

continues on next page

Table 24.382 – continued from previous page

标注	名称	类型	描述
Creation options 可选	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output raster	`OUTPUT`	[栅格]	The output raster layer with values rounded for the selected band.

Python 代码

Algorithm ID: `native:roundrastervalues`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

Sample raster values

Extracts raster values at the point locations. If the raster layer is multiband, each band is sampled.

The attribute table of the resulting layer will have as many new columns as the raster layer band count.

参数

标注	名称	类型	描述
Input Layer	`INPUT`	[vector: point]	Point vector layer to use for sampling
Raster Layer	`RASTERCOPY`	[栅格]	Raster layer to sample at the given point locations.
Output column prefix	`COLUMN_PREFIX`	[string] Default: 'SAMPLE_'	Prefix for the names of the added columns.
Sampled 可选	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the output layer containing the sampled values. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Sampled	`OUTPUT`	[vector: point]	The output layer containing the sampled values.

Python 代码

Algorithm ID: `native:rastersampling`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Zonal histogram

Appends fields representing counts of each unique value from a raster layer contained within polygon features.

The output layer attribute table will have as many fields as the unique values of the raster layer that intersects the polygon(s).

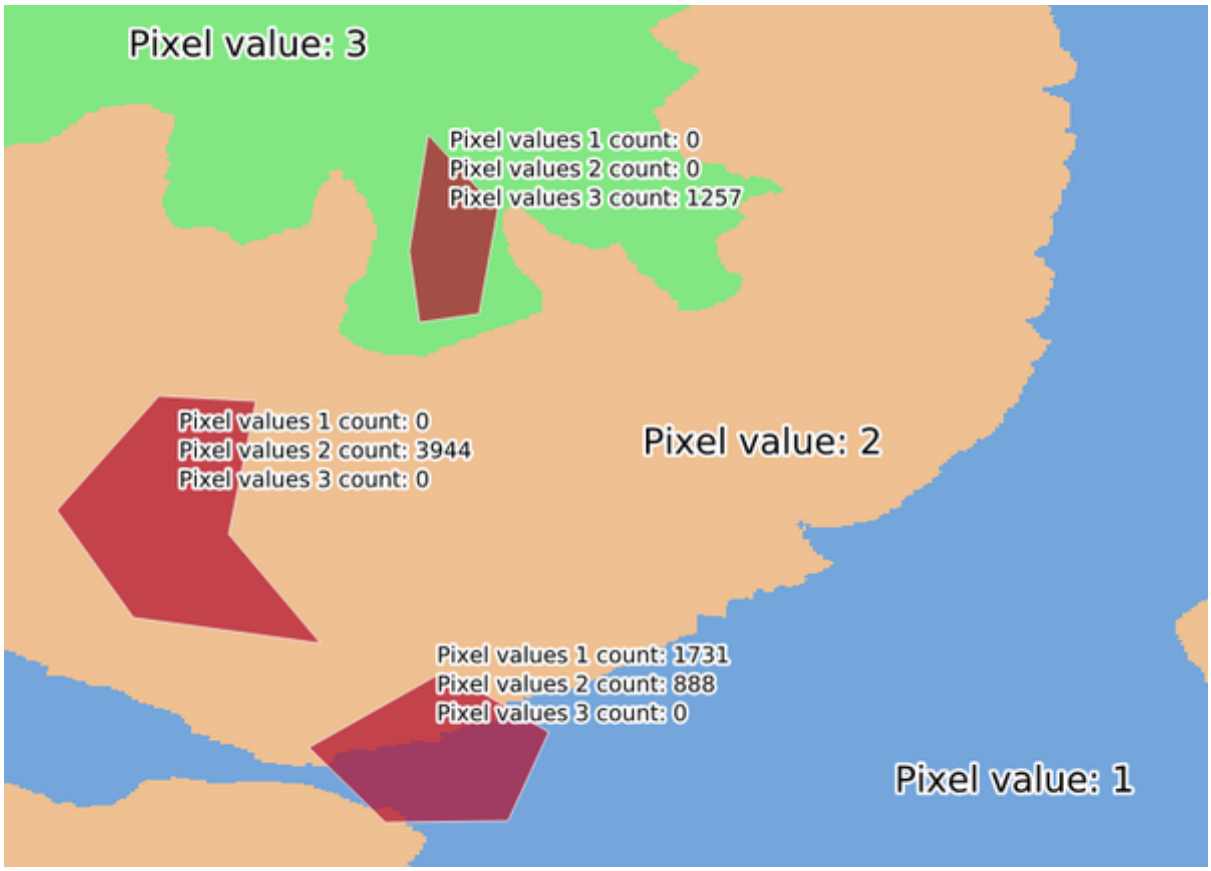


Fig. 24.59: Raster layer histogram example

参数

标注	名称	类型	描述
Raster layer	`INPUT_RASTER`	[栅格]	Input raster layer.
Band number	`RASTER_BAND`	[raster band] Default: The first band of the input layer	If the raster is multiband, choose a band.
Vector layer containing zones	`INPUT_VECTOR`	[vector: polygon]	Vector polygon layer that defines the zones.
Output column prefix	`COLUMN_PREFIX` 可选	[string] Default: 'HISTO_'	Prefix for the output columns names.

continues on next page

Table 24.385 – continued from previous page

标注	名称	类型	描述
Output zones	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output vector polygon layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Output zones	`OUTPUT`	[vector: polygon]	The output vector polygon layer.

Python 代码

Algorithm ID: `native:zonalhistogram`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Zonal Minimum/Maximum Point

Extracts point features corresponding to the minimum and maximum pixel values within polygon zones.

The output will contain one point feature for the minimum and one for the maximum raster value for every individual zonal feature from a polygon layer. The created point layer will be in the same spatial reference system as the selected raster layer.

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

参数

标注	名称	类型	描述
Input Layer	`INPUT`	[vector: polygon]	Polygon layer defining the zones.
Raster Layer	`RASTER`	[栅格]	Raster layer to extract the minimum and maximum values from.
Raster Band	`BAND`	[raster band] Default: The first band of the raster layer	If the raster has multiple bands, select the band to process.
Zonal extrema	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Zonal extrema	`OUTPUT`	[vector: point]	Output layer containing the minimum and maximum points for each zone.

Python 代码

Algorithm ID: `native:zonalminmaxpoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，*算法 id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Zonal statistics

Calculates statistics of a raster layer for each feature of an overlapping polygon vector layer.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: polygon]	Vector polygon layer that contains the zones.
Raster layer	`INPUT_RASTER`	[栅格]	Input raster layer.

continues on next page

Table 24.387 – continued from previous page

标注	名称	类型	描述
Raster band	RASTER_BAND	[raster band] Default: The first band of the input layer	If the raster is multiband, choose a band for the statistics.
Output column prefix	COLUMN_PREFIX	[string] Default: '_'	Prefix for the output columns names.
Statistics to calculate	STATISTICS	[enumeration] [list] Default: [0,1,2]	List of statistical operator for the output. Options: • 0 --- Count • 1 --- Sum • 2 --- Mean • 3 --- Median • 4 --- St. dev. • 5 --- Minimum • 6 --- Maximum • 7 --- Range • 8 --- Minority • 9 --- Majority • 10 --- Variety • 11 --- Variance
Zonal Statistics	OUTPUT	[vector: polygon] Default: [Create temporary layer]	Specify the output vector polygon layer. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Zonal Statistics	OUTPUT	[vector: polygon]	The zone vector layer with added statistics.

Python 代码

Algorithm ID: native:zonalstatisticsfb

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.19 Raster Creation

Create constant raster layer

Generates raster layer for given extent and cell size filled with the specified value.

Additionally an output data type can be specified. The algorithm will abort if a value has been entered that cannot be represented by the selected output raster data type.

参数

Basic parameters

标注	名称	类型	描述
Desired extent	`EXTENT`	[范围]	Specify the spatial extent of the output raster layer. It will internally be extended to a multiple of the tile size. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Target CRS	`TARGET_CRS`	[crs] Default: Project CRS	CRS for the output raster layer
Pixel size	`PIXEL_SIZE`	[numeric: double] 默认值：1.0	Pixel size (X=Y) in map units.
Constant value	`NUMBER`	[numeric: double] 默认值：1.0	Constant pixel value for the output raster layer.
Constant	`OUTPUT`	[栅格] 默认：[保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output raster data type	`OUTPUT_TYPE` Default: 5	[enumeration]	Defines the data type of the output raster file. Options: • 0 --- Byte • 1 --- Integer16 • 2 --- Unsigned Integer16 • 3 --- Integer32 • 4 --- Unsigned Integer32 • 5 --- Float32 • 6 --- Float64
Creation options 可选	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Constant	`OUTPUT`	[栅格]	Raster covering the desired extent with the specified pixel size and value.

Python 代码

Algorithm ID: `native:createconstantrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Create random raster layer (binomial distribution)

Generates a raster layer for given extent and cell size filled with binomially distributed random values.

By default, the values will be chosen given an N of 10 and a probability of 0.5. This can be overridden by using the advanced parameter for N and probability. The raster data type is set to Integer types (Integer16 by default). The binomial distribution random values are defined as positive integer numbers. A floating point raster will represent a cast of integer values to floating point.

参数

Basic parameters

标注	名称	类型	描述
Desired extent	`EXTENT`	[范围]	Specify the spatial extent of the output raster layer. It will internally be extended to a multiple of the tile size. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Target CRS	`TARGET_CRS`	[crs] Default: Project CRS	CRS for the output raster layer
Pixel size	`PIXEL_SIZE`	[numeric: double] 默认值：1.0	Pixel size (X=Y) in map units.
Output raster	`OUTPUT`	[栅格] 默认：[保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output raster data type	`OUTPUT_TYPE` Default: 0	[enumeration]	Defines the data type of the output raster file. Options: • 0 --- Integer16 • 1 --- Unsigned Integer16 • 2 --- Integer32 • 3 --- Unsigned Integer32 • 4 --- Float32 • 5 --- Float64
N	`N`	[numeric: integer] Default: 10	
Probability	`PROBABILITY`	[numeric: double] Default: 0.5	

continues on next page

Table 24.393 – continued from previous page

标注	名称	类型	描述
Creation options 可选	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output raster	`OUTPUT`	[栅格]	Raster covering the desired extent with the cell size filled with random values

Python 代码

Algorithm ID: `native:createrandombinomialrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

Create random raster layer (exponential distribution)

Generates a raster layer for given extent and cell size filled with exponentially distributed random values.

By default, the values will be chosen given a lambda of 1.0. This can be overridden by using the advanced parameter for lambda. The raster data type is set to Float32 by default as the exponential distribution random values are floating point numbers.

参数

Basic parameters

标注	名称	类型	描述
Desired extent	EXTENT	[范围]	Specify the spatial extent of the output raster layer. It will internally be extended to a multiple of the tile size. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as xmin, xmax, ymin, ymax
Target CRS	TARGET_CRS	[crs] Default: Project CRS	CRS for the output raster layer
Pixel size	PIXEL_SIZE	[numeric: double] 默认值：1.0	Pixel size (X=Y) in map units.
Output raster	OUTPUT	[栅格] 默认：[保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output raster data type	OUTPUT_TYPE Default: 0	[enumeration]	Defines the data type of the output raster file. Options: • 0 --- Float32 • 1 --- Float64
Lambda	LAMBDA	[numeric: double] 默认值：1.0	
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output raster	`OUTPUT`	[栅格]	Raster covering the desired extent with the cell size filled with random values

Python 代码

Algorithm ID: `native:createrandomexponentialrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Create random raster layer (gamma distribution)

Generates a raster layer for given extent and cell size filled with gamma distributed random values.

By default, the values will be chosen given an alpha and beta value of 1.0. This can be overridden by using the advanced parameter for alpha and beta. The raster data type is set to Float32 by default as the gamma distribution random values are floating point numbers.

参数

Basic parameters

标注	名称	类型	描述
Desired extent	`EXTENT`	[范围]	Specify the spatial extent of the output raster layer. It will internally be extended to a multiple of the tile size. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Target CRS	`TARGET_CRS`	[crs] Default: Project CRS	CRS for the output raster layer

continues on next page

Table 24.398 – continued from previous page

标注	名称	类型	描述
Pixel size	PIXEL_SIZE	[numeric: double] 默认值：1.0	Pixel size (X=Y) in map units.
Output raster	OUTPUT	[栅格] 默认：[保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output raster data type	OUTPUT_TYPE Default: 0	[enumeration]	Defines the data type of the output raster file. Options: • 0 --- Float32 • 1 --- Float64
Alpha	ALPHA	[numeric: double] 默认值：1.0	
Beta	BETA	[numeric: double] 默认值：1.0	
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output raster	OUTPUT	[栅格]	Raster covering the desired extent with the cell size filled with randomly distributed values

Python 代码

Algorithm ID: `native:createrandomgammarasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Create random raster layer (geometric distribution)

Generates a raster layer for given extent and cell size filled with geometrically distributed random values.

By default, the values will be chosen given a probability of 0.5. This can be overridden by using the advanced parameter for mean value. The raster data type is set to Integer types (Integer16 by default). The geometric distribution random values are defined as positive integer numbers. A floating point raster will represent a cast of integer values to floating point.

参数

Basic parameters

标注	名称	类型	描述
Desired extent	`EXTENT`	[范围]	Specify the spatial extent of the output raster layer. It will internally be extended to a multiple of the tile size. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Target CRS	`TARGET_CRS`	[crs] Default: Project CRS	CRS for the output raster layer
Pixel size	`PIXEL_SIZE`	[numeric: double] 默认值：1.0	Pixel size (X=Y) in map units.
Output raster	`OUTPUT`	[栅格] 默认：[保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output raster data type	`OUTPUT_TYPE` Default: 0	[enumeration]	Defines the data type of the output raster file. Options: • 0 --- Integer16 • 1 --- Unsigned Integer16 • 2 --- Integer32 • 3 --- Unsigned Integer32 • 4 --- Float32 • 5 --- Float64
Probability	`PROBABILITY`	[numeric: double] Default: 0.5	
Creation options 可选	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output raster	`OUTPUT`	[栅格]	Raster covering the desired extent with the cell size filled with randomly distributed values

Python 代码

Algorithm ID: `native:createrandomgeometricrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

Create random raster layer (negative binomial distribution)

Generates a raster layer for given extent and cell size filled with negative binomially distributed random values.

By default, the values will be chosen given a distribution parameter k of 10.0 and a probability of 0.5. This can be overridden by using the advanced parameters for k and probability. The raster data type is set to Integer types (Integer16 by default). The negative binomial distribution random values are defined as positive integer numbers. A floating point raster will represent a cast of integer values to floating point.

参数

Basic parameters

标注	名称	类型	描述
Desired extent	`EXTENT`	[范围]	Specify the spatial extent of the output raster layer. It will internally be extended to a multiple of the tile size. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Target CRS	`TARGET_CRS`	[crs] Default: Project CRS	CRS for the output raster layer
Pixel size	`PIXEL_SIZE`	[numeric: double] 默认值：1.0	Pixel size (X=Y) in map units.
Output raster	`OUTPUT`	[栅格] 默认：[保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output raster data type	`OUTPUT_TYPE` Default: 0	[enumeration]	Defines the data type of the output raster file. Options: • 0 --- Integer16 • 1 --- Unsigned Integer16 • 2 --- Integer32 • 3 --- Unsigned Integer32 • 4 --- Float32 • 5 --- Float64
Distribution parameter k	`K_PARAMETER`	[numeric: integer] Default: 10	
Probability	`PROBABILITY`	[numeric: double] Default: 0.5	

continues on next page

Table 24.405 – continued from previous page

标注	名称	类型	描述
Creation options 可选	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output raster	`OUTPUT`	[栅格]	Raster covering the desired extent with the cell size filled with randomly distributed values

Python 代码

Algorithm ID: `native:createrandomnegativebinomialrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

Create random raster layer (normal distribution)

Generates a raster layer for given extent and cell size filled with normally distributed random values.

By default, the values will be chosen given a mean of 0.0 and a standard deviation of 1.0. This can be overridden by using the advanced parameters for mean and standard deviation value. The raster data type is set to Float32 by default as the normal distribution random values are floating point numbers.

参数

Basic parameters

标注	名称	类型	描述
Desired extent	EXTENT	[范围]	Specify the spatial extent of the output raster layer. It will internally be extended to a multiple of the tile size. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Target CRS	TARGET_CRS	[crs] Default: Project CRS	CRS for the output raster layer
Pixel size	PIXEL_SIZE	[numeric: double] 默认值：1.0	Pixel size (X=Y) in map units.
Output raster	OUTPUT	[栅格] 默认：[保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output raster data type	OUTPUT_TYPE Default: 0	[enumeration]	Defines the data type of the output raster file. Options: • 0 --- Float32 • 1 --- Float64
Mean of normal distribution	MEAN	[numeric: double] Default: 0.0	
Standard deviation of normal distribution	STDDEV	[numeric: double] 默认值：1.0	
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output raster	`OUTPUT`	[栅格]	Raster covering the desired extent with the cell size filled with randomly distributed values

Python 代码

Algorithm ID: `native:createrandomnormalrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Create random raster layer (poisson distribution)

Generates a raster layer for given extent and cell size filled with poisson distributed random values.

By default, the values will be chosen given a mean of 1.0. This can be overridden by using the advanced parameter for mean value. The raster data type is set to Integer types (Integer16 by default). The poisson distribution random values are positive integer numbers. A floating point raster will represent a cast of integer values to floating point.

参数

Basic parameters

标注	名称	类型	描述
Desired extent	`EXTENT`	[范围]	Specify the spatial extent of the output raster layer. It will internally be extended to a multiple of the tile size. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`

continues on next page

Table 24.410 – continued from previous page

标注	名称	类型	描述
Target CRS	TARGET_CRS	[crs] Default: Project CRS	CRS for the output raster layer
Pixel size	PIXEL_SIZE	[numeric: double] 默认值：1.0	Pixel size (X=Y) in map units.
Output raster	OUTPUT	[栅格] 默认：[保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output raster data type	OUTPUT_TYPE Default: 0	[enumeration]	Defines the data type of the output raster file. Options: • 0 --- Integer16 • 1 --- Unsigned Integer16 • 2 --- Integer32 • 3 --- Unsigned Integer32 • 4 --- Float32 • 5 --- Float64
Mean	MEAN	[numeric: double] 默认值：1.0	
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output raster	OUTPUT	[栅格]	Raster covering the desired extent with the cell size filled with randomly distributed values

Python 代码

Algorithm ID: `native:createrandompoissonrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Create random raster layer (uniform distribution)

Generates a raster layer for given extent and cell size filled with random values.

By default, the values will range between the minimum and maximum value of the specified output raster type. This can be overridden by using the advanced parameters for lower and upper bound value. If the bounds have the same value or both are zero (default) the algorithm will create random values in the full value range of the chosen raster data type. Choosing bounds outside the acceptable range of the output raster type will abort the algorithm.

参数

Basic parameters

标注	名称	类型	描述
Desired extent	`EXTENT`	[范围]	Specify the spatial extent of the output raster layer. It will internally be extended to a multiple of the tile size. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Target CRS	`TARGET_CRS`	[crs] Default: Project CRS	CRS for the output raster layer
Pixel size	`PIXEL_SIZE`	[numeric: double] 默认值：1.0	Pixel size (X=Y) in map units.
Output raster	`OUTPUT`	[栅格] 默认：[保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Output raster data type	`OUTPUT_TYPE` Default: 5	[enumeration]	Defines the data type of the output raster file. Options: • 0 --- Byte • 1 --- Integer16 • 2 --- Unsigned Integer16 • 3 --- Integer32 • 4 --- Unsigned Integer32 • 5 --- Float32 • 6 --- Float64
Lower bound for random number range	`LOWER_BOUND`	[numeric: double] Default: 0.0	
Upper bound for random number range	`UPPER_BOUND`	[numeric: double] Default: 0.0	
Creation options 可选	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output raster	`OUTPUT`	[栅格]	Raster covering the desired extent with the cell size filled with randomly distributed values

Python 代码

Algorithm ID: `native:createrandomuniformrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

24.1.20 栅格地形分析

坡向

在数字地形模型中计算坡向，最终的坡向栅格图层包含了从 0 到 360 的数值去代表斜坡的倾向。坡向以正北为 (0°)，顺时针方向递增。

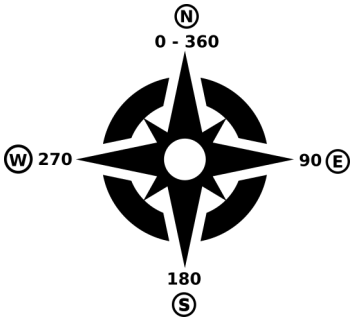


Fig. 24.60: 坡向值

下图展示了用色带重分类的坡向图层

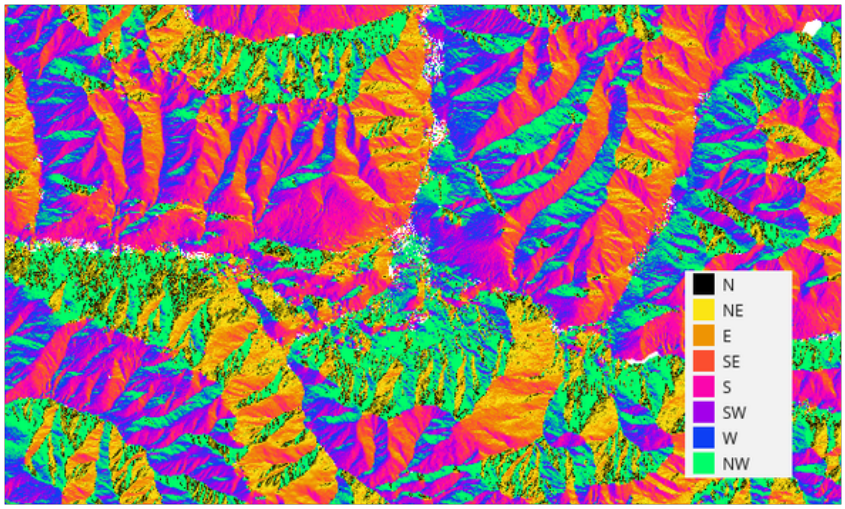


Fig. 24.61: 重分类后的坡向图层

参数

标注	名称	类型	描述
高程图层	INPUT	[栅格]	数字地形模型的栅格图层
Z 因子	Z_FACTOR	[numeric: double] 默认值：1.0	纵向拉伸。当 z 单位与 x，y 单位不同时，例如米与英尺，该参数十分有用。你可以通过这个参数去进行调整。该参数默认值为 1（没有拉伸）。
坡向	OUTPUT	[栅格] 默认：[保存到临时文件]	Specify the output aspect raster layer. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
坡向	`OUTPUT`	[栅格]	输出的坡向栅格图层

Python 代码

Algorithm ID: `native:aspect`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

DTM filter (slope-based)

Can be used to filter a digital elevation model in order to classify its cells into ground and object (non-ground) cells.

The tool uses concepts as described by Vosselman (2000) and is based on the assumption that a large height difference between two nearby cells is unlikely to be caused by a steep slope in the terrain. The probability that the higher cell might be non-ground increases when the distance between the two cells decreases. Therefore the filter defines a maximum height difference (`dz_max`) between two cells as a function of the distance (`d`) between the cells (`dz_max( d ) = d`). A cell is classified as terrain if there is no cell within the kernel radius to which the height difference is larger than the allowed maximum height difference at the distance between these two cells.

The approximate terrain slope (`s`) parameter is used to modify the filter function to match the overall slope in the study area (`dz_max( d ) = d * s`). A 5 % confidence interval (`ci = 1.65 * sqrt( 2 * stddev )`) may be used to modify the filter function even further by either relaxing (`dz_max( d ) = d * s + ci`) or amplifying (`dz_max( d ) = d * s - ci`) the filter criterium.

References: Vosselman, G. (2000): Slope based filtering of laser altimetry data. IAPRS, Vol. XXXIII, Part B3, Amsterdam, The Netherlands, 935-942

See also

This tool is a port of the SAGA DTM Filter (slope-based)

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	数字地形模型的栅格图层
Band number	`BAND`	[raster band]	The band of the DEM to consider
Kernel radius (pixels)	`RADIUS`	[numeric: integer] Default: 5	The radius of the filter kernel (in pixels). Must be large enough to reach ground cells next to non-ground objects.

continues on next page

Table 24.418 – continued from previous page

标注	名称	类型	描述
Terrain slope (%, pixel size/vertical units)	TERRAIN_SLOPE	[numeric: double] Default: 30.0	The approximate terrain slope in %. The terrain slope must be adjusted to account for the ratio of height units vs raster pixel dimensions. Used to relax the filter criterium in steeper terrain.
Filter modification	FILTER_MODIFIC	[list] Default: 0	Choose whether to apply the filter kernel without modification or to use a confidence interval to relax or amplify the height criterium. • 0 - None • 1 - Relax filter • 2 - Amplify
Standard deviation	STANDARD_DEVIA	[numeric: double] Default: 0.1	The standard deviation used to calculate a 5% confidence interval applied to the height threshold.
Output layer (ground) 可选	OUTPUT_GROUND	[栅格] 默认: [保存到临时文件]	Specify the filtered DEM containing only cells classified as ground. *One of*: • Skip Output • 保存到临时文件 • 保存到文件…
Output layer (non-ground objects) 可选	OUTPUT_NONGROU	[栅格] Default: [Skip output]	Specify the non-ground objects removed by the filter. *One of*: • Skip Output • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output layer (ground)	OUTPUT_GROUND	[栅格]	The filtered DEM containing only cells classified as ground.
Output layer (non-ground objects)	OUTPUT_NONGROU	[栅格]	The non-ground objects removed by the filter.

Python 代码

Algorithm ID: `native:dtmslopebasedfilter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Fill sinks (Wang & Liu)

`NEW in 3.44`

Uses a method proposed by Wang & Liu to identify and fill surface depressions in digital elevation models.

The method was enhanced to allow the creation of hydrologically sound elevation models, i.e. not only to fill the depression(s) but also to preserve a downward slope along the flow path. If desired, this is accomplished by preserving a minimum slope gradient (and thus elevation difference) between cells.

References: Wang, L. & H. Liu (2006): An efficient method for identifying and filling surface depressions in digital elevation models for hydrologic analysis and modelling. International Journal of Geographical Information Science, Vol. 20, No. 2: 193-213.

See also

This tool is a port of the SAGA Fill Sinks (Wang & Liu) tool.

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[栅格]	数字地形模型的栅格图层
Band number	BAND	[raster band] Default: 1	The band of the DEM to consider
Minimum slope (degrees)	MIN_SLOPE	[numeric: double] Default: 0.10	Minimum slope gradient to preserve from cell to cell; with a value of zero, sinks are filled up to the spill elevation (which results in flat areas).

continues on next page

Table 24.421 – continued from previous page

标注	名称	类型	描述
Output layer (filled DEM) 可选	OUTPUT_FILLED_	[栅格] 默认: [保存到临时文件]	Specify the ouput raster corresponding to the depression-free DEM. *One of*: • Skip Output • 保存到临时文件 • 保存到文件…
Output layer (flow directions) 可选	OUTPUT_FLOW_DI	[栅格] Default: [Skip output]	Specify the output raster with computed flow directions. *One of*: • Skip Output • 保存到临时文件 • 保存到文件…
Output layer (watershed basins) 可选	OUTPUT_WATERSH	[栅格] Default: [Skip output]	Specify the output raster corresponding to the delineated watershed basins. *One of*: • Skip Output • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Creation options 可选	CREATION_OPTIO	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output layer (filled DEM)	OUTPUT_FILLED_	[栅格]	Ouput raster corresponding to the depression-free digital elevation model.
Output layer (flow directions)	OUTPUT_FLOW_DI	[栅格]	Output raster with computed flow directions; 0=N, 1=NE, 2=E, ... 7=NW.
Output layer (watershed basins)	OUTPUT_WATERSH	[栅格]	Output raster corresponding to the delineated watershed basins.

Python 代码

Algorithm ID: `native:fillsinkswangliu`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

山体阴影

基于输入的数字地形模型计算山体阴影的栅格图层

图层的晕渲是根据太阳位置进行计算的：你可以选择更改太阳投射的水平角度（方位角）和垂直角度（太阳高度）。

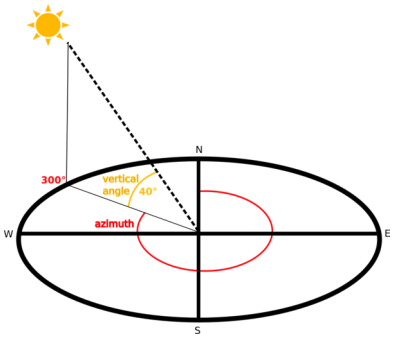


Fig. 24.62: 方位角与垂向角度

山体阴影图层的数值范围覆盖从 0（完全遮蔽）到 255（完全暴露）。山体阴影通常用于更好地理解区域中的地势。

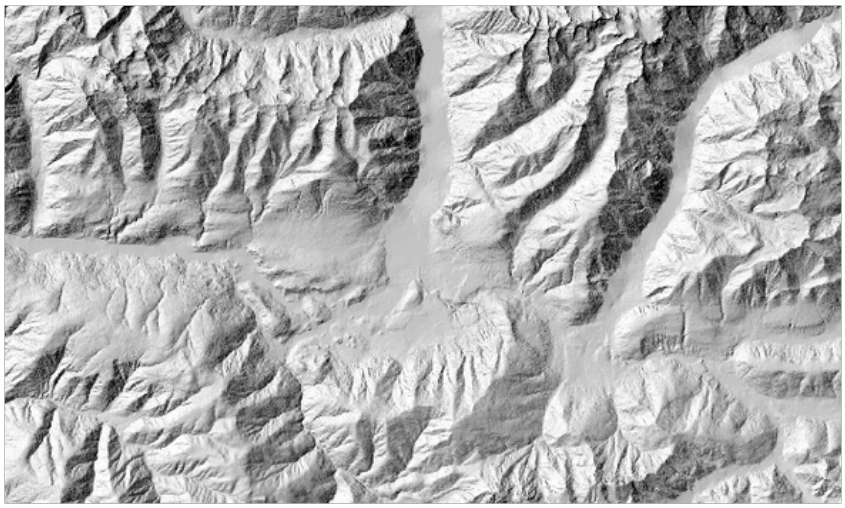


Fig. 24.63: 山体阴影图层一般设定为 300° 的方位角以及 45° 的垂向角度。

值得注意的是可以将山体阴影图层设定某个透明度值后重叠在高程栅格之上。

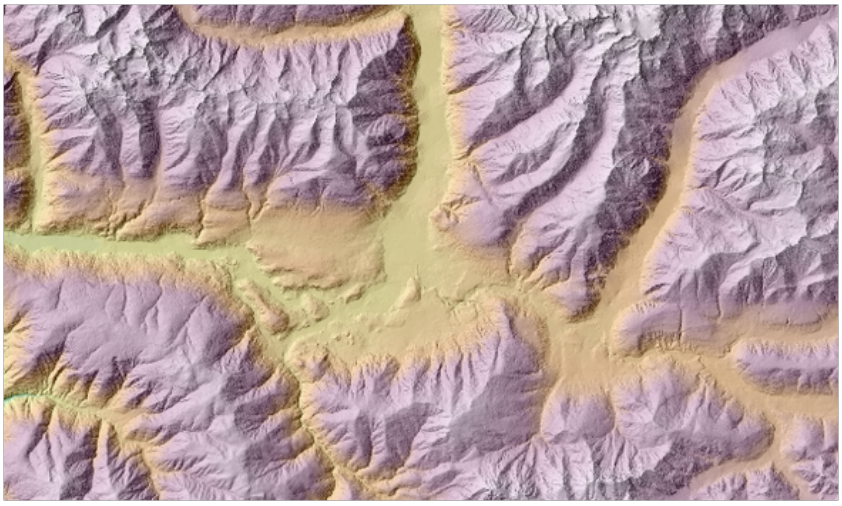


Fig. 24.64: 重叠是山体阴影叠上高程图层

参数

标注	名称	类型	描述
高程图层	INPUT	[栅格]	数字地形模型的栅格图层
Z 因子	Z_FACTOR	[numeric: double] 默认值：1.0	纵向拉伸。当 z 单位与 x，y 单位不同时，例如米与英尺，该参数十分有用。你可以通过这个参数去进行调整。随着该参数数值的增加，最终的结果会更加夸张（更加“山体化”）。该参数默认值为 1（没有拉伸）
方位角（水平角度）	AZIMUTH	[numeric: double] 默认值：300.0	Set the horizontal angle (in degrees) of the sun (clockwise direction). Range: 0 to 360. 0 is north.
仰角	V_ANGLE	[numeric: double] 默认: 40.0	Set the vertical angle (in degrees) of the sun, that is the height of the sun. Values can go from 0 (minimum elevation) to 90 (maximum elevation).
山体阴影	OUTPUT	[栅格] 默认: 保存到临时文件	Specify the output hillshade raster layer. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
山体阴影	OUTPUT	[栅格]	The output hillshade raster layer

Python 代码

Algorithm ID: `native:hillshade`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Hypsometric curves

Calculates hypsometric curves for an input Digital Elevation Model. Curves are produced as CSV files in an output folder specified by the user.

A hypsometric curve is a cumulative histogram of elevation values in a geographical area.

You can use hypsometric curves to detect differences in the landscape due to the geomorphology of the territory.

参数

标注	名称	类型	描述
DEM to analyze	`INPUT_DEM`	[栅格]	Digital Terrain Model raster layer to use for calculating altitudes
Boundary layer	`BOUNDARY_LAYER`	[vector: polygon]	Polygon vector layer with boundaries of areas used to calculate hypsometric curves
Step	`STEP`	[numeric: double] Default: 100.0	Vertical distance between curves
Use % of area instead of absolute value	`USE_PERCENTAGE`	[boolean] Default: False	Write area percentage to "Area" field of the CSV file instead of the absolute area
Hypsometric curves	`OUTPUT_DIRECTO`	[folder]	Specify the output folder for the hypsometric curves. *One of*: • Save to a Temporary Directory • Save to Directory

输出

标注	名称	类型	描述
Hypsometric curves	`OUTPUT_DIRECTO`	[folder]	Directory containing the files with the hypsometric curves. For each feature from the input vector layer, a CSV file with area and altitude values will be created. The file names start with `histogram_`, followed by layer name and feature ID.

	A	B
1	Area	Elevation
2	177475194.383	307
3	233206029.24	407
4	295553735.793	507
5	394718815.615	607
6	501801102.615	707
7	624399019.792	807
8	828877274.39	907
9	1042693465.68	1007
10	1277373021.81	1107
11	1556443975.41	1207
12	1888617494.27	1307
13	2248520437.31	1407
14	2627916813.17	1507
15	3010880212.04	1607
16	3411087555.34	1707

Python 代码

Algorithm ID: `qgis:hypsometriccurves`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Relief

Creates a shaded relief layer from digital elevation data. You can specify the relief color manually, or you can let the algorithm choose automatically all the relief classes.

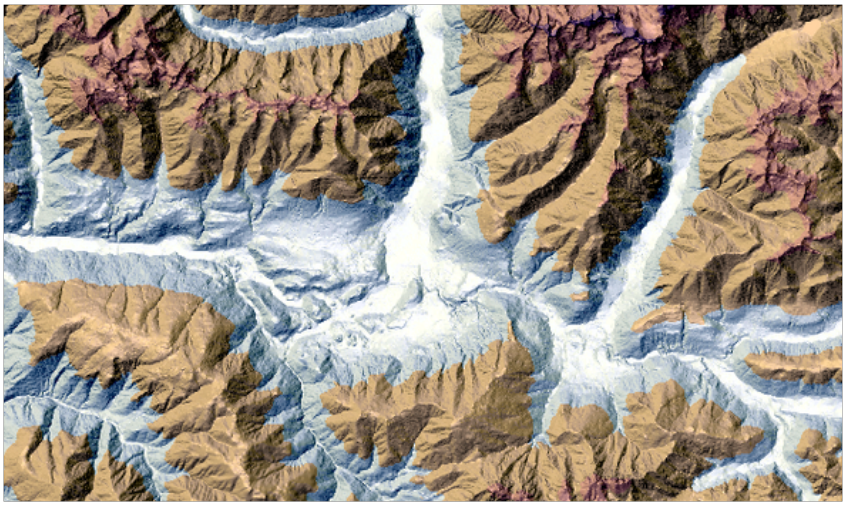


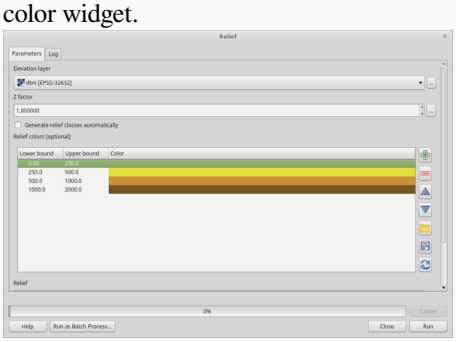
Fig. 24.65: Relief layer

参数

标注	名称	类型	描述
高程图层	INPUT	[栅格]	数字地形模型的栅格图层
Z 因子	Z_FACTOR	[numeric: double] 默认值：1.0	纵向拉伸。当 z 单位与 x，y 单位不同时，例如米与英尺，该参数十分有用。你可以通过这个参数去进行调整。随着该参数数值的增加，最终的结果会更加夸张（更加“山体化”）。该参数默认值为 1（没有拉伸）
Generate relief classes automatically	AUTO_COLORS	[boolean] Default: False	If you check this option the algorithm will create all the relief color classes automatically

continues on next page

Table 24.428 – continued from previous page

标注	名称	类型	描述
Relief colors 可选	COLORS	[table widget]	Use the table widget if you want to choose the relief colors manually. You can add as many color classes as you want: for each class you can choose the lower and upper bound and finally by clicking on the color row you can choose the color thanks to the color widget.  Fig. 24.66: Manually setting of relief color classes The buttons in the right side panel give you the chance to: add or remove color classes, change the order of the color classes already defined, open an existing file with color classes and save the current classes as file.
Relief	OUTPUT	[栅格] 默认: [保存到临时文件]	Specify the output relief raster layer. *One of*: • 保存到临时文件 • 保存到文件…
Frequency distribution 可选	FREQUENCY_DIST	[vector: table] Default: [Skip output]	Specify the CSV table for the output frequency distribution. *One of*: • Skip Output • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Relief	OUTPUT	[栅格]	The output relief raster layer
Frequency distribution	OUTPUT	[vector: table]	The output frequency distribution

Python 代码

Algorithm ID: `qgis:relief`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Ruggedness index

Calculates the quantitative measurement of terrain heterogeneity described by Riley et al. (1999). It is calculated for every location, by summarizing the change in elevation within the 3x3 pixel grid.

Each pixel contains the difference in elevation from a center cell and the 8 cells surrounding it.

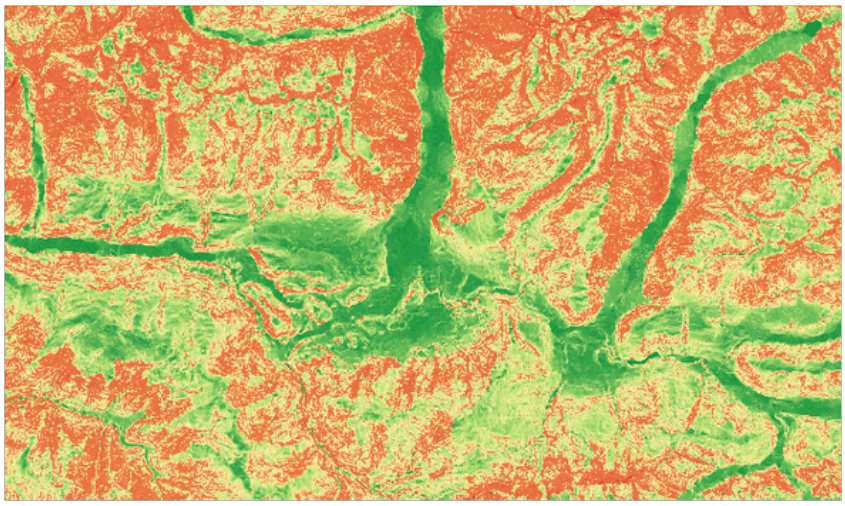


Fig. 24.67: Ruggedness layer from low (red) to high values (green)

参数

标注	名称	类型	描述
高程图层	`INPUT`	[栅格]	数字地形模型的栅格图层
Z 因子	`Z_FACTOR`	[numeric: double] 默认值：1.0	Vertical exaggeration. This parameter is useful when the Z units differ from the X and Y units, for example feet and meters. You can use this parameter to adjust for this. Increasing the value of this parameter will exaggerate the final result (making it look more rugged). The default is 1 (no exaggeration).
Ruggedness	`OUTPUT`	[栅格] 默认：[保存到临时文件]	Specify the output ruggedness raster layer. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Ruggedness	OUTPUT	[栅格]	The output ruggedness raster layer

Python 代码

Algorithm ID: `native:ruggednessindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

坡度

Calculates the slope from an input raster layer. The slope is the angle of inclination of the terrain and is expressed in **degrees**.

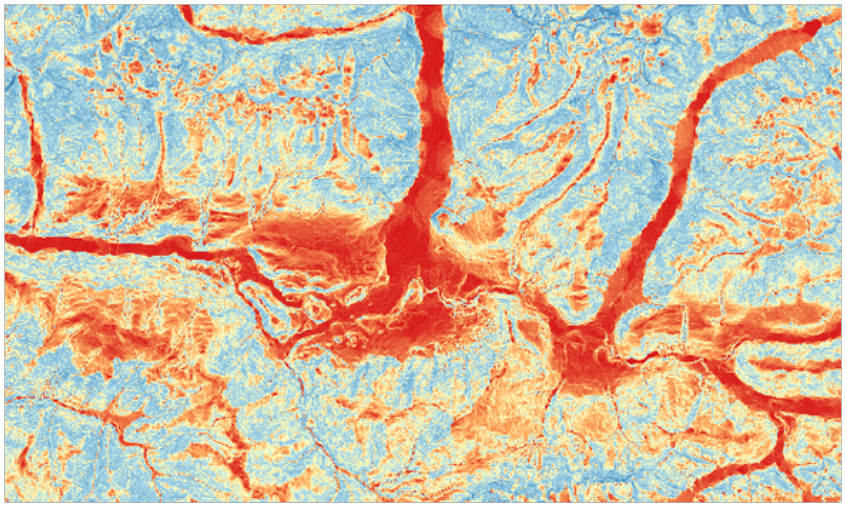


Fig. 24.68: Flat areas in red, steep areas in blue

参数

标注	名称	类型	描述
高程图层	INPUT	[栅格]	数字地形模型的栅格图层
Z 因子	Z_FACTOR	[numeric: double] 默认值：1.0	Vertical exaggeration. This parameter is useful when the Z units differ from the X and Y units, for example feet and meters. You can use this parameter to adjust for this. Increasing the value of this parameter will exaggerate the final result (making it steeper). The default is 1 (no exaggeration).

continues on next page

Table 24.432 – continued from previous page

标注	名称	类型	描述
Slope	OUTPUT	[栅格] 默认: [保存到临时文件]	Specify the output slope raster layer. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Slope	OUTPUT	[栅格]	The output slope raster layer

Python 代码

Algorithm ID: `native:slope`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.21 栅格工具

Align raster

Aligns raster by resampling it to the same cell size and reprojecting to the same CRS as a reference raster.

> **Warning**
>
> **This algorithm is ONLY available in the** *Model Designer* context. For other contexts, use instead *Align rasters*.

参数

标注	名称	类型	描述
输入图层	INPUT	[栅格]	Input raster layer to align

continues on next page

Table 24.434 – continued from previous page

标注	名称	类型	描述
Resampling method	RESAMPLING_MET	[enumeration] Default: 0	Method to use for input layer resampling. Available options are: • 0 --- Nearest Neighbour • 1 --- Bilinear (2x2 kernel) • 2 --- Cubic (4x4 kernel) • 3 --- Cubic B-Spline (4x4 kernel) • 4 --- Lanczos (6x6 kernel) • 5 --- Average • 6 --- Mode • 7 --- Maximum • 8 --- Minimum • 9 --- Median • 10 --- First quartile (Q1) • 11 --- Third quartile (Q3)
Rescale values according to the cell size	RESCALE	[boolean] Default: No	
Reference layer	REFERENCE_LAYE	[栅格]	A raster layer that will be used to fetch extent, cell size and CRS that will be applied to input layers.
Override reference CRS 可选	CRS	[crs]	CRS to be used instead of the reference layer's
Override reference cell size X 可选	CELL_SIZE_X	[numeric: double]	Cell size in X direction to be used instead of the reference layer's
Override reference cell size Y 可选	CELL_SIZE_Y	[numeric: double]	Cell size in Y direction to be used instead of the reference layer's
Override reference grid offset X 可选	GRID_OFFSET_X	[numeric: double]	Offset in X direction to apply to cells grid
Override reference grid offset Y 可选	GRID_OFFSET_Y	[numeric: double]	Offset in Y direction to apply to cells grid
Clip to extent 可选	EXTENT	[范围]	Specify the extent of the output raster layer. It will internally be extended to a multiple of the tile size. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as xmin, xmax, ymin, ymax

continues on next page

Table 24.434 – continued from previous page

标注	名称	类型	描述
Aligned raster	OUTPUT	[栅格]	output raster with pixels resampled

输出

标注	名称	类型	描述
Aligned raster	OUTPUT	[栅格]	output raster with pixels resampled

Python 代码

Algorithm ID: `native:alignsingleraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Align rasters

Aligns rasters by resampling them to the same cell size and reprojecting to the same CRS as a reference raster.

Warning

This algorithm is NOT available in the *Model Designer* context. Use instead *Align raster*.

参数

标注	名称	类型	描述
Input layers	INPUT	[raster] [list]	List of input raster layers with resampling options associated (filled as a QgsProcessingParameterAlignRasterLayers item --- done in GUI by pressing *Configure Raster...* button for each selected layer): **Input layer [string] (inputFile)** Full path of the input layer to align **Output file [string] (outputFile)** Full path of the corresponding aligned output layer **Resampling method [enumeration] (resampleMethod)** Method to use for input layer resampling. Available options are: • 0 --- Nearest Neighbour • 1 --- Bilinear (2x2 kernel) • 2 --- Cubic (4x4 kernel) • 3 --- Cubic B-Spline (4x4 kernel) • 4 --- Lanczos (6x6 kernel) • 5 --- Average • 6 --- Mode • 7 --- Maximum • 8 --- Minimum • 9 --- Median • 10 --- First quartile (Q1) • 11 --- Third quartile (Q3) **Rescale values according to the cell size** [boolean] (rescale)
Reference layer	REFERENCE_LAYE	[栅格]	A raster layer that will be used to fetch extent, cell size and CRS that will be applied to input layers.
Override reference CRS 可选	CRS	[crs]	CRS to be used instead of the reference layer's
Override reference cell size X 可选	CELL_SIZE_X	[numeric: double]	Cell size in X direction to be used instead of the reference layer's
Override reference cell size Y 可选	CELL_SIZE_Y	[numeric: double]	Cell size in Y direction to be used instead of the reference layer's
Override reference grid offset X 可选	GRID_OFFSET_X	[numeric: double]	Offset in X direction to apply to cells grid
Override reference grid offset Y 可选	GRID_OFFSET_Y	[numeric: double]	Offset in Y direction to apply to cells grid

continues on next page

Table 24.436 – continued from previous page

标注	名称	类型	描述
Clip to extent 可选	`EXTENT`	[范围]	Specify the extent of the output raster layer. It will internally be extended to a multiple of the tile size. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`

输出

标注	名称	类型	描述
Aligned rasters	`OUTPUT_LAYERS`	[raster] [list]	output rasters with pixels resampled

Python 代码

Algorithm ID: `native:alignrasters`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

将地图转换为栅格

Creates a raster image of map canvas content.

A *map theme* can be selected to render a predetermined set of layers with a defined style for each layer.

Alternatively, a single layer can be selected if no map theme is set.

If neither map theme nor layer is set, the current map content will be rendered. The minimum extent entered will internally be extended to be a multiple of the tile size.

参数

标注	名称	类型	描述
Minimum extent to render (xmin, xmax, ymin, ymax)	`EXTENT`	[范围]	Specify the extent of the output raster layer. It will internally be extended to a multiple of the tile size. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
瓦片尺寸	`TILE_SIZE`	[numeric: integer] 默认值: 1024	Size of the tile of the output raster layer. Minimum value: 64.
Map units per pixel	`MAP_UNITS_PER_`	[numeric: double] Default: 100.0	Pixel size (in map units). Minimum value: 0.0
Make background transparent	`MAKE_BACKGROUN`	[boolean] Default: False	Allows exporting the map with a transparent background. Outputs an RGBA (instead of RGB) image if set to `True`.
Map theme to render 可选	`MAP_THEME`	[enumeration]	Use an existing *map theme* for the rendering.
Single layer to render 可选	`LAYER`	[enumeration]	Choose a single layer for the rendering
Output layer	`OUTPUT`	[栅格] 默认: [保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Output layer	`OUTPUT`	[栅格]	Output raster layer

Python 代码

Algorithm ID: `native:rasterize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Fill NoData cells

Resets the NoData values in the input raster to a chosen value, resulting in raster dataset with no NoData pixels.

The algorithm respects the input raster data type, e.g. a floating point fill value will be truncated when applied to an integer raster.

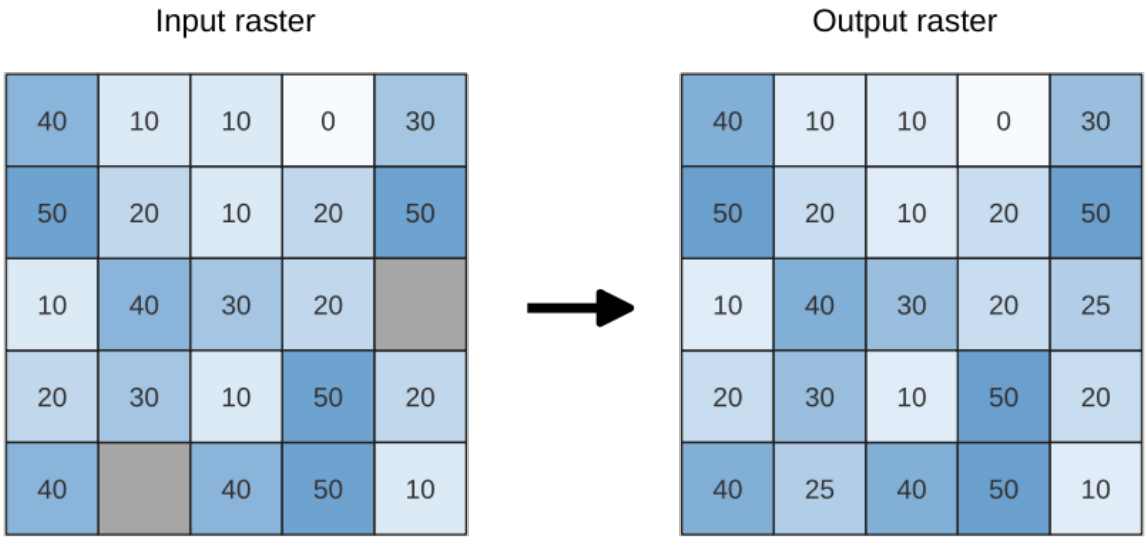


Fig. 24.69: Filling NoData values (in grey) of a raster

参数

Basic parameters

标注	名称	类型	描述
输入栅格	`INPUT`	[栅格]	The raster to process.
Band number	`BAND`	[raster band] Default: 1	The band of the raster
Fill value	`FILL_VALUE`	[numeric: double] 默认值：1.0	Set the value to use for the NoData pixels
Output raster	`OUTPUT`	[栅格] 默认：[保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Output raster	OUTPUT	[栅格]	The output raster layer with filled data cells.

Python 代码

Algorithm ID: `native:fillnodata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Generate XYZ tiles (Directory)

Generates raster "XYZ" tiles using the current QGIS project as individual images to a directory structure.

Optionally, a Leaflet HTML output file using the generated tiles as a map layer could be created.

参数

Basic parameters

标注	名称	类型	描述
Extent (xmin, xmax, ymin, ymax)	`EXTENT`	[范围]	Specify the extent of the tiles. It will internally be extended to a multiple of the tile size. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Minimum zoom	`ZOOM_MIN`	[numeric: integer] Default: 12	Minimum 0, maximum 25.
Maximum zoom	`ZOOM_MAX`	[numeric: integer] Default: 12	Minimum 0, maximum 25.
DPI	`DPI`	[numeric: integer] Default: 96	Minimum 48, maximum 600.
Background color 可选	`BACKGROUND_COL`	[color] Default: QColor(0, 0, 0, 0)	Choose the background color for the tiles
Enable antialiasing	`ANTIALIAS`	[boolean] Default: True	Determines if antialiasing should be enabled
Tile format	`TILE_FORMAT`	[enumeration] Default: 0	One of: • 0 --- PNG • 1 --- JPG
Quality (JPG only) 可选	`QUALITY`	[numeric: integer] Default: 75	Minimum 1, maximum 100.
Metatile size 可选	`METATILESIZE`	[numeric: integer] Default: 4	Specify a custom metatile size when generating XYZ tiles. Larger values may speed up the rendering of tiles and provide better labelling (fewer gaps without labels) at the expense of using more memory. Minimum 1, maximum 20.
Tile width 可选	`TILE_WIDTH`	[numeric: integer] Default: 256	Minimum 1, maximum 4096.
Tile height 可选	`TILE_HEIGHT`	[numeric: integer] Default: 256	Minimum 1, maximum 4096.
Use inverted tile Y axis (TMS conventions)	`TMS_CONVENTION`	[boolean] Default: False	

continues on next page

Table 24.443 – continued from previous page

标注	名称	类型	描述
Output directory	OUTPUT_DIRECTO	[folder] Default: [Save to temporary folder]	Specification of the output directory (for the tiles). *One of*: • Skip Output • Save to a Temporary Directory • Save to Directory
Output html (Leaflet) 可选	OUTPUT_HTML	[html] 默认: [保存到临时文件]	Specification of the output HTML file. *One of*: • Skip Output • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Leaflet HTML output title 可选	HTML_TITLE	[string] Default: Not set	HTML <title>-tag used for the Leaflet HTML output file.
Leaflet HTML output attribution 可选	HTML_ATTRIBUTI	[string] Default: Not set	Custom map attribution used for the Leaflet HTML output file. HTML links are possible.
Include OpenStreetMap basemap in Leaflet HTML output	HTML_OSM	[boolean] Default: False	An OpenStreetMap basemap layer (source: https://tile.openstreetmap.org) is included in the Leaflet HTML output file. Proper map attribution is added automatically.

输出

标注	名称	类型	描述
Output directory	OUTPUT_DIRECTO	[folder]	Output directory (for the tiles)
Output html (Leaflet)	OUTPUT_HTML	[html]	The output HTML (Leaflet) file

Python 代码

Algorithm ID: `native:tilesxyzdirectory`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Generate XYZ tiles (MBTiles)

Generates raster "XYZ" tiles using the current QGIS project as a single file in the "MBTiles" format.

参数

标注	名称	类型	描述
Extent (xmin, xmax, ymin, ymax)	`EXTENT`	[范围]	Specify the extent of the tiles. It will internally be extended to a multiple of the tile size. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Minimum zoom	`ZOOM_MIN`	[numeric: integer] Default: 12	Minimum 0, maximum 25.
Maximum zoom	`ZOOM_MAX`	[numeric: integer] Default: 12	Minimum 0, maximum 25.
DPI	`DPI`	[numeric: integer] Default: 96	Minimum 48, maximum 600.
Background color 可选	`BACKGROUND_COL`	[color] Default: QColor(0, 0, 0, 0)	Choose the background color for the tiles
Enable antialiasing	`ANTIALIAS`	[boolean] Default: True	Determines if antialiasing should be enabled
Tile format	`TILE_FORMAT`	[enumeration] Default: 0	One of: • 0 --- PNG • 1 --- JPG
Quality (JPG only) 可选	`QUALITY`	[numeric: integer] Default: 75	Minimum 1, maximum 100.
Metatile size 可选	`METATILESIZE`	[numeric: integer] Default: 4	Specify a custom metatile size when generating XYZ tiles. Larger values may speed up the rendering of tiles and provide better labelling (fewer gaps without labels) at the expense of using more memory. Minimum 1, maximum 20.

continues on next page

Table 24.446 – continued from previous page

标注	名称	类型	描述
Output file (for MBTiles)	OUTPUT_FILE	[file] 默认：[保存到临时文件]	Specification of the output file. *One of*: • Skip Output • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Output file (for MBTiles)	OUTPUT_FILE	[file]	The output file.

Python 代码

Algorithm ID: `native:tilesxyzmbtiles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.22 Vector analysis

Basic statistics for fields

Generates basic statistics for a field of the attribute table of a vector layer. Numeric, date, time and string fields are supported. The statistics returned will depend on the field type.

Statistics can be generated as a table or an HTML file and are available in the *Processing* ► *Results viewer*.

Default menu: *Vector* ► *Analysis Tools*

参数

标注	名称	类型	描述
Input vector	INPUT_LAYER	[vector: any]	Vector layer to calculate the statistics on
Field to calculate statistics on	FIELD_NAME	[tablefield: any]	Any supported table field to calculate the statistics

continues on next page

Table 24.448 – continued from previous page

标注	名称	类型	描述
Statistics 可选	`OUTPUT`	[vector: table] Default: `[Create temporary layer]`	Specify the output table for the generated statistics. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。
Statistics report 可选	`OUTPUT_HTML_FI`	[html] 默认: `[保存到临时文件]`	Specification of the file for the calculated statistics. *One of*: • Skip Output • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Statistics	`OUTPUT`	[vector: table]	Table containing the calculated statistics
Statistics report	`OUTPUT_HTML_FI`	[html]	HTML file with the calculated statistics
Count	`COUNT`	[numeric: integer]	
Number of unique values	`UNIQUE`	[numeric: integer]	
Number of empty (null) values	`EMPTY`	[numeric: integer]	
Number of non-empty values	`FILLED`	[numeric: integer]	
Minimum value	`MIN`	[same as input]	
Maximum value	`MAX`	[same as input]	
Minimum length	`MIN_LENGTH`	[numeric: integer]	
Maximum length	`MAX_LENGTH`	[numeric: integer]	
Mean length	`MEAN_LENGTH`	[numeric: double]	
Coefficient of Variation	`CV`	[numeric: double]	
Sum	`SUM`	[numeric: double]	
Mean value	`MEAN`	[numeric: double]	
Standard deviation	`STD_DEV`	[numeric: double]	
Range	`RANGE`	[numeric: double]	
Median	`MEDIAN`	[numeric: double]	
Minority (rarest occurring value)	`MINORITY`	[same as input]	
Majority (most frequently occurring value)	`MAJORITY`	[same as input]	
First quartile	`FIRSTQUARTILE`	[numeric: double]	
Third quartile	`THIRDQUARTILE`	[numeric: double]	
Interquartile Range (IQR)	`IQR`	[numeric: double]	

Python 代码

Algorithm ID: `native:basicstatisticsforfields`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Climb along line

Calculates the total climb and descent along line geometries. The input layer must have Z values present. If Z values are not available, the *Drape (set Z value from raster)* algorithm may be used to add Z values from a DEM layer.

The output layer is a copy of the input layer with additional fields that contain the total climb (`climb`), total descent (`descent`), the minimum elevation (`minelev`) and the maximum elevation (`maxelev`) for each line geometry. If the input layer contains fields with the same names as these added fields, they will be renamed (field names will be altered to "name_2", "name_3", etc, finding the first non-duplicate name).

参数

标注	名称	类型	描述
Line layer	`INPUT`	[vector: line]	Line layer to calculate the climb for. Must have Z values
Climb layer	`OUTPUT`	[vector: line] Default: `[Create temporary layer]`	Specification of the output (line) layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Climb layer	`OUTPUT`	[vector: line]	Line layer containing new attributes with the results from climb calculations.
Total climb	`TOTALCLIMB`	[numeric: double]	The sum of the climb for all the line geometries in the input layer
Total descent	`TOTALDESCENT`	[numeric: double]	The sum of the descent for all the line geometries in the input layer
Minimum elevation	`MINELEVATION`	[numeric: double]	The minimum elevation for the geometries in the layer
Maximum elevation	`MAXELEVATION`	[numeric: double]	The maximum elevation for the geometries in the layer

Python 代码

Algorithm ID: `native:climbalongline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Count points in polygon

Takes a point and a polygon layer and counts the number of points from the point layer in each of the polygons of the polygon layer.

A new polygon layer is generated, with the exact same content as the input polygon layer, but containing an additional field with the points count corresponding to each polygon.

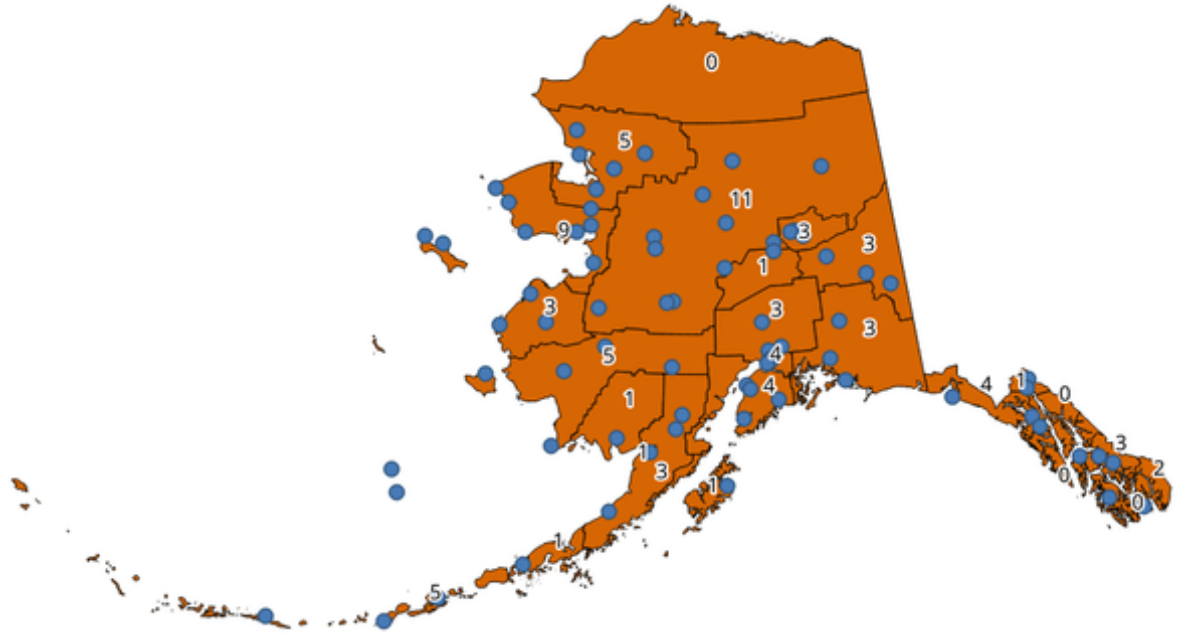


Fig. 24.70: The labels in the polygons show the point count

An optional weight field can be used to assign weights to each point. Alternatively, a unique class field can be specified. If both options are used, the weight field will take precedence and the unique class field will be ignored.

Allows *features in-place modification* of polygon features

`Default menu`: *Vector* ► *Analysis Tools*

参数

标注	名称	类型	描述
Polygons	`POLYGONS`	[vector: polygon]	Polygon layer whose features are associated with the count of points they contain
Points	`POINTS`	[vector: point]	Point layer with features to count
Weight field 可选	`WEIGHT`	[tablefield: any]	A field from the point layer. The count generated will be the sum of the weight field of the points contained by the polygon. If the weight field is not numeric, the count will be `0`.
Class field 可选	`CLASSFIELD`	[tablefield: any]	Points are classified based on the selected attribute and if several points with the same attribute value are within the polygon, only one of them is counted. The final count of the points in a polygon is, therefore, the count of different classes that are found in it.
Count field name	`FIELD`	[string] Default: 'NUMPOINTS'	The name of the field to store the count of points
Count	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Count	`OUTPUT`	[vector: polygon]	Resulting layer with the attribute table containing the new column with the points count

Python 代码

Algorithm ID: `native:countpointsinpolygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

DBSCAN clustering

Clusters point features based on a 2D implementation of Density-based spatial clustering of applications with noise (DBSCAN) algorithm.

The algorithm requires two parameters, a minimum cluster size, and the maximum distance allowed between clustered points.

See also

ST-DBSCAN clustering, *K-means clustering*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[vector: point]	Layer to analyze
Minimum cluster size	`MIN_SIZE`	[numeric: integer] Default: 5	Minimum number of features to generate a cluster
Maximum distance between clustered points	`EPS`	[numeric: double] 默认值：1.0	Distance beyond which two features can not belong to the same cluster (eps)
Clusters	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the vector layer for the result of the clustering. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Treat border points as noise (DBSCAN*)	`DBSCAN*`	[boolean] Default: False	If checked, points on the border of a cluster are themselves treated as unclustered points, and only points in the interior of a cluster are tagged as clustered.
Cluster field name	`FIELD_NAME`	[string] Default: 'CLUSTER_ID'	Name of the field where the associated cluster number shall be stored
Cluster size field name	`SIZE_FIELD_NAM`	[string] Default: 'CLUSTER_SIZE'	Name of the field with the count of features in the same cluster

输出

标注	名称	类型	描述
Clusters	`OUTPUT`	[vector: point]	Vector layer containing the original features with a field setting the cluster they belong to
Number of clusters	`NUM_CLUSTERS`	[numeric: integer]	The number of clusters discovered

Python 代码

Algorithm ID: `native:dbscanclustering`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Distance matrix

Calculates for point features distances to their nearest features in the same layer or in another layer.

Default menu: *Vector* ► *Analysis Tools*

See also

Join attributes by nearest

参数

标注	名称	类型	描述
Input point layer	`INPUT`	[vector: point]	Point layer for which the distance matrix is calculated (**from** points)
Input unique ID field	`INPUT_FIELD`	[tablefield: any]	Field to use to uniquely identify features of the input layer. Used in the output attribute table.
Target point layer	`TARGET`	[vector: point]	Point layer containing the nearest point(s) to search (**to** points)
Target unique ID field	`TARGET_FIELD`	[tablefield: any]	Field to use to uniquely identify features of the target layer. Used in the output attribute table.

continues on next page

Table 24.457 – continued from previous page

标注	名称	类型	描述
Output matrix type	MATRIX_TYPE	[enumeration] Default: 0	Different types of calculation are available: • 0 --- Linear (N * *k* x 3) distance matrix: for each input point, reports the distance to each of the *k* nearest target points. The output matrix consists of up to *k* rows per input point, and each row has three columns: *InputID*, *TargetID* and *Distance*. • 1 --- Standard (N x T) distance matrix • 2 --- Summary distance matrix (mean, std. dev., min, max): for each input point, reports statistics on the distances to its target points.
Use only the nearest (k) target points	NEAREST_POINTS	[numeric: integer] Default: 0	You can choose to calculate the distance to all the points in the target layer (*0*) or limit to a number (*k*) of closest features.
Distance matrix	OUTPUT	[vector: point] Default: [Create temporary layer]	Specification of the output vector layer. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Distance matrix	OUTPUT	[vector: point]	Point (or MultiPoint for the "Linear (N * *k* x 3)" case) vector layer containing the distance calculation for each input feature. Its features and attribute table depend on the selected output matrix type.

Python 代码

Algorithm ID: qgis:distancematrix

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Distance to nearest hub (line to hub)

Creates lines that join each feature of an input vector to the nearest feature in a destination layer. Distances are calculated based on the *center* of each feature.

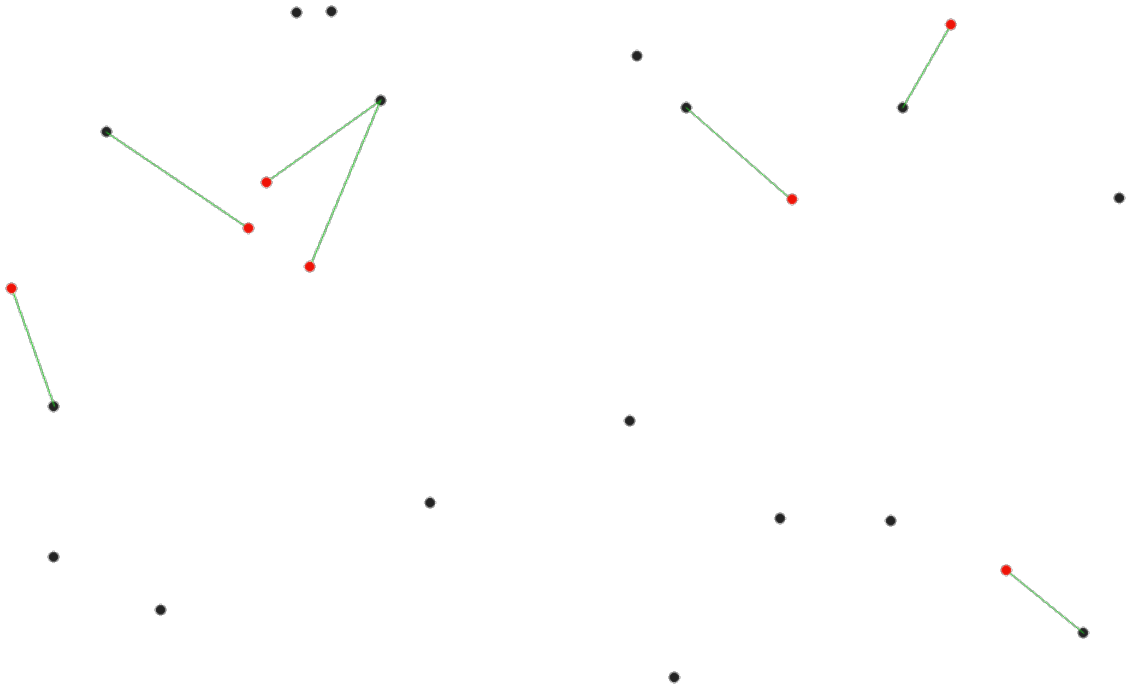


Fig. 24.71: Display the nearest hub for the red input features

See also

Distance to nearest hub (points), *Join attributes by nearest*

参数

标注	名称	类型	描述
Source points layer	`INPUT`	[vector: geometry]	Vector layer for which the nearest feature is searched
Destination hubs layer	`HUBS`	[vector: geometry]	Vector layer containing the features to search for
Hub layer name attribute	`FIELD`	[tablefield: any]	Field to use to uniquely identify features of the destination layer. Used in the output attribute table
Measurement unit	`UNIT`	[enumeration] Default: 0	Units in which to report the distance to the closest feature: • 0 --- Meters • 1 --- Feet • 2 --- Miles • 3 --- Kilometers • 4 --- Layer units

continues on next page

Table 24.459 – continued from previous page

标注	名称	类型	描述
Hub distance	`OUTPUT`	[vector: line] Default: `[Create temporary layer]`	Specify the output line vector layer connecting the matching points. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Hub distance	`OUTPUT`	[vector: line]	Line vector layer with the attributes of the input features, the identifier of their closest feature and the calculated distance.

Python 代码

Algorithm ID: `qgis:distancetonearesthublinetohub`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Distance to nearest hub (points)

Creates a point layer representing the *center* of the input features with the addition of two fields containing the identifier of the nearest feature (based on its center point) and the distance between the points.

See also

Distance to nearest hub (line to hub), *Join attributes by nearest*

参数

标注	名称	类型	描述
Source points layer	`INPUT`	[vector: any]	Vector layer for which the nearest feature is searched
Destination hubs layer	`HUBS`	[vector: any]	Vector layer containing the features to search for
Hub layer name attribute	`FIELD`	[tablefield: any]	Field to use to uniquely identify features of the destination layer. Used in the output attribute table

continues on next page

Table 24.461 – continued from previous page

标注	名称	类型	描述
Measurement unit	`UNIT`	[enumeration] Default: 0	Units in which to report the distance to the closest feature: • 0 --- Meters • 1 --- Feet • 2 --- Miles • 3 --- Kilometers • 4 --- Layer units
Hub distance	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the output point vector layer with the nearest hub. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Hub distance	`OUTPUT`	[vector: point]	Point vector layer representing the center of the source features with their attributes, the identifier of their closest feature and the calculated distance.

Python 代码

Algorithm ID: `qgis:distancetonearesthubpoints`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Join by lines (hub lines)

Creates hub and spoke diagrams by connecting lines from points on the Spoke layer to matching points in the Hub layer.

Determination of which hub goes with each point is based on a match between the Hub ID field on the hub points and the Spoke ID field on the spoke points.

If input layers are not point layers, a point on the surface of the geometries will be taken as the connecting location.

Optionally, geodesic lines can be created, which represent the shortest path on the surface of an ellipsoid. When geodesic mode is used, it is possible to split the created lines at the antimeridian (±180 degrees longitude), which can improve rendering of the lines. Additionally, the distance between vertices can be specified. A smaller distance results in a denser, more accurate line.

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

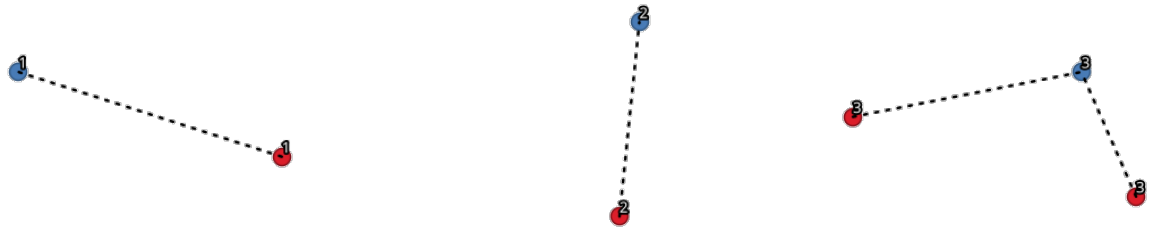


Fig. 24.72: Join points based on a common field / attribute

参数

Basic parameters

标注	名称	类型	描述
Hub layer	`HUBS`	[vector: geometry]	Input layer
Hub ID field	`HUB_FIELD`	[tablefield: any]	Field of the hub layer with ID to join
Hub layer fields to copy (leave empty to copy all fields) 可选	`HUB_FIELDS`	[tablefield: any] [list]	The field(s) of the hub layer to be copied. If no field(s) are chosen all fields are taken.
Spoke layer	`SPOKES`	[vector: geometry]	Additional spoke point layer
Spoke ID field	`SPOKE_FIELD`	[tablefield: any]	Field of the spoke layer with ID to join
Spoke layer fields to copy (leave empty to copy all fields) 可选	`SPOKE_FIELDS`	[tablefield: any] [list]	Field(s) of the spoke layer to be copied. If no fields are chosen all fields are taken.
Create geodesic lines	`GEODESIC`	[boolean] Default: False	Create geodesic lines (the shortest path on the surface of an ellipsoid)
Hub lines	`OUTPUT`	[vector: line] Default: `[Create temporary layer]`	Specify the output hub line vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Distance between vertices (geodesic lines only)	GEODESIC_DISTA	[numeric: double] Default: 1000.0 (kilometers)	Distance between consecutive vertices (in kilometers). A smaller distance results in a denser, more accurate line
Split lines at antimeridian (±180 degrees longitude)	ANTIMERIDIAN_S	[boolean] Default: False	Split lines at ±180 degrees longitude (to improve rendering of the lines)

输出

标注	名称	类型	描述
Hub lines	OUTPUT	[vector: line]	The resulting line layer connecting matching points in input layers

Python 代码

Algorithm ID: `native:hublines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

K-means clustering

Calculates the 2D distance based k-means cluster number for each input feature.

K-means clustering aims to partition the features into k clusters in which each feature belongs to the cluster with the nearest mean. The mean point is represented by the barycenter of the clustered features.

If input geometries are lines or polygons, the clustering is based on the centroid of the feature.

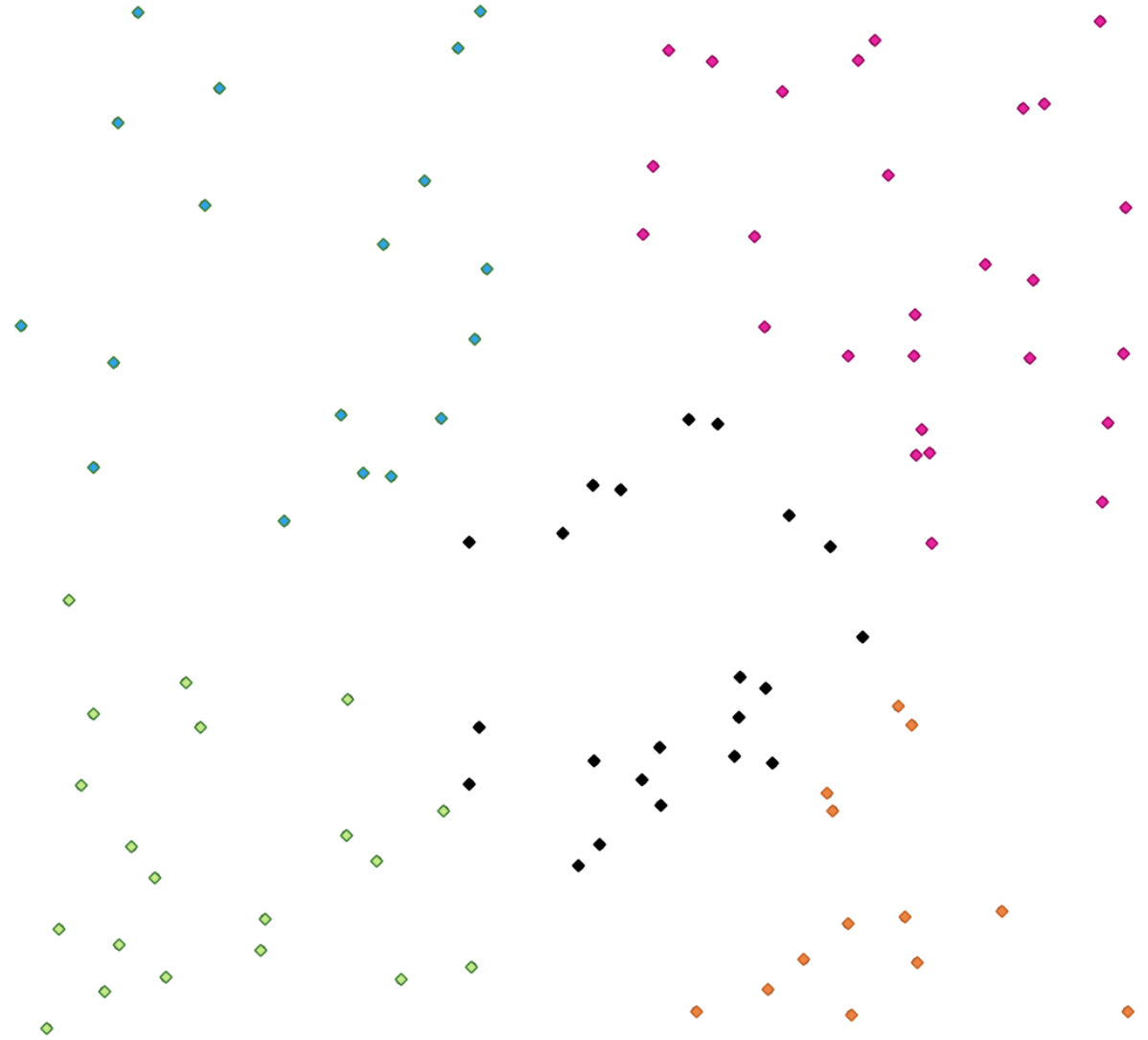


Fig. 24.73: A five class point clusters

See also

DBSCAN clustering, *ST-DBSCAN clustering*

continues on next page

Table 24.466 – continued from previous page

标注	名称	类型	描述

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	Layer to analyze
Number of clusters	`CLUSTERS`	[numeric: integer] Default: 5	Number of clusters to create with the features
Method	`METHOD`	[enumeration] Default: 0	Initial centers selection method. Possible values are: • 0 --- Farthest points: the initial centers are selected as the farthest points from each other. • 1 --- K-means++: The first center is selected uniformly at random from the set of input features. Subsequent centers are chosen based on a weighted probability distribution that favors features farther from existing centers. This ensures that the initial centers are well-dispersed, reducing the likelihood of poor clustering and improving the algorithm's performance. *References: Arthur, David & Vassilvitskii, Sergei. (2007). K-Means++: The Advantages of Careful Seeding. Proc. of the Annu. ACM-SIAM Symp. on Discrete Algorithms. 8.*
Clusters	`OUTPUT`	[vector: same as input] Default: `[Create temporary layer]`	Specify the output vector layer for the generated clusters. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Cluster field name	`FIELD_NAME`	[string] Default: 'CLUSTER_ID'	Name of the field where the associated cluster number shall be stored
Cluster size field name	`SIZE_FIELD_NAM`	[string] Default: 'CLUSTER_SIZE'	Name of the field with the count of features in the same cluster

输出

标注	名称	类型	描述
Clusters	`OUTPUT`	[vector: same as input]	Vector layer containing the original features with fields specifying the cluster they belong to and their number in it

Python 代码

Algorithm ID: `native:kmeansclustering`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

List unique values

Lists unique values of an attribute table field and counts their number.

Default menu: *Vector* ► *Analysis Tools*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	Layer to analyze
Target field(s)	`FIELDS`	[tablefield: any] [list]	Field(s) to analyze
Unique values 可选	`OUTPUT`	[vector: table] Default: `[Create temporary layer]`	Specify the summary table layer with unique values. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。
HTML report 可选	`OUTPUT_HTML_FI`	[html] Default: `[Save to temporary file]`	HTML report of unique values in the *Processing* ► *Results viewer*. *One of*: • Skip Output • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Unique values	`OUTPUT`	[vector: table]	Summary table layer with unique values
HTML report	`OUTPUT_HTML_FI`	[html]	HTML report of unique values. Can be opened from the *Processing* ► *Results viewer*
Total unique values	`TOTAL_VALUES`	[numeric: integer]	The number of unique values in the input field
Unique values concatenated	`UNIQUE_VALUES`	[string]	A string with the comma separated list of unique values found in the input field

Python 代码

Algorithm ID: `native:listuniquevalues`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Mean coordinate(s)

Computes a point layer with the center of mass of geometries in an input layer.

An attribute can be specified as containing weights to be applied to each feature when computing the center of mass.

If an attribute is selected in the parameter, features will be grouped according to values in this field. Instead of a single point with the center of mass of the whole layer, the output layer will contain a center of mass for the features in each category.

Default menu: *Vector* ► *Analysis Tools*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Weight field 可选	`WEIGHT`	[tablefield: numeric]	Field to use if you want to perform a weighted mean
Unique ID field	`UID`	[tablefield: any]	Unique field on which the calculation of the mean will be made
Mean coordinates	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the (point vector) layer for the result. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Mean coordinates	`OUTPUT`	[vector: point]	Resulting point(s) layer

Python 代码

Algorithm ID: `native:meancoordinates`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Nearest neighbour analysis

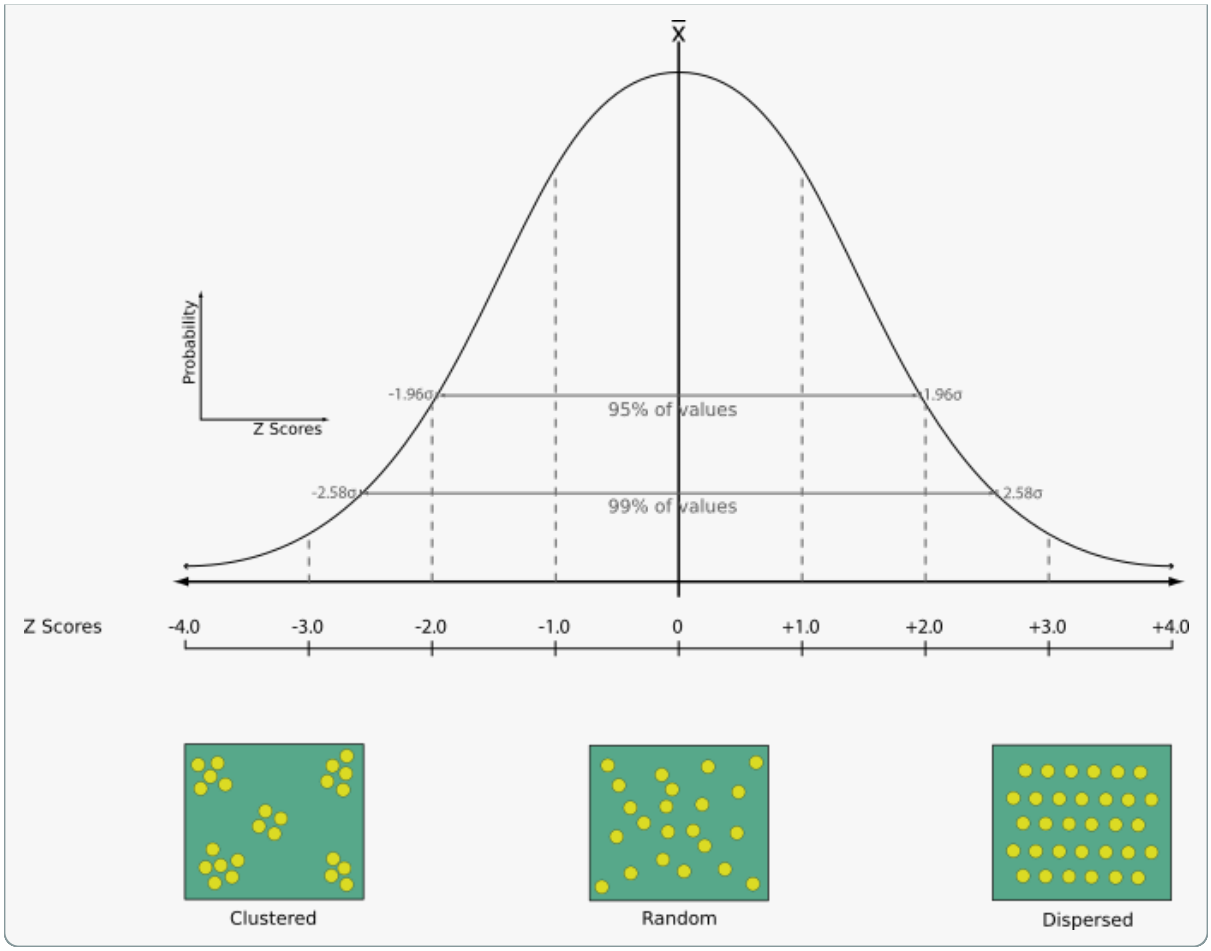
Performs nearest neighbor analysis for a point layer. The output tells you how your data are distributed (clustered, randomly or distributed).

Output is generated as an HTML file with the computed statistical values:

- Observed mean distance
- Expected mean distance
- Nearest neighbour index
- Number of points
- Z-Score: Comparing the Z-Score with the normal distribution tells you how your data are distributed. A low Z-Score means that the data are unlikely to be the result of a spatially random process, while a high Z-Score means that your data are likely to be a result of a spatially random process.

Note

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.



Default menu: *Vector* ► *Analysis Tools*

See also

Join attributes by nearest

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: point]	Point vector layer to calculate the statistics on
Nearest neighbour 可选	`OUTPUT_HTML_FI`	[html] Default: `[Save to temporary file]`	Specification of the HTML file for the computed statistics. *One of*: • Skip Output • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Nearest neighbour	OUTPUT_HTML_FI	[html]	HTML file with the computed statistics
Observed mean distance	OBSERVED_MD	[numeric: double]	Observed mean distance
Expected mean distance	EXPECTED_MD	[numeric: double]	Expected mean distance
Nearest neighbour index	NN_INDEX	[numeric: integer]	Nearest neighbour index
Number of points	POINT_COUNT	[numeric: integer]	Number of points
Z-Score	Z_SCORE	[numeric: double]	Z-Score

Python 代码

Algorithm ID: `native:nearestneighbouranalysis`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Overlap analysis

Calculates the area and percentage cover by which features from an input layer are overlapped by features from a selection of overlay layers.

New attributes are added to the output layer reporting the total area of overlap and percentage of the input feature overlapped by each of the selected overlay layers.

> **Note**
>
> This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[vector: polygon]	The input layer.
Overlay layers	LAYERS	[vector: polygon] [list]	The overlay layers.
Overlap	OUTPUT	[same as input] Default: [Create temporary layer]	Specify the output vector layer. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Grid size 可选	`GRID_SIZE`	[numeric: double] Default: Not set	If provided, the input geometries are snapped to a grid of the given size, and the result vertices are computed on that same grid.

输出

标注	名称	类型	描述
Overlap	`OUTPUT`	[same as input]	The output layer with additional fields reporting the overlap (in map units and percentage) of the input feature overlapped by each of the selected layers.

Python 代码

Algorithm ID: `native:calculatevectoroverlaps`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Shortest line between features

Creates a line layer as the shortest line between the source and the destination layer. By default only the first nearest feature of the destination layer is taken into account. The n-nearest neighboring features number can be specified. If a maximum distance is specified, then only features which are closer than this distance will be considered.

The output features will contain all the source layer attributes, all the attributes from the n-nearest feature and the additional field of the distance.

Important

This algorithm uses purely Cartesian calculations for distance, and does not consider geodetic or ellipsoid properties when determining feature proximity. The measurement and output coordinate system is based on the coordinate system of the source layer.

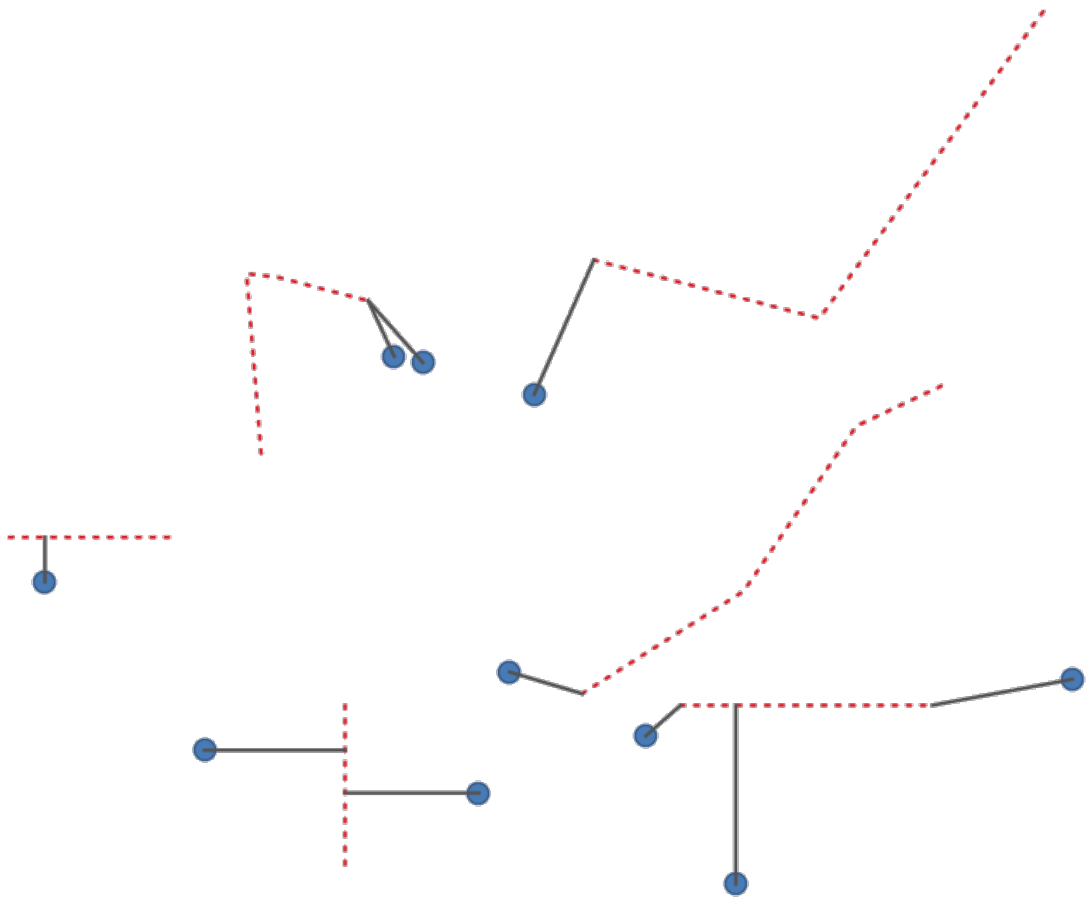


Fig. 24.74: Shortest line from point features to lines

参数

标注	名称	类型	描述
Source layer	SOURCE	[vector: geometry]	Origin layer for which to search for nearest neighbors
Destination layer	DESTINATION	[vector: geometry]	Target Layer in which to search for nearest neighbors
Method	METHOD	[enumeration] Default: 0	Shortest distance calculation method Possible values are: • 0 --- Distance to nearest point on feature • 1 --- Distance to feature centroid
Maximum number of neighbors	NEIGHBORS	[numeric: integer] Default: 1	Maximum number of neighbors to look for
Maximum distance 可选	DISTANCE	[numeric: double]	Only destination features which are closer than this distance will be considered.

continues on next page

Table 24.478 – continued from previous page

标注	名称	类型	描述
Shortest lines	`OUTPUT`	[vector: line] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Output layer	`OUTPUT`	[vector: line]	Line vector layer joining source features to their nearest neighbor(s) in the destination layer. Contains all attributes for both source and destination features, and the computed distance.

Python 代码

Algorithm ID: `native:shortestline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

ST-DBSCAN clustering

Clusters point features based on a 2D implementation of spatiotemporal Density-based clustering of applications with noise (ST-DBSCAN) algorithm.

See also

DBSCAN clustering, *K-means clustering*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[vector: point]	Layer to analyze
Date/time field	`DATETIME_FIELD`	[tablefield: date]	Field containing the temporal information
Minimum cluster size	`MIN_SIZE`	[numeric: integer] Default: 5	Minimum number of features to generate a cluster

continues on next page

Table 24.480 – continued from previous page

标注	名称	类型	描述
Maximum distance between clustered points	EPS	[numeric: double] 默认值：1.0	Distance beyond which two features can not belong to the same cluster (eps)
Maximum time duration between clustered points	EPS2	[numeric: double] Default: 0.0 (days)	Time duration beyond which two features can not belong to the same cluster (eps2). Available time units are milliseconds, seconds, minutes, hours, days and weeks.
Clusters	OUTPUT	[vector: point] Default: [Create temporary layer]	Specify the vector layer for the result of the clustering. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Treat border points as noise (DBSCAN*)	DBSCAN*	[boolean] Default: False	If checked, points on the border of a cluster are themselves treated as unclustered points, and only points in the interior of a cluster are tagged as clustered.
Cluster field name	FIELD_NAME	[string] Default: 'CLUSTER_ID'	Name of the field where the associated cluster number shall be stored
Cluster size field name	SIZE_FIELD_NAM	[string] Default: 'CLUSTER_SIZE'	Name of the field with the count of features in the same cluster

输出

标注	名称	类型	描述
Clusters	OUTPUT	[vector: point]	Vector layer containing the original features with a field setting the cluster they belong to
Number of clusters	NUM_CLUSTERS	[numeric: integer]	The number of clusters discovered

Python 代码

Algorithm ID: `native:stdbscanclustering`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Statistics by categories

Calculates statistics of a field depending on a parent class. The parent class is a combination of values from other fields.

参数

标注	名称	类型	描述
Input vector layer	`INPUT`	[vector: any]	Input vector layer with unique classes and values
Field to calculate statistics on (if empty, only count is calculated) 可选	`VALUES_FIELD_N`	[tablefield: any]	If empty only the count will be calculated
Field(s) with categories	`CATEGORIES_FIE`	[tablefield: any] [list]	The fields that (combined) define the categories
Statistics by category	`OUTPUT`	[vector: table] Default: `[Create temporary layer]`	Specify the output table for the generated statistics. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Statistics by category	`OUTPUT`	[vector: table]	Table containing the statistics

Depending on the type of the field being analyzed, the following statistics are returned for each grouped value:

统计	字符串	Numeric	日期
Count (`COUNT`)	☑	☑	☑
Unique values (`UNIQUE`)	☑		☑
Empty (null) values (`EMPTY`)	☑		☑

continues on next page

Table 24.485 – continued from previous page

统计	字符串	Numeric	日期
Non-empty values (FILLED)	☑		☑
Minimal value (MIN)	☑	☑	☑
Maximal value (MAX)	☑	☑	☑
Range (RANGE)		☑	
Sum (SUM)		☑	
Mean value (MEAN)		☑	
Median value (MEDIAN)		☑	
Standard Deviation (STD_DEV)		☑	
Coefficient of variation (CV)		☑	
Minority (rarest occurring value - MINORITY)		☑	
Majority (most frequently occurring value - MAJORITY)		☑	
First Quartile (FIRSTQUARTILE)		☑	
Third Quartile (THIRDQUARTILE)		☑	
Inter Quartile Range (IQR)		☑	
Minimum Length (MIN_LENGTH)	☑		
Mean Length (MEAN_LENGTH)	☑		
Maximum Length (MAX_LENGTH)	☑		

Python 代码

Algorithm ID: `qgis:statisticsbycategories`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Sum line lengths

Takes a polygon layer and a line layer and measures the total length of lines and the total number of them that cross each polygon.

The resulting layer has the same features as the input polygon layer, but with two additional attributes containing the length and count of the lines across each polygon.

Note

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

☑ Allows *features in-place modification* of polygon features

Default menu: *Vector* ► *Analysis Tools*

参数

标注	名称	类型	描述
Lines	`LINES`	[vector: line]	Input vector line layer
Polygons	`POLYGONS`	[vector: polygon]	Polygon vector layer
Lines length field name	`LEN_FIELD`	[string] Default: 'LENGTH'	Name of the field for the lines length
Lines count field name	`COUNT_FIELD`	[string] Default: 'COUNT'	Name of the field for the lines count
Line length	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output polygon layer with generated statistics. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Line length	`OUTPUT`	[vector: polygon]	Polygon output layer with fields of lines length and line count

Python 代码

Algorithm ID: `native:sumlinelengths`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.23 Vector coverage

Dissolve coverage

Operates on a coverage (represented as a set of polygon features with exactly matching edge geometry) to dissolve (union) the geometries. It provides a heavily optimized approach for unioning these features compared with the standard Dissolve tools.

See also

融合（*dissolve*）

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: polygon]	Input polygon vector layer
Dissolved	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Dissolved	`OUTPUT`	[vector: polygon]	The output polygon vector layer with dissolved geometries.

Python 代码

Algorithm ID: `native:coverageunion`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Simplify coverage

Operates on a coverage (represented as a set of polygon features with exactly matching edge geometry) to apply a Visvalingam–Whyatt simplification to the edges, reducing complexity in proportion with the provided tolerance, while retaining a valid coverage (i.e. no edges will cross or touch after the simplification). Geometries will never be removed, but they may be simplified down to just a triangle. Also, some geometries (such as polygons which have too few non-repeated points) will be returned unchanged.

If the input dataset is not a valid coverage due to overlaps, it will still be simplified, but invalid topology such as crossing edges will still be invalid.

Attention

Running this algorithm requires QGIS installed with GEOS >= 3.12 (see *Help* ► *About* menu).

See also

Simplify, *Smooth*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: polygon]	Input polygon vector layer
Tolerance	`TOLERANCE`	[numeric: double] Default: 0.0	The maximum distance (in unit of choice) between two consecutive vertices to get merged.
Preserve boundary	`PRESERVE_BOUND`	[boolean] Default: False	When enabled, the outside edges of the coverage will be preserved without simplification
Simplified	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Simplified	`OUTPUT`	[vector: polygon]	The output polygon vector layer, with a lower number of vertices.

Python 代码

Algorithm ID: `native:coveragesimplify`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Validate coverage

Analyzes a coverage (represented as a set of polygon features with exactly matching edge geometry) to find places where the assumption of exactly matching edges is not met. Invalidity includes polygons that overlap or that have gaps smaller than the specified gap width.

Attention

Running this algorithm requires QGIS installed with GEOS >= 3.12 (see *Help* ► *About* menu).

See also

Check validity, *Small gaps*, *Overlaps*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: polygon]	Input polygon vector layer
Gap width	`GAP_WIDTH`	[numeric: double] Default: 0.0	The maximum width of gap to detect
Invalid edges	`INVALID_EDGES`	[vector: line] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Invalid edges	`OUTPUT`	[vector: line]	The output line vector layer showing error edges for each invalid polygon.
Validity result	`IS_VALID`	[boolean]	Returns whether the coverage is valid or not.

Python 代码

Algorithm ID: `native:coveragevalidate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.24 Vector creation

Array of offset (parallel) lines

Creates copies of line features in a layer, by creating multiple offset versions of each feature. Each new version is incrementally offset by a specified distance.

Positive distance will offset lines to the left, and negative distances will offset them to the right.

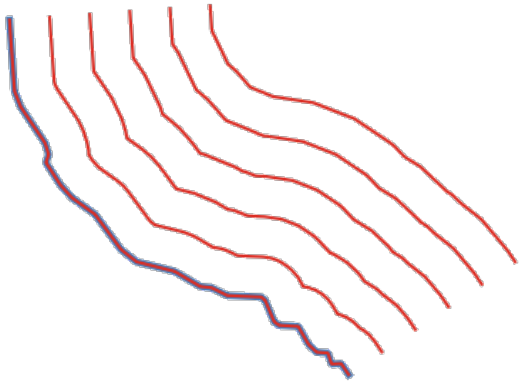


Fig. 24.75: In blue the source layer, in red the offset one

Allows *features in-place modification* of line features

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

See also

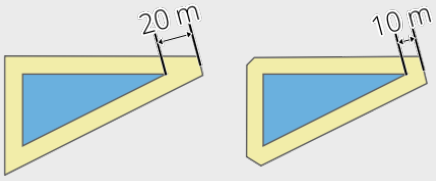
Offset lines, *Array of translated features*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line]	Input line vector layer to use for the offsets.
Number of features to create	`COUNT`	[numeric: integer] Default: 10	Number of offset copies to generate for each feature
Offset step distance	`OFFSET`	[numeric: double] 默认值：1.0	Distance between two consecutive offset copies
Offset lines	`OUTPUT`	[vector: line] Default: `[Create temporary layer]`	Specify the output line layer with offset features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Segments	`SEGMENTS`	[numeric: double] Default: 8	Number of line segments to use to approximate a quarter circle when creating rounded offsets
Join style	`JOIN_STYLE`	[enumeration] Default: 0	Specify whether round, miter or beveled joins should be used when offsetting corners in a line. One of: • 0 --- Round • 1 --- Miter • 2 --- Bevel Fig. 24.76: Round, miter, and bevel join styles
Miter limit	`MITER_LIMIT`	[numeric: double] Default: 2.0	Sets the maximum distance from the offset geometry to use when creating a mitered join as a factor of the offset distance (only applicable for miter join styles). Minimum: 1.0  Fig. 24.77: A 10m buffer with a limit of 2 and a 10m buffer with a limit of 1

输出

标注	名称	类型	描述
Offset lines	`OUTPUT`	[vector: line]	Output line layer with the original and offset features. Attributes will be copied to the corresponding outputs. The following attributes related to the input parameters will be added: `instance` indexing the `COUNT` offset lines and `offset` designating the offset distance from the original line.

Python 代码

Algorithm ID: `native:arrayoffsetlines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Array of translated features

Creates copies of features in a layer by creating multiple translated versions of each. Each copy is incrementally displaced by a preset amount in the X, Y and/or Z axis.

M values present in the geometry can also be translated.

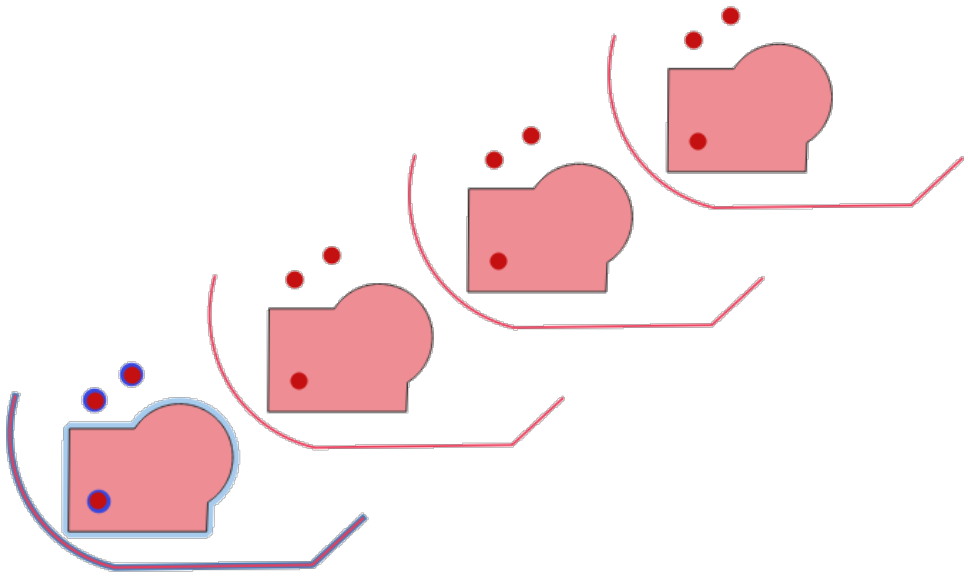


Fig. 24.78: Input layers in blue tones, output layers with translated features in red tones

Allows *features in-place modification* of point, line, and polygon features

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

See also

翻译, *Array of offset (parallel) lines*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	Input vector layer to translate
Number of features to create	`COUNT`	[numeric: integer] Default: 10	Number of copies to generate for each feature
Step distance (x-axis)	`DELTA_X`	[numeric: double] Default: 0.0	Displacement to apply on the X axis
Step distance (y-axis)	`DELTA_Y`	[numeric: double] Default: 0.0	Displacement to apply on the Y axis
Step distance (z-axis)	`DELTA_Z`	[numeric: double] Default: 0.0	Displacement to apply on the Z axis
Step distance (m values)	`DELTA_M`	[numeric: double] Default: 0.0	Displacement to apply on M
Translated	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Output vector layer with translated (moved) copies of the features. The original features are also copied. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Translated	`OUTPUT`	[same as input]	Output vector layer with translated (moved) copies of the features. The original features are also copied.

Python 代码

Algorithm ID: `native:arraytranslatedfeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Create grid

Creates a vector layer with a grid covering a given extent. Grid cells can have different shapes:

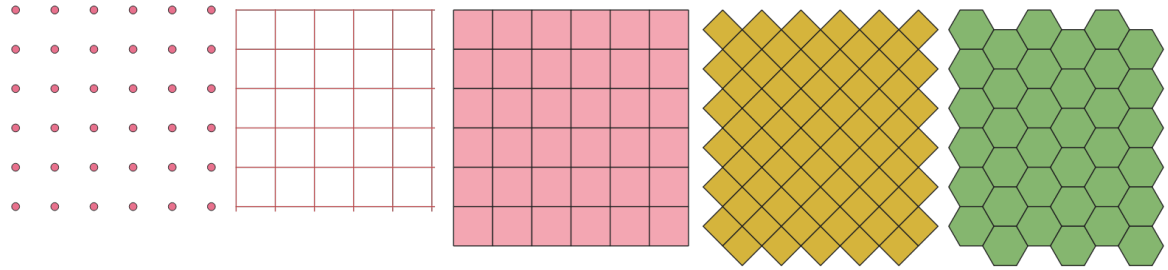


Fig. 24.79: Different grid cell shapes applied to the same extent, without overlaps

The size and/or placement of each element in the grid is defined using a horizontal and vertical spacing. The CRS of the output layer must be defined. The grid extent and the spacing values must be expressed in the coordinates and units of this CRS.

Default menu: *Vector ► Research Tools*

参数

标注	名称	类型	描述
Grid type	`TYPE`	[enumeration] Default: 0	Shape of the grid. One of: • 0 --- Point • 1 --- Line • 2 --- Rectangle (polygon) • 3 --- Diamond (polygon) • 4 --- Hexagon (polygon)
Grid extent	`EXTENT`	[范围]	Extent of the grid Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Horizontal spacing	`HSPACING`	[numeric: double] 默认值：1.0	Size of a grid cell on the X-axis
Vertical spacing	`VSPACING`	[numeric: double] 默认值：1.0	Size of a grid cell on the Y-axis
Horizontal overlay	`HOVERLAY`	[numeric: double] Default: 0.0	Overlay distance between two consecutive grid cells on the X-axis

continues on next page

Table 24.499 – continued from previous page

标注	名称	类型	描述
Vertical overlay	`VOVERLAY`	[numeric: double] Default: 0.0	Overlay distance between two consecutive grid cells on the Y-axis
Grid CRS	`CRS`	[crs] Default: *Project CRS*	Coordinate reference system to apply to the grid
Grid	`OUTPUT`	[vector: geometry] Default: `[Create temporary layer]`	Resulting vector grid layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Grid	`OUTPUT`	[vector: geometry]	Resulting vector grid layer. The output geometry type (point, line or polygon) depends on the *Grid type*. Features are created from top to bottom, left to right. The attribute table is filled with: • an `id` • coordinates on the `left`, `right`, `top` and `bottom` sides • and their placement in the grid: `row_index` and `column_index` (available for point, rectangle and hexagon grid types)

Python 代码

Algorithm ID: `native:creategrid`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Create layer from extent

Creates a new vector layer that contains a single feature with geometry matching the extent of the input layer.

It can be used in models to convert a literal extent (`xmin`, `xmax`, `ymin`, `ymax` format) into a layer which can be used for other algorithms which require a layer based input.

See also

Create layer from point

参数

标注	名称	类型	描述
Extent (xmin, xmax, ymin, ymax)	`INPUT`	[范围]	Input extent Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Extent	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Extent	`OUTPUT`	[vector: polygon]	The output (extent) vector layer

Python 代码

Algorithm ID: `native:extenttolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Create layer from point

Creates a new vector layer that contains a single feature with geometry matching a point parameter. It can be used in models to convert a point into a point layer for algorithms which require a layer based input.

See also

Create layer from extent

参数

标注	名称	类型	描述
Point	`INPUT`	[coordinates]	Input point, including CRS info (example: `397254,6214446 [EPSG:32632]`). If the CRS is not provided, the Project CRS will be used. The point can be specified by clicking on the map canvas.
Point	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the output layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Point	`OUTPUT`	[vector: point]	The output point vector layer containing the input point.

Python 代码

Algorithm ID: `native:pointtolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Create points layer from table

Creates points layer from a table with columns that contain coordinates fields.

Besides X and Y coordinates you can also specify Z and M fields.

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	Input vector layer or a table.
X field	`XFIELD`	[tablefield: any]	Field containing the X coordinate
Y field	`YFIELD`	[tablefield: any]	Field containing the Y coordinate
Z field 可选	`ZFIELD`	[tablefield: any]	Field containing the Z coordinate
M field 可选	`MFIELD`	[tablefield: any]	Field containing the M value
Target CRS	`TARGET_CRS`	[crs] Default: `EPSG:4326`	Coordinate reference system to use for layer. The provided coordinates are assumed to be compliant.
Points from table	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the resulting point layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Points from table	`OUTPUT`	[vector: point]	The resulting point layer

Python 代码

Algorithm ID: `native:createpointslayerfromtable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Generate points (pixel centroids) along line

Generates a point vector layer from an input raster and line layer.

The points correspond to the pixel centroids that intersect the line layer.

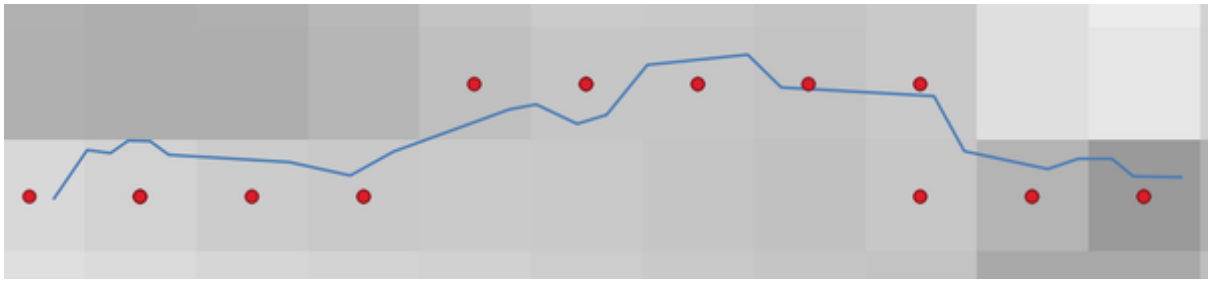


Fig. 24.80: Points of the pixel centroids

参数

标注	名称	类型	描述
Raster layer	`INPUT_RASTER`	[栅格]	Input raster layer
Vector layer	`INPUT_VECTOR`	[vector: line]	Input line vector layer
Points along line	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Resulting point layer with pixel centroids. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Points along line	`OUTPUT`	[vector: point]	Resulting point layer with pixel centroids

Python 代码

Algorithm ID: `qgis:generatepointspixelcentroidsalongline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Generate points (pixel centroids) inside polygon

Generates a point vector layer from an input raster and polygon layer.

The points correspond to the pixel centroids that intersect the polygon layer.

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

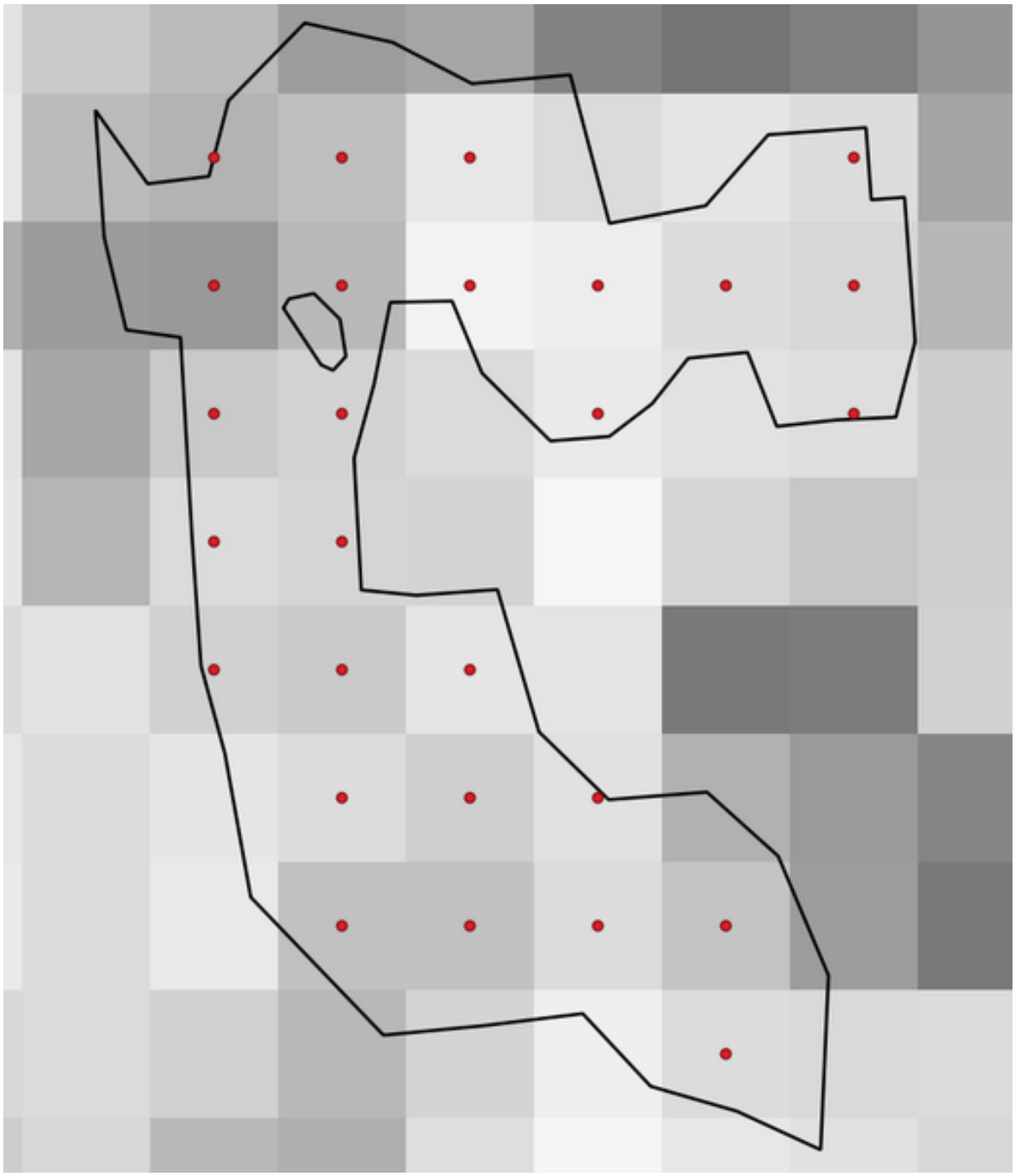


Fig. 24.81: Points of the pixel centroids

参数

标注	名称	类型	描述
Raster layer	`INPUT_RASTER`	[栅格]	Input raster layer
Vector layer	`INPUT_VECTOR`	[vector: polygon]	Input polygon vector layer
Points inside polygons	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Resulting point layer of pixel centroids. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Points inside polygons	`OUTPUT`	[vector: point]	Resulting point layer of pixel centroids

Python 代码

Algorithm ID: `native:generatepointspixelcentroidsinsidepolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Import geotagged photos

Creates a point layer corresponding to the geotagged locations from JPEG or HEIC/HEIF images from a source folder.

The point layer will contain a single PointZ feature per input file from which the geotags could be read. Any altitude information from the geotags will be used to set the point's Z value.

Besides longitude and latitude also altitude, direction and timestamp information, if present in the photo, will be added to the point as attributes.

参数

标注	名称	类型	描述
Input folder	`FOLDER`	[folder]	Path to the source folder containing the geotagged photos
Scan recursively	`RECURSIVE`	[boolean] Default: False	If checked, the folder and its subfolders will be scanned
Photos 可选	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the point vector layer for the geotagged photos. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。
Invalid photos table 可选	`INVALID`	[vector: table] Default: `[Skip output]`	Specify the table of unreadable or non-geotagged photos. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Photos	`OUTPUT`	[vector: point]	Point vector layer with geotagged photos. The form of the layer is automatically filled with paths and photo previews settings.
Invalid photos table 可选	`INVALID`	[vector: table]	Table of unreadable or non-geotagged photos can also be created.

Python 代码

Algorithm ID: `native:importphotos`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Points to path

Converts a point layer to a line layer, by joining points in an order defined by an expression or a field in the input point layer.

Points can be grouped by a field or an expression to distinguish line features.

In addition to the line vector layer, a text file is output that describes the resulting line as a start point and a sequence of bearings / directions (relative to azimuth) and distances.

Note

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

参数

标注	名称	类型	描述
Input point layer	`INPUT`	[vector: point]	Input point vector layer
Create closed paths	`CLOSE_PATH`	[boolean] Default: False	If checked, the first and last points of the line will be connected and close the generated path
Order expression 可选	`ORDER_EXPRESSI`	[expression]	Field or expression providing the order to connect the points in the path. If not set, the feature ID (`$id`) is used.
Sort text containing numbers naturally	`NATURAL_SORT`	[boolean] Default: False	If checked, naturally sorts the features based on the provided expression (i.e., 'a9' < 'a10').
Path group expression 可选	`GROUP_EXPRESSI`	[expression]	Point features of the same value in the field or expression will be grouped in the same line. If not set, a single path is drawn with all the input points.
Paths	`OUTPUT`	[vector: line] Default: `[Create temporary layer]`	Specify the line vector layer of the path. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。
Directory for text output 可选	`OUTPUT_TEXT_DI`	[folder] Default: `[Skip output]`	Specify the directory that will contain the description files of points and paths. *One of*: • Skip Output • Save to a Temporary Directory • Save to Directory

输出

标注	名称	类型	描述
Paths	`OUTPUT`	[vector: line]	Line vector layer of the path
Directory for text output	`OUTPUT_TEXT_DI`	[folder]	Directory containing description files of points and paths

Python 代码

Algorithm ID: `native:pointstopath`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Random points along line

Creates a new point layer, with points placed on the lines of another layer.

For each line in the input layer, a given number of points is added to the resulting layer. The procedure for adding a point is to:

1. randomly select a line feature from the input layer
2. if the feature is multi-part, randomly select a part of it
3. randomly select a segment of that line
4. randomly select a position on that segment.

The procedure means that curved parts of the lines (with relatively short segments) will get more points than straight parts (with relatively long segments), as demonstrated in the illustration below, where the output of the *Random points along lines* algorithm can be compared with the output of the *Random points on lines* algorithm (that produces points with an, on average, even distribution along the lines).

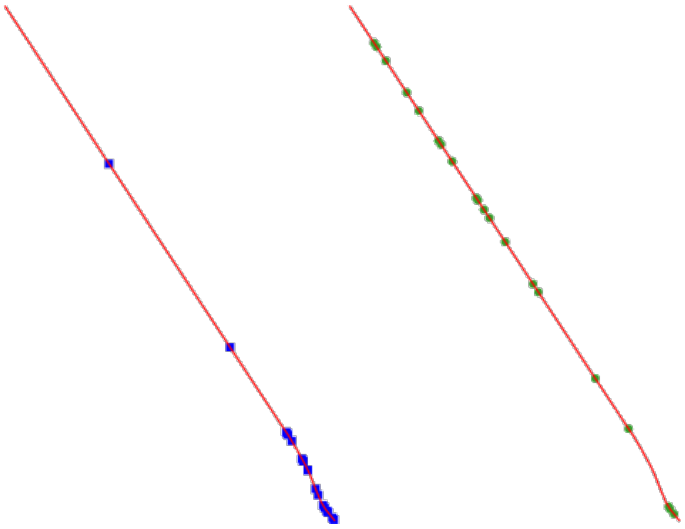


Fig. 24.82: Example algorithm output. Left: *Random points along line*, right: *Random points on lines*

A minimum distance can be specified, to avoid points being too close to each other.

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

See also

Random points on lines

参数

标注	名称	类型	描述
Input point layer	`INPUT`	[vector: line]	Input line vector layer
Number of points	`POINTS_NUMBER`	[numeric: integer] Default: 1	Number of points to create
Minimum distance between points	`MIN_DISTANCE`	[numeric: double] Default: 0.0	The minimum distance between points
Random points	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	The output random points. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Random points	`OUTPUT`	[vector: point]	The output random points layer.

Python 代码

Algorithm ID: `qgis:qgisrandompointsalongline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Random points in extent

Creates a new point layer with a given number of random points, all of them within a given extent.

A distance factor can be specified, to avoid points being too close to each other. If the minimum distance between points makes it impossible to create new points, either distance can be decreased or the maximum number of attempts may be increased.

Default menu: *Vector ► Research Tools*

参数

Basic parameters

标注	名称	类型	描述
Input extent	`EXTENT`	[范围]	Map extent for the random points Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Number of points	`POINTS_NUMBER`	[numeric: integer] Default: 1	Number of point to create
Minimum distance between points	`MIN_DISTANCE`	[numeric: double] Default: 0.0	The minimum distance between points
Target CRS	`TARGET_CRS`	[crs] Default: *Project CRS*	CRS of the random points layer
Random points	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	The output random points. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Maximum number of search attempts given the minimum distance	MAX_ATTEMPTS	[numeric: integer] Default: 200	Maximum number of attempts to place the points

输出

标注	名称	类型	描述
Random points	OUTPUT	[vector: point]	The output random points layer.

Python 代码

Algorithm ID: `native:randompointsinextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Random points in layer bounds

Creates a new point layer with a given number of random points, all of them within the extent of a given layer.

A minimum distance can be specified, to avoid points being too close to each other.

Default menu: *Vector* ► *Research Tools*

> **Warning**
>
> This algorithm drops existing primary keys or FID values and regenerates them in output layers.

参数

标注	名称	类型	描述
输入图层	INPUT	[vector: polygon]	Input polygon layer defining the area
Number of points	POINTS_NUMBER	[numeric: integer] Default: 1	Number of points to create
Minimum distance between points	MIN_DISTANCE	[numeric: double] Default: 0.0	The minimum distance between points

continues on next page

Table 24.520 – continued from previous page

标注	名称	类型	描述
Random points	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	The output random points. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Random points	`OUTPUT`	[vector: point]	The output random points layer.

Python 代码

Algorithm ID: `qgis:randompointsinlayerbounds`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Random points in polygons

Creates a point layer with points placed inside the polygons of another layer.

For each feature (polygon / multi-polygon) geometry in the input layer, the given number of points is added to the result layer.

Per feature and global minimum distances can be specified in order to avoid points being too close in the output point layer. If a minimum distance is specified, it may not be possible to generate the specified number of points for each feature. The total number of generated points and missed points are available as output from the algorithm.

The illustration below shows the effect of per feature and global minimum distances and zero/non-zero minimum distances (generated with the same seed, so at least the first point generated will be the same).

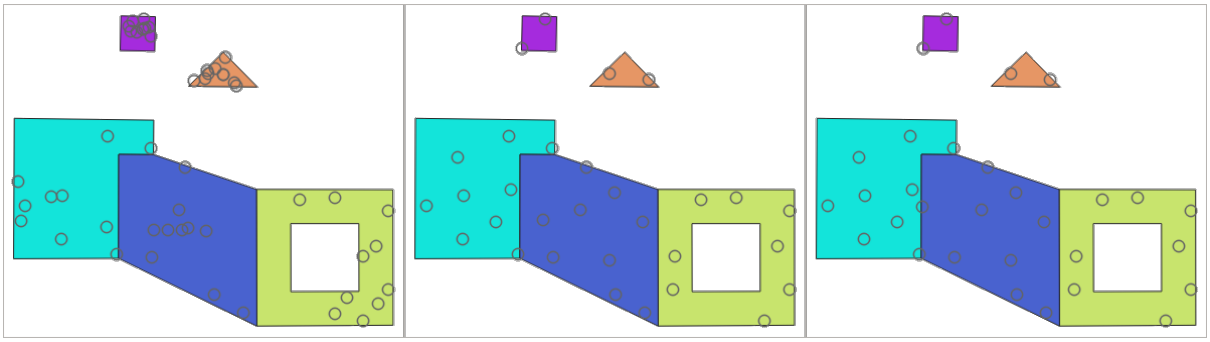


Fig. 24.83: Ten points per polygon feature, *left*: min. distances = 0, *middle*: min.distances = 1, *right*: min. distance = 1, global min. distance = 0

The maximum number of tries per point can be specified. This is only relevant for non-zero minimum distance.

A seed for the random number generator can be provided, making it possible to get identical random number sequences for different runs of the algorithm.

The attributes of the polygon feature on which a point was generated can be included (*Include polygon attributes*).

If you want approximately the same point density for all the features, you can data-define the number of points using the area of the polygon feature geometry.

See also

Random points inside polygons

参数

Basic parameters

标注	名称	类型	描述
Input polygon layer	`INPUT`	[vector: line]	Input polygon vector layer
Number of points for each feature	`POINTS_NUMBER`	[numeric: integer] Default: 1	Number of points to create
Minimum distance between points 可选	`MIN_DISTANCE`	[numeric: double] Default: 0.0	The minimum distance between points within one polygon feature
Random points in polygons	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	The output random points. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Global minimum distance between points 可选	`MIN_DISTANCE_G`	[numeric: double] Default: 0.0	The global minimum distance between points. Should be smaller than the *Minimum distance between points (per feature)* for that parameter to have an effect.
Maximum number of search attempts (for Min. dist. > 0) 可选	`MAX_TRIES_PER_`	[numeric: integer] Default: 10	The maximum number of tries per point. Only relevant if the minimum distance between points is set (and greater than 0).
Random seed 可选	`SEED`	[numeric: integer] Default: Not set	The seed to use for the random number generator.

continues on next page

Table 24.523 – continued from previous page

标注	名称	类型	描述
Include polygon attributes	INCLUDE_POLYGO	[boolean] Default: True	If set, a point will get the attributes from the line on which it is placed.

输出

标注	名称	类型	描述
Random points in polygons	OUTPUT	[vector: point]	The output random points layer.
Number of features with empty or no geometry	FEATURES_WITH_	[numeric: integer]	
Total number of points generated	OUTPUT_POINTS	[numeric: integer]	
Number of missed points	POINTS_MISSED	[numeric: integer]	The number of points that could not be generated due to the minimum distance constraint.
Number of features with missed points	POLYGONS_WITH_	[numeric: integer]	Not including features with empty or no geometry

Python 代码

Algorithm ID: `native:randompointsinpolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Random points inside polygons

Creates a new point layer with a given number of random points inside each polygon of the input polygon layer.

Two sampling strategies are available:

- Points count: number of points for each feature
- Points density: density of points for each feature

A minimum distance can be specified, to avoid points being too close to each other.

Default menu: *Vector* ► *Research Tools*

> **Warning**
>
> This algorithm drops existing primary keys or FID values and regenerates them in output layers.

See also

Random points in polygons

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: polygon]	Input polygon vector layer
Sampling strategy	`STRATEGY`	[enumeration] Default: 0	Sampling strategy to use. One of: • 0 --- Points count: number of points for each feature • 1 --- Points density: density of points for each feature
Point count or density	`VALUE`	[numeric: double] 默认值：1.0	The number or density of points, depending on the chosen *Sampling strategy*.
Minimum distance between points	`MIN_DISTANCE`	[numeric: double] Default: 0.0	The minimum distance between points
Random points	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	The output random points. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Random points	`OUTPUT`	[vector: point]	The output random points layer.

Python 代码

Algorithm ID: `qgis:randompointsinsidepolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Random points on lines

Creates a point layer with points placed on the lines of another layer.

For each feature (line / multi-line) geometry in the input layer, the given number of points is added to the result layer.

Per feature and global minimum distances can be specified in order to avoid points being too close in the output point layer. If a minimum distance is specified, it may not be possible to generate the specified number of points for each feature. The total number of generated points and missed points are available as output from the algorithm.

The illustration below shows the effect of per feature and global minimum distances and zero/non-zero minimum distances (generated with the same seed, so at least the first point generated will be the same).

Fig. 24.84: Five points per line feature, *left*: min. distances = 0, *middle*: min.distances != 0, *right*: min. distance != 0, global min. distance = 0

The maximum number of tries per point can be specified. This is only relevant for non-zero minimum distance.

A seed for the random number generator can be provided, making it possible to get identical random number sequences for different runs of the algorithm.

The attributes of the line feature on which a point was generated can be included (*Include line attributes*).

If you want approximately the same point density for all the line features, you can data-define the number of points using the length of the line feature geometry.

See also

Random points along line

参数

Basic parameters

标注	名称	类型	描述
Input line layer	`INPUT`	[vector: line]	Input line vector layer
Number of points for each feature	`POINTS_NUMBER`	[numeric: integer] Default: 1	Number of points to create
Minimum distance between points (per feature) 可选	`MIN_DISTANCE`	[numeric: double] Default: 0.0	The minimum distance between points within one line feature
Random points on lines	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	The output random points. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Global minimum distance between points 可选	`MIN_DISTANCE_G`	[numeric: double] Default: 0.0	The global minimum distance between points. Should be smaller than the *Minimum distance between points (per feature)* for that parameter to have an effect.
Maximum number of search attempts (for Min. dist. > 0) 可选	`MAX_TRIES_PER_`	[numeric: integer] Default: 10	The maximum number of tries per point. Only relevant if the minimum distance between points is set (and greater than 0).
Random seed 可选	`SEED`	[numeric: integer] Default: Not set	The seed to use for the random number generator.
Include line attributes	`INCLUDE_LINE_A`	[boolean] Default: True	If set, a point will get the attributes from the line on which it is placed.

输出

标注	名称	类型	描述
Random points on lines	`OUTPUT`	[vector: point]	The output random points layer.
Number of features with empty or no geometry	`FEATURES_WITH_`	[numeric: integer]	

continues on next page

Table 24.529 – continued from previous page

标注	名称	类型	描述
Number of features with missed points	LINES_WITH_MIS	[numeric: integer]	Not including features with empty or no geometry
Total number of points generated	POINTS_GENERAT	[numeric: integer]	
Number of missed points	POINTS_MISSED	[numeric: integer]	The number of points that could not be generated due to the minimum distance constraint.

Python 代码

Algorithm ID: `native:randompointsonlines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Raster pixels to points

Creates a vector layer of points corresponding to each pixel in a raster layer.

Converts a raster layer to a vector layer, by creating point features for each individual pixel's center in the raster layer. Any NoData pixels are skipped in the output.

参数

标注	名称	类型	描述
Raster layer	INPUT_RASTER	[栅格]	Input raster layer
Band number	RASTER_BAND	[raster band]	Raster band to extract data from
Field name	FIELD_NAME	[string] Default: 'VALUE'	Name of the field to store the raster band value
Vector points	OUTPUT	[vector: point] Default: `[Create temporary layer]`	Specify the resulting point layer of pixels centroids. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Vector points	`OUTPUT`	[vector: point]	Resulting point layer with pixels centroids

Python 代码

Algorithm ID: `native:pixelstopoints`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Raster pixels to polygons

Creates a vector layer of polygons corresponding to each pixel in a raster layer.

Converts a raster layer to a vector layer, by creating polygon features for each individual pixel's extent in the raster layer. Any NoData pixels are skipped in the output.

参数

标注	名称	类型	描述
Raster layer	`INPUT_RASTER`	[栅格]	Input raster layer
Band number	`RASTER_BAND`	[raster band]	Raster band to extract data from
Field name	`FIELD_NAME`	[string] Default: 'VALUE'	Name of the field to store the raster band value
Vector polygons	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the resulting polygon layer of pixel extents. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Vector polygons	`OUTPUT`	[vector: polygon]	Resulting polygon layer of pixel extents

Python 代码

Algorithm ID: `native:pixelstopolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Regular points

Creates a new point layer with its points placed in a regular grid within a given extent.

The grid is specified either by the spacing between the points (same spacing for all dimensions) or by the number of points to generate. In the latter case, the spacing will be determined from the extent. In order to generate a full rectangular grid, at least the number of points specified by the user is generated for the latter case.

Random offsets to the point spacing can be applied, resulting in a non-regular point pattern.

Default menu: *Vector ► Research Tools*

参数

标注	名称	类型	描述
Input extent (xmin, xmax, ymin, ymax)	`EXTENT`	[范围]	Map extent for the random points Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Point spacing/count	`SPACING`	[numeric: integer] Default: 100	Spacing between the points, or the number of points, depending on whether `Use point spacing` is checked or not.
Initial inset from corner (LH side)	`INSET`	[numeric: double] Default: 0.0	Offsets the points relative to the upper left corner. The value is used for both the X and Y axis.
Apply random offset to point spacing	`RANDOMIZE`	[boolean] Default: False	If checked the points will have a random spacing
Use point spacing	`IS_SPACING`	[boolean] Default: True	If unchecked the point spacing is not taken into account
Output layer CRS	`CRS`	[crs] Default: *Project CRS*	CRS of the random points layer

continues on next page

Table 24.534 – continued from previous page

标注	名称	类型	描述
Regular points	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the output regular point layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Regular points	`OUTPUT`	[vector: point]	The output regular point layer.

Python 代码

Algorithm ID: `qgis:regularpoints`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.25 Vector general

Assign projection

Assigns a new projection to a vector layer.

It creates a new layer with the exact same features and geometries as the input one, but assigned to a new CRS. The geometries are **not** reprojected, they are just assigned to a different CRS.

This algorithm can be used to repair layers which have been assigned an incorrect projection.

Attributes are not modified by this algorithm.

See also

Define projection, *Find projection*, *Reproject layer*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	Vector layer with wrong or missing CRS
Assigned CRS	`CRS`	[crs] Default: `EPSG:4326 - WGS84`	Select the new CRS to assign to the vector layer
Assigned CRS 可选	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Assigned CRS	`OUTPUT`	[same as input]	Vector layer with assigned projection

Python 代码

Algorithm ID: `native:assignprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Batch Nominatim geocoder

Performs batch geocoding using the Nominatim service against an input layer string field. The output layer will have a point geometry reflecting the geocoded location as well as a number of attributes associated to the geocoded location.

Allows *features in-place modification* of point features

Note

This algorithm is compliant with the usage policy of the Nominatim geocoding service provided by the OpenStreetMap Foundation.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	Vector layer to geocode the features
Address field	`FIELD`	[tablefield: string]	Field containing the addresses to geocode
Geocoded	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the output layer containing only the geocoded addresses. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Geocoded	`OUTPUT`	[vector: point]	Vector layer with point features corresponding to the geocoded addresses

Python 代码

Algorithm ID: `native:batchnominatimgeocoder`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Convert layer to spatial bookmarks

Creates spatial bookmarks corresponding to the extent of features contained in a layer.

参数

标注	名称	类型	描述
Input Layer	`INPUT`	[vector: line, polygon]	The input vector layer
Bookmark destination	`DESTINATION`	[enumeration] Default: 0	Select the destination for the bookmarks. One of: • 0 --- Project bookmarks • 1 --- User bookmarks
Name field	`NAME_EXPRESSIO`	[expression]	Field or expression that will give names to the generated bookmarks

continues on next page

Table 24.540 – continued from previous page

标注	名称	类型	描述
Group field	GROUP_EXPRESSI	[expression]	Field or expression that will provide groups for the generated bookmarks

输出

标注	名称	类型	描述
Count of bookmarks added	COUNT	[numeric: integer]	

Python 代码

Algorithm ID: native:layertobookmarks

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Convert spatial bookmarks to layer

Creates a new layer containing polygon features for stored spatial bookmarks. The export can be filtered to only bookmarks belonging to the current project, to all user bookmarks, or a combination of both.

参数

标注	名称	类型	描述
Bookmark source	SOURCE	[enumeration] [list] Default: [0,1]	Select the source(s) of the bookmarks. One or more of: • 0 --- Project bookmarks • 1 --- User bookmarks
Output CRS	CRS	[crs] Default: EPSG:4326 - WGS 84	The CRS of the output layer
Output	OUTPUT	[vector: polygon] Default: [Create temporary layer]	Specify the output layer. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Output	OUTPUT	[vector: polygon]	The output (bookmarks) vector layer

Python 代码

Algorithm ID: native:bookmarkstolayer

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Create attribute index

Creates an index against a field of the attribute table to speed up queries. The support for index creation depends on both the layer's data provider and the field type.

No outputs are created: the index is stored on the layer itself.

参数

标注	名称	类型	描述
Input Layer	INPUT	[vector: any]	Select the vector layer you want to create an attribute index for
Attribute to index	FIELD	[tablefield: any]	Field of the vector layer

输出

标注	名称	类型	描述
Indexed layer	OUTPUT	[same as input]	A copy of the input vector layer with an index for the specified field

Python 代码

Algorithm ID: native:createattributeindex

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Create spatial index

Creates an index to speed up access to the features in a layer based on their spatial location. Support for spatial index creation is dependent on the layer's data provider.

No new output layers are created.

Default menu: *Vector ► Data Management Tools*

参数

标注	名称	类型	描述
Input Layer	`INPUT`	[vector: geometry]	输入矢量图层

输出

标注	名称	类型	描述
Indexed layer	`OUTPUT`	[same as input]	A copy of the input vector layer with a spatial index

Python 代码

Algorithm ID: `native:createspatialindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Define projection

Sets an existing layer's projection to the provided CRS without reprojecting features.

Contrary to the *Assign projection* algorithm, it modifies the current layer and will not output a new layer.

Note

For Shapefile datasets, the `.prj` and `.qpj` files will be overwritten - or created if missing - to match the provided CRS.

Default menu: *Vector ► Data Management Tools*

See also

Assign projection, *Find projection*, *Reproject layer*

参数

标注	名称	类型	描述
输入图层	INPUT	[vector: geometry]	Vector layer with missing projection information
CRS	CRS	[crs]	Select the CRS to assign to the vector layer

输出

标注	名称	类型	描述
	INPUT	[same as input]	The input vector layer with the defined projection

Python 代码

Algorithm ID: `native:definecurrentprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Delete duplicate geometries

Finds and removes duplicated geometries.

Attributes are not checked, so in case two features have identical geometries but different attributes, only one of them will be added to the result layer.

Note

This algorithm does not require valid geometries as input.

See also

Drop geometries, *Remove null geometries*, *Delete duplicates by attribute*

参数

标注	名称	类型	描述
输入图层	INPUT	[vector: geometry]	The layer with duplicate geometries you want to clean

continues on next page

Table 24.550 – continued from previous page

标注	名称	类型	描述
Cleaned	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output layer containing the unique features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Count of discarded duplicate records	`DUPLICATE_COUN`	[numeric: integer]	Count of discarded duplicate records
Cleaned	`OUTPUT`	[same as input]	The output layer without any duplicated geometries
Count of retained records	`RETAINED_COUNT`	[numeric: integer]	Count of unique records

Python 代码

Algorithm ID: `native:deleteduplicategeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Delete duplicates by attribute

Deletes duplicate rows by only considering the specified field(s). The first matching row will be retained, and duplicates will be discarded.

Optionally, these duplicate records can be saved to a separate output for analysis.

See also

Delete duplicate geometries

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	The input layer
Fields to match duplicates by	`FIELDS`	[tablefield: any] [list]	Fields defining duplicates. Features with identical values for all these fields are considered duplicates.
Filtered (no duplicates)	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output layer containing the unique features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。
Filtered (duplicates) 可选	`DUPLICATES`	[same as input] Default: `[Skip output]`	Specify the output layer containing only the duplicates. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Filtered (duplicates)	`DUPLICATES`	[same as input]	Vector layer containing the duplicate features.
Count of discarded duplicate records	`DUPLICATE_COUN`	[numeric: integer]	Count of discarded duplicate records
Filtered (no duplicates)	`OUTPUT`	[same as input]	Vector layer containing the unique features.
Count of retained records	`RETAINED_COUNT`	[numeric: integer]	Count of unique records

Python 代码

Algorithm ID: `native:removeduplicatesbyattribute`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Detect dataset changes

Compares two vector layers, and determines which features are unchanged, added or deleted between the two. It is designed for comparing two different versions of the same dataset.

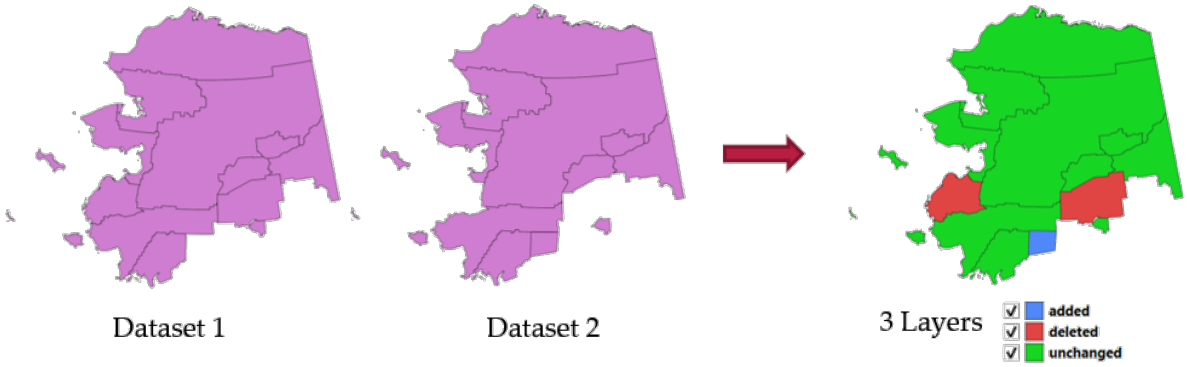


Fig. 24.85: Detect dataset change example

参数

标注	名称	类型	描述
Original layer	ORIGINAL	[vector: geometry]	The vector layer considered as the original version
Revised layer	REVISED	[vector: geometry]	The revised or modified vector layer
Attributes to consider for match 可选	COMPARE_ATTRIB	[tablefield: any] [list]	Attributes to consider for match. By default, all attributes are compared.
Geometry comparison behavior 可选	MATCH_TYPE	[enumeration] Default: 1	Defines the criteria for comparison. Options: • 0 --- Exact Match: includes the order and vertices count of geometries • 1 --- Tolerant Match (Topological Equality): geometries are considered equal
Unchanged features 可选	UNCHANGED	[vector: same as Original layer]	Specify the output vector layer containing the unchanged features. *One of*: • Skip Output • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

continues on next page

Table 24.554 – continued from previous page

标注	名称	类型	描述
Added features 可选	ADDED	[vector: same as Original layer]	Specify the output vector layer containing the added features. *One of*: • Skip Output • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。
Deleted features 可选	DELETED	[vector: same as Original layer]	Specify the output vector layer containing the deleted features. *One of*: • Skip Output • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Unchanged features	UNCHANGED	[vector: same as Original layer]	Vector layer containing the unchanged features.
Added features	ADDED	[vector: same as Original layer]	Vector layer containing the added features.
Deleted features	DELETED	[vector: same as Original layer]	Vector layer containing the deleted features.
Count of unchanged features	UNCHANGED_COUN	[numeric: integer]	Count of unchanged features.
Count of features added in revised layer	ADDED_COUNT	[numeric: integer]	Count of features added in revised layer.
Count of features deleted from original layer	DELETED_COUNT	[numeric: integer]	Count of features deleted from original layer.

Python 代码

Algorithm ID: `native:detectvectorchanges`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Drop geometries

Creates a simple *geometryless* copy of the input layer attribute table. It keeps the attribute table of the source layer.

If the file is saved in a local folder, you can choose between many file formats.

Allows *features in-place modification* of point, line, and polygon features

See also

Delete duplicate geometries, *Remove null geometries*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	The input vector layer
Dropped geometries	`OUTPUT`	[vector: table]	Specify the output geometryless layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Dropped geometries	`OUTPUT`	[vector: table]	The output geometryless layer. A copy of the original attribute table.

Python 代码

Algorithm ID: `native:dropgeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Execute SQL

Runs a simple or complex query based only on SELECT with `SQL` syntax on the source layer.

Input datasources are identified with `input1`, `input2`... `inputN` and a simple query will look like `SELECT * FROM input1`.

Beside a simple query, you can add expressions or variables within the `SQL query` parameter itself. This is particulary useful if this algorithm is executed within a Processing model and you want to use a model input as a parameter of the query. An example of a query will then be `SELECT * FROM [% @table %]` where `@table` is the variable that identifies the model input.

The result of the query will be added as a new layer.

See also

SpatiaLite execute SQL, *PostgreSQL execute SQL*

参数

标注	名称	类型	描述
Additional input datasources (called input1, .., inputN in the query)	`INPUT_DATASOUR`	[vector: any] [list]	List of layers to query. In the SQL editor you can refer these layers with their **real** name or also with **input1**, **input2**, **inputN** depending on how many layers have been chosen.
SQL query	`INPUT_QUERY`	[string]	Type the string of your SQL query, e.g. `SELECT * FROM input1`.
Unique identifier field 可选	`INPUT_UID_FIEL`	[string]	Specify the column with unique ID
Geometry field 可选	`INPUT_GEOMETRY`	[string]	Specify the geometry field
Geometry type 可选	`INPUT_GEOMETRY`	[enumeration] Default: 0	Choose the geometry of the result. By default the algorithm will autodetect it. One of: • 0 --- Autodetect • 1 --- No geometry • 2 --- Point • 3 --- LineString • 4 --- Polygon • 5 --- MultiPoint • 6 --- MultiLineString • 7 --- MultiPolygon
CRS 可选	`INPUT_GEOMETRY`	[crs]	The CRS to assign to the output layer
SQL Output	`OUTPUT`	[vector: geometry] Default: `[Create temporary layer]`	Specify the output layer created by the query. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
SQL Output	`OUTPUT`	[vector: geometry]	Vector layer created by the query

Python 代码

Algorithm ID: `qgis:executesql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Export layers to DXF

Exports layers to DXF file. For each layer, you can choose a field whose values are used to split features in generated destination layers in DXF output.

See also

新建 DXF 文件

参数

标注	名称	类型	描述
Input layers	LAYERS	[vector: geometry] [list]	List of input vector layers with options associated (filled as a QgsProcessingParameterDxfLayers item --- done in GUI by pressing *Configure Layer...* button for each selected layer): **Layer [string] (layer)** Full path of the input layer to export **Output layer attribute [tablefield: any] (attributeIndex)** Attribute index to split the input layer using unique values **Output layer name [string] (overriddenLayerName)** Overridden layer name to be used in the exported DXF file (the original project layer remains untouched) **Allow data defined symbol blocks** [boolean] (buildDataDefinedBlocks) **Maximum number of symbol blocks [numeric: integer] (dataDefinedBlocksMaximumNumberOfClasses)** −1 means no limitation.
Symbology mode	SYMBOLOGY_MODE	[enumeration] Default: 0	Type of symbology to apply to output layers. You can choose between: • 0 --- No Symbology • 1 --- Feature Symbology • 2 --- Symbol Layer Symbology
Symbology scale	SYMBOLOGY_SCAL	[scale] Default: 1:1 000 000	Default scale of data export.
Map theme 可选	MAP_THEME	[map theme]	Match layer styling to the provided map theme.
Encoding	ENCODING	[enumeration]	Encoding to apply to layers.
CRS	CRS	[crs]	Choose the CRS for the output layer.
Extent 可选	EXTENT	[范围]	Limit exported features to those with geometries intersecting the provided extent.
Use layer title as name	USE_LAYER_TITL	[boolean] Default: False	Name the output layer with the layer title (as set in layer metadata or QGIS Server properties) instead of the layer name.
Force 2D	FORCE_2D	[boolean] Default: False	
Export labels as MTEXT elements	MTEXT	[boolean] Default: True	Exports labels as MTEXT or TEXT elements
Use only selected features	SELECTED_FEATU	[boolean] Default: False	Exports only the selected features.
DXF	OUTPUT	[file] 默认: [保存到临时文件]	Specification of the output DXF file. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
DXF	`OUTPUT`	[file]	`.DXF` file containing the input layers

Python 代码

Algorithm ID: `native:dxfexport`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Extract selected features

Saves the selected features as a new layer.

> **Note**
>
> If the selected layer has no selected features, the newly created layer will be empty.

参数

标注	名称	类型	描述
Input Layer	`INPUT`	[vector: any]	Layer to save the selection from
Selected features	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the vector layer for the selected features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Selected features	`OUTPUT`	[same as input]	Vector layer with only the selected features, or no feature if none was selected.

Python 代码

Algorithm ID: `native:saveselectedfeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Extract Shapefile encoding

Extracts the attribute encoding information embedded in a Shapefile. Both the encoding specified by an optional `.cpg` file and any encoding details present in the `.dbf` LDID header block are considered.

参数

标注	名称	类型	描述
Input Layer	`INPUT`	[vector: geometry]	ESRI Shapefile (`.SHP`) Layer to extract the encoding information.

输出

标注	名称	类型	描述
Shapefile encoding	`ENCODING`	[string]	Encoding information specified in the input file
CPG encoding	`CPG_ENCODING`	[string]	Encoding information specified in any optional `.CPG` file
LDID encoding	`LDID_ENCODING`	[string]	Encoding information specified in `.dbf` LDID header block

Python 代码

Algorithm ID: `native:shpencodinginfo`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Find projection

Creates a shortlist of candidate coordinate reference systems, for instance for a layer with an unknown projection.

The area that the layer is expected to cover must be specified via the target area parameter. The coordinate reference system for this target area must be known to QGIS.

The algorithm operates by testing the layer's extent in every known reference system and then listing any for which the bounds would be near the target area if the layer was in this projection.

See also

Assign projection, *Define projection*, *Reproject layer*

参数

标注	名称	类型	描述
Input Layer	`INPUT`	[vector: geometry]	Layer with unknown projection
Target area for layer (xmin, xmax, ymin, ymax)	`TARGET_AREA`	[范围]	The area that the layer covers. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
CRS candidates	`OUTPUT`	[vector: table] Default: `[Create temporary layer]`	Specify the table (geometryless layer) for the CRS suggestions (EPSG codes). *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
CRS candidates	`OUTPUT`	[vector: table]	A table with all the CRS (EPSG codes) of the matching criteria.

Python 代码

Algorithm ID: `native:findprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Flatten relationship

Flattens a *relationship* for a vector layer, exporting a single layer containing one parent feature per related child feature. This master feature contains all the attributes for the related features. This allows to have the relation as a plain table that can be e.g. exported to CSV.

> **Warning**
>
> This algorithm drops existing primary keys or FID values and regenerates them in output layers.

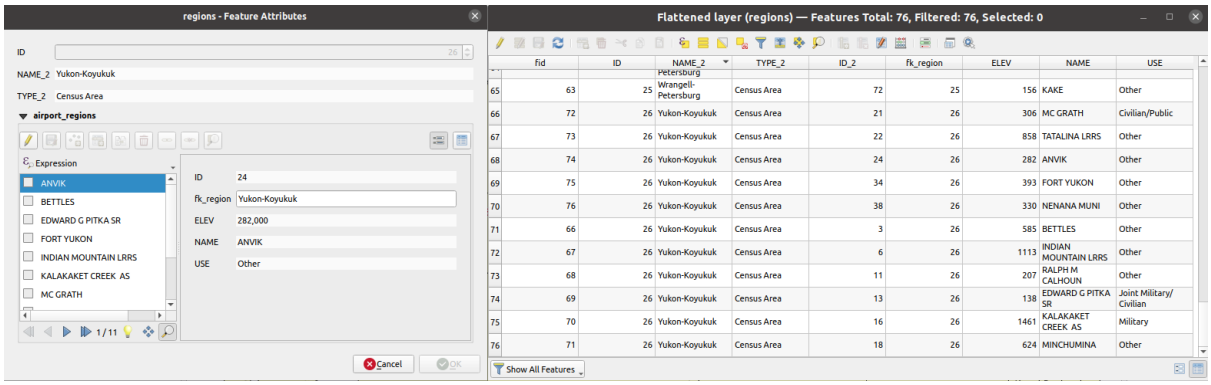


Fig. 24.86: Form of a region with related children (left) - A duplicate region feature for each related child, with joined attributes (right)

参数

标注	名称	类型	描述
Input Layer	`INPUT`	[vector: any]	Layer with the relationship that should be de-normalized
Flattened Layer 可选	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output (flattened) layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Flattened layer	`OUTPUT`	[same as input]	A layer containing master features with all the attributes for the related features

Python 代码

Algorithm ID: `native:flattenrelationships`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Join attributes by field value

Takes an input vector layer and creates a new vector layer that is an extended version of the input one, with additional attributes in its attribute table.

The additional attributes and their values are taken from a second vector layer. An attribute is selected in each of them to define the join criteria.

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

See also

Join attributes by nearest, *Join attributes by location*

参数

标注	名称	类型	描述
Input Layer	`INPUT`	[vector: any]	Input vector layer. The output layer will consist of the features of this layer with attributes from matching features in the second layer.
Table field	`FIELD`	[tablefield: any]	Field of the source layer to use for the join
Input layer 2	`INPUT_2`	[vector: any]	Layer with the attribute table to join
Table field 2	`FIELD_2`	[tablefield: any]	Field of the second (join) layer to use for the join The type of the field must be equal to (or compatible with) the input table field type.
Layer 2 fields to copy 可选	`FIELDS_TO_COPY`	[tablefield: any] [list]	Select the specific fields you want to add. By default all the fields are added.
Join type	`METHOD`	[enumeration] Default: 1	The type of the final joined layer. One of: • 0 --- Create separate feature for each matching feature (one-to-many) • 1 --- Take attributes of the first matching feature only (one-to-one)
Discard records which could not be joined	`DISCARD_NONMAT`	[boolean] Default: True	Check if you don't want to keep the features that could not be joined
Joined field prefix 可选	`PREFIX`	[string]	Add a prefix to joined fields in order to easily identify them and avoid field name collision
Joined layer 可选	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer for the join. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。
Unjoinable features from first layer 可选	`NON_MATCHING`	[same as input] Default: `[Skip output]`	Specify the output vector layer for unjoinable features from first layer. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Number of joined features from input table	JOINED_COUNT	[numeric: integer]	
Unjoinable features from first layer 可选	NON_MATCHING	[same as input]	Vector layer with the non-matched features
Joined layer 可选	OUTPUT	[same as input]	Output vector layer with added attributes from the join
Number of unjoinable features from input table 可选	UNJOINABLE_COU	[numeric: integer]	

Python 代码

Algorithm ID: `native:joinattributestable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Join attributes by location

Takes an input vector layer and creates a new vector layer that is an extended version of the input one, with additional attributes in its attribute table.

The additional attributes and their values are taken from a second vector layer. A spatial criteria is applied to select the values from the second layer that are added to each feature from the first layer.

Default menu: *Vector ► Data Management Tools*

> **Warning**
>
> This algorithm drops existing primary keys or FID values and regenerates them in output layers.

> **See also**
>
> *Join attributes by nearest*, *Join attributes by field value*, *Join attributes by location (summary)*

Exploring spatial relations

Geometric predicates are boolean functions used to determine the spatial relation a feature has with another by comparing whether and how their geometries share a portion of space.

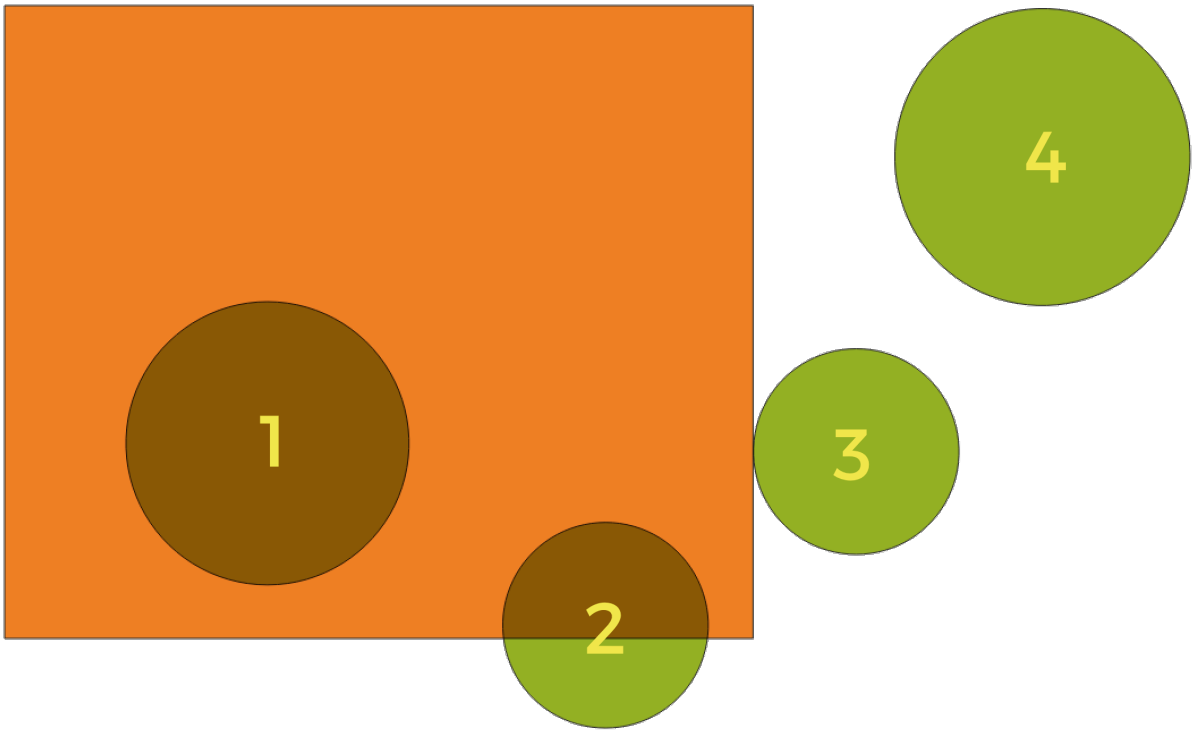


Fig. 24.87: Looking for spatial relations between layers

Using the figure above, we are looking for the green circles by spatially comparing them to the orange rectangle feature. Available geometric predicates are:

Intersect
: Tests whether a geometry intersects another. Returns 1 (true) if the geometries spatially intersect (share any portion of space - overlap or touch) and 0 if they don't. In the picture above, this will return circles 1, 2 and 3.

Contain
: Returns 1 (true) if and only if no points of b lie in the exterior of a, and at least one point of the interior of b lies in the interior of a. In the picture, no circle is returned, but the rectangle would be if you would look for it the other way around, as it contains circle 1 completely. This is the opposite of *are within*.

Disjoint
: Returns 1 (true) if the geometries do not share any portion of space (no overlap, not touching). Only circle 4 is returned.

Equal
: Returns 1 (true) if and only if geometries are exactly the same. No circles will be returned.

Touch
: Tests whether a geometry touches another. Returns 1 (true) if the geometries have at least one point in common, but their interiors do not intersect. Only circle 3 is returned.

Overlap
: Tests whether a geometry overlaps another. Returns 1 (true) if the geometries share space, are of the same dimension, but are not completely contained by each other. Only circle 2 is returned.

Are within

Tests whether a geometry is within another. Returns 1 (true) if geometry a is completely inside geometry b. Only circle 1 is returned.

Cross

Returns 1 (true) if the supplied geometries have some, but not all, interior points in common and the actual crossing is of a lower dimension than the highest supplied geometry. For example, a line crossing a polygon will cross as a line (true). Two lines crossing will cross as a point (true). Two polygons cross as a polygon (false). In the picture, no circles will be returned.

参数

标注	名称	类型	描述
Join to features in	`INPUT`	[vector: geometry]	Input vector layer. The output layer will consist of the features of this layer with attributes from matching features in the second layer.
Where the features	`PREDICATE`	[enumeration] [list] Default: [0]	Type of spatial relation the source feature should have with the target feature so that they could be joined. One or more of: • 0 --- intersect • 1 --- contain • 2 --- equal • 3 --- touch • 4 --- overlap • 5 --- are within • 6 --- cross If more than one condition is chosen, at least one of them (OR operation) has to be met for a feature to be extracted.
By comparing to	`JOIN`	[vector: geometry]	The join layer. Features of this vector layer will **add** their attributes to the source layer attribute table if they satisfy the spatial relationship.
Fields to add (leave empty to use all fields) 可选	`JOIN_FIELDS`	[tablefield: any] [list]	Select the specific fields you want to add from the join layer. By default all the fields are added.
Join type	`METHOD`	[enumeration]	The type of the final joined layer. One of: • 0 --- Create separate feature for each matching feature (one-to-many) • 1 --- Take attributes of the first matching feature only (one-to-one) • 2 --- Take attributes of the feature with largest overlap only (one-to-one)
Discard records which could not be joined	`DISCARD_NONMAT`	[boolean] Default: False	Remove from the output the input layer's features which could not be joined
Joined field prefix 可选	`PREFIX`	[string]	Add a prefix to joined fields in order to easily identify them and avoid field name collision

continues on next page

Table 24.572 – continued from previous page

标注	名称	类型	描述
Joined layer 可选	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer for the join. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。
Unjoinable features from first layer 可选	`NON_MATCHING`	[same as input] Default: `[Skip output]`	Specify the output vector layer for unjoinable features from first layer. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Number of joined features from input table	`JOINED_COUNT`	[numeric: integer]	
Unjoinable features from first layer 可选	`NON_MATCHING`	[same as input]	Vector layer of the non-matched features
Joined layer	`OUTPUT`	[same as input]	Output vector layer with added attributes from the join

Python 代码

Algorithm ID: `native:joinattributesbylocation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Join attributes by location (summary)

Takes an input vector layer and creates a new vector layer that is an extended version of the input one, with additional attributes in its attribute table.

The additional attributes and their values are taken from a second vector layer. A spatial criteria is applied to select the values from the second layer that are added to each feature from the first layer.

The algorithm calculates a statistical summary for the values from matching features in the second layer (e.g. maximum value, mean value, etc).

See also

Join attributes by location

Exploring spatial relations

Geometric predicates are boolean functions used to determine the spatial relation a feature has with another by comparing whether and how their geometries share a portion of space.

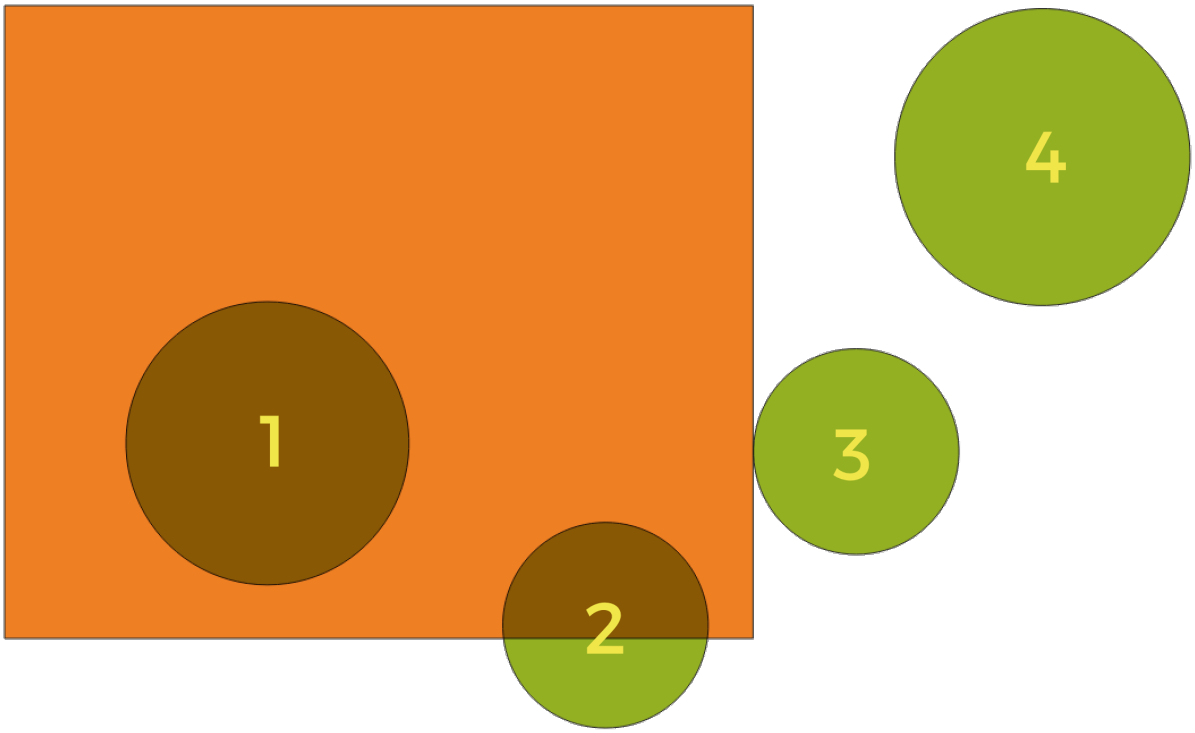


Fig. 24.88: Looking for spatial relations between layers

Using the figure above, we are looking for the green circles by spatially comparing them to the orange rectangle feature. Available geometric predicates are:

Intersect
: Tests whether a geometry intersects another. Returns 1 (true) if the geometries spatially intersect (share any portion of space - overlap or touch) and 0 if they don't. In the picture above, this will return circles 1, 2 and 3.

Contain
: Returns 1 (true) if and only if no points of b lie in the exterior of a, and at least one point of the interior of b

lies in the interior of a. In the picture, no circle is returned, but the rectangle would be if you would look for it the other way around, as it contains circle 1 completely. This is the opposite of *are within*.

Disjoint
Returns 1 (true) if the geometries do not share any portion of space (no overlap, not touching). Only circle 4 is returned.

Equal
Returns 1 (true) if and only if geometries are exactly the same. No circles will be returned.

Touch
Tests whether a geometry touches another. Returns 1 (true) if the geometries have at least one point in common, but their interiors do not intersect. Only circle 3 is returned.

Overlap
Tests whether a geometry overlaps another. Returns 1 (true) if the geometries share space, are of the same dimension, but are not completely contained by each other. Only circle 2 is returned.

Are within
Tests whether a geometry is within another. Returns 1 (true) if geometry a is completely inside geometry b. Only circle 1 is returned.

Cross
Returns 1 (true) if the supplied geometries have some, but not all, interior points in common and the actual crossing is of a lower dimension than the highest supplied geometry. For example, a line crossing a polygon will cross as a line (true). Two lines crossing will cross as a point (true). Two polygons cross as a polygon (false). In the picture, no circles will be returned.

参数

标注	名称	类型	描述
Join to features in	`INPUT`	[vector: geometry]	Input vector layer. The output layer will consist of the features of this layer with attributes from matching features in the second layer.
Where the features	`PREDICATE`	[enumeration] [list] Default: [0]	Type of spatial relation the source feature should have with the target feature so that they could be joined. One or more of: • 0 --- intersect • 1 --- contain • 2 --- equal • 3 --- touch • 4 --- overlap • 5 --- are within • 6 --- cross If more than one condition is chosen, at least one of them (OR operation) has to be met for a feature to be extracted.
By comparing to	`JOIN`	[vector: geometry]	The join layer. Features of this vector layer will **add** summaries of their attributes to the source layer attribute table if they satisfy the spatial relationship.
Fields to summarize (leave empty to use all fields) 可选	`JOIN_FIELDS`	[tablefield: any] [list]	Select the specific fields you want to add from the join layer. By default all the fields are added.

continues on next page

Table 24.574 – continued from previous page

标注	名称	类型	描述
Summaries to calculate (leave empty to use all fields) 可选	SUMMARIES	[enumeration] [list] Default: []	For each input feature, statistics are calculated on joined fields of their matching features. One or more of: • 0 --- count • 1 --- unique • 2 --- min • 3 --- max • 4 --- range • 5 --- sum • 6 --- mean • 7 --- median • 8 --- stddev • 9 --- minority • 10 --- majority • 11 --- q1 • 12 --- q3 • 13 --- iqr • 14 --- empty • 15 --- filled • 16 --- min_length • 17 --- max_length • 18 --- mean_length
Discard records which could not be joined	DISCARD_NONMAT	[boolean] Default: False	Remove from the output the input layer's features which could not be joined
Joined layer	OUTPUT	[same as input] Default: [Create temporary layer]	Specify the output vector layer for the join. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Joined layer	OUTPUT	[same as input]	Output vector layer with summarized attributes from the join

Python 代码

Algorithm ID: `native:joinbylocationsummary`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Join attributes by nearest

Takes an input vector layer and creates a new vector layer with additional fields in its attribute table. The additional attributes and their values are taken from a second vector layer. Features are joined by finding the closest features from each layer.

By default only the nearest feature is joined, but the join can also join to the k-nearest neighboring features.

If a maximum distance is specified, only features which are closer than this distance will be matched.

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

See also

Nearest neighbour analysis, *Join attributes by field value*, *Join attributes by location*, *Distance matrix*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	The input layer.
Input layer 2	`INPUT_2`	[vector: geometry]	The join layer.
Layer 2 fields to copy (leave empty to copy all fields)	`FIELDS_TO_COPY`	[fields]	Join layer fields to copy (if empty, all fields will be copied).
Discard records which could not be joined	`DISCARD_NONMAT`	[boolean] Default: False	Remove from the output the input layer records which could not be joined
Joined field prefix	`PREFIX`	[string]	Joined field prefix
Maximum nearest neighbors	`NEIGHBORS`	[numeric: integer] Default: 1	Maximum number of nearest neighbors
Maximum distance	`MAX_DISTANCE`	[numeric: double]	Maximum search distance
Joined layer 可选	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the vector layer containing the joined features. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

continues on next page

Table 24.576 – continued from previous page

标注	名称	类型	描述
Unjoinable features from first layer	NON_MATCHING	[same as input] Default: [Skip output]	Specify the vector layer containing the features that could not be joined. *One of*: • Skip Output • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Joined layer	OUTPUT	[same as input]	The output joined layer.
Unjoinable features from first layer	NON_MATCHING	[same as input]	Layer containing the features from first layer that could not be joined to any features in the join layer.
Number of joined features from input table	JOINED_COUNT	[numeric: integer]	Number of features from the input table that have been joined.
Number of unjoinable features from input table	UNJOINABLE_COU	[numeric: integer]	Number of features from the input table that could not be joined.

Python 代码

Algorithm ID: native:joinbynearest

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

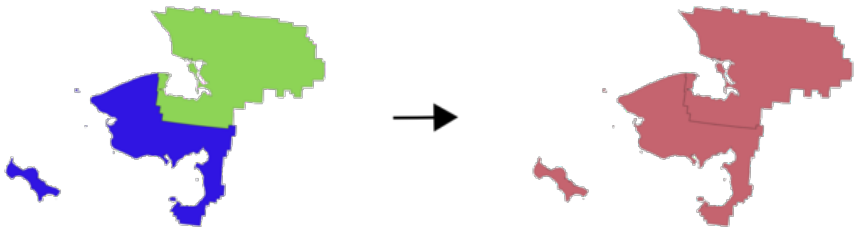
Merge vector layers

Combines multiple vector layers of the **same geometry** type into a single one.

The attribute table of the resulting layer will contain the fields from all input layers. If fields with the same name but different types are found then the exported field will be automatically converted into a string type field. Optionally, new fields storing the original layer name and source can be added.

If any input layers contain Z or M values, then the output layer will also contain these values. Similarly, if any of the input layers are multi-part, the output layer will also be a multi-part layer.

Optionally, the destination coordinate reference system (CRS) for the merged layer can be set. If it is not set, the CRS will be taken from the first input layer. All layers will be reprojected to match this CRS.



Default menu: *Vector ► Data Management Tools*

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

See also

Split vector layer

参数

标注	名称	类型	描述
Input Layers	`LAYERS`	[vector: any] [list]	The layers that are to be merged into a single layer. Layers should be of the same geometry type.
Destination CRS 可选	`CRS`	[crs]	Choose the CRS for the output layer. If not specified, the CRS of the first input layer is used.
Add source layer information (layer name and path)	`ADD_SOURCE_FIE`	[boolean] Default: True	Add fields storing the original layer name and path
Merged	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Merged	`OUTPUT`	[same as input]	Output vector layer containing all the features and attributes from the input layers.

Python 代码

Algorithm ID: `native:mergevectorlayers`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Order by expression

Sorts a vector layer according to an expression: changes the feature index according to an expression.

Be careful, it might not work as expected with some providers, the order might not be kept every time.

> **Warning**
>
> This algorithm drops existing primary keys or FID values and regenerates them in output layers.

参数

标注	名称	类型	描述
Input Layer	`INPUT`	[vector: any]	Input vector layer to sort
Expression	`EXPRESSION`	[expression]	Expression to use for the sorting
Sort ascending	`ASCENDING`	[boolean] Default: True	If checked the vector layer will be sorted from small to large values.
Sort nulls first	`NULLS_FIRST`	[boolean] Default: False	If checked, Null values are placed first
Ordered	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Ordered	`OUTPUT`	[same as input]	Output (sorted) vector layer

Python 代码

Algorithm ID: `native:orderbyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Repair Shapefile

Repairs a broken ESRI Shapefile dataset by (re)creating the SHX file.

参数

标注	名称	类型	描述
Input Shapefile	`INPUT`	[file]	Full path to the ESRI Shapefile dataset with a missing or broken SHX file

输出

标注	名称	类型	描述
Repaired layer	`OUTPUT`	[vector: geometry]	The input vector layer with the SHX file repaired

Python 代码

Algorithm ID: `native:repairshapefile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Reproject layer

Reprojects a vector layer in a different CRS. The reprojected layer will have the same features and attributes of the input layer.

Allows *features in-place modification* of point, line, and polygon features

See also

Assign projection, *Define projection*, *Find projection*

参数

Basic parameters

标注	名称	类型	描述
Input Layer	`INPUT`	[vector: geometry]	Input vector layer to reproject
Target CRS	`TARGET_CRS`	[crs] Default: `EPSG:4326 - WGS 84`	Destination coordinate reference system
Convert curved geometries to straight segments	`CONVERT_CURVED`	[boolean] Default: False	If checked, curved geometries will be converted to straight segments in the process, avoiding potential distortion issues.
Reprojected	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Coordinate Operation 可选	`OPERATION`	[string]	Specific operation to use for a particular reprojection task, instead of always forcing use of the current project's transformation settings. Useful when reprojecting a particular layer and control over the exact transformation pipeline is required. Requires proj version >= 6. Read more at 基准变换.

输出

标注	名称	类型	描述
Reprojected	`OUTPUT`	[same as input]	Output (reprojected) vector layer

Python 代码

Algorithm ID: `native:reprojectlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Save vector features to file

Saves vector features to a specified file dataset.

For dataset formats supporting layers, an optional layer name parameter can be used to specify a custom string. Optional GDAL-defined dataset and layer options can be specified. For more information on this, read the online GDAL documentation on the format.

参数

Basic parameters

标注	名称	类型	描述
Vector features	INPUT	[vector: any]	Input vector layer.
Saved features	OUTPUT	[same as input] 默认：[保存到临时文件]	Specify the file to save the features to. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Layer name 可选	LAYER_NAME	[string]	Name to use for the output layer
GDAL dataset options 可选	DATASOURCE_OPT	[string]	GDAL dataset creation options of the output format. Separate individual options with semicolons.
GDAL layer options 可选	LAYER_OPTIONS	[string]	GDAL layer creation options of the output format. Separate individual options with semicolons.
Action to take on pre-existing file	ACTION_ON_EXIS	[enumeration] Default: 0	How to manage existing features. Valid methods are: • 0 --- Create or overwrite file • 1 --- Create or overwrite layer • 2 --- Append features to existing layer, but do not create new fields • 3 --- Append features to existing layer, and create new fields if needed

输出

标注	名称	类型	描述
Saved features	OUTPUT	[same as input]	Vector layer with the saved features.
File name and path	FILE_PATH	[string]	Output file name and path.
Layer name	LAYER_NAME	[string]	Name of the layer, if any.

Python 代码

Algorithm ID: native:savefeatures

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Set layer encoding

Sets the encoding used for reading a layer's attributes. No permanent changes are made to the layer, rather it affects only how the layer is read during the current session.

> **Note**
>
> Changing the encoding is only supported for some vector layer data sources.

参数

标注	名称	类型	描述
Saved features	INPUT	[vector: geometry]	Vector layer to set the encoding.
Encoding	ENCODING	[string]	Text encoding to assign to the layer in the current QGIS session.

输出

标注	名称	类型	描述
Output layer	OUTPUT	[same as input]	Input vector layer with the set encoding.

Python 代码

Algorithm ID: `native:setlayerencoding`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Split features by character

Features are split into multiple output features by splitting a field's value at a specified character. For instance, if a layer contains features with multiple comma separated values contained in a single field, this algorithm can be used to split these values up across multiple output features. Geometries and other attributes remain unchanged in the output. Optionally, the separator string can be a regular expression for added flexibility.

Allows *features in-place modification* of point, line, and polygon features

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

参数

标注	名称	类型	描述
Input Layer	`INPUT`	[vector: any]	输入矢量图层
Split using values in the field	`FIELD`	[tablefield: any]	Field to use for splitting
Split value using character	`CHAR`	[string]	Character to use for splitting
Use regular expression separator	`REGEX`	[boolean] Default: False	
Split	`OUTPUT`	[same as input] Default: `Create temporary layer`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Split	`OUTPUT`	[same as input]	The output vector layer.

Python 代码

Algorithm ID: `native:splitfeaturesbycharacter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Split vector layer

Creates a set of vectors in an output folder based on an input layer and an attribute. The output folder will contain as many layers as the unique values found in the desired field.

The number of files generated is equal to the number of different values found for the specified attribute.

It is the opposite operation of *merging*.

Default menu: *Vector ► Data Management Tools*

See also

Merge vector layers

参数

Basic parameters

标注	名称	类型	描述
Input Layer	`INPUT`	[vector: any]	输入矢量图层
Unique ID field	`FIELD`	[tablefield: any]	Field to use for splitting
Output directory	`OUTPUT`	[folder] Default: `[Save to temporary folder]`	Specify the directory for the output layers. *One of*: • Save to a Temporary Directory • Save to Directory

Advanced parameters

标注	名称	类型	描述
Output file type 可选	`FILE_TYPE`	[enumeration] Default: `gpkg` in the dialog window	Select the extension of the output files. If not specified or invalid, the output files format will be the one set in the "Default output vector layer extension" Processing setting.

输出

标注	名称	类型	描述
Output directory	`OUTPUT`	[folder]	The directory for the output layers
Output layers	`OUTPUT_LAYERS`	[same as input] [list]	The output vector layers resulting from the split.

Python 代码

Algorithm ID: `native:splitvectorlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Truncate table

Truncates a layer, by deleting all features from within the layer.

> **Warning**
>
> This algorithm modifies the layer in place, and deleted features cannot be restored!

参数

标注	名称	类型	描述
Input Layer	`INPUT`	[vector: any]	输入矢量图层

输出

标注	名称	类型	描述
Truncated layer	`OUTPUT`	[folder]	The input layer, all features deleted

Python 代码

Algorithm ID: `native:truncatetable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.26 矢量几何图形

添加几何图形属性

计算矢量图层中要素的几何属性，并将这些属性包含在输出图层中。

它会生成一个与输入图层内容相同的新矢量图层，但会新增一些属性字段，用于存储基于所选 CRS 的几何测度。

添加到表中的属性取决于几何图形类型和输入图层维度：

- 对于**点 ** 图层：X (`xcoord`)，Y (`ycoord`)，Z (`zcoord`) 坐标，和/或 M 值 (`mvalue`)
- for **line** layers: `length` and, for the LineString and CompoundCurve geometry types, the feature `sinuosity` and straight distance (`straightdis`)
- for **polygon** layers: `perimeter` and `area`

> **Note**
>
> This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

Default menu: *Vector* ► *Geometry Tools*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Calculate using	`CALC_METHOD`	[enumeration] Default: 0	Calculation parameters to use for the geometric properties. One of: • 0 --- Layer CRS • 1 --- Project CRS • 2 --- Ellipsoidal

continues on next page

Table 24.599 – continued from previous page

标注	名称	类型	描述
Added geom info	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output (input copy with geometry) layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Added geom info	`OUTPUT`	[same as input]	Copy of the input vector layer with the addition of the geometry fields

Python 代码

Algorithm ID: `native:exportaddgeometrycolumns`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Affine transform

Applies an affine transformation to the layer geometries. Affine transformations can include translation, scaling and rotation. The operations are performed in the following order: scale, rotation, and translation.

Z and M values (if present) can be translated and scaled.

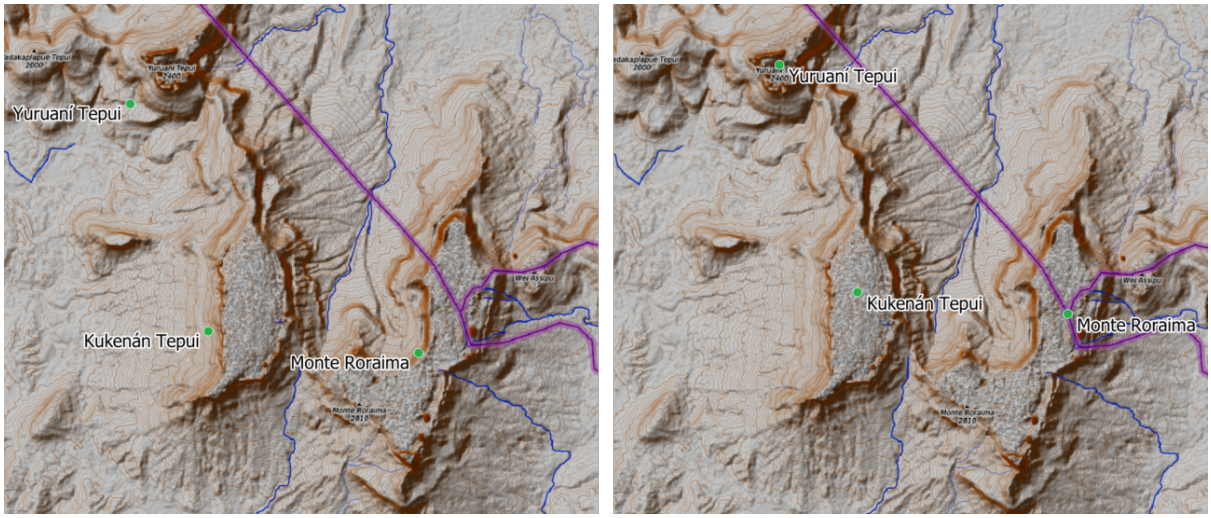


Fig. 24.89: Vector point layer (green dots) before (left), and after (right) an affine transformation (translation).

Allows *features in-place modification* of point, line, and polygon features

See also

翻译

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Translation (x-axis)	`DELTA_X`	[numeric: double] Default: 0.0	Displacement to apply on the X axis.
Translation (y-axis)	`DELTA_Y`	[numeric: double] Default: 0.0	Displacement to apply on the Y axis.
Translation (z-axis)	`DELTA_Z`	[numeric: double] Default: 0.0	Displacement to apply on the Z axis.
Translation (m-values)	`DELTA_M`	[numeric: double] Default: 0.0	Offset to apply on m values.
Scale factor (x-axis)	`SCALE_X`	[numeric: double] 默认值：1.0	Scaling value (expansion or contraction) to apply on the X axis.
Scale factor (y-axis)	`SCALE_Y`	[numeric: double] 默认值：1.0	Scaling value (expansion or contraction) to apply on the Y axis.
Scale factor (z-axis)	`SCALE_Z`	[numeric: double] 默认值：1.0	Scaling value (expansion or contraction) to apply on the Z axis.
Scale factor (m-values)	`SCALE_M`	[numeric: double] 默认值：1.0	Scaling value (expansion or contraction) to apply on m values.
Rotation around z-axis (degrees counter-clockwise)	`ROTATION_Z`	[numeric: double] Default: 0.0	Angle of the rotation in degrees.
Transformed	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Transformed	`OUTPUT`	[same as input]	Output (transformed) vector layer.

Python 代码

Algorithm ID: `native:affinetransform`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Aggregate

Takes a vector or table layer and creates a new layer by aggregating features based on a `group by` expression.

Features for which `group by` expression returns the same value are grouped together.

It is possible to group all source features together using constant value in `group by` parameter, example: NULL.

It is also possible to group features by multiple fields using Array function, example: Array("Field1", "Field2").

Geometries (if present) are combined into one multipart geometry for each group. Output attributes are computed depending on each given aggregate definition.

该算法允许使用 QGIS 表达式引擎中默认的聚合函数。

Note

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

See also

Collect geometries, 融合（*dissolve*）

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	输入矢量图层
按表达式分组	`GROUP_BY`	[tablefield: any] Default: 'NULL'	Choose the grouping field. If *NULL* all features will be grouped.

continues on next page

Table 24.603 – continued from previous page

标注	名称	类型	描述
Aggregates	AGGREGATES	[list]	List of output layer field definitions. Example of a field definition: *{'aggregate': 'sum', 'delimiter': ',', 'input': ' $area', 'length': 10, 'name': 'totarea', 'precision': 0, 'type': 6}* By default, the list contains all the fields of the input layer. In the GUI, you can edit these fields and their definitions, and you can also: • Click the button to add a new field. • Click to delete the selected field. • Use and to change order of the fields. • Click to reset to the default (the fields of the input layer). For each of the fields you'd like to retrieve information from, you need to define the following: ***Input expression*** **(`input`) [expression]** Field or expression from the input layer. ***Aggregate function*** **(`aggregate`) [enumeration]** Function to use on the input expression to return the aggregated value. These are mainly the 聚合函数, plus dedicated "first_value" and "last_value" functions for ordered list items. Default: *concatenate* (for string data type), *sum* (for numeric data type) ***Delimiter*** **(`delimiter`) [string]** Text string to separate aggregated values, for example in case of concatenation. Default: , ***Output field name*** **(`name`) [string]** Name of the aggregated field in the output layer. By default input field name is kept. ***Type*** **(`type`) [enumeration]** Data type of the output field. Available types may not be compatible with the output layer provider. One of: **Attention** For certain field types, e.g. lists, an extra `sub_type` parameter helps refine the specific type of the data. It is automatically set in the GUI but may be needed if you're running the algorithm in Python or from the command line. • 1 --- Boolean • 2 --- Integer (32bit) • 4 --- Integer (64bit) • 6 --- Decimal (double)

Table 24.603 – continued from previous page

标注	名称	类型	描述
Load fields from layer	GUI only	[vector: any]	You can load fields from another layer and use them for the aggregation
Aggregated	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output (aggregate) layer *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Aggregated	`OUTPUT`	[same as input]	Multigeometry vector layer with the aggregated values

Python 代码

Algorithm ID: `native:aggregate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Boundary

Returns the closure of the combinatorial boundary of the input geometries (i.e. the topological boundary of the geometry).

Only for polygon and line layers.

For **polygon geometries** , the boundary consists of all the lines making up the rings of the polygon.

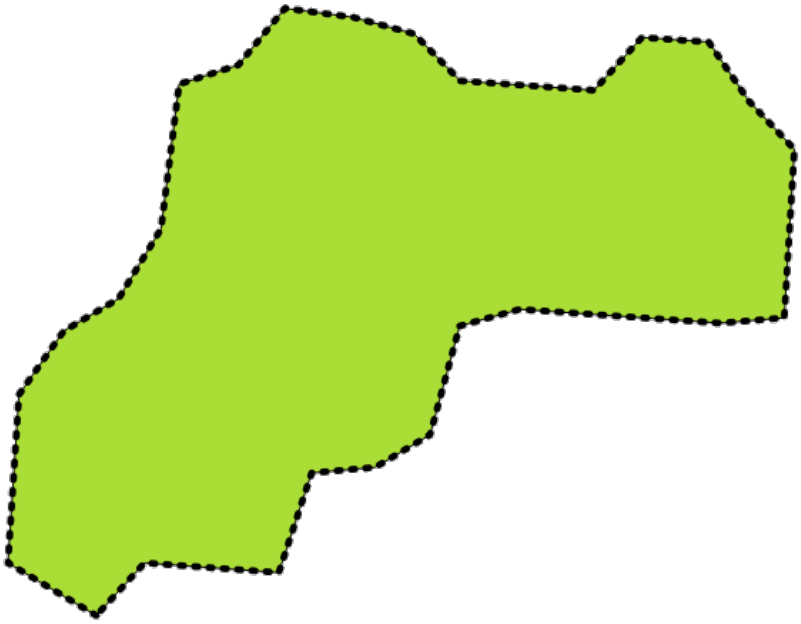


Fig. 24.90: Boundaries (black dashed line) of the source polygon layer

For **lines geometries**, the boundaries are their end points.

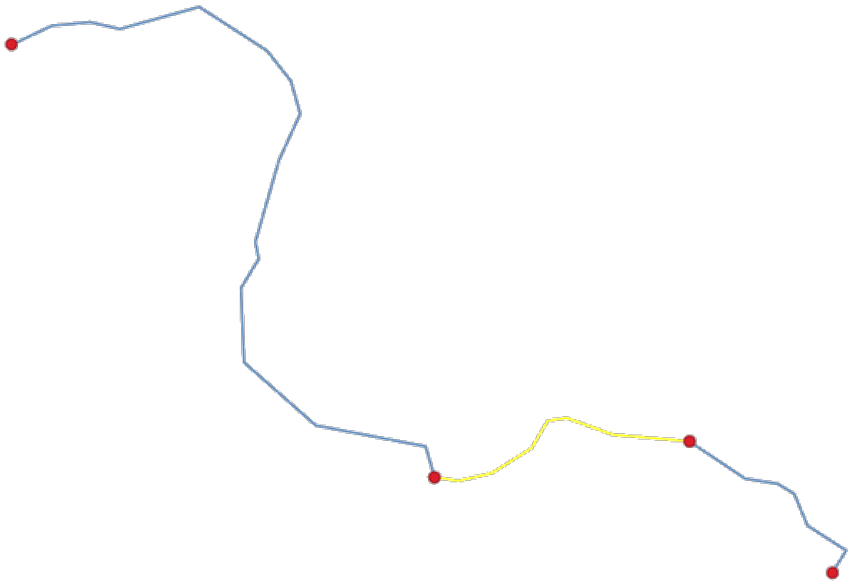


Fig. 24.91: Boundary layer (red points) for lines. In yellow a selected feature.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line, polygon]	Input line or polygon vector layer
Boundary	`OUTPUT`	[vector: point, line] Default: `[Create temporary layer]`	Specify the output (boundary) layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Boundary	`OUTPUT`	[vector: point, line]	Boundaries from the input layer (point for line, and line for polygon)

Python 代码

Algorithm ID: `native:boundary`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Bounding boxes

Calculates the bounding box (envelope) of each feature in an input layer. Polygon and line geometries are supported.

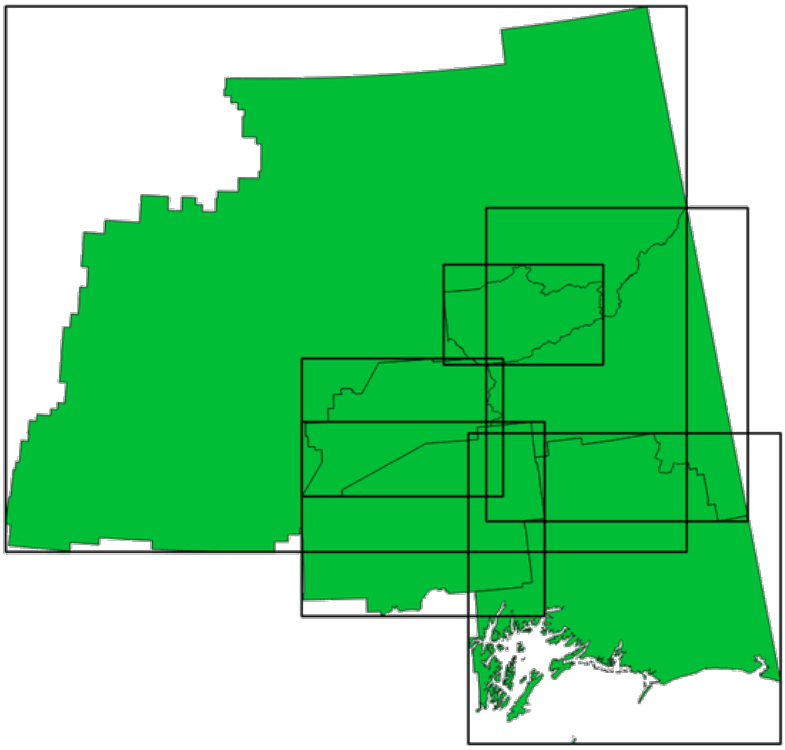


Fig. 24.92: Black lines represent the bounding boxes of each polygon feature

Allows *features in-place modification* of polygon features

See also

Minimum bounding geometry

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line, polygon]	Input line or polygon vector layer
Bounds	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output (bounding box) layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Bounds	`OUTPUT`	[vector: polygon]	Bounding boxes of input layer. Other than the input attributes, the output layer also contains following fields: `width`, `height`, `area` and `perimeter` of the generated polygon.

Python 代码

Algorithm ID: `native:boundingboxes`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Buffer

Computes a buffer area for all the features in an input layer, using a fixed or data defined distance.

It is possible to use a negative distance for polygon input layers. In this case the buffer will result in a smaller polygon (setback).

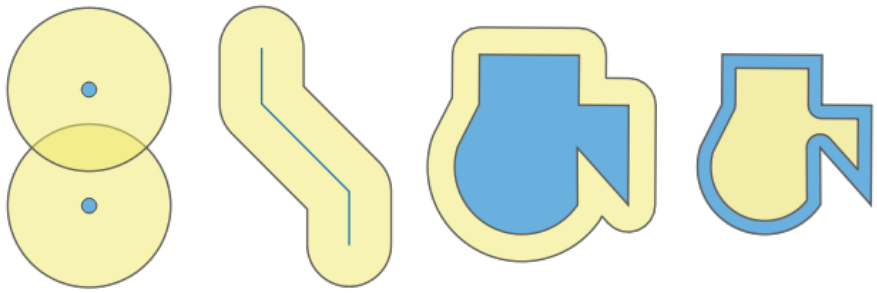


Fig. 24.93: Buffer (in yellow) of points, line, polygon with positive buffer, and polygon with negative buffer

Allows *features in-place modification* of polygon features

Default menu: *Vector* ► *Geoprocessing Tools*

Warning

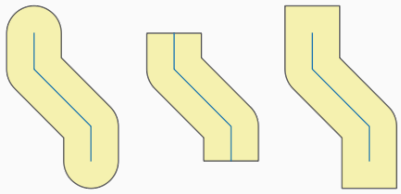
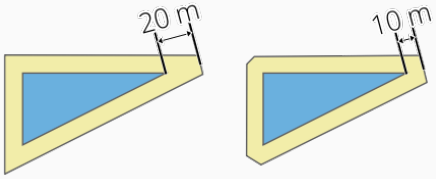
This algorithm may drop existing primary keys or FID values and regenerate them in output layers, depending on the input parameters.

See also

Variable distance buffer, *Multi-ring buffer (constant distance)*, *Variable width buffer (by M value)*

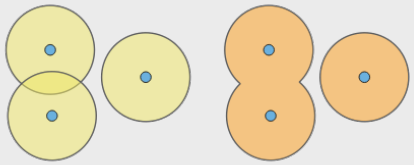
参数

Basic parameters

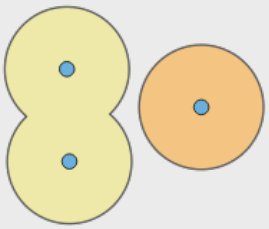
标注	名称	类型	描述
输入图层	INPUT	[vector: geometry]	输入矢量图层
Distance	DISTANCE	[numeric: double] Default: 10.0	Buffer distance (from the boundary of each feature). You can use the Data Defined button on the right to choose a field from which the radius will be calculated. This way you can have different radius for each feature.
Segments	SEGMENTS	[numeric: integer] Default: 5	Controls the number of line segments to use to approximate a quarter circle when creating rounded offsets.
End cap style	END_CAP_STYLE	[enumeration] Default: 0	Controls how line endings are handled in the buffer. One of: • 0 --- Round • 1 --- Flat • 2 --- Square  Fig. 24.94: Round, flat and square cap styles
Join style	JOIN_STYLE	[enumeration] Default: 0	Specifies whether round, miter or beveled joins should be used when offsetting corners in a line. Options are: • 0 --- Round • 1 --- Miter • 2 --- Bevel Fig. 24.95: Round, miter, and bevel join styles
Miter limit	MITER_LIMIT	[numeric: double] Default: 2.0	Sets the maximum distance from the offset geometry to use when creating a mitered join as a factor of the offset distance (only applicable for miter join styles). Minimum: 1.0  Fig. 24.96: A 10m buffer with a limit of 2 and a 10m buffer with a limit of 1

continues on next page

Table 24.609 – continued from previous page

标注	名称	类型	描述
Dissolve result	DISSOLVE	[boolean] Default: False	Dissolve the final buffer. If True (checked), overlapping buffers will be dissolved (combined) into a single multipart feature.  Fig. 24.97: Standard (three single part features - left), dissolved (1 multipart feature with 2 parts - right)
Buffered	OUTPUT	[vector: polygon] Default: [Create temporary layer]	Specify the output (buffer) layer. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Keep disjoint features separate	SEPARATE_DISJO	[boolean] Default: False	If True (checked) and dissolved is checked, features that do not overlap or touch will be exported as separate features (instead of parts of a multipart feature).  Fig. 24.98: Results in 2 single part features

输出

标注	名称	类型	描述
Buffered	OUTPUT	[vector: polygon]	Output (buffer) polygon layer

Python 代码

Algorithm ID: `native:buffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Centroids

Creates a new point layer, with points representing the centroids of the geometries of the input layer.

The centroid is a single point representing the barycenter (of all parts) of the feature, so it can be outside the feature borders. But can also be a point on each part of the feature.

The attributes of the points in the output layer are the same as for the original features.

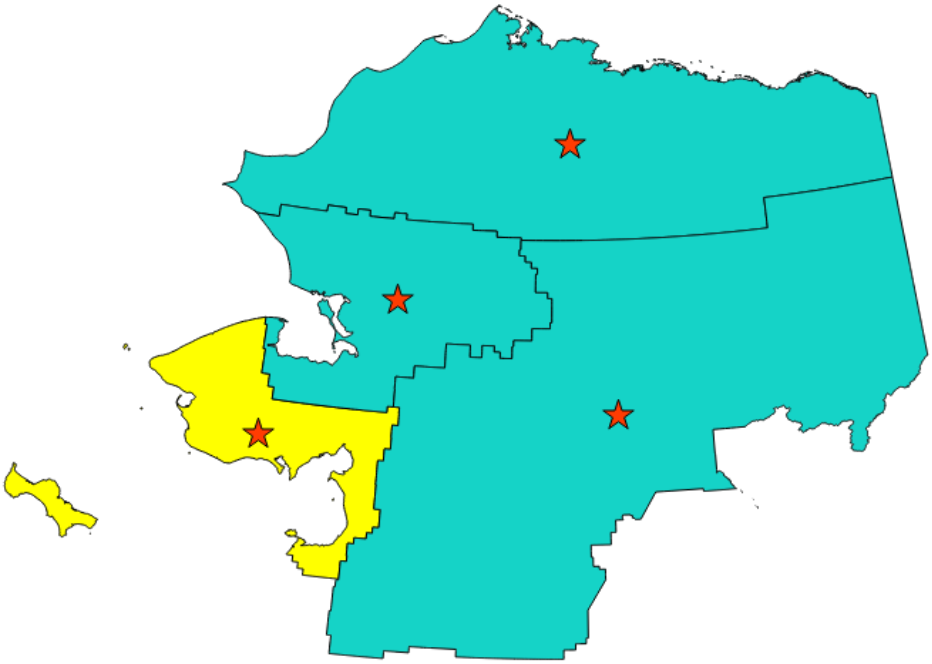


Fig. 24.99: The red stars represent the centroids of the features of the input layer.

Allows *features in-place modification* of point features

Default menu: *Vector ► Geometry Tools*

Warning

This algorithm may drop existing primary keys or FID values and regenerate them in output layers, depending on the input parameters.

See also

Point on Surface

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Create centroid for each part	`ALL_PARTS`	[boolean] Default: False	If True (checked), a centroid will be created for each part of the geometry
Centroids	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the output (centroid) layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Centroids	`OUTPUT`	[vector: point]	Output point vector layer (centroids)

Python 代码

Algorithm ID: `native:centroids`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Check validity

Performs a validity check on the geometries of a vector layer.

The geometries are classified in three groups (valid, invalid and error) and for each group, a vector layer with its features is generated:

- The **Valid output** layer contains only the valid features (without topological errors).
- The **Invalid output** layer contains all the invalid features found by the algorithm.
- The **Error output** layer is a point layer that points to where the invalid features were found.

The attribute tables of the generated layers will contain some additional information (”message” for the **error** layer, ”FID” and ”_errors” for the **invalid** layer and only ”FID” for the **valid** layer):

The attribute table of each generated vector layer will contain some additional information (number of errors found and types of error):

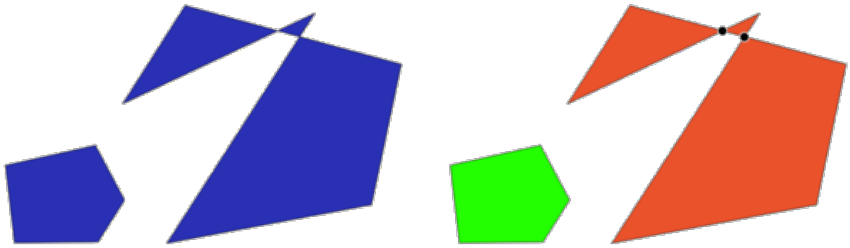


Fig. 24.100: Left: the input layer. Right: the valid layer (green), the invalid layer (orange)

Default menu: *Vector ► Geometry Tools*

See also

Check Geometry, *Fix geometries*, *Validate coverage*

参数

标注	名称	类型	描述
输入图层	`INPUT_LAYER`	[vector: geometry]	输入矢量图层
Method	`METHOD`	[enumeration] Default: 2	Method to use to check validity. Options: • 0: The one selected in digitizing settings • 1: QGIS • 2: GEOS
Ignore ring self intersection	`IGNORE_RING_SE`	[boolean] Default: False	Ignore self intersecting rings when checking for validity.
Valid output	`VALID_OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the vector layer to contain a copy of the valid features of the source layer. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。
Invalid output	`INVALID_OUTPUT`	[same as input] Default: `[Create temporary layer]`	Vector layer containing copy of the invalid features of the source layer with the field `_errors` listing the summary of the error(s) found. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

continues on next page

Table 24.614 – continued from previous page

标注	名称	类型	描述
Error output	ERROR_OUTPUT	[vector: point] Default: [Create temporary layer]	Point layer of the exact position of the validity problems detected with the message field describing the error(s) found. *One of*: • Skip Output • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Count of errors	ERROR_COUNT	[numeric: integer]	The number of geometries that caused errors.
Error output	ERROR_OUTPUT	[vector: point]	Point layer of the exact position of the validity problems detected with the message field describing the error(s) found.
Count of invalid features	INVALID_COUNT	[numeric: integer]	The number of invalid geometries.
Invalid output	INVALID_OUTPUT	[same as input]	Vector layer containing copy of the invalid features of the source layer with the field _errors listing the summary of the error(s) found.
Count of valid features	VALID_COUNT	[numeric: integer]	The number of valid geometries.
Valid output	VALID_OUTPUT	[same as input]	Vector layer containing a copy of the valid features of the source layer.

Python 代码

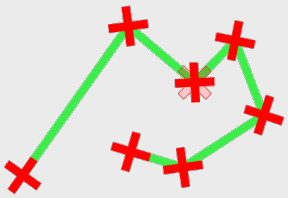
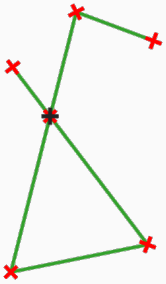
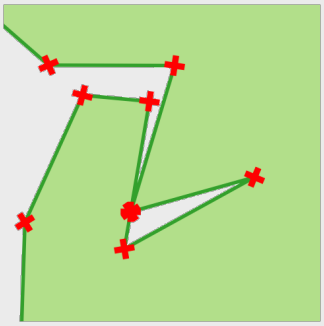
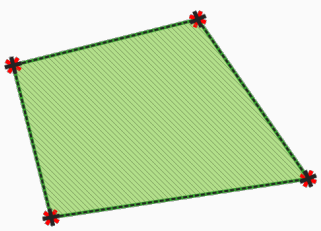
Algorithm ID: native:checkvalidity

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

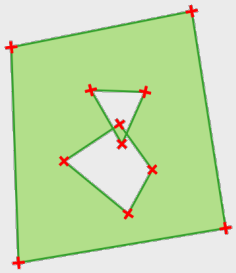
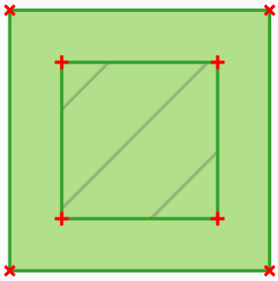
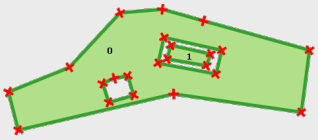
Types of error messages and their meanings

Table 24.616: QGIS 中几何图形检查失败的一些例子

几何图形上下文	GEOS 验证和错误消息	QGIS 验证和错误消息
一条直线上有相同坐标的连续点 	*Repeated point*	*Line a contains x duplicate node(s) at b*
一条线的各个线段彼此相交 		*Segments a and b of line c intersect at d*
Polygon geometry touches itself and generates a ring 	*Ring self-intersection*	*Ring self-intersection*
Two rings (exterior or interior) of a polygon geometry are identical 	*Duplicate rings*	

continues on next page

Table 24.616 – continued from previous page

几何图形上下文	GEOS 验证和错误消息	QGIS 验证和错误消息
Geometry touches itself 	✗ *Self-intersection*	
A polygon geometry is on top of another polygon geometry 	✗ *Nested shell*	
Part of a MultiPolygon geometry is within a hole of a MultiPolygon geometry 		✗ *Polygon a lies inside polygon b*
Point geometry does not have a proper coordinate pair. The coordinate pair does not contain a latitude value and a longitude value in that order.	✗ *Invalid coordinate*	

Collect geometries

Takes a vector layer and collects its geometries into new multipart geometries.

One or more attributes can be specified to collect only geometries belonging to the same class (having the same value for the specified attributes), alternatively all geometries can be collected.

All output geometries will be converted to multi geometries, even those with just a single part. This algorithm does not dissolve overlapping geometries - they will be collected together without modifying the shape of each geometry part.

See the 'Promote to multipart' or 'Aggregate' algorithms for alternative options.

Default menu: *Vector ► Geometry Tools*

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

See also

Aggregate, *Promote to multipart*, 融合（*dissolve*）

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Unique ID fields	`FIELD`	[tablefield: any] [list]	Choose one or more attributes to collect the geometries
Collected	`OUTPUT`	[same as input]	Vector layer with collected geometries

输出

标注	名称	类型	描述
Collected	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer for the collected geometries. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Python 代码

Algorithm ID: `native:collect`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Concave hull

Computes the concave hull of the features from an input point layer.

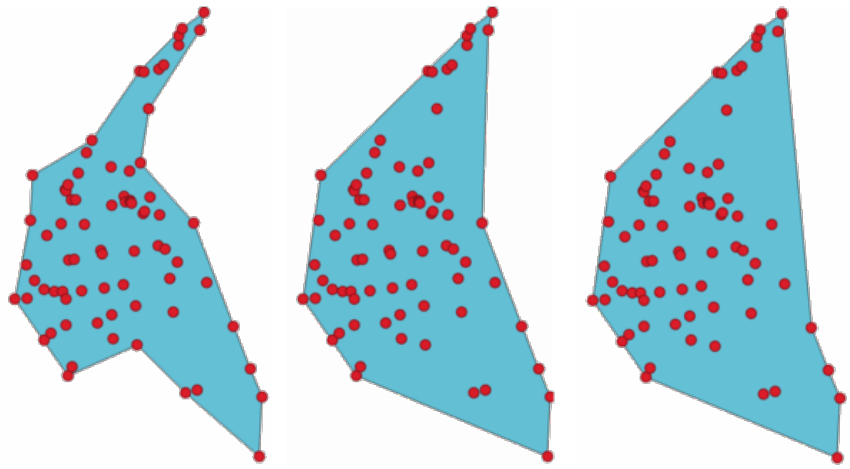


Fig. 24.101: Concave hulls with different thresholds (0.3, 0.6, 0.9)

See also

Convex hull

参数

标注	名称	类型	描述
Input point layer	`INPUT`	[vector: point]	Input point vector layer
Threshold	`ALPHA`	[numeric: double] Default: 0.3	Number from 0 (maximum concave hull) to 1 (convex hull).
Allow holes	`HOLES`	[boolean] Default: True	Choose whether to allow holes in the final concave hull
Split multipart geometry into singlepart geometries	`NO_MULTIGEOMET`	[boolean] Default: False	Check if you want to have singlepart geometries instead of multipart ones.

continues on next page

Table 24.619 – continued from previous page

标注	名称	类型	描述
Concave hull	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Concave hull	`OUTPUT`	[vector: polygon]	The output vector layer

Python 代码

Algorithm ID: `native:concavehull`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Convert geometry type

Generates a new layer based on an existing one, with a different type of geometry.

The attribute table of the output layer is the same as the one of the input layer.

Not all conversions are possible. For instance, a line layer can be converted to a point layer, but a point layer cannot be converted to a line layer.

See also

Polygonize, *Lines to polygons*, *Polygons to lines*, *Points to path*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层

continues on next page

Table 24.621 – continued from previous page

标注	名称	类型	描述
New geometry type	TYPE	[enumeration] Default: 0	Geometry type to apply to the output features. One of: • 0 --- Centroids • 1 --- Nodes • 2 --- Linestrings • 3 --- Multilinestrings • 4 --- Polygons
Converted	OUTPUT	[vector: geometry] Default: [Create temporary layer]	Specify the output vector layer. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Converted	OUTPUT	[vector: geometry]	Output vector layer - the type depends on the parameters

Python 代码

Algorithm ID: native:convertgeometrytype

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Convert to curved geometries

Converts a geometry into its curved geometry equivalent.

Already curved geometries will be retained without change.

Allows *features in-place modification* of line and polygon features

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line or polygon]	输入矢量图层
Maximum distance tolerance	`DISTANCE`	[numeric: double] Default: 0.000001	The maximum distance allowed between the original location of vertices and where they would fall on the converted curved geometries
Maximum angle tolerance	`ANGLE`	[numeric: double] Default: 0.000001	Segments are considered as suitable for replacing with an arc if the points are all regularly spaced on the candidate arc. This parameter specifies the maximum angular deviation (in degrees) allowed when testing for regular point spacing. Between 0 and 45°.
Curves	`OUTPUT`	[vector: compoundcurve or curvepolygon] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Curves	`OUTPUT`	[vector: compoundcurve or curvepolygon]	Output vector layer with curved geometries

Python 代码

Algorithm ID: `native:converttocurves`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Convex hull

Calculates the convex hull for each feature in an input layer.

See the 'Minimum bounding geometry' algorithm for a convex hull calculation which covers the whole layer or grouped subsets of features.

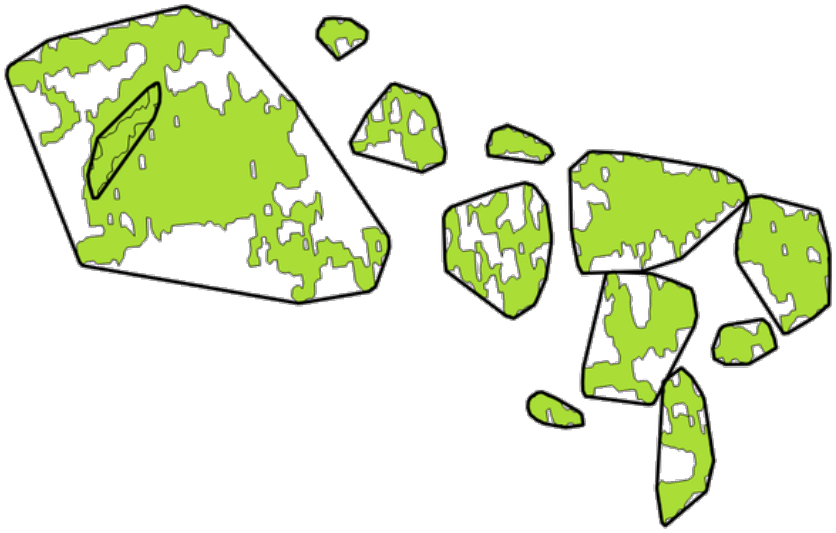


Fig. 24.102: Black lines identify the convex hull for each layer feature

Allows *features in-place modification* of polygon features

Default menu: *Vector* ► *Geoprocessing Tools*

See also

Minimum bounding geometry, *Concave hull*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Convex hull	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Convex hull	`OUTPUT`	[vector: polygon]	The output (convex hull) vector layer. Other than the input attributes, the output layer also contains the following fields: `area` and `perimeter` of the generated polygon.

Python 代码

Algorithm ID: `native:convexhull`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Create wedge buffers

Creates wedge shaped buffers from input points.

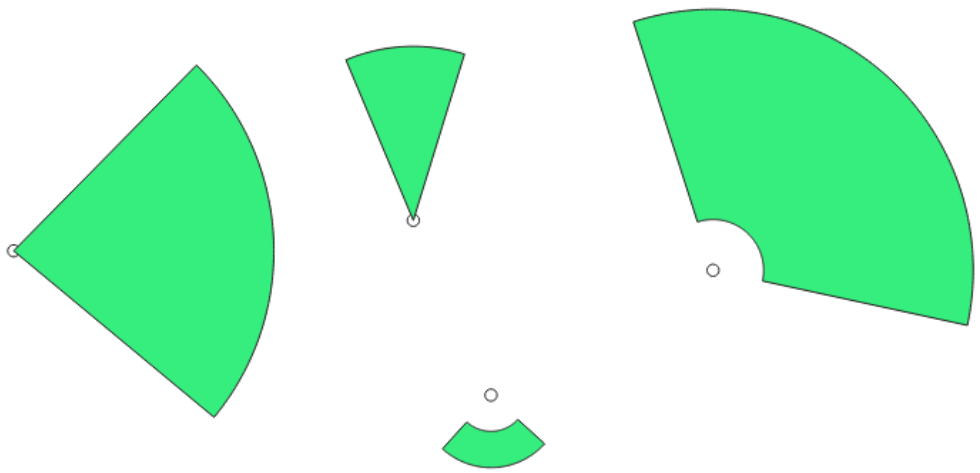


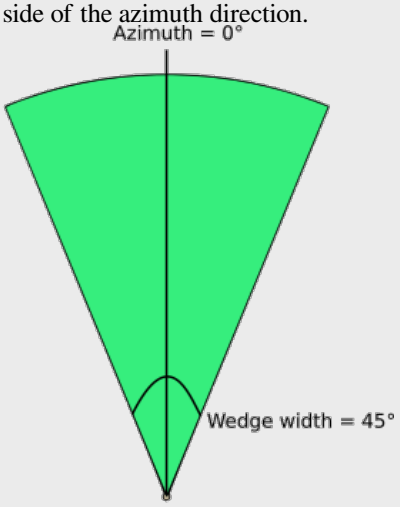
Fig. 24.103: Wedge buffers

The native output from this algorithm are CurvePolygon geometries, but these may be automatically segmentized to Polygons depending on the output format.

See also

Buffer, *Variable width buffer (by M value)*, *Tapered buffers*

参数

标注	名称	类型	描述
输入图层	INPUT	[vector: point]	Input point vector layer
Azimuth (degrees from North)	AZIMUTH	[numeric: double] Default: 0.0	Angle (in degrees) as the middle value of the wedge
Wedge width (in degrees)	WIDTH	[numeric: double] Default: 45.0	Width (in degrees) of the buffer. The wedge will extend to half of the angular width either side of the azimuth direction.  Fig. 24.104: Azimuth and width values of the wedge buffer
Outer radius	OUTER_RADIUS	[numeric: double] 默认值：1.0	The outer *size* (length) of the wedge: the size is meant from the source point to the edge of the wedge shape.
Inner radius 可选	INNER_RADIUS	[numeric: double] Default: 0.0	Inner radius value. If 0 the wedge will begin from the source point.
Buffers	OUTPUT	[vector: polygon] Default: [Create temporary layer]	Specify the output vector layer. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Buffers	OUTPUT	[vector: polygon]	The output (wedge buffer) vector layer

Python 代码

Algorithm ID: `native:wedgebuffers`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Delaunay triangulation

Creates a polygon layer with the Delaunay triangulation corresponding to the input point layer.

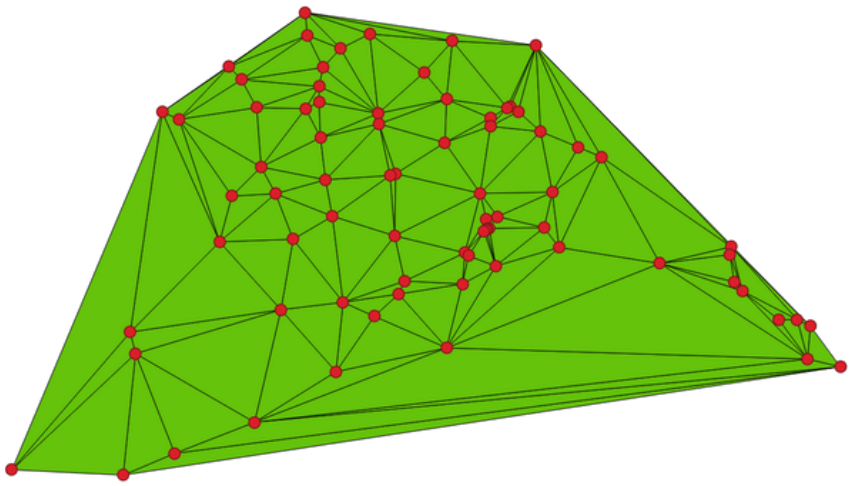


Fig. 24.105: Delaunay triangulation on points

Default menu: *Vector ► Geometry Tools*

参数

标注	名称	类型	描述
输入图层	INPUT	[vector: point]	Input point vector layer
Tolerance 可选	TOLERANCE	[numeric: double] Default: 0.0	Specifies an optional snapping tolerance which can be used to improve the robustness of the triangulation.

continues on next page

Table 24.629 – continued from previous page

标注	名称	类型	描述
Add point IDs to output	`ADD_ATTRIBUTES`	[boolean] Default: True	Specifies whether fields storing involved point features ID should be added to the output. If False, an `id` field is used to identify the polygons.
Delaunay triangulation	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Delaunay triangulation	`OUTPUT`	[vector: polygon]	The output (Delaunay triangulation) vector layer

Python 代码

Algorithm ID: `native:delaunaytriangulation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Delete holes

Takes a polygon layer and removes holes in polygons. It creates a new vector layer in which polygons with holes have been replaced by polygons with only their external ring. Attributes are not modified.

An optional minimum area parameter allows removing only holes which are smaller than a specified area threshold. Leaving this parameter at `0.0` results in all holes being removed.

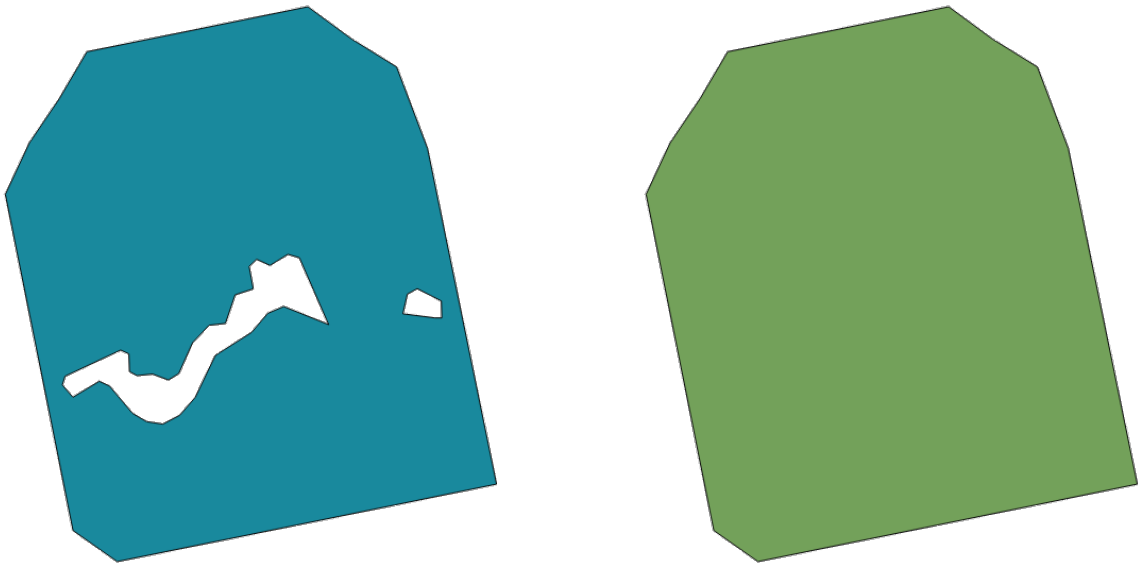


Fig. 24.106: Before and after the cleaning

Allows *features in-place modification* of polygon features

See also

qgisfixgeometryhole

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: polygon]	Input polygon vector layer
Remove holes with area less than 可选	`MIN_AREA`	[numeric: double] Default: 0.0	Only holes with an area less than this threshold will be deleted. With a value of `0.0`, **all** the holes will be deleted.
Cleaned	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Cleaned	`OUTPUT`	[same as input]	The output (cleaned) vector layer

Python 代码

Algorithm ID: `native:deleteholes`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Densify by count

Takes a polygon or line layer and generates a new one in which the geometries have a larger number of vertices than the original one.

If the geometries have Z or M values present then these will be linearly interpolated at the added vertices.

The number of new vertices to add to each segment is specified as an input parameter.

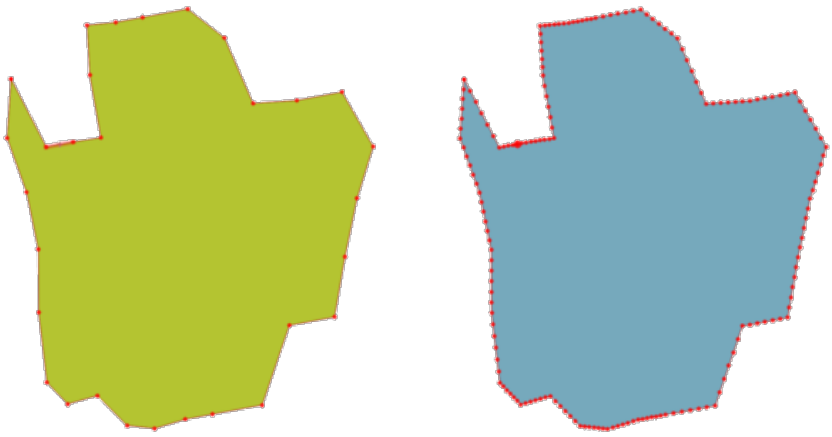


Fig. 24.107: Red points show the vertices before and after the densify

Allows *features in-place modification* of line and polygon features

Default menu: *Vector ► Geometry Tools*

See also

Densify by interval

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line, polygon]	Input line or polygon vector layer
Vertices to add	`VERTICES`	[numeric: integer] Default: 1	Number of vertices to add to each segment
Densified	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Densified	`OUTPUT`	[same as input]	The output (densified) vector layer

Python 代码

Algorithm ID: `native:densifygeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Densify by interval

Takes a polygon or line layer and generates a new one in which the geometries have a larger number of vertices than the original one.

The geometries are densified by adding regularly placed extra vertices inside each segment so that the maximum distance between any two vertices does not exceed the specified distance.

If the geometries have Z or M values present then these will be linearly interpolated at the added vertices.

Example

Specifying a distance of 3 would cause the segment `[0 0] -> [10 0]` to be converted to `[0 0] -> [2.5 0] -> [5 0] -> [7.5 0] -> [10 0]`, since 3 extra vertices are required on the segment and spacing these at 2.5 increments allows them to be evenly spaced over the segment.

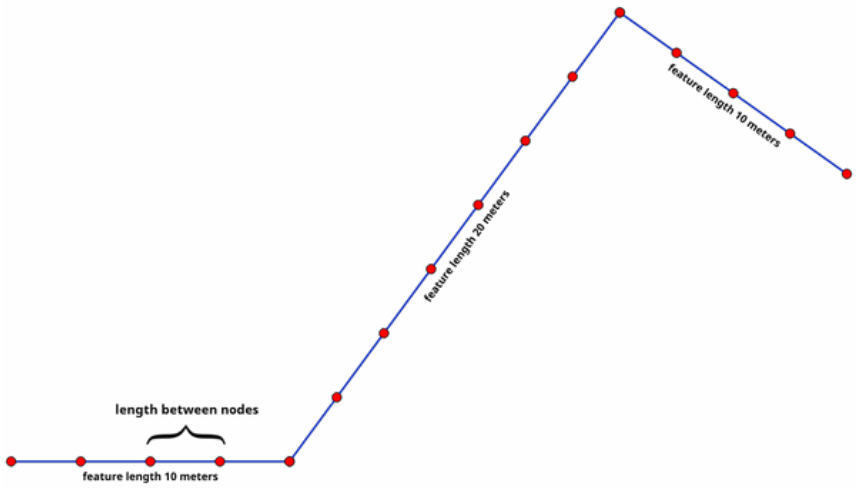


Fig. 24.108: Densify geometry at a given interval

Allows *features in-place modification* of line and polygon features

See also

Densify by count

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line, polygon]	Input line or polygon vector layer
Interval between vertices to add	`INTERVAL`	[numeric: double] 默认值：1.0	Maximum distance between two consecutive vertices
Densified	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Densified	`OUTPUT`	[same as input]	The output (densified) vector layer

Python 代码

Algorithm ID: `native:densifygeometriesgivenaninterval`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

融合（dissolve）

Takes a vector layer and combines its features into new features. One or more attributes can be specified to dissolve features belonging to the same class (having the same value for the specified attributes), alternatively all features can be dissolved to a single feature.

All output geometries will be converted to multi geometries. In case the input is a polygon layer, common boundaries of adjacent polygons being dissolved will get erased. If enabled, the optional "Keep disjoint features separate" setting will cause features and parts that do not overlap or touch to be exported as separate features (instead of parts of a single multipart feature).

The resulting attribute table will have the same fields as the input layer. The values in the output layer's fields are the ones of the first input feature that happens to be processed.

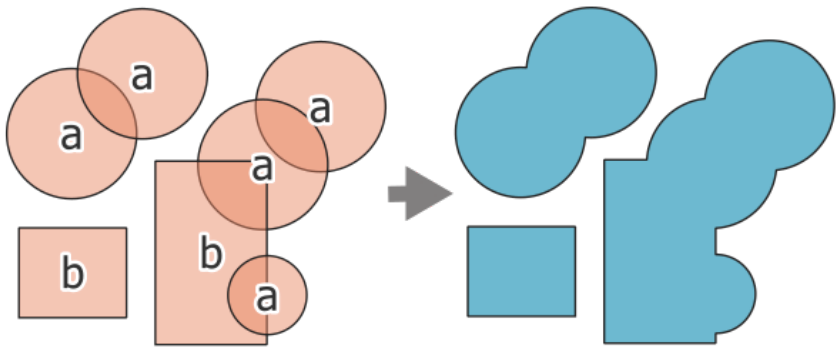


Fig. 24.109: Dissolving a layer into a single multipart feature

Default menu: *Vector* ► *Geoprocessing Tools*

Warning

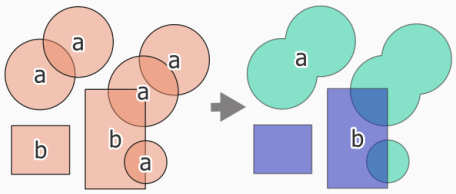
This algorithm drops existing primary keys or FID values and regenerates them in output layers.

See also

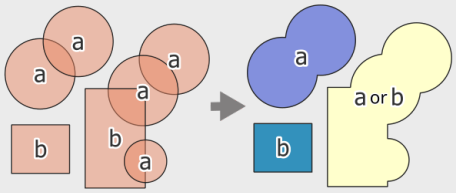
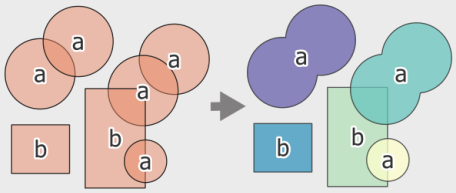
Dissolve coverage, *Aggregate*, *Collect geometries*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Dissolve field(s) 可选	`FIELD`	[tablefield: any] [list] Default: []	Features having the same value for the selected field(s) will be replaced with a single one and their geometries are merged. If no field is provided then all the features are dissolved, resulting in a single (multipart) feature.  Fig. 24.110: Dissolve the polygon layer on a common attribute (2 multipart features)
Dissolved	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Keep disjoint features separate	SEPARATE_DISJO	[boolean] Default: False	Parts of dissolved features are exported as separate features (instead of parts of a multipart feature).  Fig. 24.111: source (left), dissolve all (3 distinct features - right)  Fig. 24.112: source (left), dissolve on field (5 distinct features - right)

输出

标注	名称	类型	描述
Dissolved	OUTPUT	[same as input]	The output vector layer with dissolved geometries

Python 代码

Algorithm ID: `native:dissolve`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Drape (set Z value from raster)

Uses values sampled from a band within a raster layer to set the Z value for every overlapping vertex in the feature geometry. The raster values can optionally be scaled by a preset amount.

If Z values already exist in the layer, they will be overwritten with the new value. If no Z values exist, the geometry will be upgraded to include the Z dimension.

Allows *features in-place modification* of point, line, and polygon features with Z enabled

See also

Set M value from raster, *Set Z value*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Raster layer	`RASTER`	[栅格]	Raster layer with Z values
Band number	`BAND`	[raster band] Default: 1	The raster band to take the Z values from
Value for No-Data or non-intersecting vertices	`NODATA`	[numeric: double] Default: 0.0	Value to use in case the vertex does not intersect (a valid pixel of) the raster
Scale factor	`SCALE`	[numeric: double] 默认值：1.0	Scaling value: the band values are multiplied by this value.
Offset	`OFFSET`	[numeric: double] Default: 0.0	Offset value: it is algebraically added to the band values after applying the "Scale factor".
Updated	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer (with Z values from the raster layer). *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Updated	`OUTPUT`	[same as input]	The output vector layer with Z values from the raster layer

Python 代码

Algorithm ID: `native:setzfromraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Drop M/Z values

Removes M (measure) or Z (altitude) values from input geometries.

See also

Set M value, *Set Z value*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	Input vector layer with M or Z values
Drop M Values	`DROP_M_VALUES`	[boolean] Default: False	Removes the M values from the geometries
Drop Z Values	`DROP_Z_VALUES`	[boolean] Default: False	Removes the Z values from the geometries
Z/M Dropped	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Z/M Dropped	`OUTPUT`	[same as input]	The output vector layer (identical to the input layer, except that the M and/or Z dimensions have been removed from the geometries).

Python 代码

Algorithm ID: `native:dropmzvalues`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Eliminate selected polygons

Combines selected polygons of the input layer with certain adjacent polygons by erasing their common boundary. The adjacent polygon can be either the one with the largest or smallest area or the one sharing the largest common boundary with the polygon to be eliminated.

Eliminate is normally used to get rid of sliver polygons, i.e. tiny polygons that are a result of polygon intersection processes where boundaries of the inputs are similar but not identical.

Default menu: *Vector ► Geoprocessing Tools*

See also

Fix geometries

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: polygon]	Input polygon vector layer
Merge selection with the neighboring polygon with the	`MODE`	[enumeration] Default: Not set	Choose the parameter to use in order to get rid of the selected polygons: • 0 --- Largest Area • 1 --- Smallest Area • 2 --- Largest Common Boundary
Eliminated	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Eliminated	`OUTPUT`	[vector: polygon]	The output polygon vector layer.

Python 代码

Algorithm ID: `qgis:eliminateselectedpolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Explode lines

Takes a lines layer and creates a new one in which each line layer is replaced by a set of lines representing the segments in the original line.

Each line in the resulting layer contains only a start and an end point, with no intermediate vertices between them.

If the input layer consists of CircularStrings or CompoundCurves, the output layer will be of the same type and contain only single curve segments.

Note

- This algorithm drops existing primary keys or FID values and regenerates them in output layers.
- This algorithm does not require valid geometries as input.

Fig. 24.113: The original line layer and the exploded one

Allows *features in-place modification* of line features

See also

Subdivide, *Line substring*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line]	Input line vector layer
Exploded	`OUTPUT`	[vector: line] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Exploded	`OUTPUT`	[vector: line]	The output line vector layer with features representing each segment of the input layer.

Python 代码

Algorithm ID: `native:explodelines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Extend lines

Extends line geometry by a specified amount at the start and end of the line.

Lines are extended using the bearing of the first and last segment in the line.

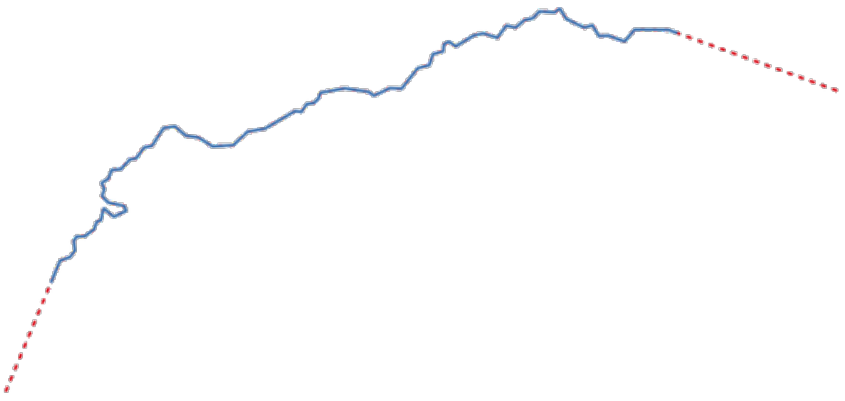


Fig. 24.114: The red dashes represent the initial and final extension of the original layer

Allows *features in-place modification* of line features

See also

Line substring

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line]	Input line vector layer
Start distance	`START_DISTANCE`	[numeric: double] Default: 0.0	Distance by which to extend the first segment of the line (starting point)
End distance	`END_DISTANCE`	[numeric: double] Default: 0.0	Distance by which to extend the last segment of the line (ending point)
Extended	`OUTPUT`	[vector: line] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Extended	`OUTPUT`	[vector: line]	The output (extended) line vector layer.

Python 代码

Algorithm ID: `native:extendlines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Extract M values

Extracts M values from geometries into feature attributes.

By default only the M value from the first vertex of each feature is extracted, however the algorithm can optionally calculate statistics on all of the geometry's M values, including sum, mean, minimum and maximum.

See also

Extract Z values, *Set M value*, *Drop M/Z values*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Summaries to calculate	`SUMMARIES`	[enumeration] Default: [0]	Statistics on the M values of a geometry. One or more of: • 0 --- First • 1 --- Last • 2 --- Count • 3 --- Sum • 4 --- Mean • 5 --- Median • 6 --- St.dev (pop) • 7 --- Minimum • 8 --- Maximum • 9 --- Range • 10 --- Minority • 11 --- Majority • 12 --- Variety • 13 --- Q1 • 14 --- Q3 • 15 --- IQR
Output column prefix	`COLUMN_PREFIX`	[string] Default: 'm_'	The prefix for the output (M) column
Extracted	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Extracted	OUTPUT	[same as input]	The output vector layer (with M values)

Python 代码

Algorithm ID: native:extractmvalues

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Extract specific vertices

Takes a vector layer and generates a point layer with points representing specific vertices in the input geometries.

For instance, this algorithm can be used to extract the first or last vertices in the geometry. The attributes associated to each point are the same ones associated to the feature that the vertex belongs to.

The vertex indices parameter accepts a comma separated string specifying the indices of the vertices to extract. The first vertex corresponds to an index of 0, the second vertex has an index of 1, etc. Negative indices can be used to find vertices at the end of the geometry, e.g., an index of -1 corresponds to the last vertex, -2 corresponds to the second last vertex, etc.

Additional fields are added to the vertices indicating the specific vertex position (e.g., 0, -1, etc), the original vertex index, the vertex ' s part and its index within the part (as well as its ring for polygons), distance along the original geometry and bisector angle of vertex for the original geometry.

Allows *features in-place modification* of point features

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

See also

Extract vertices, *Filter vertices by M value*, *Filter vertices by Z value*

参数

标注	名称	类型	描述
输入图层	INPUT	[vector: geometry]	输入矢量图层
Vertex indices	VERTICES	[string] Default: '0'	Comma-separated string of the indices of the vertices to extract.

continues on next page

Table 24.652 – continued from previous page

标注	名称	类型	描述
Vertices	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Vertices	`OUTPUT`	[vector: point]	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 代码

Algorithm ID: `native:extractspecificvertices`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Extract vertices

Takes a vector layer and generates a point layer with points representing the vertices in the input geometries.

The attributes associated to each point are the same ones associated to the feature that the vertex belongs to.

Additional fields are added to the vertices indicating the vertex index (beginning at 0), the feature' s part and its index within the part (as well as its ring for polygons), distance along original geometry and bisector angle of vertex for original geometry.

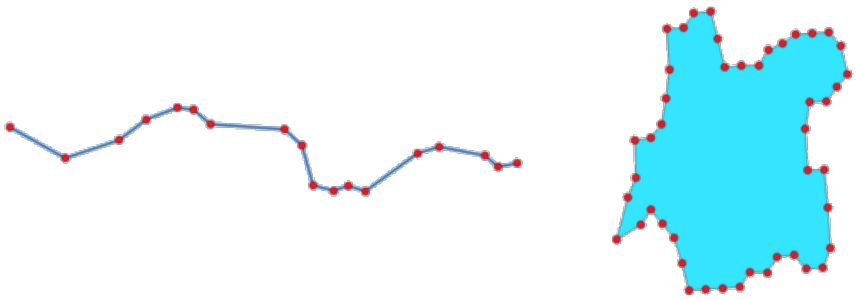


Fig. 24.115: Vertices extracted for line and polygon layer

Allows *features in-place modification* of point features

Default menu: *Vector* ► *Geometry Tools*

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

See also

Extract specific vertices, *Filter vertices by M value*, *Filter vertices by Z value*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层

continues on next page

Table 24.654 – continued from previous page

标注	名称	类型	描述
Vertices	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Vertices	`OUTPUT`	[vector: point]	The output (point) vector layer containing the vertices from the input layer geometries. Other than the input attributes, the output layer also contains the following fields: • `vertex_index`: the index of the vertex within the geometry. The first vertex is indexed as 0. • `vertex_part`: the part number of the geometry that the vertex belongs to. For single-part geometries, this is always 0. For multi-part geometries (e.g. MultiPolygon or MultiLine), this indicates which part the vertex comes from. • `vertex_part_ring`: for polygon geometries, this indicates whether the vertex belongs to the outer ring (0) or one of the inner rings (1, 2, ...). • `vertex_part_index`: the index of the vertex in its specific ring (outer or inner). • `distance`: the distance from the previous vertex. • `angle`: returns approximate angle at a vertex. Usually the average angle between adjacent segments, and can be pictured as the orientation of a line following the curvature of the geometry at the specified vertex.

Python 代码

Algorithm ID: `native:extractvertices`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Extract Z values

Extracts Z values from geometries into feature attributes.

By default only the Z value from the first vertex of each feature is extracted, however the algorithm can optionally calculate statistics on all of the geometry's Z values, including sum, mean, minimum and maximum.

See also

Extract M values, *Set Z value*, *Drop M/Z values*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Summaries to calculate	`SUMMARIES`	[enumeration] Default: [0]	Statistics on the Z values of a geometry. One or more of: • 0 --- First • 1 --- Last • 2 --- Count • 3 --- Sum • 4 --- Mean • 5 --- Median • 6 --- St.dev (pop) • 7 --- Minimum • 8 --- Maximum • 9 --- Range • 10 --- Minority • 11 --- Majority • 12 --- Variety • 13 --- Q1 • 14 --- Q3 • 15 --- IQR
Output column prefix	`COLUMN_PREFIX`	[string] Default: 'z_'	The prefix for the output (Z) column

continues on next page

Table 24.656 – continued from previous page

标注	名称	类型	描述
Extracted	OUTPUT	[same as input] Default: [Create temporary layer]	Specify the output layer. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Extracted	OUTPUT	[same as input]	The output vector layer (with Z values)

Python 代码

Algorithm ID: native:extractzvalues

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Filter vertices by M value

Filters away vertices based on their M value, returning geometries with only vertex points that have a M value greater than or equal to the specified minimum value and/or less than or equal to the maximum value.

If the minimum value is not specified then only the maximum value is tested, and similarly if the maximum value is not specified then only the minimum value is tested.

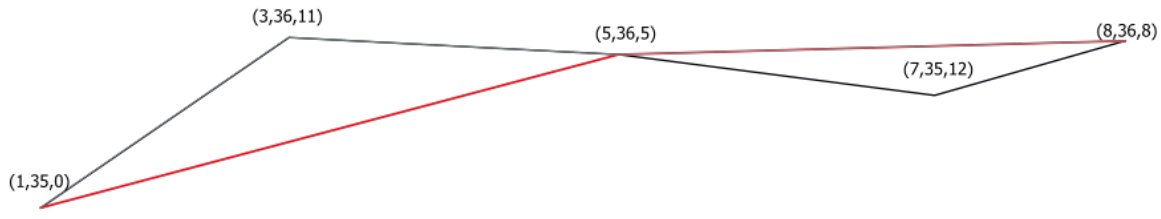


Fig. 24.116: The red line represents the black line with only vertices whose M value is <=10.

Allows *features in-place modification* of point, line and polygon features with M enabled

Warning

Depending on the input geometry attributes and the filters used, the geometries created by this algorithm may no longer be valid.

See also

Filter vertices by Z value, *Extract vertices*, *Extract specific vertices*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	Input vector layer to remove vertices from
Minimum 可选	`MIN`	[numeric: double] Default: Not set	Minimum of M values allowed
Maximum 可选	`MAX`	[numeric: double] Default: Not set	Maximum of M values allowed
Filtered	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Filtered	`OUTPUT`	[same as input]	The output vector layer of features with only the filtered vertices.

Python 代码

Algorithm ID: `native:filterverticesbym`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Filter vertices by Z value

Filters away vertices based on their Z value, returning geometries with only vertex points that have a Z value greater than or equal to the specified minimum value and/or less than or equal to the maximum value.

If the minimum value is not specified then only the maximum value is tested, and similarly if the maximum value is not specified then only the minimum value is tested.

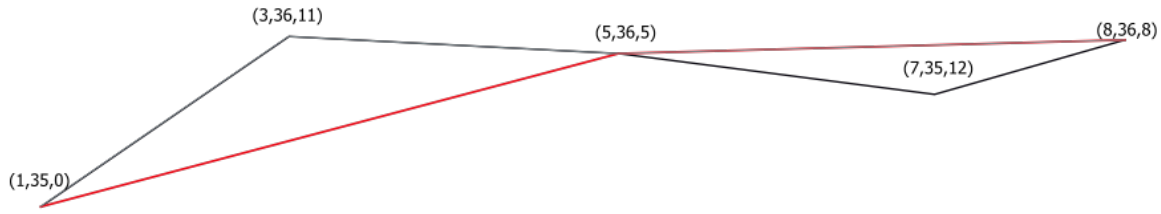


Fig. 24.117: The red line represents the black line with only vertices whose Z value is <=10.

Allows *features in-place modification* of point, line and polygon features with Z enabled

Warning

Depending on the input geometry attributes and the filters used, the geometries created by this algorithm may no longer be valid. You may need to run the *Fix geometries* algorithm to ensure their validity.

See also

Filter vertices by M value, *Extract vertices*, *Extract specific vertices*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	Input vector layer to remove vertices from
Minimum 可选	`MIN`	[numeric: double] Default: Not set	Minimum of Z values allowed
Maximum 可选	`MAX`	[numeric: double] Default: Not set	Maximum of Z values allowed
Filtered	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Filtered	`OUTPUT`	[same as input]	The output vector layer of features with only the filtered vertices.

Python 代码

Algorithm ID: `native:filterverticesbyz`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Fix geometries

Attempts to create a valid representation of a given invalid geometry without losing any of the input vertices. Already valid geometries are returned without further intervention. Always outputs multi-geometry layer.

Allows *features in-place modification* of point, line, and polygon features without M enabled

Warning

M values will be dropped from the output.

See also

Check validity, *Fix Geometry*

continues on next page

Table 24.662 – continued from previous page

标注	名称	类型	描述

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Repair method	`METHOD`	[enumeration] Default: 1	Method used to repair the geometries. One of: • 0 --- `Linework`: combines all rings into a set of noded lines and then extracts valid polygons from that linework • 1 --- `Structure`: first makes all rings valid and then merges shells and subtracts holes from shells to generate valid result. Assumes that holes and shells are correctly categorized. **Attention** Using this method requires QGIS installed with GEOS >= 3.10 (see *Help* ► *About* menu).
Fixed geometries	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Fixed geometries	`OUTPUT`	[same as input]	The output vector layer with fixed geometries.

Python 代码

Algorithm ID: `native:fixgeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Force right-hand-rule

Forces polygon geometries to respect the Right-Hand-Rule, in which the area that is bounded by a polygon is to the right of the boundary. In particular, the exterior ring is oriented in a clockwise direction and any interior rings in a counter-clockwise direction.

Allows *features in-place modification* of polygon features

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: polygon]	输入矢量图层
Reoriented	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Reoriented	`OUTPUT`	[vector: polygon]	The output vector layer with reoriented geometries.

Python 代码

Algorithm ID: `native:forcerhr`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Geodesic line split at antimeridian

Splits a line into multiple geodesic segments, whenever the line crosses the antimeridian (±180 degrees longitude).

Splitting at the antimeridian helps the visual display of the lines in some projections. The returned geometry will always be a multi-part geometry.

Whenever line segments in the input geometry cross the antimeridian, they will be split into two segments, with the latitude of the breakpoint being determined using a geodesic line connecting the points either side of this segment. The current project ellipsoid setting will be used when calculating this breakpoint.

If the input geometry contains M or Z values, these will be linearly interpolated for the new vertices created at the antimeridian.

> **Note**
>
> This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

Allows *features in-place modification* of line features

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line]	Input line vector layer
Split	`OUTPUT`	[vector: line] Default: `[Create temporary layer]`	Specify the output line vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Split	`OUTPUT`	[vector: line]	The output line vector layer split at the antimeridian.

Python 代码

Algorithm ID: `native:antimeridiansplit`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Geometry by expression

Updates existing geometries (or creates new geometries) for input features by use of a QGIS expression.

This allows complex geometry modifications which can utilize all the flexibility of the QGIS expression engine to manipulate and create geometries for output features.

For help with QGIS expression functions, see the inbuilt help available in the *expression builder*.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	输入矢量图层
Output geometry type	`OUTPUT_GEOMETR`	[enumeration] Default: 0	The output geometry strongly depends on the expression: for instance, if you create a buffer the geometry type has to be polygon. One of: • 0 --- Polygon • 1 --- Line • 2 --- Point
Output geometry has z values	`WITH_Z`	[boolean] Default: False	Choose if the output geometry should include the Z dimension
Output geometry has m values	`WITH_M`	[boolean] Default: False	Choose if the output geometry should include the M dimension
Geometry expression	`EXPRESSION`	[expression] Default: '$geometry'	Add the geometry expression you want to use. You can use the button to open the Expression Dialog. The dialog lists all the relevant expressions, together with their help and guide.
Modified geometry	`OUTPUT`	[vector: geometry] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Modified geometry	`OUTPUT`	[vector: geometry]	The output vector layer

Python 代码

Algorithm ID: `native:geometrybyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，*算法 id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Interpolate point on line

Creates a point geometry interpolated at a set distance along line or curve geometries.

Z and M values are linearly interpolated from existing values.

If a multipart geometry is encountered, only the first part is considered when calculating the substring.

If the specified distance is greater than the input feature's length, the resultant feature will have a null geometry.

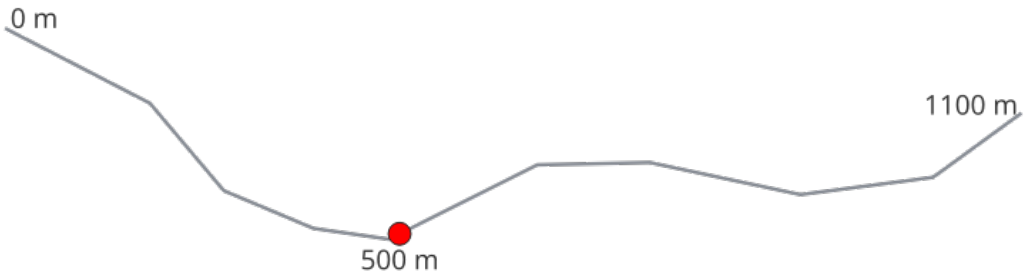


Fig. 24.118: Interpolated point at 500m of the beginning of the line

See also

Points along geometry

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line, polygon]	Input line or polygon vector layer
Distance	`DISTANCE`	[numeric: double] Default: 0.0	Distance from the beginning of the line
Interpolated points	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Interpolated points	`OUTPUT`	[vector: point]	The output point vector layer with features at a set distance along the line or polygon boundary

Python 代码

Algorithm ID: `native:interpolatepoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Keep N biggest parts

Takes a layer with polygons or multipolygons and returns a new layer in which only the *n* largest polygons of each multipolygon feature are kept. If a feature has *n* or fewer parts, the feature will just be copied.

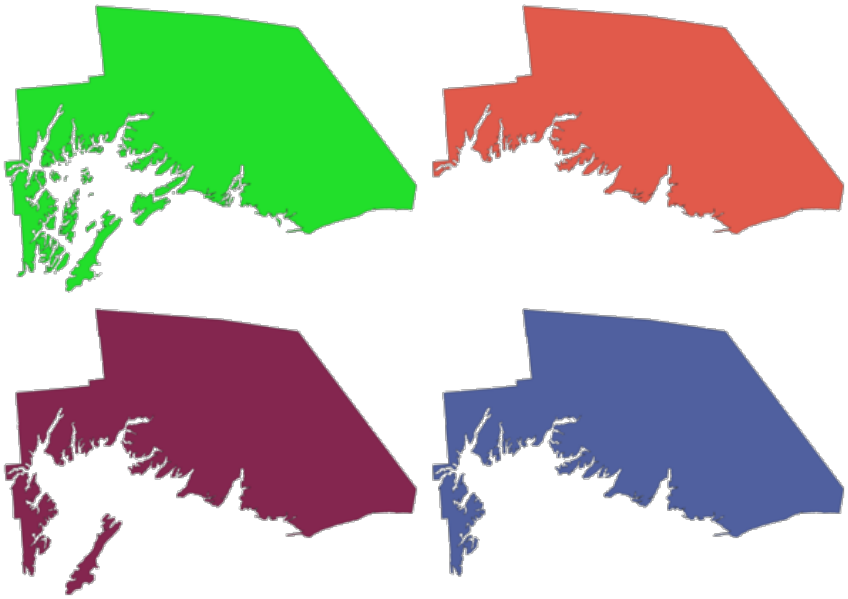


Fig. 24.119: Clockwise from top left: original multipart feature, one, two and three biggest parts kept

参数

标注	名称	类型	描述
Polygons	`INPUT`	[vector: polygon]	Input polygon vector layer
Parts to keep	`PARTS`	[numeric: integer] Default: 1	Number of parts to keep. If 1, only the biggest part of the feature will be kept.
Parts	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output polygon vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Parts	`OUTPUT`	[vector: polygon]	The output polygon vector layer with the N biggest parts of each feature

Python 代码

Algorithm ID: `native:keepnbiggestparts`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Line substring

Returns the portion of a line (or curve) which falls between the specified start and end distances (measured from the beginning of the line).

Z and M values are linearly interpolated from existing values.

If a multipart geometry is encountered, only the first part is considered when calculating the substring.

Fig. 24.120: Substring line with starting distance set at 0 meters and the ending distance at 250 meters.

Allows *features in-place modification* of line features

See also

Extend lines

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line]	Input line vector layer
Start distance	`START_DISTANCE`	[numeric: double] Default: 0.0	Distance along the input line to the start point of the output feature
End distance	`END_DISTANCE`	[numeric: double] 默认值：1.0	Distance along the input line to the end point of the output feature
Substring	`OUTPUT`	[vector: line] Default: `[Create temporary layer]`	Specify the output line vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Substring	`OUTPUT`	[vector: line]	The output line vector layer.

Python 代码

Algorithm ID: `native:linesubstring`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Lines to polygons

Generates a polygon layer using as polygon rings the lines from an input line layer.

The attribute table of the output layer is the same as the one of the input layer.

Default menu: *Vector* ► *Geometry Tools*

See also

Polygons to lines, *Polygonize*, *Convert geometry type*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line]	Input line vector layer
Polygons	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output polygon vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Polygons	`OUTPUT`	[vector: polygon]	The output polygon vector layer.

Python 代码

Algorithm ID: `qgis:linestopolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Merge lines

Joins all connected parts of MultiLineString geometries into single LineString geometries.

If any parts of the input MultiLineString geometries are not connected, the resultant geometry will be a MultiLineString containing any lines which could be merged and any non-connected line parts.

Allows *features in-place modification* of line features

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line]	Input line vector layer
Merged	`OUTPUT`	[vector: line] Default: `[Create temporary layer]`	Specify the output line vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Merged	`OUTPUT`	[vector: line]	The output (merged) line vector layer.

Python 代码

Algorithm ID: `native:mergelines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Minimum bounding geometry

Creates geometries which enclose the features from an input layer. The features can be grouped by a field. The output layer will then contain one feature per group value with a geometry (MBB) that covers the geometries of the features with matching value.

The following enclosing geometry types are supported:

- bounding box (envelope)
- oriented rectangle
- circle
- convex hull

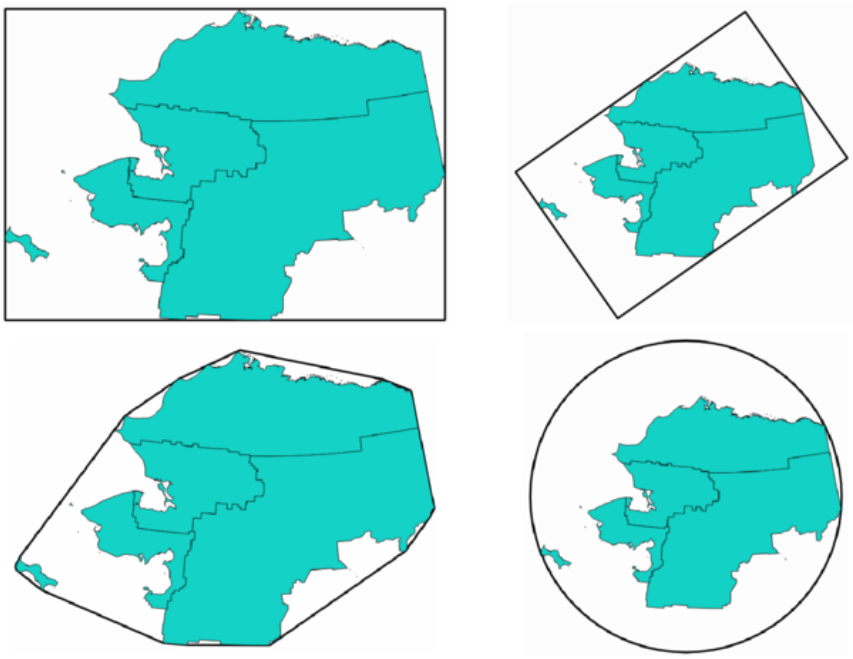


Fig. 24.121: Clockwise from top left: envelope, oriented rectangle, circle, convex hull

See also

Minimum enclosing circles

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
字段 可选	`FIELD`	[tablefield: any]	Features can be grouped by a field. If set, this causes the output layer to contain one feature per grouped value with a minimal geometry covering only the features with matching values.
Geometry type	`TYPE`	[enumeration] Default: 0	Enclosing geometry types. One of: • 0 --- Envelope (Bounding Box) • 1 --- Minimum Oriented Rectangle • 2 --- Minimum Enclosing Circle • 3 --- Convex Hull
Bounding geometry	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output polygon vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Bounding geometry	`OUTPUT`	[vector: polygon]	The output (bounding) polygon vector layer. It has the fields from the input layer and, depending on the selected parameter `TYPE`, the output also has the attributes: • `width`, `height`, `area` and `perimeter` of the Envelope (Bounding Box) • `width`, `height`, `area`, `angle` and `perimeter` of the Minimum Oriented Rectangle • `radius` and `area` of the Minimum Enclosing Circle • `area` and `perimeter` of the Convex Hull

Python 代码

Algorithm ID: `qgis:minimumboundinggeometry`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Minimum enclosing circles

Calculates the minimum enclosing circles of the features in the input layer.

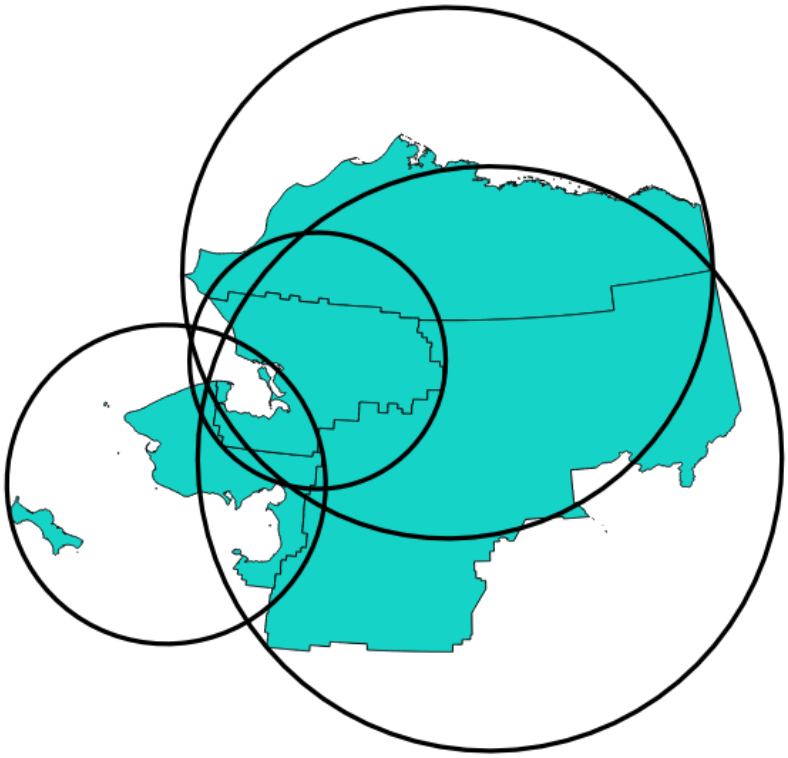


Fig. 24.122: Enclosing circles for each feature

Allows *features in-place modification* of polygon features

See also

Minimum bounding geometry

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Number of segments in circles	`SEGMENTS`	[numeric: integer] Default: 72	The number of segments used to approximate a circle. Minimum 8, maximum 100000.
Minimum enclosing circles	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output polygon vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Minimum enclosing circles	`OUTPUT`	[vector: polygon]	The output polygon vector layer. Other than the input attributes, the output layer also contains the following fields: `area` and `radius` of the generated polygon.

Python 代码

Algorithm ID: `native:minimumenclosingcircle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Multi-ring buffer (constant distance)

Computes multi-ring (*donut*) buffer for the features of the input layer, using a fixed or dynamic distance and number of rings.

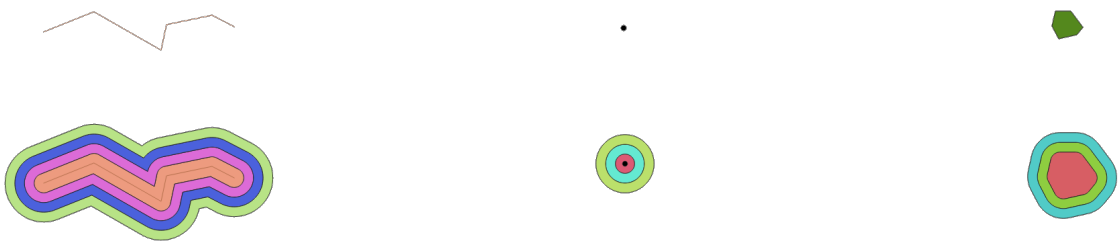


Fig. 24.123: Multi-ring buffer for a line, point and polygon layer

Allows *features in-place modification* of polygon features

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

See also

Buffer, *Variable distance buffer*, *Rectangles, ovals, diamonds*, *Single sided buffer*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Number of rings	`RINGS`	[numeric: integer] Default: 1	The number of rings. It can be a unique value (same number of rings for all the features) or it can be taken from features data (the number of rings depends on feature values).
Distance between rings	`DISTANCE`	[numeric: double] 默认值：1.0	Distance between the rings. It can be a unique value (same distance for all the features) or it can be taken from features data (the distance depends on feature values).
Multi-ring buffer (constant distance)	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output polygon vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Multi-ring buffer (constant distance)	`OUTPUT`	[vector: polygon]	The output polygon vector layer. Other than the input attributes, the output layer also contains the following fields: `ringId` and `distance` of the generated buffer polygon.

Python 代码

Algorithm ID: `native:multiringconstantbuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Multipart to singleparts

Splits multipart features in the input layer into singlepart features.

The attributes of the output layer are the same as the original ones but divided into single features.

Fig. 24.124: Left the multipart source layer and right the single part output result

Allows *features in-place modification* of point, line, and polygon features

Default menu: *Vector* ► *Geometry Tools*

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

See also

Collect geometries, *Promote to multipart*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Single parts	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output polygon vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Single parts	`OUTPUT`	[same as input]	The output vector layer.

Python 代码

Algorithm ID: `native:multiparttosingleparts`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Offset lines

Offsets lines by a specified distance. Positive distances will offset lines to the left, and negative distances will offset them to the right.

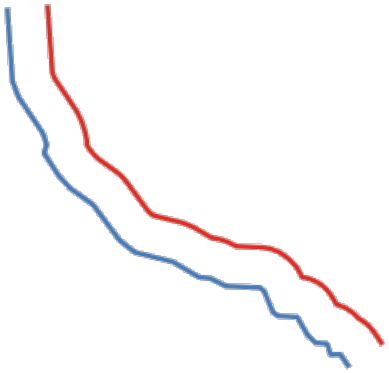


Fig. 24.125: In blue the source layer, in red the offset one

Allows *features in-place modification* of line features

See also

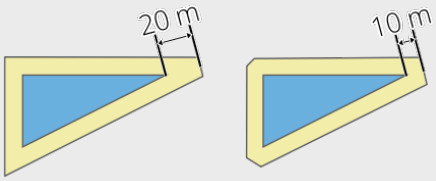
Array of offset (parallel) lines, 翻译

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line]	Input line vector layer
Distance	`DISTANCE`	[numeric: double] Default: 10.0	Offset distance. You can use the Data Defined button on the right to choose a field from which the radius will be calculated. This way you can have different radius for each feature (see *Variable distance buffer*).
Segments	`SEGMENTS`	[numeric: integer] Default: 8	Controls the number of line segments to use to approximate a quarter circle when creating rounded offsets.

continues on next page

Table 24.688 – continued from previous page

标注	名称	类型	描述
Join style	`JOIN_STYLE`	[enumeration] Default: 0	Specifies whether round, miter or beveled joins should be used when offsetting corners in a line. Options are: • 0 --- Round • 1 --- Miter • 2 --- Bevel Fig. 24.126: Round, miter, and bevel join styles
Miter limit	`MITER_LIMIT`	[numeric: double] Default: 2.0	Sets the maximum distance from the offset geometry to use when creating a mitered join as a factor of the offset distance (only applicable for miter join styles). Minimum: 1.0  Fig. 24.127: A 10m buffer with a limit of 2 and a 10m buffer with a limit of 1
Offset	`OUTPUT`	[vector: line] Default: `[Create temporary layer]`	Specify the output (offset) layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Offset	`OUTPUT`	[vector: line]	Output (offset) line layer

Python 代码

Algorithm ID: `native:offsetline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Oriented minimum bounding box

Calculates the minimum area rotated rectangle for each feature in the input layer.

Note

For singlepart point features, the output corresponds to the bounding box of the geometry.

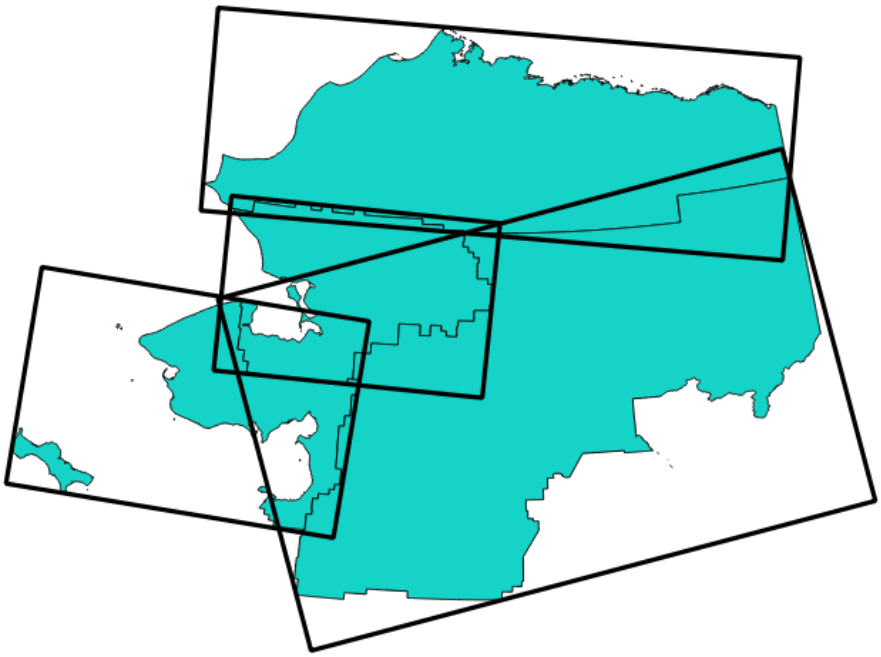


Fig. 24.128: Oriented minimum bounding box

Allows *features in-place modification* of polygon features

See also

Minimum bounding geometry

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Bounding boxes	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output polygon vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Bounding boxes	`OUTPUT`	[vector: polygon]	The output polygon vector layer. Other than the input attributes, the output layer also contains the following fields: `width`, `height`, `angle`, `area` and `perimeter` of the generated polygon.

Python 代码

Algorithm ID: `native:orientedminimumboundingbox`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Orthogonalize

Attempts to orthogonalize the geometries of the input line or polygon layer. This process shifts the vertices in the geometries to try to make every angle in the geometry either a right angle or a straight line.

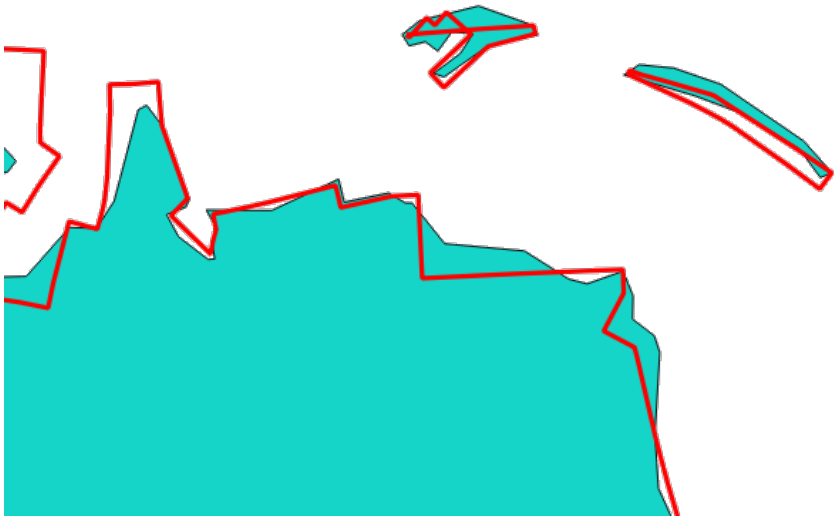


Fig. 24.129: In blue the source layer and in the red orthogonalized result

Allows *features in-place modification* of line and polygon features

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line, polygon]	Input line or polygon vector layer
Maximum angle tolerance (degrees)	`ANGLE_TOLERANC`	[numeric: double] Default: 15.0	Specify the maximum deviation from a right angle or straight line a vertex can have for it to be adjusted. Smaller tolerances mean that only vertices which are already closer to right angles will be adjusted, and larger tolerances mean that vertices which deviate further from right angles will also be adjusted.
Maximum algorithm iterations	`MAX_ITERATIONS`	[numeric: integer] Default: 1000	Setting a larger number for the maximum number of iterations will result in a more orthogonal geometry at the cost of extra processing time.
Orthogonalized	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output polygon vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Orthogonalized	`OUTPUT`	[same as input]	The output polygon vector layer with adjusted angles.

Python 代码

Algorithm ID: `native:orthogonalize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Point on Surface

For each feature of the input layer, returns a point that is guaranteed to lie on the surface of the feature geometry.

Allows *features in-place modification* of point features

Warning

This algorithm may drop existing primary keys or FID values and regenerate them in output layers, depending on the input parameters.

See also

Centroids

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Create point on surface for each part	`ANGLE_TOLERANC`	[boolean]	If checked, a point will be created for each part of the geometry.
Point	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the output point vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Point	`OUTPUT`	[vector: point]	The output point vector layer.

Python 代码

Algorithm ID: `native:pointonsurface`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Points along geometry

Creates points at regular intervals along line or polygon geometries. Created points will have new attributes added for the distance along the geometry and the angle of the line at the point.

An optional start and end offset can be specified, which controls how far from the start and end of the geometry the points should be created.

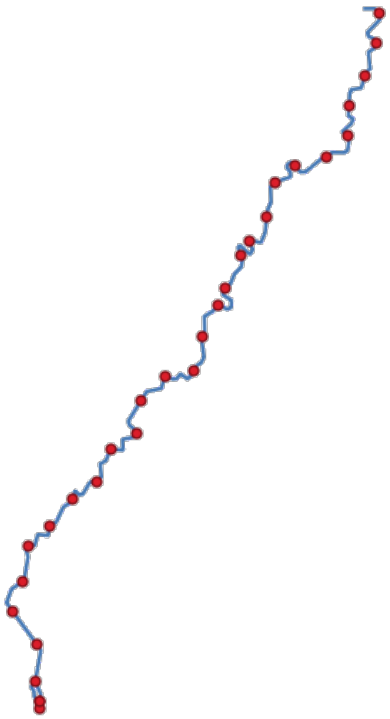


Fig. 24.130: Points created along the source line layer

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

See also

Interpolate point on line

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line, polygon]	Input line or polygon vector layer
Distance	`DISTANCE`	[numeric: double] 默认值：1.0	Distance between two consecutive points along the line
Start offset	`START_OFFSET`	[numeric: double] Default: 0.0	Distance from the beginning of the input line, representing the position of the first point.
End offset	`END_OFFSET`	[numeric: double] Default: 0.0	Distance from the end of the input line, representing the position beyond which no point feature shoud be created.
Interpolated points	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Interpolated points	`OUTPUT`	[vector: point]	Point vector layer with features placed along lines or polygon boundaries of the input layer. Other than the input attributes on every related point, the output layer also contains the following fields: `distance` and `angle` of the generated point.

Python 代码

Algorithm ID: `native:pointsalonglines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Points displacement

Given a distance of proximity, identifies nearby point features and radially distributes them over a circle whose center represents their barycenter. A convenient tool to scatter overlaid features.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: point]	Input point vector layer
Minimum distance to other points	`PROXIMITY`	[numeric: double] 默认值：1.0	Distance below which point features are considered close. Close features are distributed altogether.
Displacement distance	`DISTANCE`	[numeric: double] 默认值：1.0	Radius of the circle on which close features are placed
Horizontal distribution for two point case	`HORIZONTAL`	[boolean] Default: False	When only two points are identified as close, aligns them horizontally on the circle instead of vertically.
Displaced	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Displaced	`OUTPUT`	[vector: point]	Output point vector layer

Python 代码

Algorithm ID: `qgis:pointsdisplacement`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Pole of inaccessibility

Calculates the pole of inaccessibility for a polygon layer, which is the most distant internal point from the boundary of the surface.

This algorithm uses the 'polylabel' algorithm (Vladimir Agafonkin, 2016), which is an iterative approach guaranteed to find the true pole of inaccessibility within a specified tolerance. A more precise tolerance (lower value) requires more iterations and will take longer to calculate.

Fig. 24.131: Pole of inaccessibility

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: polygon]	输入矢量图层
Tolerance	`TOLERANCE`	[numeric: double] 默认值：1.0	Set the tolerance for the calculation
Point	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the output polygon vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Point	`OUTPUT`	[vector: point]	The output point vector layer. It will contain a new attribute with the distance `dist_pole` from the calculated pole to the polygon boundary.

Python 代码

Algorithm ID: `native:poleofinaccessibility`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Polygonize

Creates a polygon layer whose features boundaries are generated from a line layer of **closed** features.

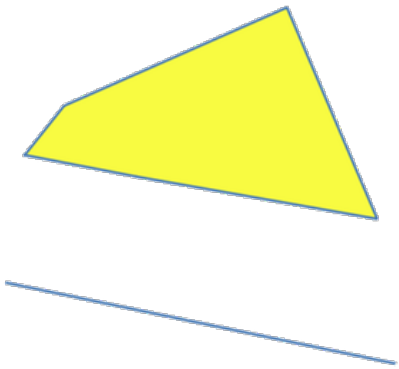


Fig. 24.132: The yellow polygons generated from the closed lines

Note

The line layer must have closed shapes in order to be transformed into a polygon.

See also

Polygons to lines, *Lines to polygons*, *Convert geometry type*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line]	Input line vector layer
Keep fields from the input layer	`KEEP_FIELDS`	[boolean] Default: False	Check to keep the fields (only the table structure, not the values) of the input layer

continues on next page

Table 24.702 – continued from previous page

标注	名称	类型	描述
Polygons from lines	OUTPUT	[vector: polygon] Default: [Create temporary layer]	Specify the output polygon vector layer. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Polygons from lines	OUTPUT	[vector: polygon]	The output polygon vector layer from lines

Python 代码

Algorithm ID: native:polygonize

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Polygons to lines

Takes a polygon layer and creates a line layer, with lines representing the boundaries of the polygons in the input layer.

The attribute table of the output layer is the same as the one of the input layer.

Fig. 24.133: Black lines as the result of the algorithm

Default menu: *Vector* ► *Geometry Tools*

See also

Lines to polygons, *Polygonize*, *Convert geometry type*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: polygon]	Input polygon vector layer
Lines	`OUTPUT`	[vector: line] Default: `[Create temporary layer]`	Specify the output line vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Lines	`OUTPUT`	[vector: line]	The output line vector layer from polygons

Python 代码

Algorithm ID: `native:polygonstolines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Project points (Cartesian)

Projects point geometries by a specified distance and bearing (azimuth).

Allows *features in-place modification* of point features

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: point]	Input point vector layer
Bearing (degrees from North)	`BEARING`	[numeric: double] Default: 0.0	Clockwise angle starting from North, in degree (°) unit
Distance	`DISTANCE`	[numeric: double] 默认值：1.0	Distance to offset geometries, in layer units

continues on next page

Table 24.706 – continued from previous page

标注	名称	类型	描述
Projected	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the output point vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Projected	`OUTPUT`	[vector: point]	The output (projected) point vector layer

Python 代码

Algorithm ID: `native:projectpointcartesian`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Promote to multipart

Takes a vector layer with singlepart geometries and generates a new one in which all geometries are multipart.

Input features which are already multipart features will remain unchanged.

This algorithm can be used to force geometries to multipart types in order to be compatible with data providers that require multipart features.

Allows *features in-place modification* of point, line, and polygon features

See also

Aggregate, *Collect geometries*, *Convert to strictly multipart*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Multiparts	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output multipart vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Multiparts	`OUTPUT`	[same as input]	The output multipart vector layer

Python 代码

Algorithm ID: `native:promotetomulti`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Rectangles, ovals, diamonds

Creates a buffer area with a rectangle, oval or diamond shape for each feature of the input point layer.

The shape parameters can be fixed for all features or dynamic using a field or an expression.

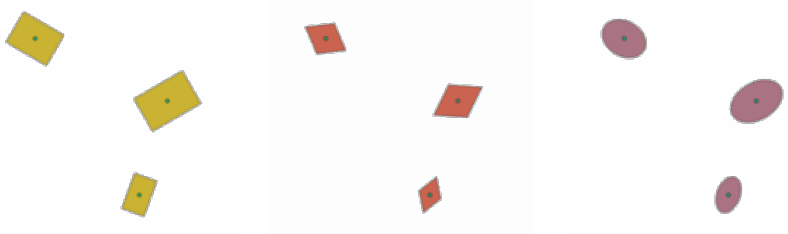


Fig. 24.134: Different buffer shapes with dynamic parameters

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: point]	Input point vector layer
Buffer shape	`SHAPE`	[enumeration]	The shape to use. One of: • 0 --- Rectangles • 1 --- Ovals • 2 --- Diamonds
Width	`WIDTH`	[numeric: double] 默认值：1.0	Width of the buffer shape
Height	`HEIGHT`	[numeric: double] 默认值：1.0	Height of the buffer shape
Rotation 可选	`ROTATION`	[numeric: double] Default: 0.0	Rotation of the buffer shape
Number of segments	`SEGMENTS`	[numeric: integer] Default: 36	Number of segments for a full circle (*Ovals* shape)
Output	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Output	`OUTPUT`	[vector: polygon]	The output vector layer (with the buffer shapes)

Python 代码

Algorithm ID: `native:rectanglesovalsdiamonds`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Remove duplicate vertices

Removes duplicate vertices from features, wherever removing the vertices does not result in a degenerate geometry.

The tolerance parameter specifies the tolerance for coordinates when determining whether vertices are identical.

By default, Z values are not considered when detecting duplicate vertices. E.g. two vertices with the same X and Y coordinate but different Z values will still be considered duplicate and one will be removed. If the *Use Z Value* parameter is true, then the Z values are also tested and vertices with the same X and Y but different Z will be maintained.

Allows *features in-place modification* of point, line, and polygon features

Note

Duplicate vertices are not tested between different parts of a multipart geometry, e.g. a multipoint geometry with overlapping points will not be changed by this method.

See also

Extract vertices, *Extract specific vertices*, *Delete duplicate geometries*, *Delete duplicated vertices*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Tolerance	`TOLERANCE`	[numeric: double] Default: 0.000001	Vertices closer than the specified distance are considered duplicates
Use Z value	`USE_Z_VALUE`	[boolean] Default: False	If the *Use Z Value* parameter is true, then the Z values are also tested and vertices with the same X and Y but different Z will be maintained.
Cleaned	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Cleaned	`OUTPUT`	[same as input]	The output vector layer (without duplicate vertices)

Python 代码

Algorithm ID: `native:removeduplicatevertices`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Remove null geometries

Removes any features which do not have a geometry from a vector layer. All other features will be copied unchanged.

The features with null geometries can be saved to a separate layer.

If *Also remove empty geometries* is checked, the algorithm removes features whose geometries have no coordinates, i.e., geometries that are empty. In that case, also the null output will reflect this option, containing both null and empty geometries.

See also

Delete duplicate geometries

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	Input vector layer (with non-NULL geometries)
Also remove empty geometries	`REMOVE_EMPTY`	[boolean]	
Non null geometries	`OUTPUT` 可选	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer for the non-NULL (and non-empty) geometries. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

continues on next page

Table 24.714 – continued from previous page

标注	名称	类型	描述
Null geometries 可选	NULL_OUTPUT	[same as input] Default: [Skip output]	Specify the output vector layer for the NULL (and empty) geometries. *One of*: • Skip Output • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Null geometries	NULL_OUTPUT	[same as input]	Output vector layer (for NULL and, if chosen, empty geometries)
Non null geometries	OUTPUT	[same as input]	The output vector layer (without NULL and, if chosen, empty geometries)

Python 代码

Algorithm ID: native:removenullgeometries

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Reverse line direction

Inverts the direction of a line layer.

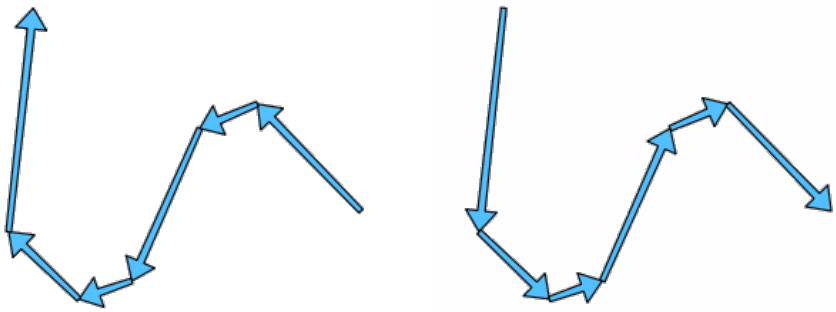


Fig. 24.135: Before and after the direction inversion

Allows *features in-place modification* of line features

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line]	Input line vector layer
Reversed	`OUTPUT`	[vector: line] Default: `[Create temporary layer]`	Specify the output line vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Reversed	`OUTPUT`	[vector: line]	The output line vector layer (with reversed lines)

Python 代码

Algorithm ID: `native:reverselinedirection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Rotate

Rotates feature geometries by the specified angle clockwise. The rotation occurs around each feature's centroid, or optionally around a unique preset point.

Allows *features in-place modification* of point, line, and polygon features

See also

翻译, *Swap X and Y coordinates*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Rotation (degrees clockwise)	`ANGLE`	[numeric: double] Default: 0.0	Angle of the rotation in degrees
Rotation anchor point (x, y) 可选	`ANCHOR`	[coordinates] Default: Not set	X,Y coordinates of the point to rotate the features around. If not set the rotation occurs around each feature's centroid.
Rotated	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer (with rotated geometries). *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Rotated	`OUTPUT`	[same as input]	The output vector layer with rotated geometries

Python 代码

Algorithm ID: `native:rotatefeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Roundness

Calculates the roundness of each feature and stores it as a new field `roundness`. The input vector layer must contain polygons.

The roundness of a polygon is defined as $4\pi \times$ polygon area / perimeter^2. The roundness value varies between 0 and 1. A perfect circle has a roundness of 1, while a completely flat polygon has a roundness of 0.

Note

The algorithm returns NULL for multipart polygon features.

Allows *features in-place modification* of polygon features

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: polygon]	输入矢量图层
Roundness	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output vector layer (with roundness field). *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Roundness	`OUTPUT`	[same as input]	The output vector layer with `roundness` value in a field

Python 代码

Algorithm ID: `native:roundness`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Segmentize by maximum angle

Segmentizes a geometry by converting curved sections to linear sections.

The segmentization is performed by specifying the maximum allowed radius angle between vertices on the straightened geometry (e.g the angle of the arc created from the original arc center to consecutive output vertices on the linearized geometry). Non-curved geometries will be retained without change.

See also

Segmentize by maximum distance, *Simplify*, *Smooth*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line, polygon]	Input line or polygon vector layer
Maximum angle between vertices (degrees)	`ANGLE`	[numeric: double] Default: 5.0	Maximum allowed radius angle between vertices on the straightened geometry
Segmentized	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer (with segmentized geometries). *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Segmentized	`OUTPUT`	[same as input]	The output vector layer with segmentized geometries

Python 代码

Algorithm ID: `native:segmentizebymaxangle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Segmentize by maximum distance

Segmentizes a geometry by converting curved sections to linear sections.

The segmentization is performed by specifying the maximum allowed offset distance between the original curve and the segmentized representation. Non-curved geometries will be retained without change.

See also

Segmentize by maximum angle, *Simplify*, *Smooth*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line, polygon]	Input line or polygon vector layer
Maximum offset distance	`DISTANCE`	[numeric: double] 默认值：1.0	Maximum allowed offset distance between the original curve and the segmentized representation, in the layer units.
Segmentized	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer (with segmentized geometries). *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Segmentized	`OUTPUT`	[same as input]	The output vector layer with segmentized geometries

Python 代码

Algorithm ID: `native:segmentizebymaxdistance`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Set M value

Sets the M value for geometries in a layer.

If M values already exist in the layer, they will be overwritten with the new value. If no M values exist, the geometry will be upgraded to include M values and the specified value used as the initial M value for all geometries.

Allows *features in-place modification* of point, line, and polygon features with M enabled

Tip

Use the Identify Features button to check the added M value: the results are available in the *Identify Results* dialog.

See also

Set M value from raster, *Set Z value*, *Drop M/Z values*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
M Value	`M_VALUE`	[numeric: double] Default: 0.0	M value to assign to the feature geometries
M Added	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
M Added	`OUTPUT`	[same as input]	The output vector layer (with M values assigned to the geometries)

Python 代码

Algorithm ID: `native:setmvalue`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Set M value from raster

Uses values sampled from a band within a raster layer to set the M value for every overlapping vertex in the feature geometry. The raster values can optionally be scaled by a preset amount.

If M values already exist in the layer, they will be overwritten with the new value. If no M values exist, the geometry will be upgraded to include M values.

Allows *features in-place modification* of point, line, and polygon features with M enabled

See also

Drape (set Z value from raster), *Set M value*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Raster layer	`RASTER`	[栅格]	Raster layer with M values
Band number	`BAND`	[raster band] Default: 1	The raster band from which the M values are taken
Value for No-Data or non-intersecting vertices	`NODATA`	[numeric: double] Default: 0.0	Value to use in case the vertex does not intersect (a valid pixel of) the raster
Scale factor	`SCALE`	[numeric: double] 默认值：1.0	Scaling value: the band values are multiplied by this value.
Offset	`OFFSET`	[numeric: double] Default: 0.0	Offset value: it is algebraically added to the band values after applying the "Scale factor".
Updated	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer (with updated M values). *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Updated	`OUTPUT`	[same as input]	The output vector layer (with updated M values)

Python 代码

Algorithm ID: `native:setmfromraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Set Z value

Sets the Z value for geometries in a layer.

If Z values already exist in the layer, they will be overwritten with the new value. If no Z values exist, the geometry will be upgraded to include Z values and the specified value used as the initial Z value for all geometries.

Allows *features in-place modification* of point, line, and polygon features with Z enabled

Tip

Use the Identify Features button to check the added Z value: the results are available in the *Identify Results* dialog.

See also

Drape (set Z value from raster), *Set M value*, *Drop M/Z values*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Z Value	`Z_VALUE`	[numeric: double] Default: 0.0	Z value to assign to the feature geometries
Z Added	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Z Added	`OUTPUT`	[same as input]	The output vector layer (with Z values assigned)

Python 代码

Algorithm ID: `native:setzvalue`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Simplify

Simplifies the geometries in a line or polygon layer. It creates a new layer with the same features as the ones in the input layer, but with geometries containing a lower number of vertices.

The algorithm gives a choice of simplification methods, including distance based (the "Douglas-Peucker" algorithm), area based ("Visvalingam" algorithm) and snapping geometries to grid.

Fig. 24.136: From left to right, source layer and increasing simplification tolerances

Allows *features in-place modification* of line and polygon features

Default menu: *Vector ► Geometry Tools*

See also

Simplify coverage, *Smooth*, *Densify by count*, *Densify by interval*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line, polygon]	Input line or polygon vector layer
Simplification method	`METHOD`	[enumeration] Default: 0	Simplification method. One of: • 0 --- Distance (Douglas-Peucker) • 1 --- Snap to grid • 2 --- Area (Visvalingam)

continues on next page

Table 24.732 – continued from previous page

标注	名称	类型	描述
Tolerance	TOLERANCE	[numeric: double] 默认值：1.0	Threshold tolerance (in units of the layer): if the distance between two nodes is smaller than the tolerance value, the segment will be simplified and vertices will be removed.
Simplified	OUTPUT	[same as input] Default: [Create temporary layer]	Specify the output (simplified) vector layer. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Simplified	OUTPUT	[same as input]	The output (simplified) vector layer

Python 代码

Algorithm ID: native:simplifygeometries

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Single sided buffer

Computes a buffer on lines by a specified distance on one side of the line only.

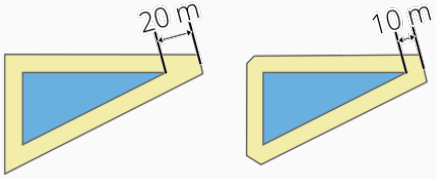
Buffer always results in a polygon layer.

Fig. 24.137: Left versus right side buffer on the same vector line layer

See also

Buffer

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line]	Input line vector layer
Distance	`DISTANCE`	[numeric: double] Default: 10.0	Buffer distance.
Side	`SIDE`	[enumeration] Default: 0	Which side to create the buffer on. One of: • 0 -- Left • 1 -- Right
Segments	`SEGMENTS`	[numeric: integer] Default: 8	Controls the number of line segments to use to approximate a quarter circle when creating rounded offsets.
Join style	`JOIN_STYLE`	[enumeration] Default: 0	Specifies whether round, miter or beveled joins should be used when offsetting corners in a line. Options are: • 0 --- Round • 1 --- Miter • 2 --- Bevel Fig. 24.138: Round, miter, and bevel join styles
Miter limit	`MITER_LIMIT`	[numeric: double] Default: 2.0	Sets the maximum distance from the offset geometry to use when creating a mitered join as a factor of the offset distance (only applicable for miter join styles). Minimum: 1.0  Fig. 24.139: A 10m buffer with a limit of 2 and a 10m buffer with a limit of 1
Buffer	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output (buffer) layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Buffer	`OUTPUT`	[vector: polygon]	Output (buffer) polygon layer

Python 代码

Algorithm ID: `native:singlesidedbuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Smooth

Smooths the geometries in a line or polygon layer by adding more **vertices and corners** to the feature geometries.

The iterations parameter dictates how many smoothing iterations will be applied to each geometry. A higher number of iterations results in smoother geometries with the cost of greater number of nodes in the geometries.

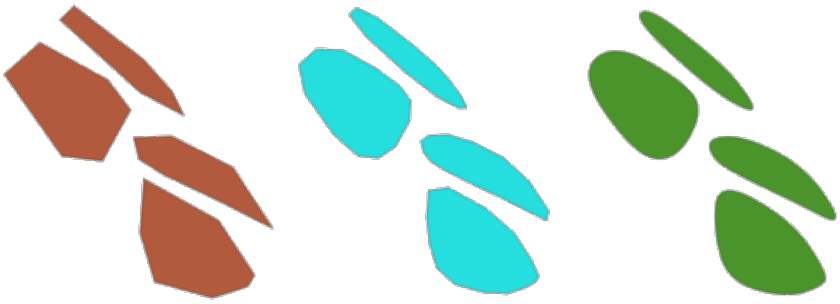


Fig. 24.140: Increasing number of iterations causes smoother geometries

The offset parameter controls how "tightly" the smoothed geometries follow the original geometries. Smaller values results in a tighter fit, and larger values will create a looser fit.

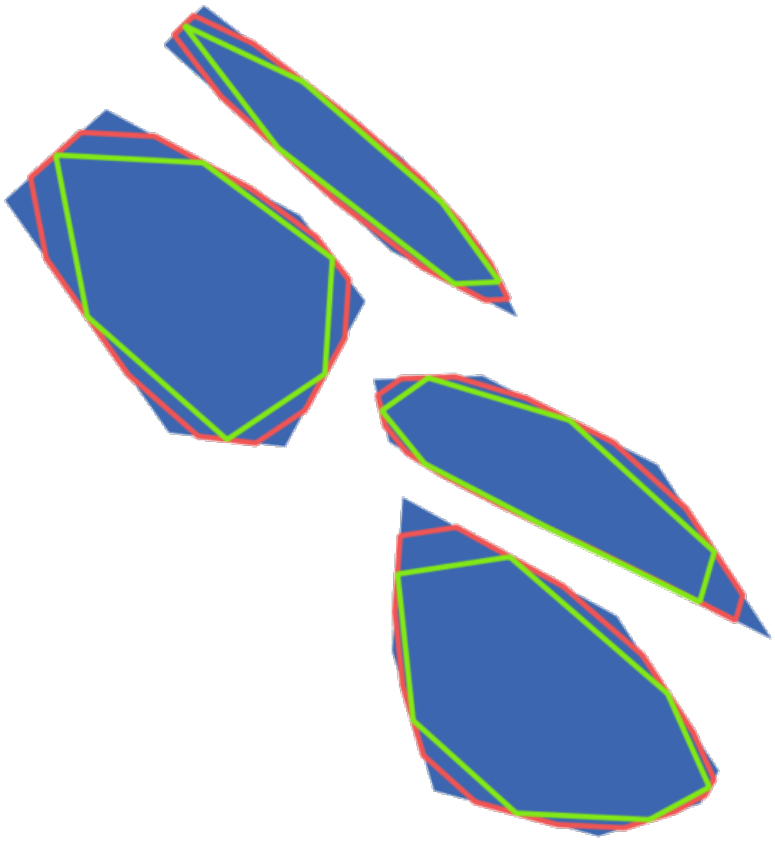


Fig. 24.141: Blue: the input layer. Offset 0.25 gives the red line, while offset 0.50 gives the green line.

The maximum angle parameter can be used to prevent smoothing of nodes with large angles. Any node where the angle of the segments to either side is larger than this will not be smoothed. For example, setting the maximum angle to 90 degrees or lower would preserve right angles in the geometry.

Allows *features in-place modification* of line and polygon features

See also

Simplify, *Simplify coverage*, *Densify by count*, *Densify by interval*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line, polygon]	Input line or polygon vector layer
Iterations	`ITERATIONS`	[numeric: integer] Default: 1	Increasing the number of iterations will give smoother geometries (and more vertices).
Offset	`OFFSET`	[numeric: double] Default: 0.25	Increasing values will *move* the smoothed lines / boundaries further away from the input lines / boundaries.

continues on next page

Table 24.736 – continued from previous page

标注	名称	类型	描述
Maximum node angle to smooth	`MAX_ANGLE`	[numeric: double] Default: 180.0	Every node below this value will be smoothed
Smoothed	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output (smoothed) layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Smoothed	`OUTPUT`	[same as input]	Output (smoothed) vector layer

Python 代码

Algorithm ID: `native:smoothgeometry`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Snap geometries to layer

Snaps the geometries in a layer either to the geometries from another layer, or to geometries within the same layer.

Matching is done based on a tolerance distance, and vertices will be inserted or removed as required to make the geometries match the reference geometries.

Allows *features in-place modification* of point, line, and polygon features

See also

Snap points to grid

参数

标注	名称	类型	描述
输入图层	INPUT	[vector: geometry]	输入矢量图层
Reference layer	REFERENCE_LAYE	[vector: geometry]	Vector layer to snap to
Tolerance	TOLERANCE	[numeric: double] Default: 10.0	Control how close input vertices need to be to the reference layer geometries before they are snapped.
Behavior	BEHAVIOR	[enumeration] Default: 0	Snapping can be done to an existing node or a segment (its closest point to the vertex to move). Available snapping options: • 0 --- Prefer aligning nodes, insert extra vertices where required Prefer to snap to nodes, even when a segment may be closer than a node. New nodes will be inserted to make geometries follow each other exactly when inside allowable tolerance. • 1 --- Prefer closest point, insert extra vertices where required Snap to closest point, regardless of it is a node or a segment. New nodes will be inserted to make geometries follow each other exactly when inside allowable tolerance. • 2 --- Prefer aligning nodes, don't insert new vertices Prefer to snap to nodes, even when a segment may be closer than a node. No new nodes will be inserted. • 3 --- Prefer closest point, don't insert new vertices Snap to closest point, regardless of it is a node or a segment. No new nodes will be inserted. • 4 --- Move end points only, prefer aligning nodes Only snap start/end points of lines (point features will also be snapped, polygon features will not be modified), prefer to snap to nodes. • 5 --- Move end points only, prefer closest point Only snap start/end points of lines (point features will also be snapped, polygon features will not be modified), snap to closest point • 6 --- Snap end points to end points only Only snap the start/end points of lines to other start/end points of lines • 7 --- Snap to anchor nodes (single layer only)

continues on next page

Table 24.738 – continued from previous page

标注	名称	类型	描述
Snapped geometry	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output (snapped) layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Snapped geometry	`OUTPUT`	[same as input]	Output (snapped) vector layer

Python 代码

Algorithm ID: `native:snapgeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Snap points to grid

Modifies the coordinates of geometries in a vector layer, so that all points or vertices are snapped to the closest point of a grid.

If the snapped geometry cannot be calculated (or is totally collapsed) the feature's geometry will be cleared.

Snapping can be performed on the X, Y, Z or M axis. A grid spacing of 0 for any axis will disable snapping for that axis.

Allows *features in-place modification* of point, line, and polygon features

Warning

Snapping to grid may generate an invalid geometry in some corner cases.

See also

Snap geometries to layer

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
X Grid Spacing	`HSPACING`	[numeric: double] 默认值：1.0	Grid spacing on the X axis
Y Grid Spacing	`VSPACING`	[numeric: double] 默认值：1.0	Grid spacing on the Y axis
Z Grid Spacing	`ZSPACING`	[numeric: double] Default: 0.0	Grid spacing on the Z axis
M Grid Spacing	`MSPACING`	[numeric: double] Default: 0.0	Grid spacing on the M axis
Snapped	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output (snapped) layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Snapped	`OUTPUT`	[same as input]	Output (snapped) vector layer

Python 代码

Algorithm ID: `native:snappointstogrid`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Split lines by maximum length

Takes a line (or curve) layer and splits each feature into multiple parts, where each part is of a specified maximum length. Z and M values at the start and end of the new line substrings are linearly interpolated from existing values.

Allows *features in-place modification* of line features

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line]	The input line vector layer
Maximum line length	`LENGTH`	[numeric: double] Default: 10.0	The maximum length of a line in the output.
Split	`OUTPUT`	[vector: line] Default: `[Create temporary layer]`	Specify the output line vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Split	`OUTPUT`	[vector: line]	The new line vector layer. The length of the feature geometries is less than or equal to the length specified in the LENGTH parameter. An additional field containing the `order` will be added.

Python 代码

Algorithm ID: `native:splitlinesbylength`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Subdivide

Subdivides the geometry. The returned geometry will be a collection containing subdivided parts from the original geometry, where no part has more than the specified maximum number of nodes.

This is useful for dividing a complex geometry into less complex parts, easier to spatially index and faster to perform spatial operations. Curved geometries will be segmentized before subdivision.

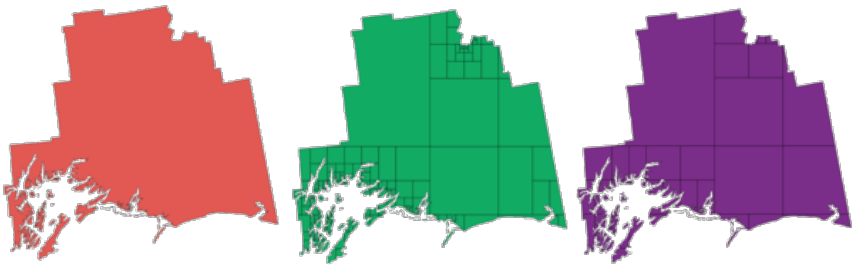


Fig. 24.142: Left the input layer, middle maximum nodes value is 100 and right maximum value is 200

Allows *features in-place modification* of point, line, and polygon features

Note

Subdividing a geometry can generate geometry parts that may not be valid and may contain self-intersections.

See also

Explode lines, *Line substring*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	The input vector layer
Maximum nodes in parts	`MAX_NODES`	[numeric: integer] Default: 256	Maximum number of vertices each new geometry part is allowed to have. Fewer *sub-parts* for higher values.
Subdivided	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output (subdivided) vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Subdivided	`OUTPUT`	[same as input]	Output vector layer

Python 代码

Algorithm ID: `native:subdivide`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Swap X and Y coordinates

Switches the X and Y coordinate values in input geometries.

It can be used to repair geometries which have accidentally had their latitude and longitude values reversed.

Allows *features in-place modification* of point, line, and polygon features

See also

翻译, *Rotate*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	The input vector layer
Swapped	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Swapped	`OUTPUT`	[same as input]	Output (swapped) vector layer

Python 代码

Algorithm ID: `native:swapxy`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Tapered buffers

Creates tapered buffer along line geometries, using a specified start and end buffer diameter.

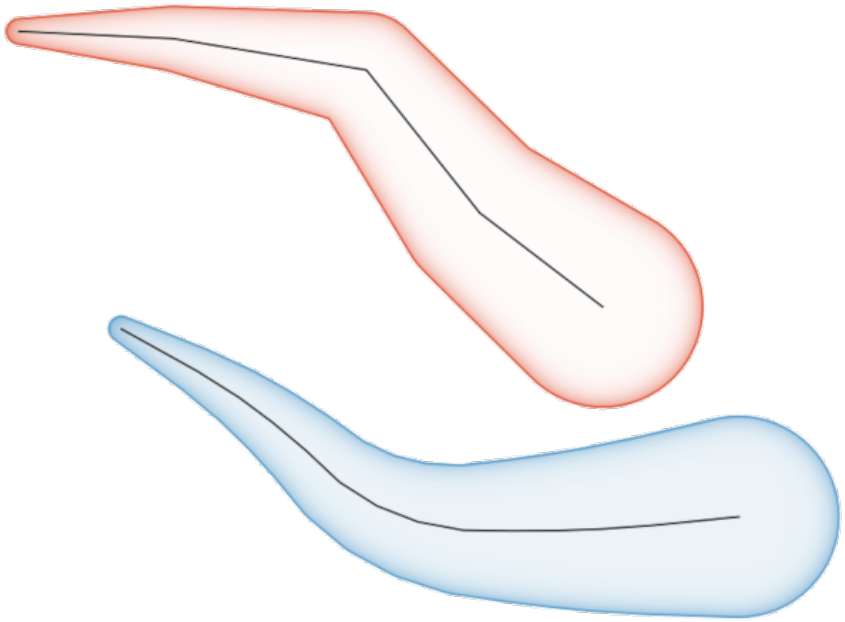


Fig. 24.143: Tapered buffer example

See also

Variable width buffer (by M value), *Buffer*, *Create wedge buffers*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line]	Input line vector layer
Start width	`START_WIDTH`	[numeric: double] Default: 0.0	Represents the radius of the buffer applied at the start point of the line feature
End width	`END_WIDTH`	[numeric: double] Default: 0.0	Represents the radius of the buffer applied at the end point of the line feature.
Segments	`SEGMENTS`	[numeric: integer] Default: 16	Controls the number of line segments to use to approximate a quarter circle when creating rounded offsets.
Buffered	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output (buffer) layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Buffered	`OUTPUT`	[vector: polygon]	Output (buffer) polygon layer

Python 代码

Algorithm ID: `native:taperedbuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Tessellate

Tessellates a polygon geometry layer, dividing the geometries into triangular components.

The output layer consists of multipolygon geometries for each input feature, with each multipolygon consisting of multiple triangle component polygons.

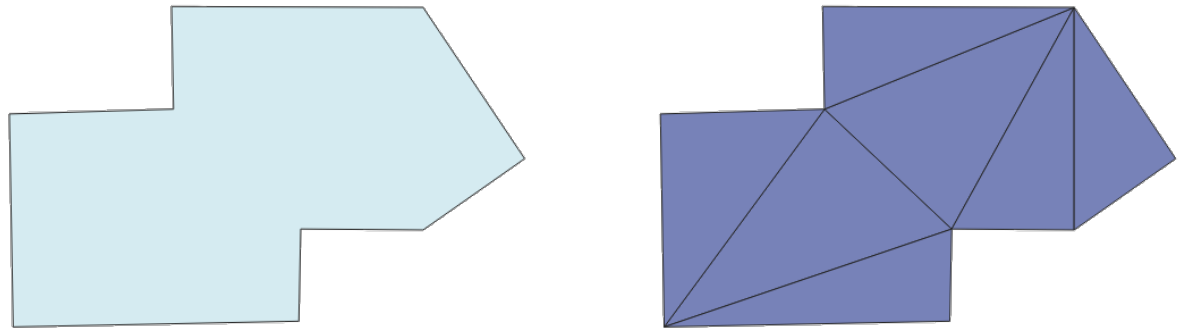


Fig. 24.144: Tessellated polygon (right)

Allows *features in-place modification* of polygon features

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: polygon]	Input polygon vector layer
Tesselated	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Tesselated	`OUTPUT`	[vector: polygon]	Output multipolygonZ layer

Python 代码

Algorithm ID: `3d:tessellate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Transect

Creates transects on vertices for (multi)linestring.

A transect is a line oriented from an angle (by default perpendicular) to the input polylines (at vertices).

Field(s) from feature(s) are returned in the transect with these new fields:

- TR_FID: ID of the original feature
- TR_ID: ID of the transect. Each transect have an unique ID
- TR_SEGMENT: ID of the segment of the linestring
- TR_ANGLE: Angle in degrees from the original line at the vertex
- TR_LENGTH: Total length of the transect returned
- TR_ORIENT: Side of the transect (only on the left or right of the line, or both side)

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

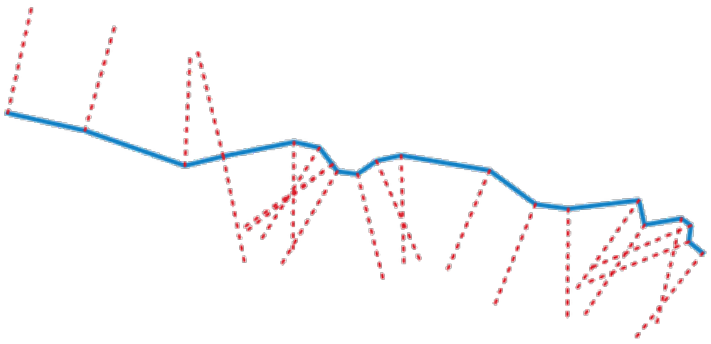


Fig. 24.145: Dashed red lines represent the transect of the input line layer

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line]	Input line vector layer
Length of the transect	`LENGTH`	[numeric: double] Default: 5.0	Length in map unit of the transect
Angle in degrees from the original line at the vertices	`ANGLE`	[numeric: double] Default: 90.0	Change the angle of the transect
Side to create the transect	`SIDE`	[enumeration]	Choose the side of the transect. Available options are: • 0 --- Left • 1 --- Right • 2 --- Both

continues on next page

Table 24.752 – continued from previous page

标注	名称	类型	描述
Transect	`OUTPUT`	[vector: line] Default: `[Create temporary layer]`	Specify the output line layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Transect	`OUTPUT`	[vector: line]	Output line layer

Python 代码

Algorithm ID: `native:transect`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

翻译

Moves the geometries within a layer, by offsetting with a predefined X and Y displacement.

Z and M values present in the geometry can also be translated.

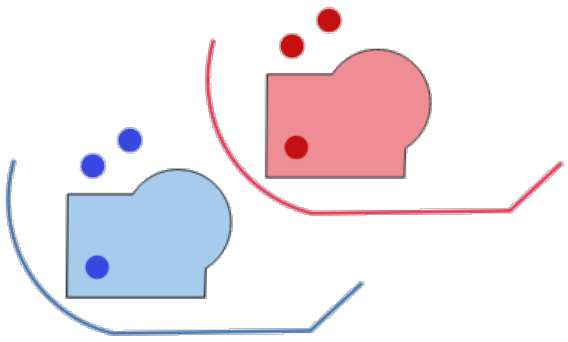


Fig. 24.146: Features in red represent the translated geometry of the input (blue) layer

Allows *features in-place modification* of point, line, and polygon features

See also

Array of translated features, *Offset lines*, *Rotate*, *Swap X and Y coordinates*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Offset distance (x-axis)	`DELTA_X`	[numeric: double] Default: 0.0	Displacement to apply on the X axis
Offset distance (y-axis)	`DELTA_Y`	[numeric: double] Default: 0.0	Displacement to apply on the Y axis
Offset distance (z-axis)	`DELTA_Z`	[numeric: double] Default: 0.0	Displacement to apply on the Z axis
Offset distance (m values)	`DELTA_M`	[numeric: double] Default: 0.0	Displacement to apply on the M axis
Translated	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Translated	`OUTPUT`	[same as input]	Output vector layer

Python 代码

Algorithm ID: `native:translategeometry`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Variable width buffer (by M value)

Creates variable width buffers along lines, using the M value of the line geometries as the diameter of the buffer at each vertex.

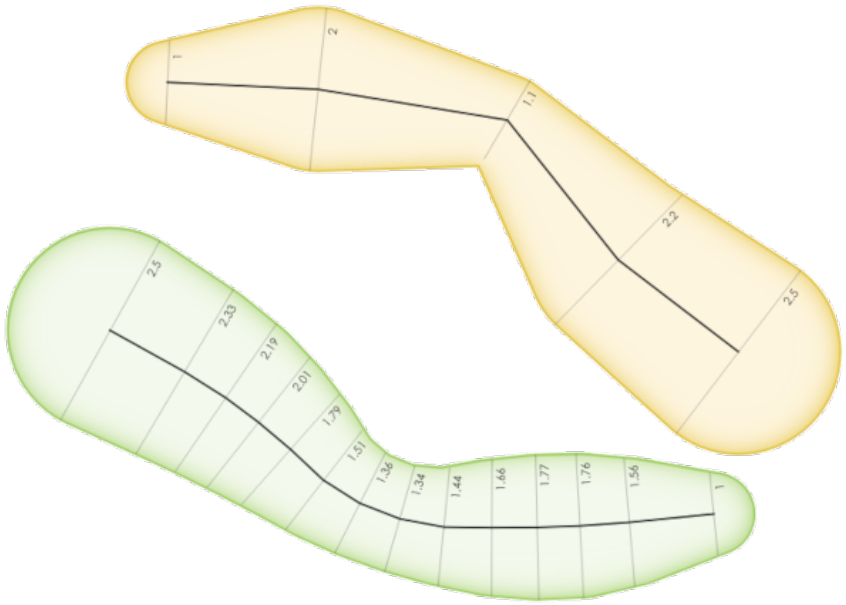


Fig. 24.147: Variable buffer example

See also

Tapered buffers, *Buffer*, *Set M value*, *Variable distance buffer*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line]	Input line vector layer
Segments	`SEGMENTS`	[numeric: integer] Default: 16	Number of the buffer segments per quarter circle. It can be a unique value (same value for all the features), or it can be taken from features data (the value can depend on feature attributes).
Buffered	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output (buffer) layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Buffered	`OUTPUT`	[vector: polygon]	Variable buffer polygon layer

Python 代码

Algorithm ID: `native:bufferbym`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Voronoi polygons

Takes a point layer and generates a polygon layer containing the Voronoi polygons (known also as Thiessen polygons) corresponding to those input points.

Any location within a Voronoi polygon is closer to the associated point than to any other point.

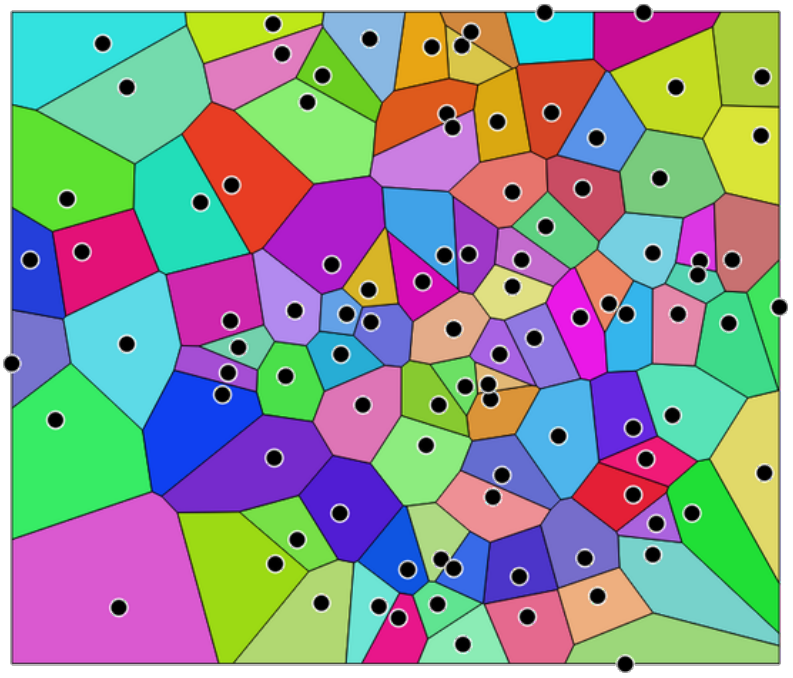


Fig. 24.148: Voronoi polygons

Default menu: *Vector ► Geometry Tools*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: point]	Input point vector layer
Buffer region (% of extent)	`BUFFER`	[numeric: double] Default: 0.0	The extent of the output layer will be this much bigger than the extent of the input layer
Tolerance 可选	`TOLERANCE`	[numeric: double] Default: 0.0	Specifies an optional snapping tolerance which can be used to improve the robustness of the voronoi.
Copy attributes from input features	`COPY_ATTRIBUTE`	[boolean] Default: True	Specifies whether fields storing involved point features ID should be added to the output. If False, an `id` field is created to identify the polygons.
Voronoi polygons	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specify the output layer (with the Voronoi polygons). *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Voronoi polygons	`OUTPUT`	[vector: polygon]	Voronoi polygons of the input point vector layer

Python 代码

Algorithm ID: `native:voronoipolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.27 Vector overlay

Clip

Clips a vector layer using the features of an additional polygon layer.

Only the parts of the features in the input layer that fall within the polygons of the overlay layer will be added to the resulting layer.

Warning

Geometry modification only

This operation modifies only the features geometry. The attribute values of the features are **not modified**, although properties such as area or length of the features will be modified by the overlay operation. If such properties are stored as attributes, those attributes will have to be manually updated.

This algorithm uses spatial indexes on the providers, prepared geometries and apply a clipping operation if the geometry isn't wholly contained by the mask geometry.

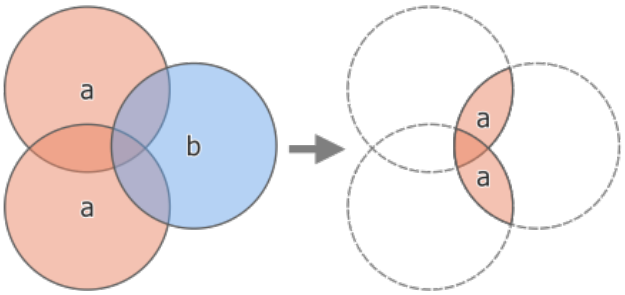


Fig. 24.149: Clipping operation between a two-feature input layer 'a' and a single feature overlay layer 'b' (left) - resulting in a new layer with the modified 'a' features (right)

Allows *features in-place modification* of point, line, and polygon features

Default menu: *Vector* ► *Geoprocessing Tools*

See also

Intersection, *Difference*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	Layer containing the features to be clipped
Overlay layer	`OVERLAY`	[vector: polygon]	Layer containing the clipping features
Clipped	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the layer to contain the features from the input layer that are inside the overlay (clipping) layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Clipped	`OUTPUT`	[same as input]	Layer containing features from the input layer split by the overlay layer.

Python 代码

Algorithm ID: `native:clip`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Difference

Extracts features from the input layer that don't fall within the boundaries of the overlay layer.

Input layer features that partially overlap the overlay layer feature(s) are split along the boundary of those feature(s) and only the portions outside the overlay layer features are retained.

Warning

Geometry modification only

This operation modifies only the features geometry. The attribute values of the features are **not modified**, although properties such as area or length of the features will be modified by the overlay operation. If such properties are stored as attributes, those attributes will have to be manually updated.

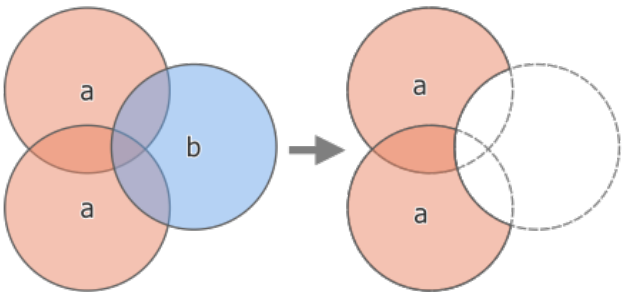


Fig. 24.150: Difference operation between a two-features input layer 'a' and a single feature overlay layer 'b' (left) - resulting in a new layer with the modified 'a' features (right)

Allows *features in-place modification* of point, line, and polygon features

Default menu: *Vector* ► *Geoprocessing Tools*

See also

Difference (multiple), *Symmetrical difference*, *Clip*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	Layer to extract (parts of) features from.
Overlay layer	`OVERLAY`	[vector: geometry]	Layer containing the geometries that will be subtracted from the input layer geometries. It is expected to have at least as many dimensions (point: 0D, line: 1D, polygon: 2D, volume: 3D) as the input layer geometries.
Difference	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the layer to contain the (parts of) features from the input layer that are not inside the overlay layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Grid size 可选	`GRID_SIZE`	[numeric: double] Default: Not set	If provided, the input geometries are snapped to a grid of the given size, and the result vertices are computed on that same grid.

输出

标注	名称	类型	描述
Difference	`OUTPUT`	[same as input]	Layer containing (parts of) features from the input layer not overlapping the overlay layer.

Python 代码

Algorithm ID: `native:difference`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Difference (multiple)

Extracts features from the input layer that fall completely outside or only partially overlap the features from any of the overlay layer(s).

For each overlay layer the difference is calculated between the result of all previous difference operations and this overlay layer. Input layer features that partially overlap feature(s) in the overlay layer are split along those features' boundary and only the portions outside the overlay layer features are retained.

Warning

Geometry modification only

This operation modifies only the features geometry. The attribute values of the features are **not modified**, although properties such as area or length of the features will be modified by the overlay operation. If such properties are stored as attributes, those attributes will have to be manually updated.

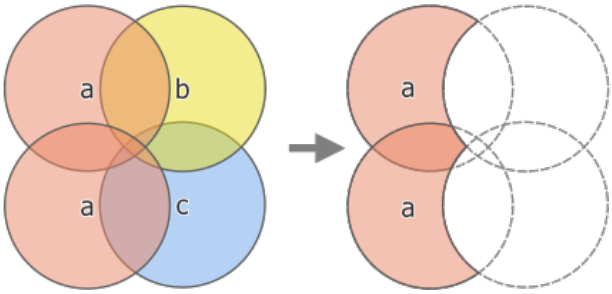


Fig. 24.151: Difference operation between a two-feature input layer 'a' and single feature overlay layers 'b' and 'c' (left) - resulting in a new layer with the modified 'a' features (right)

See also

Difference, *Symmetrical difference*, *Clip*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	Layer to extract (parts of) features from.
Overlay layers	`OVERLAYS`	[vector: geometry] [list]	List of layers containing the geometries that will be subtracted from the input layer geometries. They are expected to have at least as many dimensions (point: 0D, line: 1D, polygon: 2D, volume: 3D) as the input layer geometries.

continues on next page

Table 24.765 – continued from previous page

标注	名称	类型	描述
Difference	OUTPUT	[same as input] Default: [Create temporary layer]	Specify the layer to contain the (parts of) features from the input layer that do not overlap features of the overlay layers. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Difference	OUTPUT	[same as input]	Layer containing (parts of) features from the input layer not overlapping features from the overlay layers.

Python 代码

Algorithm ID: native:multidifference

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Extract/clip by extent

Creates a new vector layer that only contains features which fall within a specified extent. Any features which intersect the extent will be included by default.

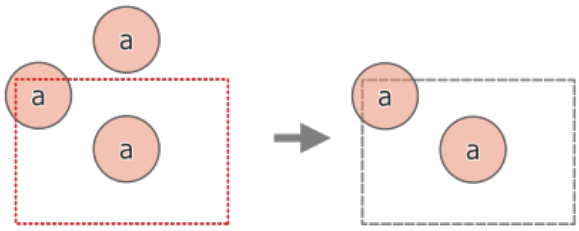


Fig. 24.152: Extract operation between a three-feature input layer 'a' and a dashed extent (left) - resulting features with dashed extent for reference (right)

Optionally, feature geometries can also be clipped to the extent.

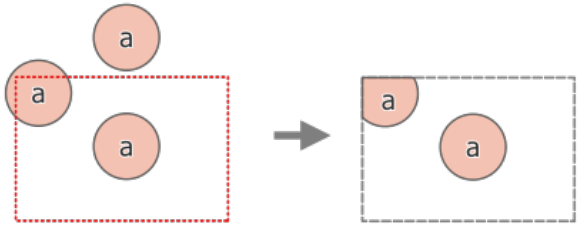


Fig. 24.153: Extract operation between a three-feature input layer 'a' and a dashed extent (left) - resulting features with dashed extent for reference (right)

See also

Clip

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	Layer to extract (parts of) features from.
Extent (xmin, xmax, ymin, ymax)	`EXTENT`	[范围]	Extent for clipping. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Clip features to extent	`CLIP`	[boolean] Default: False	If checked, the geometries will be clipped to the extent chosen instead of taking the whole geometry as output. Moreover, output geometries will be automatically converted to multi geometries to ensure uniform output types.
Extracted	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the layer to contain the features from the input layer that are inside the clip extent. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Extracted	OUTPUT	[same as input]	Layer containing the clipped features.

Python 代码

Algorithm ID: `native:extractbyextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Intersection

Extracts the portions of features from the input layer that overlap features in the overlay layer.

Features in the intersection layer are assigned the attributes of the overlapping features from both the input and overlay layers.

Warning

Geometry modification only

This operation modifies only the features geometry. The attribute values of the features are **not modified**, although properties such as area or length of the features will be modified by the overlay operation. If such properties are stored as attributes, those attributes will have to be manually updated.

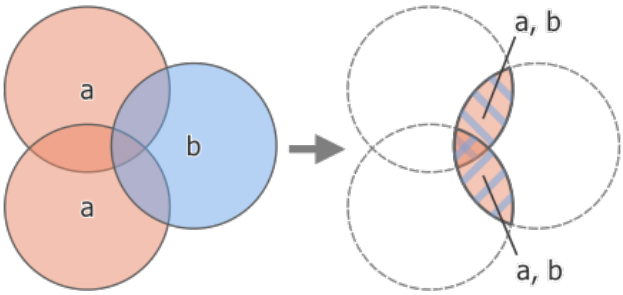


Fig. 24.154: Intersection operation between a two-feature input layer 'a' and a single feature overlay layer 'b' (left) - overlapping areas become a new two-feature layer with both layers' attributes (right)

Default menu: *Vector ► Geoprocessing Tools*

Warning

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

See also

Intersection (multiple), *Clip*, *Difference*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	Layer to extract (parts of) features from.
Overlay layer	`OVERLAY`	[vector: geometry]	Layer containing the features to check for overlap. Its features' geometry is expected to have at least as many dimensions (point: 0D, line: 1D, polygon: 2D, volume: 3D) as the input layer's.
Input fields to keep (leave empty to keep all fields) 可选	`INPUT_FIELDS`	[tablefield: any] [list] Default: Not set	Field(s) of the input layer to keep in the output. If no fields are chosen all fields are taken.
Overlay fields to keep (leave empty to keep all fields) 可选	`OVERLAY_FIELDS`	[tablefield: any] [list] Default: Not set	Field(s) of the overlay layer to keep in the output. If no fields are chosen all fields are taken. Duplicate field names will be appended a count suffix to avoid collision.
Intersection	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the layer to contain (the parts of) the features from the input layer that overlap one or more features from the overlay layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Overlay fields prefix 可选	`OVERLAY_FIELDS_`	[string]	Add a prefix to identify fields of the overlay layer. Duplicate field names will be appended a count suffix to avoid collision.
Grid size 可选	`GRID_SIZE`	[numeric: double] Default: Not set	If provided, the input geometries are snapped to a grid of the given size, and the result vertices are computed on that same grid.

输出

标注	名称	类型	描述
Intersection	`OUTPUT`	[same as input]	Layer containing (parts of) features from the input layer that overlap the overlay layer.

Python 代码

Algorithm ID: `native:intersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Intersection (multiple)

Extracts the overlapping portions of features in the input and all overlay layers.

Features in the output layer are assigned the attributes of the overlapping features from both the input and overlay layers.

> **Warning**
>
> **Geometry modification only**
>
> This operation modifies only the features geometry. The attribute values of the features are **not modified**, although properties such as area or length of the features will be modified by the overlay operation. If such properties are stored as attributes, those attributes will have to be manually updated.

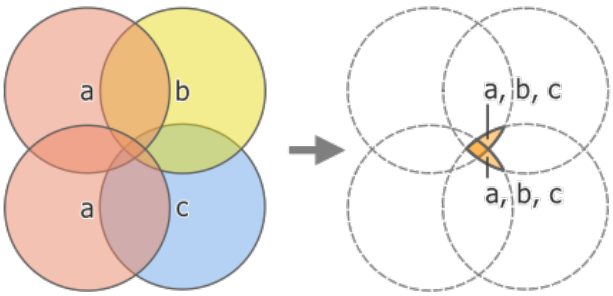


Fig. 24.155: Intersection operation between a two-feature input layer 'a' and single feature overlay layers 'b' and 'c' (left) - overlapping areas become a new two-feature layer with all layers' attributes (right)

> **Warning**
>
> This algorithm drops existing primary keys or FID values and regenerate them in output layers.

See also

Intersection, *Clip*, *Difference*

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[vector: geometry]	Layer to extract (parts of) features from.
Overlay layers	OVERLAYS	[vector: geometry] [list]	Layers containing the features to check for overlap. The features' geometry is expected to have at least as many dimensions (point: 0D, line: 1D, polygon: 2D, volume: 3D) as the input layer's.

Advanced parameters

标注	名称	类型	描述
Overlay fields prefix 可选	OVERLAY_FIELDS_	[string]	Add a prefix to identify fields of the overlay layers. Duplicate field names will be appended a count suffix to avoid collision.

输出

标注	名称	类型	描述
Intersection	OUTPUT	[same as input]	Layer containing (parts of) features from the input layer that overlap all the overlay layers.

Python 代码

Algorithm ID: `native:multiintersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Line intersections

Creates point features where the lines from the two layers intersect.

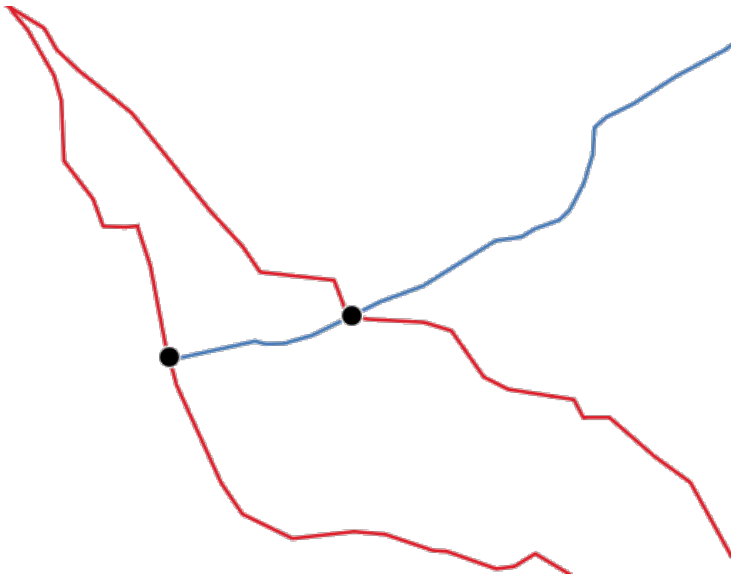


Fig. 24.156: Points of intersection

Default menu: *Vector* ► *Analysis Tools*

> **Warning**
>
> This algorithm drops existing primary keys or FID values and regenerate them in output layers.

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line]	Input line layer.
Intersect layer	`INTERSECT`	[vector: line]	Layer to use to find line intersections.
Input fields to keep (leave empty to keep all fields) 可选	`INPUT_FIELDS`	[tablefield: any] [list] Default: Not set	Field(s) of the input layer to keep in the output. If no fields are chosen all fields are taken.
Intersect fields to keep (leave empty to keep all fields) 可选	`INTERSECT_FIEL`	[tablefield: any] [list] Default: Not set	Field(s) of the intersect layer to keep in the output. If no fields are chosen all fields are taken. Duplicate field names will be appended a count suffix to avoid collision.

continues on next page

Table 24.775 – continued from previous page

标注	名称	类型	描述
Intersection	OUTPUT	[vector: point] Default: [Create temporary layer]	Specify the layer to contain the intersection points of the lines from the input and overlay layers. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Intersect fields prefix 可选	INTERSECT_FIEL	[string]	Add a prefix to identify fields of the intersect layer.

输出

标注	名称	类型	描述
Intersections	OUTPUT	[vector: point]	Point vector layer of the lines intersections, with both layers' attributes.

Python 代码

Algorithm ID: native:lineintersections

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Split with lines

Splits the lines or polygons in one layer using the lines or polygon rings in another layer to define the breaking points. Intersection between geometries in both layers are considered as split points.

Output will contain multi geometries for split features.

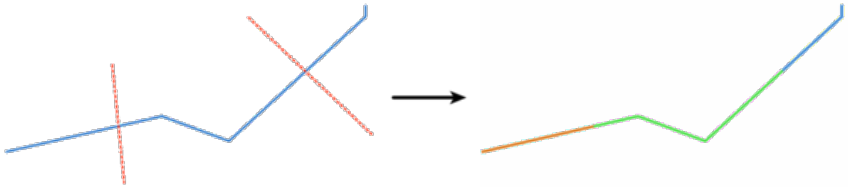


Fig. 24.157: Split lines

Allows *features in-place modification* of line and polygon features

Warning

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line, polygon]	Layer containing the lines or polygons to split.
Split layer	`LINES`	[vector: line, polygon]	Layer whose lines or rings are used to define the breaking points.
Split	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the layer to contain the splitted (in case they are intersected by a line in the split layer) line/polygon features from the input layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Split	`OUTPUT`	[same as input]	Output vector layer with split lines or polygons from input layer.

Python 代码

Algorithm ID: `native:splitwithlines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Symmetrical difference

Creates a layer containing features from both the input and overlay layers but with the overlapping areas between the two layers removed.

The attribute table of the symmetrical difference layer contains attributes and fields from both the input and overlay layers.

Warning

Geometry modification only

This operation modifies only the features geometry. The attribute values of the features are **not modified**, although properties such as area or length of the features will be modified by the overlay operation. If such properties are stored as attributes, those attributes will have to be manually updated.

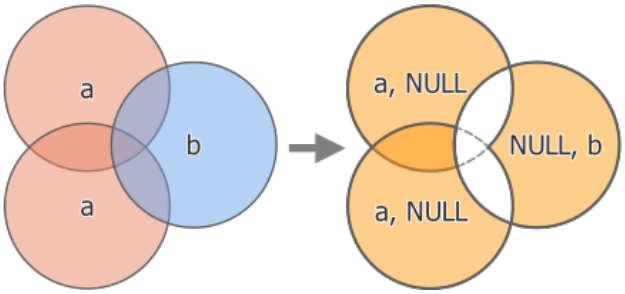


Fig. 24.158: Symmetrical difference operation between a two-features input layer 'a' and a single feature overlay layer 'b' (left) - resulting three-feature layer with both layers' attributes (right)

Default menu: *Vector ► Geoprocessing Tools*

Warning

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

See also

Difference, *Clip*, *Intersection*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	First layer to extract (parts of) features from.
Overlay layer	`OVERLAY`	[vector: geometry]	Second layer to extract (parts of) features from. Ideally the geometry type should be the same as input layer.

continues on next page

Table 24.780 – continued from previous page

标注	名称	类型	描述
Symmetrical difference	OUTPUT	[same as input] Default: [Create temporary layer]	Specify the layer to contain (the parts of) the features from the input and overlay layers that do not overlap features from the other layer. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Overlay fields prefix 可选	OVERLAY_FIELDS_	[string]	Add a prefix to identify fields of the overlay layer. Duplicate field names will be appended a count suffix to avoid collision.
Grid size 可选	GRID_SIZE	[numeric: double] Default: Not set	If provided, the input geometries are snapped to a grid of the given size, and the result vertices are computed on that same grid.

输出

标注	名称	类型	描述
Symmetrical difference	OUTPUT	[same as input]	Layer containing (parts of) features from each layer not overlapping the other layer, with both layers' attributes.

Python 代码

Algorithm ID: native:symmetricaldifference

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Union

Checks overlaps between features within the input layer and creates separate features for overlapping and non-overlapping parts. The area of overlap will create as many identical overlapping features as there are features that participate in that overlap.

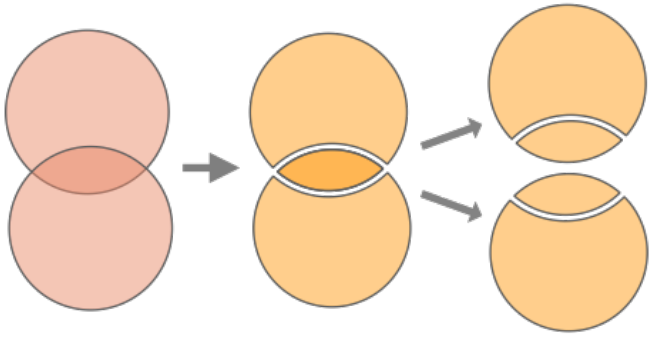


Fig. 24.159: Union operation with a single input layer with two overlapping features (left) - resulting in four features (middle) - features moved for clarity (right)

An overlay layer can also be used, in which case features from each layer are split at their overlap with features from the other one, creating a layer containing all the portions from both input and overlay layers. Features on the same layer will not split each other. The attribute table of the union layer is filled with attribute values from the respective original layer for non-overlapping features, and attribute values from both layers for overlapping features.

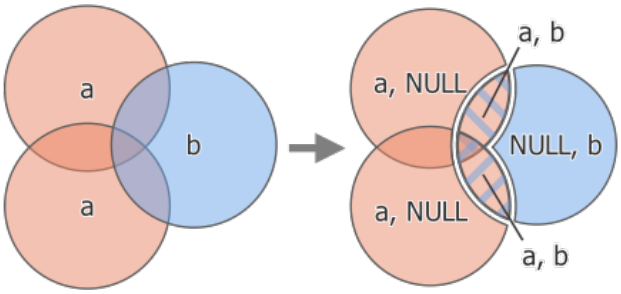


Fig. 24.160: Union operation between a two-feature input layer 'a' and a single feature overlay layer 'b' (left) - resulting five-feature layer with attributes from both layers (right)

Note

With an overlay layer, features on the same layer will not split each other. If you want to split overlaps on the same layer as well as other layers, first run the algorithm with multiple layers then run the algorithm again with only the previous output.

Default menu: *Vector* ► *Geoprocessing Tools*

Warning

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

See also

Union (multiple), *Clip*, *Difference*, *Intersection*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	Input vector layer to split at any intersections.
Overlay layer 可选	`OVERLAY`	[vector: geometry]	Layer that will be combined to the first one. Ideally the geometry type should be the same as input layer.
Union	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the layer to contain the (split and duplicated) features from the input layer and the overlay layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Overlay fields prefix 可选	`OVERLAY_FIELDS_`	[string]	Add a prefix to identify fields of the overlay layer. Duplicate field names will be appended a count suffix to avoid collision.
Grid size 可选	`GRID_SIZE`	[numeric: double] Default: Not set	If provided, the input geometries are snapped to a grid of the given size, and the result vertices are computed on that same grid.

输出

标注	名称	类型	描述
Union	`OUTPUT`	[same as input]	Layer containing all the overlapping and non-overlapping parts from the processed layer(s).

Python 代码

Algorithm ID: `native:union`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Union (multiple)

Checks overlaps between features within the input layer and creates separate features for overlapping and non-overlapping parts. The area of overlap will create as many identical overlapping features as there are features that participate in that overlap.

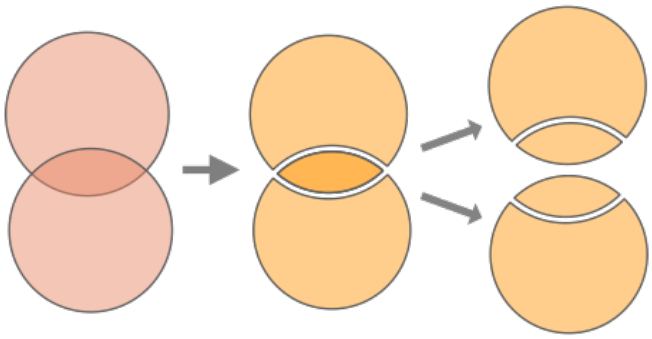


Fig. 24.161: Union operation with a single input layer with two overlapping features (left) - resulting in four features (middle) - features moved for clarity (right)

Multiple overlay layers can also be used, in which case features from each layer are split at their overlap with features from all other layers, creating a layer containing all the portions from both input and overlay layers. Features on the same layer will not split each other. The attribute table of the Union layer is filled with attribute values from the respective original layer for non-overlapping features, and attribute values from overlay layers for overlapping features.

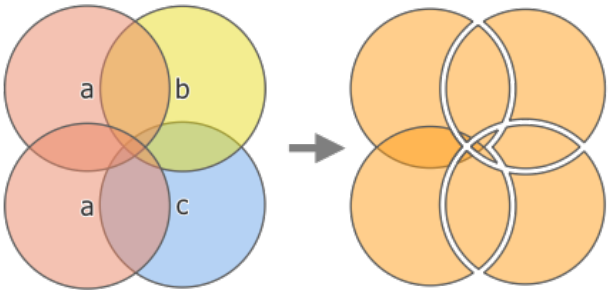


Fig. 24.162: Union operation between a two-feature input layer 'a' and single feature overlay layers 'b' and 'c' (left) - resulting eleven-feature layer with attributes from all layers (right)

Note

With an overlay layer, features on the same layer will not split each other. If you want to split overlaps on the same layer as well as other layers, first run the algorithm with multiple layers then run the algorithm again with only the previous output.

Warning

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

See also

Union, *Clip*, *Difference*, *Intersection*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	Input vector layer to split at any intersections.
Overlay layers 可选	`OVERLAYS`	[vector: geometry] [list]	Layers that will be combined to the first one. Ideally the geometry type should be the same as input layer.
Union	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the layer to contain the (split and duplicated) features from the input layer and the overlay layers. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

Advanced parameters

标注	名称	类型	描述
Overlay fields prefix 可选	`OVERLAY_FIELDS_`	[string]	Add a prefix to identify fields of the overlay layers. Duplicate field names will be appended a count suffix to avoid collision.

输出

标注	名称	类型	描述
Union	`OUTPUT`	[same as input]	Layer containing all the overlapping and non-overlapping parts from the processed layer(s), with all layers' attributes.

Python 代码

Algorithm ID: `native:multiunion`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.28 Vector selection

Extract by attribute

Creates two vector layers from an input layer: one will contain only matching features while the second will contain all the non-matching features.

The criteria for adding features to the resulting layer is based on the values of an attribute from the input layer.

See also

Select by attribute

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	Layer to extract features from.
Selection attribute	`FIELD`	[tablefield: any]	Filtering field of the layer
Operator	`OPERATOR`	[enumeration] Default: 0	Many different operators are available: • 0 --- = • 1 --- ≠ • 2 --- >>• 3 --- >= • 4 --- < • 5 --- <= • 6 --- begins with • 7 --- contains • 8 --- is null • 9 --- is not null • 10 --- does not contain
Value 可选	`VALUE`	[string]	Value to be evaluated
Extracted (attribute)	`OUTPUT`	[same as input] Default: `[Create Temporary Layer]`	Specify the output vector layer for matching features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

continues on next page

Table 24.789 – continued from previous page

标注	名称	类型	描述
Extracted (non-matching)	FAIL_OUTPUT	[same as input] Default: [Skip output]	Specify the output vector layer for non-matching features. *One of*: • Skip Output • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Extracted (attribute)	OUTPUT	[same as input]	Vector layer with matching features from the input layer
Extracted (non-matching)	FAIL_OUTPUT	[same as input]	Vector layer with non-matching features from the input layer

Python 代码

Algorithm ID: native:extractbyattribute

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Extract by expression

Creates two vector layers from an input layer: one will contain only matching features while the second will contain all the non-matching features.

The criteria for adding features to the resulting layer is based on a QGIS expression. For more information about expressions see the 表达式.

See also

Select by expression

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	输入矢量图层
Expression	`EXPRESSION`	[expression]	Expression to filter the vector layer
Matching features	`OUTPUT`	[same as input] Default: `[Create Temporary Layer]`	Specify the output vector layer for matching features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。
Non-matching	`FAIL_OUTPUT`	[same as input] Default: `[Skip output]`	Specify the output vector layer for non-matching features. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Matching features	`OUTPUT`	[same as input]	Vector layer with matching features from the input layer
Non-matching	`FAIL_OUTPUT`	[same as input]	Vector layer with non-matching features from the input layer

Python 代码

Algorithm ID: `native:extractbyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Extract by location

Creates a new vector layer that only contains matching features from an input layer.

The criteria for adding features to the resulting layer is based on the spatial relationship between each feature and the features in an additional layer.

See also

Select by location, *Extract within distance*

Exploring spatial relations

Geometric predicates are boolean functions used to determine the spatial relation a feature has with another by comparing whether and how their geometries share a portion of space.

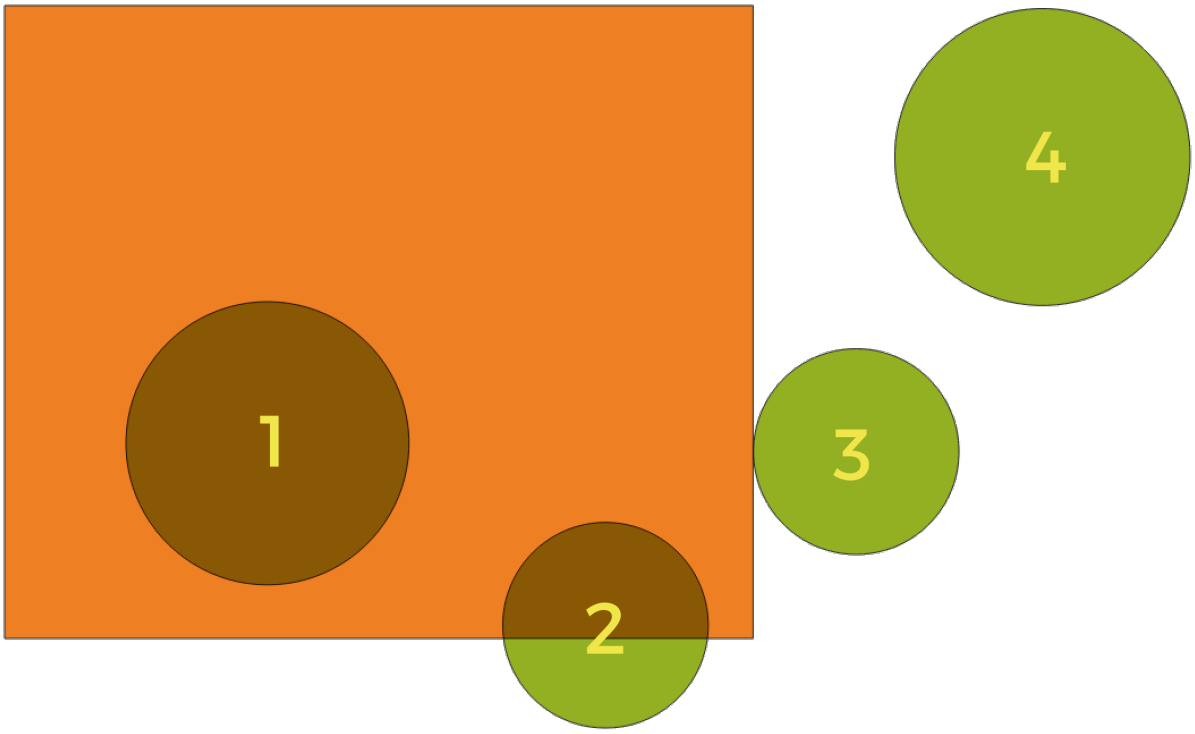


Fig. 24.163: Looking for spatial relations between layers

Using the figure above, we are looking for the green circles by spatially comparing them to the orange rectangle feature. Available geometric predicates are:

Intersect
: Tests whether a geometry intersects another. Returns 1 (true) if the geometries spatially intersect (share any portion of space - overlap or touch) and 0 if they don't. In the picture above, this will return circles 1, 2 and 3.

Contain
: Returns 1 (true) if and only if no points of b lie in the exterior of a, and at least one point of the interior of b lies in the interior of a. In the picture, no circle is returned, but the rectangle would be if you would look for it the other way around, as it contains circle 1 completely. This is the opposite of *are within*.

Disjoint
: Returns 1 (true) if the geometries do not share any portion of space (no overlap, not touching). Only circle 4

is returned.

Equal
Returns 1 (true) if and only if geometries are exactly the same. No circles will be returned.

Touch
Tests whether a geometry touches another. Returns 1 (true) if the geometries have at least one point in common, but their interiors do not intersect. Only circle 3 is returned.

Overlap
Tests whether a geometry overlaps another. Returns 1 (true) if the geometries share space, are of the same dimension, but are not completely contained by each other. Only circle 2 is returned.

Are within
Tests whether a geometry is within another. Returns 1 (true) if geometry a is completely inside geometry b. Only circle 1 is returned.

Cross
Returns 1 (true) if the supplied geometries have some, but not all, interior points in common and the actual crossing is of a lower dimension than the highest supplied geometry. For example, a line crossing a polygon will cross as a line (true). Two lines crossing will cross as a point (true). Two polygons cross as a polygon (false). In the picture, no circles will be returned.

参数

标注	名称	类型	描述
Extract features from	`INPUT`	[vector: geometry]	输入矢量图层
Where the features (geometric predicate)	`PREDICATE`	[enumeration] [list] Default: [0]	Type of spatial relation the input feature should have with an intersect feature so that it could be selected. One or more of: • 0 --- intersect • 1 --- contain • 2 --- disjoint • 3 --- equal • 4 --- touch • 5 --- overlap • 6 --- are within • 7 --- cross If more than one condition is chosen, at least one of them (OR operation) has to be met for a feature to be extracted.
By comparing to the features from	`INTERSECT`	[vector: geometry]	Intersection vector layer
Extracted (location)	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer for the features that have the chosen spatial relationship(s) with one or more features in the comparison layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Extracted (location)	`OUTPUT`	[same as input]	Vector layer with features from the input layer that have the chosen spatial relationship(s) with one or more features in the comparison layer.

Python 代码

Algorithm ID: `native:extractbylocation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Extract within distance

Creates a new vector layer that only contains matching features from an input layer. Features are copied wherever they are within the specified maximum distance from the features in an additional reference layer.

See also

Select within distance, *Extract by location*

参数

标注	名称	类型	描述
Extract features from	`INPUT`	[vector: geometry]	Input vector layer to copy features from
By comparing to the features from	`REFERENCE`	[vector: geometry]	Vector layer whose features closeness is used
Where the features are within	`DISTANCE`	[numeric: double] Default: 100.0	The maximum distance around reference features to select input features within
Modify current selection by	`METHOD`	[enumeration] Default: 0	How the selection of the algorithm should be managed. One of: • 0 --- creating new selection • 1 --- adding to current selection • 2 --- selecting within current selection • 3 --- removing from current selection

continues on next page

Table 24.795 – continued from previous page

标注	名称	类型	描述
Extracted (location)	OUTPUT	[same as input] Default: [Create temporary layer]	Specify the output vector layer for the features that are within the set distance from reference features. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Extracted (location)	OUTPUT	[same as input]	Vector layer with features from the input layer matching the condition of distance from reference features

Python 代码

Algorithm ID: `native:extractwithindistance`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Filter by geometry type

Filters features by their geometry type. Incoming features will be directed to different outputs based on whether they have a point, line or polygon geometry.

参数

标注	名称	类型	描述
输入图层	INPUT	[vector: any]	Layer to evaluate

输出

标注	名称	类型	描述
Point features 可选	`POINTS`	[vector: point]	Layer with points
Line features 可选	`LINES`	[vector: line]	Layer with lines
Polygon features 可选	`POLYGONS`	[vector: polygon]	Layer with polygons
Features with no geometry 可选	`NO_GEOMETRY`	[vector: table]	Geometry-less vector layer

Python 代码

Algorithm ID: `native:filterbygeometry`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Random extract

Takes a vector layer and generates a new one that contains only a subset of the features in the input layer.

The subset is defined randomly, based on feature IDs, using a percentage or count value to define the total number of features in the subset.

Warning

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

See also

Random selection

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	Source vector layer to select the features from
Method	`METHOD`	[enumeration] Default: 0	Random selection methods. One of: • 0 --- Number of selected features • 1 --- Percentage of selected features

continues on next page

Table 24.799 – continued from previous page

标注	名称	类型	描述
Number/percentage of selected features	NUMBER	[numeric: integer] Default: 10	Number or percentage of features to select
Extracted (random)	OUTPUT	[same as input] Default: [Create temporary layer]	Specify the output vector layer for the randomly selected features. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Extracted (random)	OUTPUT	[same as input]	Vector layer containing randomly selected features from the input layer

Python 代码

Algorithm ID: native:randomextract

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Random extract within subsets

Takes a vector layer and generates a new one that contains only a subset of the features in the input layer.

The subset is defined randomly, based on feature IDs, using a percentage or count value to define the total number of features in the subset. The percentage/count value is not applied to the whole layer, but instead to each category. Categories are defined according to a given attribute.

See also

Random selection within subsets

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	Vector layer to select the features from
ID field	`FIELD`	[tablefield: any]	Category of the source vector layer to select the features from
Method	`METHOD`	[enumeration] Default: 0	Random selection method. One of: • 0 --- Number of selected features • 1 --- Percentage of selected features
Number/percentage of selected features	`NUMBER`	[numeric: integer] Default: 10	Number or percentage of features to select
Extracted (random stratified)	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer for the randomly selected features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Extracted (random stratified)	`OUTPUT`	[same as input]	Vector layer containing randomly selected features from the input layer

Python 代码

Algorithm ID: `qgis:randomextractwithinsubsets`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Random selection

Takes a vector layer and selects a subset of its features. No new layer is generated by this algorithm.

The subset is defined randomly, based on feature IDs, using a percentage or count value to define the total number of features in the subset.

Default menu: *Vector ► Research Tools*

See also

Random extract

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	Vector layer for the selection
Method	`METHOD`	[enumeration] Default: 0	Random selection method. One of: • 0 --- Number of selected features • 1 --- Percentage of selected features
Number/percentage of selected features	`NUMBER`	[numeric: integer] Default: 10	Number or percentage of features to select

输出

标注	名称	类型	描述
输入图层	`INPUT`	[same as input]	The input layer with features selected

Python 代码

Algorithm ID: `qgis:randomselection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Random selection within subsets

Takes a vector layer and selects a subset of its features. No new layer is generated by this algorithm.

The subset is defined randomly, based on feature IDs, using a percentage or count value to define the total number of features in the subset.

The percentage/count value is not applied to the whole layer, but instead to each category.

Categories are defined according to a given attribute, which is also specified as an input parameter for the algorithm.

No new outputs are created.

Default menu: *Vector ► Research Tools*

See also

Random extract within subsets

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	Vector layer to select features in
ID field	`FIELD`	[tablefield: any]	Category of the input layer to select the features from
Method	`METHOD`	[enumeration] Default: 0	Random selection method. One of: • 0 --- Number of selected features • 1 --- Percentage of selected features
Number/percentage of selected features	`NUMBER`	[numeric: integer] Default: 10	Number or percentage of features to select

输出

标注	名称	类型	描述
输入图层	`INPUT`	[same as input]	The input layer with features selected

Python 代码

Algorithm ID: `qgis:randomselectionwithinsubsets`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Select by attribute

Creates a selection in a vector layer.

The criteria for selecting features is based on the values of an attribute from the input layer.

See also

Extract by attribute

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	Vector layer to select features in
Selection attribute	`FIELD`	[tablefield: any]	Filtering field of the layer
Operator	`OPERATOR`	[enumeration] Default: 0	Many different operators are available: • 0 --- = • 1 --- ≠ • 2 --- >> • 3 --- >= • 4 --- < • 5 --- <= • 6 --- begins with • 7 --- contains • 8 --- is null • 9 --- is not null • 10 --- does not contain
Value 可选	`VALUE`	[string]	Value to be evaluated
Modify current selection by	`METHOD`	[enumeration] Default: 0	How the selection of the algorithm should be managed. One of: • 0 --- creating new selection • 1 --- adding to current selection • 2 --- removing from current selection • 3 --- selecting within current selection

输出

标注	名称	类型	描述
输入图层	`INPUT`	[same as input]	The input layer with features selected

Python 代码

Algorithm ID: `qgis:selectbyattribute`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Select by expression

Creates a selection in a vector layer.

The criteria for selecting features is based on a QGIS expression. For more information about expressions see the 表达式.

See also

Extract by expression

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	输入矢量图层
Expression	`EXPRESSION`	[expression]	Expression to filter the input layer
Modify current selection by	`METHOD`	[enumeration] Default: 0	How the selection of the algorithm should be managed. One of: • 0 --- creating new selection • 1 --- adding to current selection • 2 --- removing from current selection • 3 --- selecting within current selection

输出

标注	名称	类型	描述
输入图层	`INPUT`	[same as input]	The input layer with features selected

Python 代码

Algorithm ID: `qgis:selectbyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Select by location

Creates a selection in a vector layer.

The criteria for selecting features is based on the spatial relationship between each feature and the features in an additional layer.

Default menu: *Vector ► Research Tools*

See also

Extract by location, *Select within distance*

Exploring spatial relations

Geometric predicates are boolean functions used to determine the spatial relation a feature has with another by comparing whether and how their geometries share a portion of space.

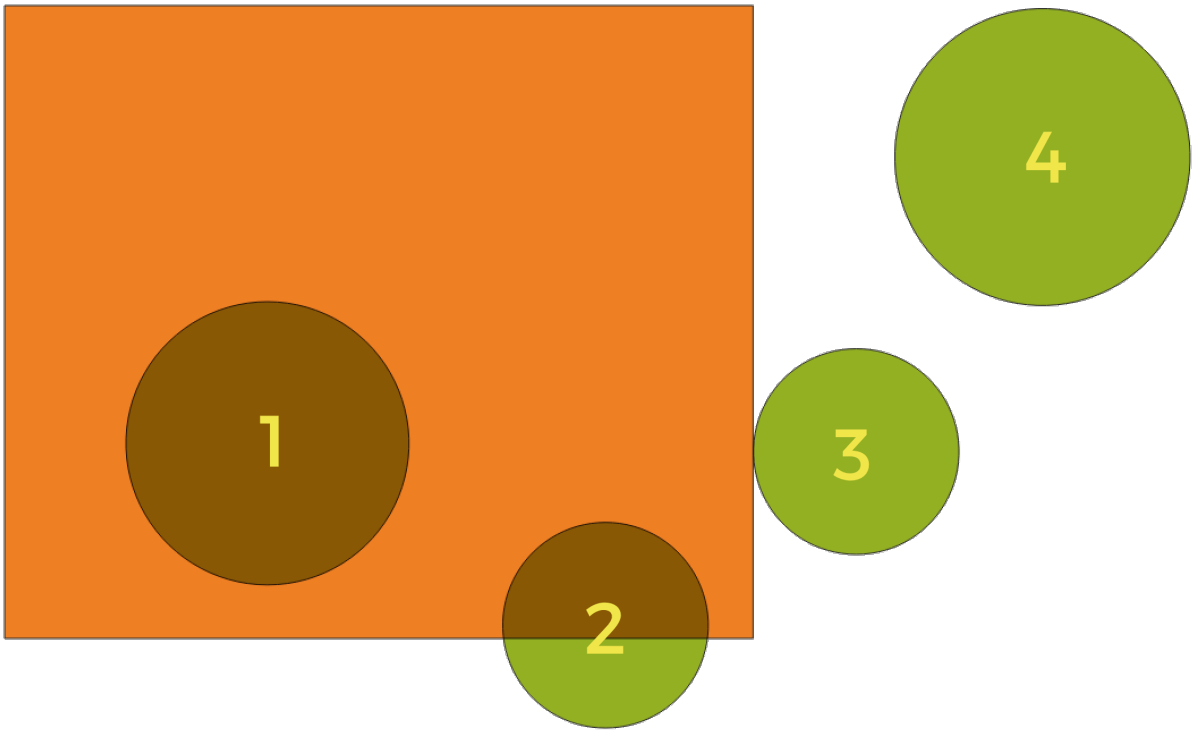


Fig. 24.164: Looking for spatial relations between layers

Using the figure above, we are looking for the green circles by spatially comparing them to the orange rectangle feature. Available geometric predicates are:

Intersect
: Tests whether a geometry intersects another. Returns 1 (true) if the geometries spatially intersect (share any portion of space - overlap or touch) and 0 if they don't. In the picture above, this will return circles 1, 2 and 3.

Contain
: Returns 1 (true) if and only if no points of b lie in the exterior of a, and at least one point of the interior of b lies in the interior of a. In the picture, no circle is returned, but the rectangle would be if you would look for it the other way around, as it contains circle 1 completely. This is the opposite of *are within*.

Disjoint
Returns 1 (true) if the geometries do not share any portion of space (no overlap, not touching). Only circle 4 is returned.

Equal
Returns 1 (true) if and only if geometries are exactly the same. No circles will be returned.

Touch
Tests whether a geometry touches another. Returns 1 (true) if the geometries have at least one point in common, but their interiors do not intersect. Only circle 3 is returned.

Overlap
Tests whether a geometry overlaps another. Returns 1 (true) if the geometries share space, are of the same dimension, but are not completely contained by each other. Only circle 2 is returned.

Are within
Tests whether a geometry is within another. Returns 1 (true) if geometry a is completely inside geometry b. Only circle 1 is returned.

Cross
Returns 1 (true) if the supplied geometries have some, but not all, interior points in common and the actual crossing is of a lower dimension than the highest supplied geometry. For example, a line crossing a polygon will cross as a line (true). Two lines crossing will cross as a point (true). Two polygons cross as a polygon (false). In the picture, no circles will be returned.

参数

标注	名称	类型	描述
Select features from	`INPUT`	[vector: geometry]	输入矢量图层
Where the features (geometric predicate)	`PREDICATE`	[enumeration] [list] Default: [0]	Type of spatial relation the input feature should have with an intersect feature so that it could be selected. One or more of: • 0 --- intersect • 1 --- contain • 2 --- disjoint • 3 --- equal • 4 --- touch • 5 --- overlap • 6 --- are within • 7 --- cross If more than one condition is chosen, at least one of them (OR operation) has to be met for a feature to be extracted.
By comparing to the features from	`INTERSECT`	[vector: geometry]	Intersection vector layer
Modify current selection by	`METHOD`	[enumeration] Default: 0	How the selection of the algorithm should be managed. One of: • 0 --- creating new selection • 1 --- adding to current selection • 2 --- selecting within current selection • 3 --- removing from current selection

输出

标注	名称	类型	描述
输入图层	`INPUT`	[same as input]	The input layer with features selected

Python 代码

Algorithm ID: `native:selectbylocation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Select within distance

creates a selection in a vector layer. Features are selected wherever they are within the specified maximum distance from the features in an additional reference layer.

See also

Extract within distance, *Select by location*

参数

标注	名称	类型	描述
Select features from	`INPUT`	[vector: geometry]	Input vector layer to select features from
By comparing to the features from	`REFERENCE`	[vector: geometry]	Vector layer whose features closeness is used
Where the features are within	`DISTANCE`	[numeric: double] Default: 100.0	The maximum distance around reference features to select input features
Modify current selection by	`METHOD`	[enumeration] Default: 0	How the selection of the algorithm should be managed. One of: • 0 --- creating new selection • 1 --- adding to current selection • 2 --- selecting within current selection • 3 --- removing from current selection

输出

标注	名称	类型	描述
输入图层	INPUT	[same as input]	The input layer with features selected

Python 代码

Algorithm ID: native:selectwithindistance

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.29 Vector table

Add autoincremental field

Adds a new integer field to a vector layer, with a sequential value for each feature.

This field can be used as a unique ID for features in the layer. The new attribute is not added to the input layer but a new layer is generated instead.

The initial starting value for the incremental series can be specified. Optionally, the incremental series can be based on grouping fields and a sort order for features can also be specified.

参数

标注	名称	类型	描述
输入图层	INPUT	[vector: any]	The input vector layer.
Field name	FIELD_NAME	[string] Default: 'AUTO'	Name of the field with autoincremental values
Start values at 可选	START	[numeric: integer] Default: 0	Choose the initial number of the incremental count
Modulus value 可选	MODULUS	[numeric: integer] Default: 0	Specifying an optional modulus value will restart the count to START whenever the field value reaches the modulus value. 0 means no restart.
Group values by 可选	GROUP_FIELDS	[tablefield: any] [list]	Select grouping field(s): instead of a single count run for the whole layer, a separate count is processed for each value returned by the combination of these fields.
Sort expression 可选	SORT_EXPRESSIO	[expression]	Use an expression to sort the features in the layer either globally or if set, based on group fields.
Sort ascending	SORT_ASCENDING	[boolean] Default: True	When a sort expression is set, use this option to control the order in which features are assigned values.

continues on next page

Table 24.815 – continued from previous page

标注	名称	类型	描述
Sort nulls first	SORT_NULLS_FIR	[boolean] Default: False	When a sort expression is set, use this option to set whether *Null* values are counted first or last.
Incremented	OUTPUT	[same as input] Default: [Create temporary layer]	Specify the output vector layer with the auto increment field. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Incremented	OUTPUT	[same as input]	Vector layer with auto incremental field

Python 代码

Algorithm ID: native:addautoincrementalfield

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Add field to attributes table

Adds a new field to a vector layer.

The name and characteristics of the attribute are defined as parameters.

The new attribute is not added to the input layer but a new layer is generated instead.

参数

标注	名称	类型	描述
输入图层	INPUT	[vector: any]	The input layer
Field name	FIELD_NAME	[string]	Name of the new field

continues on next page

Table 24.817 – continued from previous page

标注	名称	类型	描述
Field type	FIELD_TYPE	[enumeration] Default: 0	Type of the new field. You can choose between: • 0 --- Integer (32 bit) • 1 --- Decimal (double) • 2 --- Text (string) • 3 --- Boolean • 4 --- Date • 5 --- Time • 6 --- Date & Time • 7 --- Binary Object (BLOB) • 8 --- String List • 9 --- Integer List • 10 --- Decimal (double) List
Field length	FIELD_LENGTH	[numeric: integer] Default: 10	Length of the field
Field precision	FIELD_PRECISIO	[numeric: integer] Default: 0	Precision of the field. Useful with Float field type.
Field alias 可选	FIELD_ALIAS	[string]	Set a name to use as alias for the field. Not supported by all format types.
Field comment 可选	FIELD_COMMENT	[string]	Store a comment describing the field. Not supported by all format types.
Added	OUTPUT	[same as input] Default: [Create temporary layer]	Specify the output vector layer. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Added	OUTPUT	[same as input]	Vector layer with new field added

Python 代码

Algorithm ID: native:addfieldtoattributestable

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Add unique value index field

Takes a vector layer and an attribute and adds a new numeric field.

Values in this field correspond to values in the specified attribute, so features with the same value for the attribute will have the same value in the new numeric field.

This creates a numeric equivalent of the specified attribute, which defines the same classes.

The new attribute is not added to the input layer but a new layer is generated instead.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	The input layer.
Class field	`FIELD`	[tablefield: any]	Features that have the same value for this field will get the same index.
Output field name	`FIELD_NAME`	[string] Default: 'NUM_FIELD'	Name of the new field containing the indexes.
Layer with index field	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Vector layer with the numeric field containing indexes. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。
Class summary	`SUMMARY_OUTPUT`	[vector: table] Default: `[Skip output]`	Specify the table to contain the summary of the class field mapped to the corresponding unique value. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Layer with index field	`OUTPUT`	[same as input]	Vector layer with the numeric field containing indexes.
Class summary	`SUMMARY_OUTPUT`	[vector: table]	Table with summary of the class field mapped to the corresponding unique value.

Python 代码

Algorithm ID: `native:adduniquevalueindexfield`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Add X/Y fields to layer

Adds X and Y (or latitude/longitude) fields to a point layer. The X/Y fields can be calculated in a different CRS to the layer (e.g. creating latitude/longitude fields for a layer in a projected CRS).

Allows *features in-place modification* of point features

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: point]	The input layer.
Coordinate system	`CRS`	[crs] Default: "EPSG:4326"	Coordinate reference system to use for the generated x and y fields.
Field prefix 可选	`PREFIX`	[string]	Prefix to add to the new field names to avoid name collisions with fields in the input layer.
Added fields	`OUTPUT`	[vector: point] Default: `[Create temporary layer]`	Specify the output layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Added fields	`OUTPUT`	[vector: point]	The output layer - identical to the input layer but with two new double fields, `x` and `y`.

Python 代码

Algorithm ID: `native:addxyfields`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Advanced Python field calculator

Adds a new attribute to a vector layer, with values resulting from applying an expression to each feature.

The expression is defined as a Python function.

Warning

This algorithm is a potential security risk if executed with unchecked inputs, and may result in system damage or data leaks.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	输入矢量图层
Result field name	`FIELD_NAME`	[string] Default: 'NewField'	Name of the new field
Field type	`FIELD_TYPE`	[enumeration] Default: 0	Type of the new field. One of: • 0 --- Integer (32 bit) • 1 --- Decimal (double) • 2 --- Text (string) • 3 --- Boolean • 4 --- Date • 5 --- Time • 6 --- Date & Time • 7 --- Binary Object (BLOB) • 8 --- String List • 9 --- Integer List • 10 --- Decimal (double) List
Field length	`FIELD_LENGTH`	[numeric: integer] Default: 10	Length of the field
Field precision	`FIELD_PRECISIO`	[numeric: integer] Default: 3	Precision of the field. Useful with Float field type.
Global expression 可选	`GLOBAL`	[string]	The code in the global expression section will be executed only once before the calculator starts iterating through all the features of the input layer. Therefore, this is the correct place to import necessary modules or to calculate variables that will be used in subsequent calculations.

continues on next page

Table 24.823 – continued from previous page

标注	名称	类型	描述

Formula	FORMULA	[string]	The Python formula to evaluate. Example: To calculate the area of an input polygon layer you can add:
			`value = $geom.area()`
			To access an existing field use its name enclosed in the angle brackets. For example, to uppercase values of the "address" field:
			`value = <address>.upper()`

Calculated	OUTPUT	[same as input] Default: `[Create temporary layer]`	Specify the vector layer with the new calculated field. One of: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Calculated	OUTPUT	[same as input]	Vector layer with the new calculated field

Python 代码

Algorithm ID: `qgis:advancedpythonfieldcalculator`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Drop field(s)

Takes a vector layer and generates a new one that has the same features but without the selected columns.

See also

Retain fields

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	Input vector layer to drop field(s) from
Fields to drop	`COLUMN`	[tablefield: any] [list]	The field(s) to drop
Remaining fields	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer with the remaining fields. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Remaining fields	`OUTPUT`	[same as input]	Vector layer with the remaining fields

Python 代码

Algorithm ID: `native:deletecolumn`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Explode HStore Field

Creates a copy of the input layer and adds a new field for every unique key in the HStore field.

The expected field list is an optional comma separated list. If this list is specified, only these fields are added and the HStore field is updated. By default, all unique keys are added.

The PostgreSQL HStore is a simple key-value store used in PostgreSQL and GDAL (when reading an OSM file with the `other_tags` field.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
HStore field	`FIELD`	[tablefield: any]	The field(s) to drop
Expected list of fields separated by a comma 可选	`EXPECTED_FIELD`	[string] Default: ”	Comma-separated list of fields to extract. The HStore field will be updated by removing these keys.
Exploded	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specify the output vector layer. One of: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Exploded	`OUTPUT`	[same as input]	Output vector layer

Python 代码

Algorithm ID: `native:explodehstorefield`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

Extract binary field

Extracts contents from a binary field, saving them to individual files. Filenames can be generated using values taken from an attribute in the source table or based on a more complex expression.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	Input vector layer containing the binary data
Binary field	`FIELD`	[tablefield: any]	Field containing the binary data
File name	`FILENAME`	[expression]	Field or expression-based text to name each output file
Destination folder	`FOLDER`	[folder] Default: `[Save to temporary folder]`	Folder in which to store the output files. *One of*: • Save to a Temporary Directory • Save to Directory

输出

标注	名称	类型	描述
Folder	`FOLDER`	[folder]	The folder that contains the output files.

Python 代码

Algorithm ID: `native:extractbinary`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Field calculator

Opens the field calculator (see 表达式). You can use all the supported expressions and functions.

A new layer is created with the result of the expression.

The field calculator is very useful when used in *The model designer*.

> **Note**
>
> This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	The layer to calculate on
Output field name	`FIELD_NAME`	[string]	The name of the field for the results
Output field type	`FIELD_TYPE`	[enumeration] Default: 0	The type of the field. One of: • 0 --- Decimal (double) • 1 --- Integer (32 bit) • 2 --- Text (string) • 3 --- Date • 4 --- Time • 5 --- Date & Time • 6 --- Boolean • 7 --- Binary Object (BLOB) • 8 --- String List • 9 --- Integer List • 10 --- Decimal (double) List
Output field width	`FIELD_LENGTH`	[numeric: integer] Default: 0	The length of the result field (minimum 0)
Field precision	`FIELD_PRECISIO`	[numeric: integer] Default: 0	The precision of the result field (minimum 0, maximum 15)
Create new field	`NEW_FIELD`	[boolean] Default: True	Should the result field be a new field
Formula	`FORMULA`	[expression]	The formula to use to calculate the result
Calculated	`OUTPUT`	[same as input] Default: `[Create temporary layer]`	Specification of the output layer. • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Calculated	`OUTPUT`	[same as input]	Output layer with the calculated field values

Python 代码

Algorithm ID: `native:fieldcalculator`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Refactor fields

Allows editing the structure of the attribute table of a vector layer.

Fields can be modified in their type and name, using a fields mapping.

The original layer is not modified. A new layer is generated, which contains a modified attribute table, according to the provided fields mapping.

> **Note**
>
> When using a template layer with *constraints* on fields, the information is displayed in the widget with a coloured background and tooltip. Treat this information as a hint during configuration. No constraints will be added on an output layer nor will they be checked or enforced by the algorithm.

The Refactor fields algorithm allows to:

- Change field names and types
- Add and remove fields
- Reorder fields
- Calculate new fields based on expressions
- Load field list from another layer

> **Note**
>
> This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

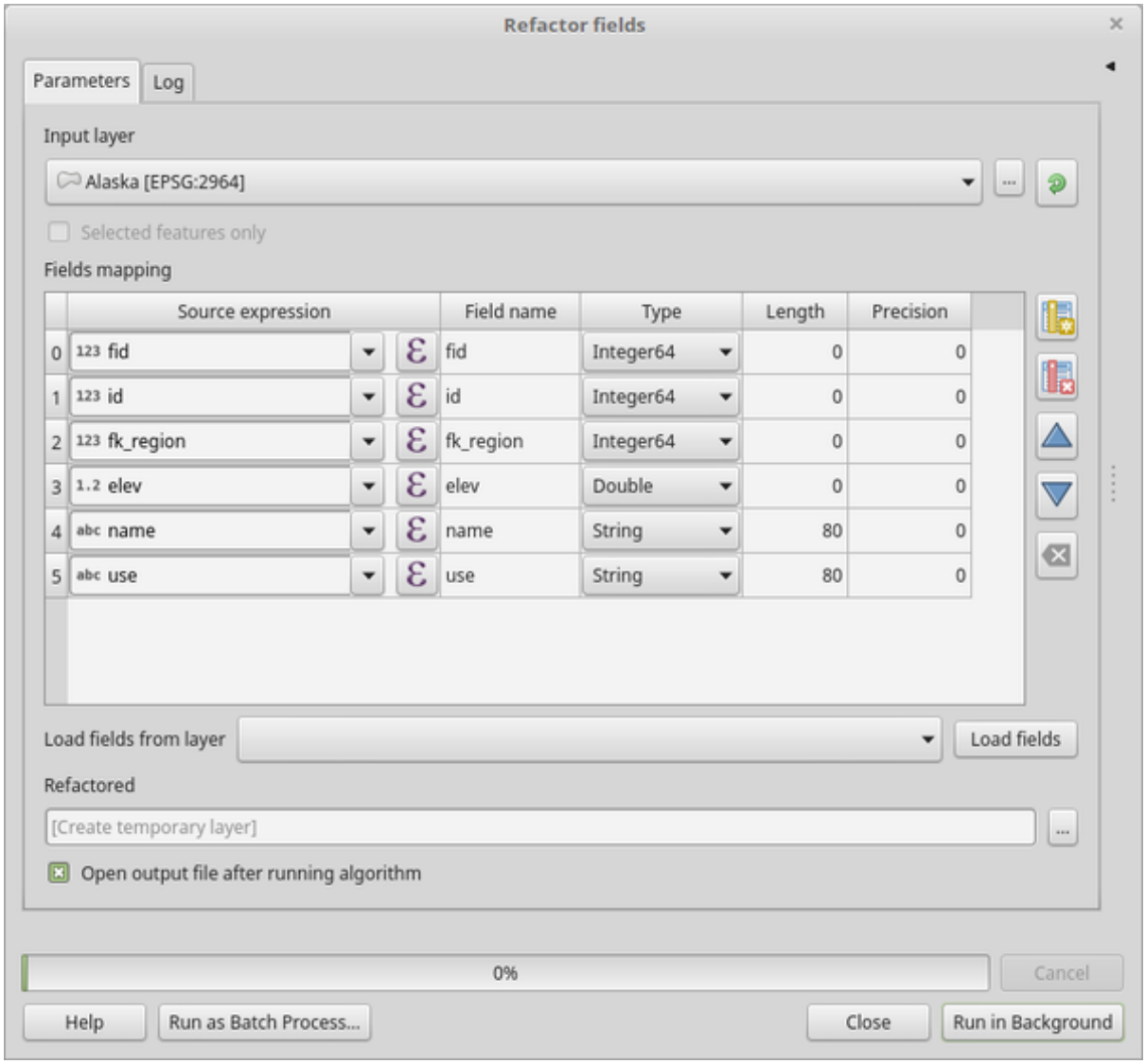


Fig. 24.165: Refactor fields dialog

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	The layer to modify

continues on next page

Table 24.833 – continued from previous page

标注	名称	类型	描述
Fields mapping	`FIELDS_MAPPING`	[list]	List of output fields with their definitions. The embedded table lists all the fields of the source layer and allows you to edit them: • Click to create a new field. • Click to remove a field. • Use and to change the selected field order. • Click to reset to the default view. • Click to invert the selection in the fields list. For each of the fields you'd like to reuse, you need to fill the following options: ***Source expression*** **(`expression`) [expression]** Field or expression from the input layer. ***Name*** **(`name`) [string]** Name of the field in the output layer. By default input field name is kept. ***Type*** **(`type`) [enumeration]** Data type of the output field. Available types may not be compatible with the output layer provider. One of: **Attention** For certain field types, e.g. lists, an extra `sub_type` parameter helps refine the specific type of the data. It is automatically set in the GUI but may be needed if you're running the algorithm in Python or from the command line. • 1 --- Boolean • 2 --- Integer (32bit) • 4 --- Integer (64bit) • 6 --- Decimal (double) • 9 --- Integer list • 9 --- Integer (64bit) list • 9 --- Decimal (double) list • 10 --- Text (string) • 11 --- String list • 12 --- Binary Object (BLOB) • 14 --- Date • 15 --- Time • 16 --- Date & Time ***Sub-type*** **(`sub_type`) [enumeration]** For certain field types, e.g. lists, this parameter helps refine the specific `type` of the data. It is automatically set in the GUI but may be needed if you're running the algorithm in Python or from the command line. *One of*: • 2 --- Integer • 4 --- Integer (64bit) list • 6 --- Decimal (double) list • 10 --- String list

Table 24.833 – continued from previous page

标注	名称	类型	描述
Refactored	OUTPUT	[same as input] Default: [Create temporary layer]	Specification of the output layer. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Refactored	OUTPUT	[same as input]	Output layer with refactored fields

Python 代码

Algorithm ID: native:refactorfields

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Rename field

Renames an existing field from a vector layer.

The original layer is not modified. A new layer is generated where the attribute table contains the renamed field.

See also

Refactor fields

参数

标注	名称	类型	描述
输入图层	INPUT	[vector: any]	The input vector layer
Field to rename	FIELD	[tablefield: any]	The field to be altered
New field name	NEW_NAME	[string]	The new field name

continues on next page

Table 24.835 – continued from previous page

标注	名称	类型	描述
Renamed	OUTPUT	[same as input] Default: [Create temporary layer]	Specification of the output layer. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Renamed	OUTPUT	[same as input]	Output layer with the renamed field

Python 代码

Algorithm ID: native:renametablefield

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Retain fields

Takes a vector layer and generates a new one that retains only the selected fields. All other fields will be dropped.

See also

Drop field(s)

参数

标注	名称	类型	描述
输入图层	INPUT	[vector: any]	The input vector layer
Fields to retain	FIELDS	[tablefield: any] [list]	List of fields to keep in the layer
Retained fields	OUTPUT	[same as input] Default: [Create temporary layer]	Specification of the output layer. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Retained fields	`OUTPUT`	[same as input]	Output layer with the retained fields

Python 代码

Algorithm ID: `native:retainfields`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Text to float

Modifies the type of a given attribute in a vector layer, converting a text attribute containing numeric strings into a numeric attribute (e.g. '1' to `1.0`).

The algorithm creates a new vector layer so the source one is not modified.

If the conversion is not possible the selected column will have `NULL` values.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	The input vector layer.
Text attribute to convert to float	`FIELD`	[tablefield: string]	The string field for the input layer that is to be converted to a float field.
Float from text	`OUTPUT`	[same as input] Default: `[Create Temporary Layer]`	Specify the output layer. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • 保存到文件… • Save to Geopackage… • Save to Database Table… • Append to Layer… 文件编码也可以在此更改。

输出

标注	名称	类型	描述
Float from text	`OUTPUT`	[same as input]	Output vector layer with the string field converted into a float field

Python 代码

Algorithm ID: `qgis:texttofloat`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.1.30 Vector Tiles

Download vector tiles

Downloads vector tiles of an input vector tile layer and saves them in a local vector tile file.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector tiles]	A vector tile layer to extract some tiles from
Extent	`EXTENT`	[范围]	Specify the spatial extent of the downloaded area. It will internally be extended to a multiple of the tile size. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Maximum zoom level to download	`MAX_ZOOM`	[numeric: integer] Default: 10	Defines how far to zoom in and fetch data from the tiles
Tile limit	`TILE_LIMIT`	[numeric: integer] Default: 100	Maximum number of tiles to download, considering the zoom levels and the extent.
Output	`OUTPUT`	[vector tiles] Default: [Save to temporary file]	Specification of the output vector tile file. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Output	`OUTPUT`	[vector tiles]	A local vector tile file storing the downloaded tiles.

Python 代码

Algorithm ID: `native:downloadvectortiles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Write vector tiles (MBTiles)

Exports one or more vector layers to vector tiles, a data format optimized for fast map rendering and small data size.

MBTiles is a specification for storing tiled map data in SQLite databases for immediate usage and for transfer. MBTiles files are known as tilesets.

参数

标注	名称	类型	描述
Input layers	`INPUT`	[vector: any] [list]	A list of layers to combine to generate the vector tiles
Minimum zoom level	`MIN_ZOOM`	[numeric: integer] Default: 0	The lowest zoom level for which the tileset provides data. Set between 0 and 24.
Maximum zoom level	`MAX_ZOOM`	[numeric: integer] Default: 3	The highest zoom level for which the tileset provides data. Set between 0 and 24.
Extent 可选	`EXTENT`	[范围] Default: Not set	The maximum extent of the rendered map area. Bounds must define an area covered by all zoom levels.
Metadata: Name 可选	`META_NAME`	[string]	Name of the tileset
Metadata: Description 可选	`META_DESCRIPTI`	[string]	A description of the tileset's contents
Metadata: Attribution 可选	`META_ATTRIBUTI`	[string]	An attribution string, which explains the sources of data and/or style for the map.
Metadata: Version 可选	`META_VERSION`	[string]	The version of the tileset. This refers to a revision of the tileset itself, not of the MBTiles specification.
Metadata: Type 可选	`META_TYPE`	[string]	Type of tileset. Possible values are `overlay` or `baselayer`.

continues on next page

Table 24.843 – continued from previous page

标注	名称	类型	描述
Metadata: Center 可选	`META_CENTER`	[string]	The center (string of comma-separated numbers: the longitude, latitude, and zoom level) of the default view of the map. Example: `-122.1906,37.7599,11`
Destination MBTiles	`OUTPUT`	[vector tiles] Default: [Save to temporary file]	Specification of the output MBTiles file. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Destination MBTiles	`OUTPUT`	[file]	Output vector tiles `.mbtiles` file.

Python 代码

Algorithm ID: `native:writevectortiles_mbtiles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Write vector tiles (XYZ)

Exports one or more vector layers to vector tiles, a data format optimized for fast map rendering and small data size.

参数

标注	名称	类型	描述
File template	`XYZ_TEMPLATE`	[string] Default: '{z}/{x}/{y}.pbf'	Template to generate the vector tiles url
Input layers	`INPUT`	[vector: any] [list]	A list of layers to combine to generate the vector tiles
Minimum zoom level	`MIN_ZOOM`	[numeric: integer] Default: 0	The lowest zoom level for which the tileset provides data. Set between 0 and 24.
Maximum zoom level	`MAX_ZOOM`	[numeric: integer] Default: 3	The highest zoom level for which the tileset provides data. Set between 0 and 24.
Extent 可选	`EXTENT`	[范围] Default: Not set	The maximum extent of the rendered map area. Bounds must define an area covered by all zoom levels.

continues on next page

Table 24.845 – continued from previous page

标注	名称	类型	描述
Output directory	OUTPUT_DIRECTO	[folder] Default: [Save to temporary folder]	Specification of the output vector tiles folder. *One of*: • Save to a Temporary Directory • Save to Directory

输出

标注	名称	类型	描述
Output directory	OUTPUT_DIRECTO	[folder]	A folder containing different subsets of the vector tiles files (`.pbf`) stored in subfolders corresponding to the zoom levels.

Python 代码

Algorithm ID: `native:writevectortiles_xyz`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.2 GDAL algorithm provider

GDAL (Geospatial Data Abstraction Library) is a translator library for raster and vector geospatial data formats. Algorithms in the Processing Framework are derived from the GDAL raster programs and GDAL vector programs.

24.2.1 栅格分析

坡向

Generates an aspect map from any GDAL-supported elevation raster. Aspect is the compass direction that a slope faces. The pixels will have a value from 0-360° measured in degrees from north indicating the azimuth. On the northern hemisphere, the north side of slopes is often shaded (small azimuth from 0°-90°), while the southern side receives more solar radiation (higher azimuth from 180°-270°).

This algorithm is derived from the GDAL DEM utility.

Default menu: *Raster* ► *Analysis*

参数

标注	名称	类型	描述
输入图层	INPUT	[栅格]	Input elevation raster layer
Band number	BAND	[raster band] Default: 1	The number of the band to use as elevation
Return trigonometric angle instead of azimuth	TRIG_ANGLE	[boolean] Default: False	Activating the trigonometric angle results in different categories: 0° (East), 90° (North), 180° (West), 270° (South).
Return 0 for flat instead of -9999	ZERO_FLAT	[boolean] Default: False	Activating this option will insert a 0-value for the value -9999 on flat areas.
Compute edges	COMPUTE_EDGES	[boolean] Default: False	Generates edges from the elevation raster
Use Zevenbergen&Thorne formula instead of the Horn's one	ZEVENBERGEN	[boolean] Default: False	Activates Zevenbergen&Thorne formula for smooth landscapes
坡向	OUTPUT	[栅格] 默认: [保存到临时文件]	Output raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Additional command-line parameters 可选	EXTRA	[string] Default: Not set	Add extra GDAL command line options

输出

标注	名称	类型	描述
坡向	OUTPUT	[栅格]	Output raster with angle values in degrees

Python 代码

Algorithm ID: `gdal:aspect`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Color relief

Generates a color relief map from any GDAL-supported elevation raster. Color reliefs can particularly be used to depict elevations. The Algorithm outputs a 4-band raster with values computed from the elevation and a text-based color configuration file. By default, the colors between the given elevation values are blended smoothly and the result is a nice colorized elevation raster.

This algorithm is derived from the GDAL DEM utility.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	Input elevation raster layer
Band number	`BAND`	[raster band] Default: 1	The number of the band to use as elevation
Compute edges	`COMPUTE_EDGES`	[boolean] Default: False	Generates edges from the elevation raster
Color configuration file	`COLOR_TABLE`	[file]	A text-based color configuration file
Matching mode	`MATCH_MODE`	[enumeration] Default: 2	One of: • 0 --- Use strict color matching • 1 --- Use closest RGBA quadruples • 2 --- Use smoothly blended colours
Additional creation options 可选	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Additional command-line parameters 可选	`EXTRA`	[string] Default: Not set	Add extra GDAL command line options
Color relief	`OUTPUT`	[栅格] 默认: [保存到临时文件]	Output raster layer. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Color relief	`OUTPUT`	[栅格]	A 4-band output raster

Python 代码

Algorithm ID: `gdal:colorrelief`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Fill NoData

Fill raster regions with NoData values by interpolation from edges. The values for the NoData regions are calculated by the surrounding pixel values using inverse distance weighting. After the interpolation a smoothing of the results takes place. Input can be any GDAL-supported raster layer. This algorithm is generally suitable for interpolating missing regions of fairly continuously varying rasters (such as elevation models for instance). It is also suitable for filling small holes and cracks in more irregularly varying images (like airphotos). It is generally not so great for interpolating a raster from sparse point data.

This algorithm is derived from the GDAL fillnodata utility.

Default menu: *Raster* ► *Analysis*

参数

Basic parameters

标注	名称	类型	描述
标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	Input raster layer
Band number	`BAND`	[raster band] Default: 1	The band to operate on. NoData values must be represented by the value 0.
Maximum distance (in pixels) to search out for values to interpolate	`DISTANCE`	[numeric: integer] Default: 10	The number of pixels to search in all directions to find values to interpolate from
Number of smoothing iterations to run after the interpolation	`ITERATIONS`	[numeric: integer] Default: 0	The number of 3x3 filter passes to run (0 or more) to smoothen the results of the interpolation.
Validity mask	`MASK_LAYER`	[栅格]	A raster layer that defines the areas to fill.

continues on next page

Table 24.852 – continued from previous page

标注	名称	类型	描述
Filled	OUTPUT	[栅格] 默认: [保存到临时文件]	Specification of the output raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Additional command-line parameters 可选	EXTRA	[string] Default: Not set	Add extra GDAL command line options

输出

标注	名称	类型	描述
Filled	OUTPUT	[栅格]	Output raster

Python 代码

Algorithm ID: `gdal:fillnodata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

Grid (Data metrics)

Computes some data metrics using the specified window and output grid geometry.

This algorithm is derived from the GDAL grid utility.

Default menu: *Raster* ► *Analysis*

See also

GDAL grid tutorial

参数

Basic parameters

标注	名称	类型	描述
Point layer	`INPUT`	[vector: point]	Input point vector layer
Data metric to use	`METRIC`	[enumeration] Default: 0	One of: • 0 --- Minimum, minimum value found in grid node search ellipse • 1 --- Maximum, maximum value found in grid node search ellipse • 2 --- Range, a difference between the minimum and maximum values found in grid node search ellipse • 3 --- Count, a number of data points found in grid node search ellipse • 4 --- Average distance, an average distance between the grid node (center of the search ellipse) and all of the data points found in grid node search ellipse • 5 --- Average distance between points, an average distance between the data points found in grid node search ellipse. The distance between each pair of points within ellipse is calculated and average of all distances is set as a grid node value
The first radius of search ellipse	`RADIUS_1`	[numeric: double] Default: 0.0	The first radius (X axis if rotation angle is 0) of the search ellipse
The second radius of search ellipse	`RADIUS_2`	[numeric: double] Default: 0.0	The second radius (Y axis if rotation angle is 0) of the search ellipse
Angle of search ellipse rotation in degrees (counter clockwise)	`ANGLE`	[numeric: double] Default: 0.0	Angle of ellipse rotation in degrees. Ellipse rotated counter clockwise.
Minimum number of data points to use	`MIN_POINTS`	[numeric: double] Default: 0.0	Minimum number of data points to average. If less amount of points found the grid node considered empty and will be filled with NoData marker.

continues on next page

Table 24.855 – continued from previous page

标注	名称	类型	描述
NoData	NODATA	[numeric: double] Default: 0.0	NoData marker to fill empty points
Interpolated (data metrics)	OUTPUT	[栅格] 默认: [保存到临时文件]	Specify the output raster layer with interpolated values. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Z value from field 可选	Z_FIELD	[tablefield: numeric]	Field for the interpolation
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Additional command-line parameters 可选	EXTRA	[string] Default: Not set	Add extra GDAL command line options
Output data type	DATA_TYPE	[enumeration] Default: 5	Defines the data type of the output raster file. Options: • 0 --- Byte (Eight bit unsigned integer (quint8)) • 1 --- Int16 (Sixteen bit signed integer (qint16)) • 2 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 3 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 4 --- Int32 (Thirty two bit signed integer (qint32)) • 5 --- Float32 (Thirty two bit floating point (float)) • 6 --- Float64 (Sixty four bit floating point (double)) • 7 --- CInt16 (Complex Int16) • 8 --- CInt32 (Complex Int32) • 9 --- CFloat32 (Complex Float32) • 10 --- CFloat64 (Complex Float64) • 11 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)

输出

标注	名称	类型	描述
Interpolated (data metrics)	`OUTPUT`	[栅格]	Output raster with interpolated values

Python 代码

Algorithm ID: `gdal:griddatametrics`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Grid (IDW with nearest neighbor searching)

Computes the Inverse Distance to a Power gridding combined to the nearest neighbor method. Ideal when a maximum number of data points to use is required.

This algorithm is derived from the GDAL grid utility.

See also

GDAL grid tutorial

参数

标注	名称	类型	描述
Point layer	`INPUT`	[vector: point]	Input point vector layer
Weighting power	`POWER`	[numeric: double] Default: 2.0	Weighting power
Smoothing	`SMOOTHING`	[numeric: double] Default: 0.0	Smoothing parameter
The radius of the search circle	`RADIUS`	[numeric: double] 默认值：1.0	The radius of the search circle
Maximum number of data points to use	`MAX_POINTS`	[numeric: integer] Default: 12	Do not search for more points than this number.
Minimum number of data points to use	`MIN_POINTS`	[numeric: integer] Default: 0	Minimum number of data points to average. If less amount of points found the grid node considered empty and will be filled with NoData marker.
NoData	`NODATA`	[numeric: double] Default: 0.0	NoData marker to fill empty points

continues on next page

Table 24.858 – continued from previous page

标注	名称	类型	描述
Interpolated (IDW with NN search)	OUTPUT	[栅格] 默认: [保存到临时文件]	Specify the output raster layer with interpolated values. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Z value from field 可选	Z_FIELD	[tablefield: numeric]	Field for the interpolation
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Additional command-line parameters 可选	EXTRA	[string] Default: Not set	Add extra GDAL command line options
Output data type	DATA_TYPE	[enumeration] Default: 5	Defines the data type of the output raster file. Options: • 0 --- Byte (Eight bit unsigned integer (quint8)) • 1 --- Int16 (Sixteen bit signed integer (qint16)) • 2 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 3 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 4 --- Int32 (Thirty two bit signed integer (qint32)) • 5 --- Float32 (Thirty two bit floating point (float)) • 6 --- Float64 (Sixty four bit floating point (double)) • 7 --- CInt16 (Complex Int16) • 8 --- CInt32 (Complex Int32) • 9 --- CFloat32 (Complex Float32) • 10 --- CFloat64 (Complex Float64) • 11 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)

输出

标注	名称	类型	描述
Interpolated (IDW with NN search)	`OUTPUT`	[栅格]	Output raster with interpolated values

Python 代码

Algorithm ID: `gdal:gridinversedistancenearestneighbor`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Grid (Inverse distance to a power)

The Inverse Distance to a Power gridding method is a weighted average interpolator.

You should supply the input arrays with the scattered data values including coordinates of every data point and output grid geometry. The function will compute interpolated value for the given position in output grid.

This algorithm is derived from the GDAL grid utility.

Default menu: *Raster* ► *Analysis*

See also

GDAL grid tutorial

参数

Basic parameters

标注	名称	类型	描述
Point layer	`INPUT`	[vector: point]	Input point vector layer
Weighting power	`POWER`	[numeric: double] Default: 2.0	Weighting power
Smothing	`SMOOTHING`	[numeric: double] Default: 0.0	Smoothing parameter
The first radius of search ellipse	`RADIUS_1`	[numeric: double] Default: 0.0	The first radius (X axis if rotation angle is 0) of the search ellipse
The second radius of search ellipse	`RADIUS_2`	[numeric: double] Default: 0.0	The second radius (Y axis if rotation angle is 0) of the search ellipse
Angle of search ellipse rotation in degrees (counter clockwise)	`ANGLE`	[numeric: double] Default: 0.0	Angle of ellipse rotation in degrees. Ellipse rotated counter clockwise.

continues on next page

Table 24.861 – continued from previous page

标注	名称	类型	描述
Maximum number of data points to use	MAX_POINTS	[numeric: integer] Default: 0	Do not search for more points than this number.
Minimum number of data points to use	MIN_POINTS	[numeric: integer] Default: 0	Minimum number of data points to average. If less amount of points found the grid node considered empty and will be filled with NoData marker.
NoData	NODATA	[numeric: double] Default: 0.0	NoData marker to fill empty points
Interpolated (IDW)	OUTPUT	[栅格] 默认: [保存到临时文件]	Specify the output raster layer with interpolated values. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Z value from field 可选	Z_FIELD	[tablefield: numeric]	Field for the interpolation
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Additional command-line parameters 可选	EXTRA	[string] Default: Not set	Add extra GDAL command line options

continues on next page

Table 24.862 – continued from previous page

标注	名称	类型	描述
Output data type	DATA_TYPE	[enumeration] Default: 5	Defines the data type of the output raster file. Options: • 0 --- Byte (Eight bit unsigned integer (quint8)) • 1 --- Int16 (Sixteen bit signed integer (qint16)) • 2 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 3 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 4 --- Int32 (Thirty two bit signed integer (qint32)) • 5 --- Float32 (Thirty two bit floating point (float)) • 6 --- Float64 (Sixty four bit floating point (double)) • 7 --- CInt16 (Complex Int16) • 8 --- CInt32 (Complex Int32) • 9 --- CFloat32 (Complex Float32) • 10 --- CFloat64 (Complex Float64) • 11 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)

输出

标注	名称	类型	描述
Interpolated (IDW)	OUTPUT	[栅格]	Output raster with interpolated values

Python 代码

Algorithm ID: `gdal:gridinversedistance`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Grid (Linear)

The Linear method perform linear interpolation by computing a Delaunay triangulation of the point cloud, finding in which triangle of the triangulation the point is, and by doing linear interpolation from its barycentric coordinates within the triangle. If the point is not in any triangle, depending on the radius, the algorithm will use the value of the nearest point or the NoData value.

This algorithm is derived from the GDAL grid utility.

参数

Basic parameters

标注	名称	类型	描述
Point layer	`INPUT`	[vector: point]	Input point vector layer
Search distance	`RADIUS`	[numeric: double] Default: -1.0	In case the point to be interpolated does not fit into a triangle of the Delaunay triangulation, use that maximum distance to search a nearest neighbour, or use NoData otherwise. If set to `-1`, the search distance is infinite. If set to `0`, NoData value will be used.
NoData	`NODATA`	[numeric: double] Default: 0.0	NoData marker to fill empty points
Interpolated (Linear)	`OUTPUT`	[栅格] 默认: [保存到临时文件]	Specify the output raster layer with interpolated values. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Z value from field 可选	`Z_FIELD`	[tablefield: numeric]	Field for the interpolation
Additional creation options 可选	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (`|`).
Additional command-line parameters 可选	`EXTRA`	[string] Default: Not set	Add extra GDAL command line options

continues on next page

Table 24.865 – continued from previous page

标注	名称	类型	描述
Output data type	DATA_TYPE	[enumeration] Default: 5	Defines the data type of the output raster file. Options: • 0 --- Byte (Eight bit unsigned integer (quint8)) • 1 --- Int16 (Sixteen bit signed integer (qint16)) • 2 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 3 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 4 --- Int32 (Thirty two bit signed integer (qint32)) • 5 --- Float32 (Thirty two bit floating point (float)) • 6 --- Float64 (Sixty four bit floating point (double)) • 7 --- CInt16 (Complex Int16) • 8 --- CInt32 (Complex Int32) • 9 --- CFloat32 (Complex Float32) • 10 --- CFloat64 (Complex Float64) • 11 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)

输出

标注	名称	类型	描述
Interpolated (Linear)	OUTPUT	[栅格]	Output raster with interpolated values

Python 代码

Algorithm ID: `gdal:gridlinear`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Grid (Moving average)

The Moving Average is a simple data averaging algorithm. It uses a moving window of elliptic form to search values and averages all data points within the window. Search ellipse can be rotated by specified angle, the center of ellipse located at the grid node. Also the minimum number of data points to average can be set, if there are not enough points in window, the grid node considered empty and will be filled with specified NoData value.

This algorithm is derived from the GDAL grid utility.

Default menu: *Raster* ► *Analysis*

See also

GDAL grid tutorial

参数

Basic parameters

标注	名称	类型	描述
Point layer	`INPUT`	[vector: point]	Input point vector layer
The first radius of search ellipse	`RADIUS_1`	[numeric: double] Default: 0.0	The first radius (X axis if rotation angle is 0) of the search ellipse
The second radius of search ellipse	`RADIUS_2`	[numeric: double] Default: 0.0	The second radius (Y axis if rotation angle is 0) of the search ellipse
Angle of search ellipse rotation in degrees (counter clockwise)	`ANGLE`	[numeric: double] Default: 0.0	Angle of ellipse rotation in degrees. Ellipse rotated counter clockwise.
Minimum number of data points to use	`MIN_POINTS`	[numeric: integer] Default: 0	Minimum number of data points to average. If less amount of points found the grid node considered empty and will be filled with NoData marker.
NoData	`NODATA`	[numeric: double] Default: 0.0	NoData marker to fill empty points
Interpolated (moving average)	`OUTPUT`	[栅格] 默认: [保存到临时文件]	Specify the output raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Z value from field 可选	`Z_FIELD`	[tablefield: numeric]	Field for the interpolation

continues on next page

Table 24.868 – continued from previous page

标注	名称	类型	描述
Additional creation options 可选	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Additional command-line parameters 可选	`EXTRA`	[string] Default: Not set	Add extra GDAL command line options
Output data type	`DATA_TYPE`	[enumeration] Default: 5	Defines the data type of the output raster file. Options: • 0 --- Byte (Eight bit unsigned integer (quint8)) • 1 --- Int16 (Sixteen bit signed integer (qint16)) • 2 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 3 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 4 --- Int32 (Thirty two bit signed integer (qint32)) • 5 --- Float32 (Thirty two bit floating point (float)) • 6 --- Float64 (Sixty four bit floating point (double)) • 7 --- CInt16 (Complex Int16) • 8 --- CInt32 (Complex Int32) • 9 --- CFloat32 (Complex Float32) • 10 --- CFloat64 (Complex Float64) • 11 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)

输出

标注	名称	类型	描述
Interpolated (moving average)	`OUTPUT`	[栅格]	Output raster with interpolated values

Python 代码

Algorithm ID: `gdal:gridaverage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Grid (Nearest neighbor)

The Nearest Neighbor method doesn't perform any interpolation or smoothing, it just takes the value of nearest point found in grid node search ellipse and returns it as a result. If there are no points found, the specified NoData value will be returned.

This algorithm is derived from the GDAL grid utility.

Default menu: *Raster* ► *Analysis*

See also

GDAL grid tutorial

参数

Basic parameters

标注	名称	类型	描述
Point layer	`INPUT`	[vector: point]	Input point vector layer
The first radius of search ellipse	`RADIUS_1`	[numeric: double] Default: 0.0	The first radius (X axis if rotation angle is 0) of the search ellipse
The second radius of search ellipse	`RADIUS_2`	[numeric: double] Default: 0.0	The second radius (Y axis if rotation angle is 0) of the search ellipse
Angle of search ellipse rotation in degrees (counter clockwise)	`ANGLE`	[numeric: double] Default: 0.0	Angle of ellipse rotation in degrees. Ellipse rotated counter clockwise.
NoData	`NODATA`	[numeric: double] Default: 0.0	NoData marker to fill empty points
Interpolated (Nearest neighbour)	`OUTPUT`	[栅格] 默认: [保存到临时文件]	Specify the output raster layer with interpolated values. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Z value from field 可选	`Z_FIELD`	[tablefield: numeric]	Field for the interpolation
Additional creation options 可选	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Additional command-line parameters 可选	`EXTRA`	[string] Default: Not set	Add extra GDAL command line options
Output data type	`DATA_TYPE`	[enumeration] Default: 5	Defines the data type of the output raster file. Options: • 0 --- Byte (Eight bit unsigned integer (quint8)) • 1 --- Int16 (Sixteen bit signed integer (qint16)) • 2 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 3 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 4 --- Int32 (Thirty two bit signed integer (qint32)) • 5 --- Float32 (Thirty two bit floating point (float)) • 6 --- Float64 (Sixty four bit floating point (double)) • 7 --- CInt16 (Complex Int16) • 8 --- CInt32 (Complex Int32) • 9 --- CFloat32 (Complex Float32) • 10 --- CFloat64 (Complex Float64) • 11 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)

输出

标注	名称	类型	描述
Interpolated (Nearest neighbour)	`OUTPUT`	[栅格]	Output raster with interpolated values

Python 代码

Algorithm ID: `gdal:gridnearestneighbor`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

山体阴影

Outputs a raster with a nice shaded relief effect. It' s very useful for visualizing the terrain. You can optionally specify the azimuth and altitude of the light source, a vertical exaggeration factor and a scaling factor to account for differences between vertical and horizontal units.

This algorithm is derived from the GDAL DEM utility .

Default menu: *Raster* ► *Analysis*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	Input Elevation raster layer
Band number	`BAND`	[raster band] Default: 1	Band containing the elevation information
Z factor (vertical exaggeration)	`Z_FACTOR`	[numeric: double] 默认值：1.0	The factor exaggerates the height of the output elevation raster
Scale (ratio of vertical units to horizontal)	`SCALE`	[numeric: double] 默认值：1.0	The ratio of vertical units to horizontal units
Azimuth of the light	`AZIMUTH`	[numeric: double] Default: 315.0	Defines the azimuth of the light shining on the elevation raster in degrees. If it comes from the top of the raster the value is 0, if it comes from the east it is 90 a.s.o.
Altitude of the light	`ALTITUDE`	[numeric: double] Default: 45.0	Defines the altitude of the light, in degrees. 90 if the light comes from above the elevation raster, 0 if it is raking light.
Compute edges	`COMPUTE_EDGES`	[boolean] Default: False	Generates edges from the elevation raster
Use Zevenbergen&Thorne formula (instead of the Horn's one)	`ZEVENBERGEN`	[boolean] Default: False	Activates Zevenbergen&Thorne formula for smooth landscapes
Combined shading	`COMBINED`	[boolean] Default: False	
Multidirectional shading	`MULTIDIRECTION`	[boolean] Default: False	

continues on next page

Table 24.873 – continued from previous page

标注	名称	类型	描述
山体阴影	OUTPUT	[栅格] 默认: [保存到临时文件]	Specify the output raster layer with interpolated values. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Additional command-line parameters 可选	EXTRA	[string] Default: Not set	Add extra GDAL command line options

输出

标注	名称	类型	描述
山体阴影	OUTPUT	[栅格]	Output raster with interpolated values

Python 代码

Algorithm ID: `gdal:hillshade`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

Near black

Converts nearly black/white borders to black.

This algorithm will scan an image and try to set all pixels that are nearly or exactly black, white or one or more custom colors around the collar to black or white. This is often used to ”fix up” lossy compressed airphotos so that color pixels can be treated as transparent when mosaicking.

This algorithm is derived from the GDAL nearblack utility.

Default menu: *Raster* ► *Analysis*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	Input Elevation raster layer
How far from black (white)	`NEAR`	[numeric: integer] Default: 15	Select how far from black, white or custom colors the pixel values can be and still considered near black, white or custom color.
Search for nearly white pixels instead of nearly black	`WHITE`	[boolean] Default: False	Search for nearly white (255) pixels instead of nearly black pixels
Nearblack	`OUTPUT`	[栅格] 默认: [保存到临时文件]	Specify the output raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Additional command-line parameters 可选	`EXTRA`	[string] Default: Not set	Add extra GDAL command line options

输出

标注	名称	类型	描述
Nearblack	OUTPUT	[栅格]	Output raster

Python 代码

Algorithm ID: `gdal:nearblack`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Proximity (raster distance)

Generates a raster proximity map indicating the distance from the center of each pixel to the center of the nearest pixel identified as a target pixel. Target pixels are those in the source raster for which the raster pixel value is in the set of target pixel values.

This algorithm is derived from the GDAL proximity utility.

Default menu: *Raster* ► *Analysis*

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[栅格]	Input Elevation raster layer
Band number	BAND	[raster band] Default: 1	Band containing the elevation information
List of target pixels 可选	VALUES	[string] Default: ''	Comma-separated list of pixel values in the source image to consider as target pixels. If not specified, all non-zero pixels will be considered target pixels.
Distance units	UNITS	[enumeration] Default: 1	Indicate whether distances generated should be in pixel or georeferenced coordinates. One of: • 0 --- Georeferenced coordinates • 1 --- Pixel coordinates
The maximum distance to generate 可选	MAX_DISTANCE	[numeric: double] Default: 0.0	The maximum distance to be generated. The NoData value will be used for pixels beyond this distance. Distance is interpreted according to the value of the *Distance units* parameter.

continues on next page

Table 24.879 – continued from previous page

标注	名称	类型	描述
Value to apply to pixels within the maximum distance 可选	REPLACE	[numeric: double] Default: 0.0	Specify a value to apply to all pixels that are within the maximum distance from target pixels (including the target pixels) instead of a distance value.
NoData value to use for the destination proximity raster 可选	NODATA	[numeric: double] Default: 0.0	Specify the NoData value to use for the pixels beyond the maximum distance. If a NoData value is not provided, the output band will be queried for its NoData value. If the output band does not have a NoData value, then the value 65535 will be used.
Proximity map	OUTPUT	[栅格] 默认: [保存到临时文件]	Specify the output raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Additional command-line parameters 可选	EXTRA	[string] Default: Not set	Add extra GDAL command line options

continues on next page

Table 24.880 – continued from previous page

标注	名称	类型	描述
Output data type	DATA_TYPE	[enumeration] Default: 5	Defines the data type of the output raster file. Options: • 0 --- Byte (Eight bit unsigned integer (quint8)) • 1 --- Int16 (Sixteen bit signed integer (qint16)) • 2 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 3 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 4 --- Int32 (Thirty two bit signed integer (qint32)) • 5 --- Float32 (Thirty two bit floating point (float)) • 6 --- Float64 (Sixty four bit floating point (double)) • 7 --- CInt16 (Complex Int16) • 8 --- CInt32 (Complex Int32) • 9 --- CFloat32 (Complex Float32) • 10 --- CFloat64 (Complex Float64) • 11 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)

输出

标注	名称	类型	描述
Proximity map	OUTPUT	[栅格]	Output raster

Python 代码

Algorithm ID: `gdal:proximity`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Roughness

Outputs a single-band raster with values computed from the elevation. Roughness is the degree of irregularity of the surface. It's calculated by the largest inter-cell difference of a central pixel and its surrounding cell. The determination of the roughness plays a role in the analysis of terrain elevation data, it's useful for calculations of the river morphology, in climatology and physical geography in general.

This algorithm is derived from the GDAL DEM utility.

Default menu: *Raster* ► *Analysis*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	Input elevation raster layer
Band number	`BAND`	[raster band] Default: 1	The number of the band to use as elevation
Compute edges	`COMPUTE_EDGES`	[boolean] Default: False	Generates edges from the elevation raster
Roughness	`OUTPUT`	[栅格] 默认: [保存到临时文件]	Specify the output raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Roughness	`OUTPUT`	[栅格]	Single-band output roughness raster. The value -9999 is used as NoData value.

Python 代码

Algorithm ID: `gdal:roughness`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Sieve

Removes raster polygons smaller than a provided threshold size (in pixels) and replaces them with the pixel value of the largest neighbour polygon. It is useful if you have a large amount of small areas on your raster map.

This algorithm is derived from the GDAL sieve utility.

Default menu: *Raster* ► *Analysis*

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[栅格]	Input elevation raster layer
Threshold	THRESHOLD	[numeric: integer] Default: 10	Only raster polygons smaller than this size will be removed
Use 8-connectedness	EIGHT_CONNECTE	[boolean] Default: False	If False, consider as neighbours only the (4) pixels that connect the edges horizontally or vertically. If True, every pixel that touches an edge or a corner is considered as neighbour.
Do not use the default validity mask for the input band	NO_MASK	[boolean] Default: False	
Validity mask 可选	MASK_LAYER	[栅格]	Validity mask to use instead of the default
Sieved	OUTPUT	[栅格] 默认: [保存到临时文件]	Specify the output raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional command-line parameters 可选	EXTRA	[string] Default: Not set	Add extra GDAL command line options

输出

标注	名称	类型	描述
Sieved	`OUTPUT`	[栅格]	Output raster layer.

Python 代码

Algorithm ID: `gdal:sieve`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

坡度

Generates a slope map from any GDAL-supported elevation raster. Slope is the angle of inclination to the horizontal. You have the option of specifying the type of slope value you want: degrees or percent slope.

This algorithm is derived from the GDAL DEM utility.

Default menu: *Raster ► Analysis*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	Input Elevation raster layer
Band number	`BAND`	[raster band] Default: 1	Band containing the elevation information
Ratio of vertical units to horizontal	`SCALE`	[numeric: double] 默认值：1.0	The ratio of vertical units to horizontal units
Express slope as percent (instead of degrees)	`AS_PERCENT`	[boolean] Default: False	Express slope as percent instead of degrees
Compute edges	`COMPUTE_EDGES`	[boolean] Default: False	Generates edges from the elevation raster
Use Zevenbergen&Thorne formula (instead of the Horn's one)	`ZEVENBERGEN`	[boolean] Default: False	Activates Zevenbergen&Thorne formula for smooth landscapes
Slope	`OUTPUT`	[栅格] 默认：[保存到临时文件]	Specify the output raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Additional command-line parameters 可选	EXTRA	[string] Default: Not set	Add extra GDAL command line options

输出

标注	名称	类型	描述
Slope	OUTPUT	[栅格]	Output raster

Python 代码

Algorithm ID: `gdal:slope`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Terrain Ruggedness Index (TRI)

Outputs a single-band raster with values computed from the elevation. TRI stands for Terrain Ruggedness Index, which is defined as the mean difference between a central pixel and its surrounding cells.

This algorithm is derived from the GDAL DEM utility.

Default menu: *Raster* ► *Analysis*

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[栅格]	Input elevation raster layer
Band number	BAND	[raster band] Default: 1	The number of the band to use as elevation

continues on next page

Table 24.891 – continued from previous page

标注	名称	类型	描述
Compute edges	COMPUTE_EDGES	[boolean] Default: False	Generates edges from the elevation raster
Terrain Ruggedness Index	OUTPUT	[栅格] 默认: [保存到临时文件]	Specify the output raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Terrain Ruggedness Index	OUTPUT	[栅格]	Output ruggedness raster. The value -9999 is used as NoData value.

Python 代码

Algorithm ID: `gdal:triterrainruggednessindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Topographic Position Index (TPI)

Outputs a single-band raster with values computed from the elevation. TPI stands for Topographic Position Index, which is defined as the difference between a central pixel and the mean of its surrounding cells.

This algorithm is derived from the GDAL DEM utility.

Default menu: *Raster* ► *Analysis*

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[栅格]	Input elevation raster layer
Band number	BAND	[raster band] Default: 1	The number of the band to use for elevation values
Compute edges	COMPUTE_EDGES	[boolean] Default: False	Generates edges from the elevation raster
Terrain Ruggedness Index	OUTPUT	[栅格] 默认: [保存到临时文件]	Specify the output raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

输出

标注	名称	类型	描述
Terrain Ruggedness Index	OUTPUT	[栅格]	Output raster.

Python 代码

Algorithm ID: `gdal:tpitopographicpositionindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

24.2.2 Raster conversion

gdal2xyz

Converts raster data to XYZ ASCII file format.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	Raster layer to convert
Band number	`BAND`	[raster band] Default: The first band of the input layer	If the raster is multiband, choose the band you want to convert
Input pixel value to treat as NoData 可选	`NODATA_INPUT`	[numeric: integer] Default: Not set	Input pixel value to treat as "NoData" **Attention** Using this parameter requires QGIS installed with GDAL >= 3.7 (see *Help* ► *About* menu).
Destination NoData 可选	`NODATA_OUTPUT`	[numeric: integer] Default: Not set	Assign specified "NoData" value to output. **Attention** Using this parameter requires QGIS installed with GDAL >= 3.7 (see *Help* ► *About* menu).
Do not output NoData values	`SKIP_NODATA`	[boolean] Default: False	Do not output "NoData" values. **Attention** Using this parameter requires QGIS installed with GDAL >= 3.3 (see *Help* ► *About* menu).
Output comma-separated values	`CSV`	[boolean] Default: False	Sets whether the output file should be of type comma-separated values (csv).
XYZ ASCII file	`OUTPUT`	[file] 默认: [保存到临时文件]	Specification of the output file. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
XYZ ASCII file	`INPUT`	[table]	Table file containing the values exported from the raster band.

Python 代码

Algorithm ID: `gdal:gdal2xyz`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

PCT to RGB

Converts an 8 bit paletted image to a 24 bit RGB. It will convert a pseudocolor band from the input file to an RGB file of the desired format.

This algorithm is derived from the GDAL pct2rgb utility.

Default menu: *Raster* ► *Conversion*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	Input 8 bit raster image
Band number	`BAND`	[raster band] Default: The first band of the input layer	If the raster is multiband, choose the band you want to convert
Generate a RGBA file	`RGBA`	[boolean] Default: False	Sets whether the output file should be of type RGBA.
PCT to RGB	`OUTPUT`	[file] 默认：[保存到临时文件]	Specification of the output file. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
PCT to RGB	`OUTPUT`	[栅格]	24 bit RGB raster image

Python 代码

Algorithm ID: `gdal:pcttorgb`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Polygonize (raster to vector)

Creates vector polygons for all connected regions of pixels in the raster sharing a common pixel value. Each polygon is created with an attribute indicating the pixel value of that polygon.

This algorithm is derived from the GDAL polygonize utility.

Default menu: *Raster* ► *Conversion*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	Input raster layer
Band number	`BAND`	[raster band] Default: The first band of the input layer	If the raster is multiband, choose the band you want to use
Name of the field to create	`FIELD`	[string] Default: 'DN'	Specify the field name for the attributes of the connected regions.
Use 8-connectedness	`EIGHT_CONNECTE`	[boolean] Default: False	If not set, raster cells must have a common border to be considered connected (*4-connected*). If set, touching raster cells are also considered connected (*8-connected*).
Vectorized	`OUTPUT`	[vector: polygon] 默认：[保存到临时文件]	Specification of the output (polygon) vector layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional command-line parameters 可选	`EXTRA`	[string] Default: Not set	Add extra GDAL command line options

输出

标注	名称	类型	描述
Vectorized	OUTPUT	[vector: polygon]	Output vector layer

Python 代码

Algorithm ID: `gdal:polygonize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Rearrange bands

Creates a new raster using selected band(s) from a given raster layer. The algorithm also makes it possible to reorder the bands for the newly-created raster.

This algorithm is derived from the GDAL translate utility.

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[栅格]	Input raster layer
Selected band(s)	BANDS	[raster band] [list] Default: Not set	Ordered list of the bands to use to create the new raster
Converted	OUTPUT	[栅格] Default: [Save to temporary file]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

continues on next page

Table 24.905 – continued from previous page

标注	名称	类型	描述
Output data type	DATA_TYPE	[enumeration] Default: 0	Defines the data type of the output raster file. Options: • 0 --- Use Input Layer Data Type • 1 --- Byte (Eight bit unsigned integer (quint8)) • 2 --- Int16 (Sixteen bit signed integer (qint16)) • 3 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 4 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 5 --- Int32 (Thirty two bit signed integer (qint32)) • 6 --- Float32 (Thirty two bit floating point (float)) • 7 --- Float64 (Sixty four bit floating point (double)) • 8 --- CInt16 (Complex Int16) • 9 --- CInt32 (Complex Int32) • 10 --- CFloat32 (Complex Float32) • 11 --- CFloat64 (Complex Float64) • 12 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)

输出

标注	名称	类型	描述
Converted	OUTPUT	[栅格]	Output raster layer with rearranged bands.

Python 代码

Algorithm ID: `gdal:rearrange_bands`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

RGB to PCT

Converts a 24 bit RGB image into a 8 bit paletted. Computes an optimal pseudo-color table for the given RGB-image using a median cut algorithm on a downsampled RGB histogram. Then it converts the image into a pseudo-colored image using the color table. This conversion utilizes Floyd-Steinberg dithering (error diffusion) to maximize output image visual quality.

If you want to classify a raster map and want to reduce the number of classes it can be helpful to downsample your image with this algorithm before.

This algorithm is derived from the GDAL rgb2pct utility.

Default menu: *Raster* ► *Conversion*

参数

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	Input (RGB) raster layer
Number of colors	`NCOLORS`	[numeric: integer] Default: 2	The number of colors the resulting image will contain. A value from 2-256 is possible.
RGB to PCT	`OUTPUT`	[栅格] 默认: [保存到临时文件]	Specification of the output raster. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
RGB to PCT	`OUTPUT`	[栅格]	Output raster layer.

Python 代码

Algorithm ID: `gdal:rgbtopct`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Translate (convert format)

Converts raster data between different formats.

This algorithm is derived from the GDAL translate utility.

Default menu: *Raster* ► *Conversion*

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[栅格]	Input raster layer
Override the projection of the output file 可选	TARGET_CRS	[crs]	Specify a projection for the output file
Assign a specified NoData value to output bands 可选	NODATA	[numeric: double] Default: Not set	Defines the value to use for NoData in the output raster
Copy all subdatasets of this file to individual output files	COPY_SUBDATASE	[boolean] Default: False	Create individual files for subdatasets
Converted	OUTPUT	[栅格] 默认: [保存到临时文件]	Specification of the output (translated) raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Additional command-line parameters 可选	EXTRA	[string] Default: Not set	Add extra GDAL command line options

continues on next page

Table 24.910 – continued from previous page

标注	名称	类型	描述
Output data type	DATA_TYPE	[enumeration] Default: 0	Defines the data type of the output raster file. Options: • 0 --- Use Input Layer Data Type • 1 --- Byte (Eight bit unsigned integer (quint8)) • 2 --- Int16 (Sixteen bit signed integer (qint16)) • 3 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 4 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 5 --- Int32 (Thirty two bit signed integer (qint32)) • 6 --- Float32 (Thirty two bit floating point (float)) • 7 --- Float64 (Sixty four bit floating point (double)) • 8 --- CInt16 (Complex Int16) • 9 --- CInt32 (Complex Int32) • 10 --- CFloat32 (Complex Float32) • 11 --- CFloat64 (Complex Float64) • 12 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)

输出

标注	名称	类型	描述
Converted	OUTPUT	[栅格]	Output (translated) raster layer.

Python 代码

Algorithm ID: `gdal:translate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.2.3 Raster extraction

Clip raster by extent

Clips any GDAL-supported raster file to a given extent.

This algorithm is derived from the GDAL translate utility.

Default menu: *Raster* ► *Extraction*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	The input raster
Clipping extent	`EXTENT`	[范围]	Extent that should be used for the output raster. Only pixels within the specified bounding box will be included in the output. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Override the projection for the output file	`OVERCRS`	[boolean] Default: False	If checked, the output file is assigned the input layer CRS.
Assign a specified NoData value to output bands 可选	`NODATA`	[numeric: double] Default: Not set	Defines a value that should be inserted for the NoData values in the output raster
Clipped (extent)	`OUTPUT`	[栅格] 默认: [保存到临时文件]	Specification of the output raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Output data type	DATA_TYPE	[enumeration] Default: 0	Defines the format of the output raster file. Options: • 0 --- Use Input Layer Data Type • 1 --- Byte (Eight bit unsigned integer (quint8)) • 2 --- Int16 (Sixteen bit signed integer (qint16)) • 3 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 4 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 5 --- Int32 (Thirty two bit signed integer (qint32)) • 6 --- Float32 (Thirty two bit floating point (float)) • 7 --- Float64 (Sixty four bit floating point (double)) • 8 --- CInt16 (Complex Int16) • 9 --- CInt32 (Complex Int32) • 10 --- CFloat32 (Complex Float32) • 11 --- CFloat64 (Complex Float64) • 12 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Additional command-line parameters 可选	EXTRA	[string] Default: Not set	Add extra GDAL command line options

输出

标注	名称	类型	描述
Clipped (extent)	OUTPUT	[栅格]	Output raster layer clipped by the given extent

Python 代码

Algorithm ID: `gdal:cliprasterbyextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Clip raster by mask layer

Clips any GDAL-supported raster by a vector mask layer.

This algorithm is derived from the GDAL warp utility.

Default menu: *Raster* ► *Extraction*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	The input raster
Mask layer	`MASK`	[vector: polygon]	Vector mask for clipping the raster
Source CRS	`SOURCE_CRS`	[crs]	Set the coordinate reference to use for the input raster
Target CRS	`TARGET_CRS`	[crs]	Set the coordinate reference to use for the mask layer
Target extent 可选	`TARGET_EXTENT`	[范围]	Extent of the output file to be created Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`

continues on next page

Table 24.915 – continued from previous page

标注	名称	类型	描述
Assign a specified NoData value to output bands 可选	NODATA	[numeric: double] Default: Not set	Defines a value that should be inserted for the NoData values in the output raster
Create an output alpha band	ALPHA_BAND	[boolean] Default: False	Creates an alpha band for the result. The alpha band then includes the transparency values of the pixels.
Match the extent of the clipped raster to the extent of the mask layer	CROP_TO_CUTLIN	[boolean] Default: True	Applies the vector layer extent to the output raster if checked.
Keep resolution of input raster	KEEP_RESOLUTIO	[boolean] Default: False	The resolution of the output raster will not be changed
Set output file resolution	SET_RESOLUTION	[boolean] Default: False	Shall the output resolution (cell size) be specified
X Resolution to output bands 可选	X_RESOLUTION	[numeric: double] Default: Not set	The width of the cells in the output raster
Y Resolution to output band 可选	Y_RESOLUTION	[numeric: double] Default: Not set	The height of the cells in the output raster
Use multithreaded warping implementation	MULTITHREADING	[boolean] Default: False	Two threads will be used to process chunks of image and perform input/output operation simultaneously. Note that computation is not multithreaded itself.
Clipped (mask)	OUTPUT	[栅格] 默认: [保存到临时文件]	Specification of the output raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

continues on next page

Table 24.916 – continued from previous page

标注	名称	类型	描述
Output data type	DATA_TYPE	[enumeration] Default: 0	Defines the format of the output raster file. Options: • 0 --- Use Input Layer Data Type • 1 --- Byte (Eight bit unsigned integer (quint8)) • 2 --- Int16 (Sixteen bit signed integer (qint16)) • 3 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 4 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 5 --- Int32 (Thirty two bit signed integer (qint32)) • 6 --- Float32 (Thirty two bit floating point (float)) • 7 --- Float64 (Sixty four bit floating point (double)) • 8 --- CInt16 (Complex Int16) • 9 --- CInt32 (Complex Int32) • 10 --- CFloat32 (Complex Float32) • 11 --- CFloat64 (Complex Float64) • 12 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Additional command-line parameters 可选	EXTRA	[string] Default: Not set	Add extra GDAL command line options

输出

标注	名称	类型	描述
Clipped (mask)	OUTPUT	[栅格]	Output raster layer clipped by the vector layer

Python 代码

Algorithm ID: `gdal:cliprasterbymasklayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Contour

Extracts contour lines from any GDAL-supported elevation raster.

This algorithm is derived from the GDAL contour utility.

Default menu: *Raster* ► *Extraction*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	Input raster
Band number	`BAND`	[raster band] Default: 1	Raster band to create the contours from
Interval between contour lines	`INTERVAL`	[numeric: double] Default: 10.0	Defines the interval between the contour lines in the given units of the elevation raster (minimum value 0)
Attribute name (if not set, no elevation attribute is attached) 可选	`FIELD_NAME`	[string] Default: 'ELEV'	Provides a name for the attribute in which to put the elevation.
Offset from zero relative to which to interpret intervals 可选	`OFFSET`	[numeric: double] Default: 0.0	
Contours	`OUTPUT`	[vector: line] 默认: [保存到临时文件]	Specification of the output vector layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Produce 3D vector	`CREATE_3D`	[boolean] Default: False	Forces production of 3D vectors instead of 2D. Includes elevation at every vertex.
Treat all raster values as valid	`IGNORE_NODATA`	[boolean] Default: False	Ignores any NoData values in the dataset.
Input pixel value to treat as "NoData" 可选	`NODATA`	[numeric: double] Default: Not set	Defines a value that should be inserted for the NoData values in the output raster
Additional command-line parameters 可选	`EXTRA`	[string] Default: Not set	Add extra GDAL command line options. Refer to the corresponding GDAL utility documentation.

输出

标注	名称	类型	描述
Contours	OUTPUT	[vector: line]	Output vector layer with contour lines

Python 代码

Algorithm ID: `gdal:contour`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Contour Polygons

Extracts contour polygons from any GDAL-supported elevation raster.

This algorithm is derived from the GDAL contour utility.

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[栅格]	Input raster
Band number	BAND	[raster band] Default: 1	Raster band to create the contours from
Interval between contour lines	INTERVAL	[numeric: double] Default: 10.0	Defines the interval between the contour lines in the given units of the elevation raster (minimum value 0)
Offset from zero relative to which to interpret intervals 可选	OFFSET	[numeric: double] Default: 0.0	
Attribute name for minimum elevation of contour polygon 可选	FIELD_NAME_MIN	[string] Default: 'ELEV_MIN'	Provides a name for the attribute in which to put the minimum elevation of contour polygon. If not provided no minimum elevation attribute is attached.
Attribute name for maximum elevation of contour polygon 可选	FIELD_NAME_MAX	[string] Default: 'ELEV_MAX'	Provides a name for the attribute in which to put the maximum elevation of contour polygon. If not provided no maximum elevation attribute is attached.
Contours	OUTPUT	[vector: polygon] 默认：[保存到临时文件]	Specification of the output vector layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Produce 3D vector	CREATE_3D	[boolean] Default: False	Forces production of 3D vectors instead of 2D. Includes elevation at every vertex.
Treat all raster values as valid	IGNORE_NODATA	[boolean] Default: False	Ignores any NoData values in the dataset.
Input pixel value to treat as "No-Data" 可选	NODATA	[numeric: double] Default: Not set	Defines a value that should be inserted for the NoData values in the output raster
Additional command-line parameters 可选	EXTRA	[string] Default: Not set	Add extra GDAL command line options. Refer to the corresponding GDAL utility documentation.

输出

标注	名称	类型	描述
Contours	OUTPUT	[vector: polygon]	Output vector layer with contour polygons

Python 代码

Algorithm ID: `gdal:contour_polygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.2.4 Raster miscellaneous

Build overviews (pyramids)

To speed up rendering time of raster layers overviews (pyramids) can be created. Overviews are lower resolution copies of the data which QGIS uses depending of the level of zoom.

This algorithm is derived from the GDAL addo utility.

Default menu: *Raster* ► *Miscellaneous*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	Input raster layer
Remove all existing overviews	`CLEAN`	[boolean] Default: False	Removes existing overviews from the raster. By default these are not removed.

Advanced parameters

标注	名称	类型	描述
Overview levels	`LEVELS`	[string] Default: '2 4 8 16'	Defines the number of overview levels calculated by the original resolution of the input raster layer. By default 4 levels will be taken into consideration.
Resampling method 可选	`RESAMPLING`	[enumeration] Default: 0	Calculates the overviews with a defined resampling method. Possible resampling methods are: • 0 -- Nearest Neighbour (`nearest`) • 1 -- Average (`average`) • 2 -- Gaussian (`gauss`) • 3 -- Cubic (4x4 kernel) (`cubic`) • 4 -- Cubic B-Spline (4x4 kernel) (`cubicspline`) • 5 -- Lanczos (6x6 kernel) (`lanczos`) • 6 -- Average MP (`average_mp`) • 7 -- Average in Mag/Phase Space (`average_magphase`) • 8 -- Mode (`mode`)
Overviews format 可选	`FORMAT`	[enumeration] Default: 0	The overviews can be stored internally, or externally as GTiff or ERDAS Imagine file. By default the overviews are stored in the output raster. Possible formats methods are: • 0 -- Internal (if possible) • 1 -- External (GTiff .ovr) • 2 -- External (ERDAS Imagine .aux)
Additional command-line parameters 可选	`EXTRA`	[string] Default: Not set	Add extra GDAL command line options

输出

标注	名称	类型	描述
Pyramidized	`OUTPUT`	[栅格]	Output raster layer with overviews

Python 代码

Algorithm ID: `gdal:overviews`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Build virtual raster

Builds a VRT (Virtual Dataset) that is a mosaic of the list of input GDAL-supported rasters. With a mosaic you can merge several raster files.

This algorithm is derived from the GDAL buildvrt utility.

Default menu: *Raster ► Miscellaneous*

参数

Basic parameters

标注	名称	类型	描述
Input layers	`INPUT`	[raster] [list]	GDAL-supported raster layers.
Resolution	`RESOLUTION`	[enumeration] Default: 0	The output resolution of the mosaic. By default the average resolution of the raster files will be chosen. Options: • 0 --- Average (`average`) • 1 --- Highest (`highest`) • 2 --- Lowest (`lowest`)
Place each input file into a separate band	`SEPARATE`	[boolean] Default: False	With 'True' you can define that each raster file goes into a separated stacked band in the VRT band.
Allow projection difference	`PROJ_DIFFERENC`	[boolean] Default: False	Allows that the output bands have different projections derived from the projection of the input raster layers.
Virtual	`OUTPUT`	[栅格] 默认：[保存到临时文件]	Specification of the output raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Add alpha mask band to VRT when source raster has none	ADD_ALPHA	[boolean] Default: False	Adds an alpha mask band to the VRT when the source raster has none.
Override projection for the output file 可选	ASSIGN_CRS	[crs] Default: Not set	Overrides the projection for the output file. No reprojection is done.
Resampling algorithm	RESAMPLING	[enumeration] Default: 0	The resampling algorithm to use. Options: • 0 --- Nearest Neighbour (nearest) • 1 --- Bilinear (2x2 kernel) (bilinear) • 2 --- Cubic (4x4 kernel) (cubic) • 3 --- Cubic B-Spline (4x4 kernel) (cubicspline) • 4 --- Lanczos (6x6 kernel) (lanczos) • 5 --- Average (average) • 6 --- Mode (mode)
NoData value(s) for input bands (space separated) 可选	SRC_NODATA	[string] Default: Not set	Space separated NoData value(s) for input band(s)
Additional command-line parameters	EXTRA	[string] Default: Not set	Add extra GDAL command line options

输出

标注	名称	类型	描述
Virtual	OUTPUT	[栅格]	Output raster layer

Python 代码

Algorithm ID: gdal:buildvirtualraster

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，*算法 id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

gdal2tiles

Generates a directory with small tiles and metadata, following the OSGeo Tile Map Service Specification. See also the OpenGIS Web Map Tile Service Implementation Standard. Simple web pages with viewers based on Google Maps, OpenLayers and Leaflet are generated as well. To explore your maps on-line in the web browser, you only need to upload the generated directory onto a web server.

This algorithm also creates the necessary metadata for Google Earth (KML SuperOverlay), in case the supplied map uses `EPSG:4326` projection.

ESRI world files and embedded georeferencing is used during tile generation, but you can publish a picture without proper georeferencing too.

This algorithm is derived from the GDAL gdal2tiles utility.

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	GDAL-supported raster layer.
Tile cutting profile	`PROFILE`	[enumeration] Default: 0	One of: • 0 --- Mercator (`mercator`) • 1 --- Geodetic (`geodetic`) • 2 --- Raster (`raster`)
Zoom levels to render 可选	`ZOOM`	[string] Default: ”	
Web viewer to generate	`VIEWER`	[enumeration] Default: 0	One of: • 0 --- All (`all`) • 1 --- GoogleMaps (`google`) • 2 --- OpenLayers (`openlayers`) • 3 --- Leaflet (`leaflet`) • 4 --- None (`none`)
Title of the map 可选	`TITLE`	[string] Default: ”	
Copyright of the map	`COPYRIGHT`	[string] Default: ”	
Output directory	`OUTPUT`	[folder] Default: `[Save to temporary folder]`	Specify the output folder for the tiles. *One of*: • Save to a Temporary Directory • Save to Directory

Advanced parameters

标注	名称	类型	描述
Resampling method	RESAMPLING	[enumeration] Default: 0	The resampling algorithm to use. Options: • 0 --- Average (average) • 1 --- Nearest Neighbour (near) • 2 --- Bilinear (2x2 kernel) (bilinear) • 3 --- Cubic (4x4 kernel) (cubic) • 4 --- Cubic B-Spline (4x4 kernel) (cubicspline) • 5 --- Lanczos (6x6 kernel) (lanczos) • 6 --- Antialias (antialias)
The spatial reference system used for the source input data 可选	SOURCE_CRS	[crs] Default: Not set	
Transparency value to assign to the input data 可选	NODATA	[numeric: double] Default: 0.0	
URL address where the generated tiles are going to be published 可选	URL	[string] Default: ”	
Google Maps API key (http://code.google.c 可选	GOOGLE_KEY	[string] Default: ”	Your Google maps API key.
Bing Maps API key (https://www.bingm 可选	BING_KEY	[string] Default: ”	Your Bing maps API key.
Generate only missing files	RESUME	[boolean] Default: False	
Generate KML for Google Earth	KML	[boolean] Default: False	
Avoid automatic generation of KML files for EPSG:4326	NO_KML	[boolean] Default: False	

输出

标注	名称	类型	描述
Output directory	`OUTPUT`	[folder]	The output folder (for the tiles)

Python 代码

Algorithm ID: `gdal:gdal2tiles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Merge

Merges raster files in a simple way. Here you can use a pseudocolor table from an input raster and define the output raster type. All the images must be in the same coordinate system.

This algorithm is derived from the GDAL merge utility.

Default menu: *Raster* ► *Miscellaneous*

参数

Basic parameters

标注	名称	类型	描述
Input layers	`INPUT`	[raster] [list]	Input raster layers
Grab pseudocolor table from first layer	`PCT`	[boolean] Default: False	The pseudocolor table from the first layer will be used for the coloring
Place each input file into a separate band	`SEPARATE`	[boolean] Default: False	Place each input file into a separate band

continues on next page

Table 24.933 – continued from previous page

标注	名称	类型	描述
Output data type	DATA_TYPE	[enumeration] Default: 5	Defines the format of the output raster file. Options: • 0 --- Byte (Eight bit unsigned integer (quint8)) • 1 --- Int16 (Sixteen bit signed integer (qint16)) • 2 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 3 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 4 --- Int32 (Thirty two bit signed integer (qint32)) • 5 --- Float32 (Thirty two bit floating point (float)) • 6 --- Float64 (Sixty four bit floating point (double)) • 7 --- CInt16 (Complex Int16) • 8 --- CInt32 (Complex Int32) • 9 --- CFloat32 (Complex Float32) • 10 --- CFloat64 (Complex Float64) • 11 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Merged	OUTPUT	[栅格] 默认: [保存到临时文件]	Specification of the output raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Input pixel value to treat as "No-Data" 可选	NODATA_INPUT	[numeric: double] Default: Not set	Ignores pixels from files being merged in with this pixel value
Assign specified "NoData" value to output 可选	NODATA_OUTPUT	[numeric: double] Default: Not set	Assigns the specified NoData value to output bands.
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

continues on next page

Table 24.934 – continued from previous page

标注	名称	类型	描述
Additional command-line parameters	EXTRA	[string] Default: Not set	Add extra GDAL command line options

输出

标注	名称	类型	描述
Merged	OUTPUT	[栅格]	Output raster layer

Python 代码

Algorithm ID: `gdal:merge`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Pansharpening

Performs a pan-sharpening operation. It can create a "classic" output dataset (such as GeoTIFF), or a VRT dataset describing the pan-sharpening operation.

See GDAL Pansharpen.

参数

Basic parameters

标注	名称	类型	描述
Spectral dataset	SPECTRAL	[栅格]	Input (spectral) raster layer
Panchromatic dataset	PANCHROMATIC	[栅格]	Input (panchromatic) raster layer
Output	OUTPUT	[栅格] 默认: [保存到临时文件]	Specify the output (sharpened) raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Resampling algorithm	RESAMPLING	[enumeration] Default: 2	The resampling algorithm to use. Options: • 0 --- Nearest Neighbour (nearest) • 1 --- Bilinear (2x2 kernel) (bilinear) • 2 --- Cubic (4x4 kernel) (cubic) • 3 --- Cubic B-Spline (4x4 kernel) (cubicspline) • 4 --- Lanczos (6x6 kernel) (lanczos) • 5 --- Average (average)
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Additional command-line parameters 可选	EXTRA	[string] Default: Not set	Add extra GDAL command line options

输出

标注	名称	类型	描述
Output	OUTPUT	[栅格]	Output (sharpened) raster layer

Python 代码

Algorithm ID: gdal:pansharp

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Raster calculator

Command line raster calculator with numpy syntax. Use any basic arithmetic supported by numpy arrays, such as +, -, *, and / along with logical operators, such as >. Note that all input rasters must have the same dimensions, but no projection checking is performed.

See the GDAL Raster Calculator utility docs.

See also

Raster calculator, *Raster calculator (virtual)*

参数

Basic parameters

标注	名称	类型	描述
Input layer A	`INPUT_A`	[栅格]	First input raster layer (mandatory)
Number of raster band for A	`BAND_A`	[raster band]	Band for input layer A (mandatory)
Input layer B 可选	`INPUT_B`	[栅格] Default: Not set	Second input raster layer
Number of raster band for B 可选	`BAND_B`	[raster band]	Band for input layer B
Input layer C 可选	`INPUT_C`	[栅格] Default: Not set	Third input raster layer
Number of raster band for C 可选	`BAND_C`	[raster band]	Band for input layer C
Input layer D 可选	`INPUT_D`	[栅格] Default: Not set	Fourth input raster layer
Number of raster band for D 可选	`BAND_D`	[raster band]	Band for input layer D
Input layer E 可选	`INPUT_E`	[栅格] Default: Not set	Fifth input raster layer
Number of raster band for E 可选	`BAND_E`	[raster band]	Band for input layer E
Input layer F 可选	`INPUT_F`	[栅格]	Sixth input raster layer
Number of raster band for F 可选	`BAND_F`	[raster band] Default: Not set	Band for input layer F

continues on next page

Table 24.939 – continued from previous page

标注	名称	类型	描述
Calculation in gdalnumeric syntax using +-/* or any numpy array functions (i.e. logical_and())	`FORMULA`	[string] Default: ”	The calculation formula. Examples: • `A*(A>0)` --- outputs the value of the raster A if the value of A is greater than 0. If not, outputs 0. • `A*(A>0 and A>B)`--- outputs the value of A if that value is bigger than 0 and bigger than the value of B. If not, outputs 0. • `A*logical_or(A<=177, A>=185)` --- outputs the value of A if A <= 177 or A >= 185. If not, outputs 0. • `sqrt(A*A+B*B)` --- Outputs the square root of the sum of the value of A squared and the value of B squared.
Set output NoData value 可选	`NO_DATA`	[numeric: double] Default: Not set	Value to use for NoData
Handling of extent differences	`EXTENT_OPT`	[enumeration] Default: 0	Determines how to handle rasters with different extents. Supported options are: • 0 --- Ignore • 1 --- Fail • 2 --- Union • 3 --- Intersect **Attention** Using this parameter requires QGIS installed with GDAL >= 3.3 (see *Help* ► *About* menu).
Output extent 可选	`INPUT`	[范围]	Custom extent of the output raster. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax` **Attention** Using this parameter requires QGIS installed with GDAL >= 3.3 (see *Help* ► *About* menu).

continues on next page

Table 24.939 – continued from previous page

标注	名称	类型	描述
Output raster type	RTYPE	[enumeration] Default: 5	Defines the data type of the output raster file. Options: • 0 --- Byte (Eight bit unsigned integer (quint8)) • 1 --- Int16 (Sixteen bit signed integer (qint16)) • 2 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 3 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 4 --- Int32 (Thirty two bit signed integer (qint32)) • 5 --- Float32 (Thirty two bit floating point (float)) • 6 --- Float64 (Sixty four bit floating point (double)) • 7 --- CInt16 (Complex Int16) • 8 --- CInt32 (Complex Int32) • 9 --- CFloat32 (Complex Float32) • 10 --- CFloat64 (Complex Float64) • 11 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Calculated	OUTPUT	[栅格] 默认: [保存到临时文件]	Specify the output (calculated) raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Additional command-line parameters 可选	EXTRA	[string] Default: ”	Add extra GDAL command line options

输出

标注	名称	类型	描述
Calculated	OUTPUT	[栅格]	Output (calculated) raster layer

Python 代码

Algorithm ID: `gdal:rastercalculator`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Raster information

The gdalinfo program lists various information about a GDAL supported raster dataset.

This algorithm is derived from the GDAL info utility.

Default menu: *Raster ► Miscellaneous*

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[栅格]	Input raster layer
Force computation of the actual min/max values for each band	MIN_MAX	[boolean] Default: False	Forces computation of the actual min/max values for each band in the dataset
Read and display image statistics (force computation if necessary)	STATS	[boolean] Default: False	Reads and displays image statistics. Forces computation if no statistics are stored in an image.
Suppress GCP info	NO_GCP	[boolean] Default: False	Suppresses ground control points list printing. It may be useful for datasets with huge amount of GCPs, such as L1B AVHRR or HDF4 MODIS which contain thousands of them.
Suppress metadata info	NO_METADATA	[boolean] Default: False	Suppresses metadata printing. Some datasets may contain a lot of metadata strings.
Layer information	OUTPUT	[html] 默认: [保存到临时文件]	Specify the HTML file for output. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional command-line parameters 可选	EXTRA	[string] Default: Not set	Add extra GDAL command line options

输出

标注	名称	类型	描述
Layer information	OUTPUT	[html]	The HTML file containing information about the input raster layer

Python 代码

Algorithm ID: `gdal:gdalinfo`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Retile

Retiles a set of input tiles. All the input tiles must be georeferenced in the same coordinate system and have a matching number of bands. Optionally pyramid levels are generated.

This algorithm is derived from the GDAL Retile utility.

参数

Basic parameters

标注	名称	类型	描述
Input files	INPUT	[raster] [list]	The input raster files
Tile width	TILE_SIZE_X	[numeric: integer] Default: 256	Width of the tiles in pixels (minimum 0)
Tile height	TILE_SIZE_Y	[numeric: integer] Default: 256	Height of the tiles in pixels (minimum 0)
Overlap in pixels between consecutive tiles	OVERLAP	[numeric: integer] Default: 0	
Number of pyramid levels to build	LEVELS	[numeric: integer] Default: 1	Minimum: 0

continues on next page

Table 24.945 – continued from previous page

标注	名称	类型	描述
Output directory	OUTPUT	[folder] Default: [Save to temporary folder]	Specify the output folder for the tiles. *One of*: • Save to a Temporary Directory • Save to Directory
CSV file containing the tile(s) georeferencing information	OUTPUT_CSV	[file] Default: [Skip output]	Specify the output file for the tiles. *One of*: • Skip Output • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Source coordinate reference system 可选	SOURCE_CRS	[crs] Default: Not set	
Resampling method	RESAMPLING	[enumeration] Default: 0	The resampling algorithm to use. Options: • 0 --- Nearest Neighbour (nearest) • 1 --- Bilinear (2x2 kernel) (bilinear) • 2 --- Cubic (4x4 kernel) (cubic) • 3 --- Cubic B-Spline (4x4 kernel) (cubicspline) • 4 --- Lanczos (6x6 kernel) (lanczos)
Column delimiter used in the CSV file 可选	DELIMITER	[string] Default: ';'	Delimiter to use in the CSV file containing the tile(s) georeferencing information
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Additional command-line parameters 可选	EXTRA	[string] Default: ''	Add extra GDAL command line options

continues on next page

Table 24.946 – continued from previous page

标注	名称	类型	描述
Output data type	DATA_TYPE	[enumeration] Default: 5	Defines the format of the output raster file. Options: • 0 --- Byte (Eight bit unsigned integer (quint8)) • 1 --- Int16 (Sixteen bit signed integer (qint16)) • 2 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 3 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 4 --- Int32 (Thirty two bit signed integer (qint32)) • 5 --- Float32 (Thirty two bit floating point (float)) • 6 --- Float64 (Sixty four bit floating point (double)) • 7 --- CInt16 (Complex Int16) • 8 --- CInt32 (Complex Int32) • 9 --- CFloat32 (Complex Float32) • 10 --- CFloat64 (Complex Float64) • 11 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Build only the pyramids	ONLY_PYRAMIDS	[boolean] Default: False	
Use separate directory for each tile row	DIR_FOR_ROW	[boolean] Default: False	

输出

标注	名称	类型	描述
Output directory	OUTPUT	[folder]	The output folder for the tiles.
CSV file containing the tile(s) georeferencing information	OUTPUT_CSV	[file]	The CSV file with georeferencing information for the tiles.

Python 代码

Algorithm ID: `gdal:retile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Tile index

Builds a vector layer with a record for each input raster file, an attribute containing the filename, and a polygon geometry outlining the raster. This output is suitable for use with MapServer as a raster tileindex.

This algorithm is derived from the GDAL Tile Index utility.

Default menu: *Raster* ► *Miscellaneous*

参数

Basic parameters

标注	名称	类型	描述
Input files	`LAYERS`	[raster] [list]	The input raster files. Can be multiple files.
Field name to hold the file path to the indexed rasters	`PATH_FIELD_NAM` Optional	[string] Default: 'location'	The output field name to hold the file path/location to the indexed rasters.
Store absolute path to the indexed rasters	`ABSOLUTE_PATH`	[boolean] Default: False	Set whether the absolute path to the raster files is stored in the tile index file. By default the raster filenames will be put in the file exactly as they are specified in the command.
Skip files with different projection reference	`PROJ_DIFFERENC`	[boolean] Default: False	Only files with same projection as files already inserted in the tile index will be inserted. Default does not check projection and accepts all inputs.
Tile index	`OUTPUT`	[vector: polygon] 默认: [保存到临时文件]	Specify the polygon vector layer to write the index to. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Transform geometries to the given CRS 可选	`TARGET_CRS`	[crs]	Geometries of input files will be transformed to the specified target coordinate reference system. Default creates simple rectangular polygons in the same coordinate reference system as the input rasters.
The name of the field to store the SRS of each tile 可选	`CRS_FIELD_NAME`	[string]	The name of the field to store the SRS of each tile
The format in which the CRS of each tile must be written	`CRS_FORMAT`	[enumeration] Default: 0	Format for the CRS. One of: • 0 -- Auto (`AUTO`) • 1 -- Well-known text (`WKT`) • 2 -- EPSG (`EPSG`) • 3 -- Proj.4 (`PROJ`)

输出

标注	名称	类型	描述
Tile index	`OUTPUT`	[vector: polygon]	The polygon vector layer with the tile index.

Python 代码

Algorithm ID: `gdal:tileindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Viewshed

Calculates a viewshed raster from an input raster DEM using method defined in Wang2000 for a user defined point.

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	Input elevation raster layer
Band number	`BAND`	[raster band] Default: 1	The number of the band to use as elevation
Observer location	`OBSERVER`	[coordinates]	The location of the observer
Observer height	`OBSERVER_HEIGH`	[numeric: double] 默认值：1.0	The altitude of the observer, in the DEM units
Target height	`TARGET_HEIGHT`	[numeric: double] 默认值：1.0	The altitude of the target element, in the DEM units
Maximum distance from observer to compute visibility	`MAX_DISTANCE`	[numeric: double] Default: 100.0	Maximum distance from observer to compute visibility, in the DEM units
Output	`OUTPUT`	[栅格] 默认：[保存到临时文件]	Output raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Additional command-line parameters	EXTRA	[string] Default: Not set	Add extra GDAL command line options

输出

标注	名称	类型	描述
Output	OUTPUT	[栅格]	The raster layer displaying the viewshed.

Python 代码

Algorithm ID: `gdal:viewshed`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.2.5 Raster projections

Assign projection

Applies a coordinate system to a raster dataset.

This algorithm is derived from the GDAL edit utility.

Default menu: *Raster* ► *Projections*

参数

标注	名称	类型	描述
输入图层	`INPUT_LAYER`	[栅格]	Input raster layer
Desired CRS	`CRS`	[crs]	The projection (CRS) of the output layer

输出

标注	名称	类型	描述
Layer with projection	`OUTPUT`	[栅格]	The output raster layer (with the new projection information)

Python 代码

Algorithm ID: `gdal:assignprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Extract projection

Extracts the projection of a raster file and writes it into a *world* file with extension `.wld`.

This algorithm is derived from the GDAL srsinfo utility.

Default menu: *Raster ► Projections*

参数

标注	名称	类型	描述
Input file	`INPUT_LAYER`	[栅格]	Input raster The raster layer has to be file based, as the algorithm uses the path to the raster file as the location of the generated `.wld` file. Using a non-file raster layer will lead to an error.
Create also .prj file	`PRJ_FILE_CREAT`	[boolean] Default: False	If this is activated a `.prj` file containing the projection information is also created.

输出

标注	名称	类型	描述
World file	`WORLD_FILE`	[file]	Text file with extension `.wld` containing transformation parameters for the raster file.
ESRI Shapefile prj file	`PRJ_FILE`	[file]	Text file with `.prj` extension that describes the CRS. Will be `None` if *Create also .prj file* is False.

Python 代码

Algorithm ID: `gdal:extractprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Warp (reproject)

Reprojects a raster layer into another Coordinate Reference System (CRS). The output file resolution and the resampling method can be chosen.

This algorithm is derived from the GDAL warp utility.

Default menu: *Raster* ► *Projections*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[栅格]	Input raster layer to reproject
Source CRS 可选	`SOURCE_CRS`	[crs]	Defines the CRS of the input raster layer
Target CRS 可选	`TARGET_CRS`	[crs] Default: `EPSG:4326`	The CRS of the output layer

continues on next page

Table 24.958 – continued from previous page

标注	名称	类型	描述
Resampling method to use	RESAMPLING	[enumeration] Default: 0	Pixel value resampling method to use. Options: • 0 --- Nearest Neighbour • 1 --- Bilinear (2x2 kernel) • 2 --- Cubic (4x4 kernel) • 3 --- Cubic B-Spline (4x4 kernel) • 4 --- Lanczos (6x6 kernel) • 5 --- Average • 6 --- Mode • 7 --- Maximum • 8 --- Minimum • 9 --- Median • 10 --- First quartile (Q1) • 11 --- Third quartile (Q3)
NoData value for output bands 可选	NODATA	[numeric: double] Default: Not set	Sets NoData value for output bands. If not provided, then NoData values will be copied from the source dataset.
Output file resolution in target georeferenced units 可选	TARGET_RESOLUT	[numeric: double] Default: Not set	Defines the output file resolution of reprojection result
Reprojected	OUTPUT	[栅格] 默认: [保存到临时文件]	Specification of the output raster layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ''	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

continues on next page

Table 24.959 – continued from previous page

标注	名称	类型	描述
Output data type	`DATA_TYPE`	[enumeration] Default: 0	Defines the format of the output raster file. Options: • 0 --- Use Input Layer Data Type • 1 --- Byte (Eight bit unsigned integer (quint8)) • 2 --- Int16 (Sixteen bit signed integer (qint16)) • 3 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 4 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 5 --- Int32 (Thirty two bit signed integer (qint32)) • 6 --- Float32 (Thirty two bit floating point (float)) • 7 --- Float64 (Sixty four bit floating point (double)) • 8 --- CInt16 (Complex Int16) • 9 --- CInt32 (Complex Int32) • 10 --- CFloat32 (Complex Float32) • 11 --- CFloat64 (Complex Float64) • 12 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Georeferenced extents of output file to be created 可选	`TARGET_EXTENT`	[范围]	Sets the georeferenced extent of the output file to be created (in the *Target CRS* by default. In the *CRS of the target raster extent*, if specified). Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
CRS of the target raster extent 可选	`TARGET_EXTENT_`	[crs]	Specifies the CRS in which to interpret the coordinates given for the extent of the output file. This must not be confused with the target CRS of the output dataset. It is instead a convenience e.g. when knowing the output coordinates in a geodetic long/lat CRS, but wanting a result in a projected coordinate system.

continues on next page

Table 24.959 – continued from previous page

标注	名称	类型	描述
Use multithreaded warping implementation	MULTITHREADING	[boolean] Default: False	Two threads will be used to process chunks of the image and perform input/output operations simultaneously. Note that the computation itself is not multithreaded.
Additional command-line parameters 可选	EXTRA	[string] Default: Not set	Add extra GDAL command line options.

输出

标注	名称	类型	描述
Reprojected	OUTPUT	[栅格] 默认: [保存到临时文件]	Reprojected output raster layer

Python 代码

Algorithm ID: `gdal:warpreproject`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.2.6 Vector conversion

Convert format

Converts any OGR-supported vector layer into another OGR-supported format.

This algorithm is derived from the ogr2ogr utility.

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[vector: any]	输入矢量图层
Convert all layers from dataset	CONVERT_ALL_LA	[boolean] Default: False	Converts the whole dataset. Supported output formats for this option are GPKG and GML.

continues on next page

Table 24.961 – continued from previous page

标注	名称	类型	描述
Converted	OUTPUT	[same as input]	Specification of the output vector layer. *One of*: • 保存到临时文件 • 保存到文件… For Save to File, the output format has to be specified. All GDAL vector formats are supported. For Save to a Temporary File the QGIS default vector format will be used.

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ” (no additional options)	Additional GDAL creation options.

输出

标注	名称	类型	描述
Converted	OUTPUT	[same as input]	The output vector layer

Python 代码

Algorithm ID: gdal:convertformat

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

Rasterize (overwrite with attribute)

Overwrites a raster layer with values from a vector layer. New values are assigned based on the attribute value of the overlapping vector feature.

This algorithm is derived from the GDAL rasterize utility.

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[vector: geometry]	输入矢量图层
Input raster layer	INPUT_RASTER	[栅格]	Input raster layer
Field to use for a burn-in value 可选	FIELD	[tablefield: numeric]	Defines the attribute field to use to set the pixels values

Advanced parameters

标注	名称	类型	描述
Add burn in values to existing raster values	ADD	[boolean] Default: False	If False, pixels are assigned the selected field's value. If True, the selected field's value is added to the value of the input raster layer.
Additional command-line parameters 可选	EXTRA	[string] Default: ''	Add extra GDAL command line options

输出

标注	名称	类型	描述
Rasterized	OUTPUT	[栅格]	The overwritten input raster layer

Python 代码

Algorithm ID: `gdal:rasterize_over`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，*算法 id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Rasterize (overwrite with fixed value)

Overwrites parts of a raster layer with a fixed value. The pixels to overwrite are chosen based on the supplied (overlapping) vector layer.

This algorithm is derived from the GDAL rasterize utility.

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[vector: geometry]	输入矢量图层
Input raster layer	INPUT_RASTER	[栅格]	Input raster layer
A fixed value to burn	BURN	[numeric: double] Default: 0.0	The value to burn

Advanced parameters

标注	名称	类型	描述
Add burn in values to existing raster values	ADD	[boolean] Default: False	If False, pixels are assigned the fixed value. If True, the fixed value is added to the value of the input raster layer.
Additional command-line parameters 可选	EXTRA	[string] Default: ”	Add extra GDAL command line options

输出

标注	名称	类型	描述
Rasterized	OUTPUT	[栅格]	The overwritten input raster layer

Python 代码

Algorithm ID: `gdal:rasterize_over_fixed_value`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

Rasterize (vector to raster)

Converts vector geometries (points, lines and polygons) into a raster image.

This algorithm is derived from the GDAL rasterize utility.

Default menu: *Raster* ► *Conversion*

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Field to use for a burn-in value 可选	`FIELD`	[tablefield: numeric]	Defines the attribute field from which the attributes for the pixels should be chosen
A fixed value to burn 可选	`BURN`	[numeric: double] Default: 0.0	A fixed value to burn into a band for all features.
Burn value extracted from the "Z" values of the feature	`USE_Z`	[boolean] Default: False	Indicates that a burn value should be extracted from the "Z" values of the feature. Works with points and lines (linear interpolation along each segment). For polygons, works properly only if they are flat (same Z value for all vertices)
Output raster size units	`UNITS`	[enumeration] Default: 0	Units to use when defining the output raster size/resolution. One of: • 0 --- Pixels • 1 --- Georeferenced units
Width/Horizontal resolution	`WIDTH`	[numeric: double] Default: 0.0	Sets the width (if size units is "Pixels") or horizontal resolution (if size units is "Georeferenced units") of the output raster. Minimum value: 0.0.
Height/Vertical resolution	`HEIGHT`	[numeric: double] Default: 0.0	Sets the height (if size units is "Pixels") or vertical resolution (if size units is "Georeferenced units") of the output raster.

continues on next page

Table 24.970 – continued from previous page

标注	名称	类型	描述
Output extent 可选	EXTENT	[范围]	Extent of the output raster layer. If the extent is not specified, the minimum extent that covers the selected reference layer(s) will be used. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Assign a specified NoData value to output bands 可选	NODATA	[numeric: double] Default: 0.0	Assigns a specified NoData value to output bands
Rasterized	OUTPUT	[栅格] 默认: [保存到临时文件]	Specification of the output raster layer. *One of*: • 保存到临时文件 • 保存到文件… For `Save to File`, the output format has to be specified. All GDAL raster formats are supported. For `Save to a Temporary File` the QGIS default raster format will be used.

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ”	For adding one or more creation options that control the raster to be created (colors, block size, file compression...). For convenience, you can rely on predefined profiles (see *GDAL driver options section*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

continues on next page

Table 24.971 – continued from previous page

标注	名称	类型	描述
Output data type	DATA_TYPE	[enumeration] Default: 5	Defines the format of the output raster file. Options: • 0 --- Byte (Eight bit unsigned integer (quint8)) • 1 --- Int16 (Sixteen bit signed integer (qint16)) • 2 --- UInt16 (Sixteen bit unsigned integer (quint16)) • 3 --- UInt32 (Thirty two bit unsigned integer (quint32)) • 4 --- Int32 (Thirty two bit signed integer (qint32)) • 5 --- Float32 (Thirty two bit floating point (float)) • 6 --- Float64 (Sixty four bit floating point (double)) • 7 --- CInt16 (Complex Int16) • 8 --- CInt32 (Complex Int32) • 9 --- CFloat32 (Complex Float32) • 10 --- CFloat64 (Complex Float64) • 11 --- Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Pre-initialize the output image with value 可选	INIT	[numeric: double]	Pre-initializes the output image bands with this value. Not marked as the NoData value in the output file. The same value is used in all the bands.
Invert rasterization	INVERT	[boolean] Default: False	Burns the fixed burn value, or the burn value associated with the first feature into all parts of the image not inside the provided polygon.
Additional command-line parameters 可选	EXTRA	[string] Default: ”	Add extra GDAL command line options

输出

标注	名称	类型	描述
Rasterized	OUTPUT	[栅格]	Output raster layer

Python 代码

Algorithm ID: `gdal:rasterize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

24.2.7 Vector geoprocessing

Buffer vectors

Create buffers around the features of a vector layer.

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[vector: geometry]	The input vector layer
Geometry column name	GEOMETRY	[string] Default: 'geometry'	The name of the input layer geometry column to use
Buffer distance	DISTANCE	[numeric: double] Default: 10.0	Minimum: 0.0
Dissolve by attribute 可选	FIELD	[tablefield: any] Default: Not set	Field to use for dissolving
Dissolve results	DISSOLVE	[boolean] Default: False	If set, the result is dissolved. If no field is set for dissolving, all the buffers are dissolved into one feature.
Produce one feature for each geometry in any kind of geometry collection in the source file	EXPLODE_COLLEC	[boolean] Default: False	
Buffer	OUTPUT	[vector: polygon] 默认：[保存到临时文件]	Specify the output buffer layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: '' (no additional options)	Additional GDAL creation options.

输出

标注	名称	类型	描述
Buffer	OUTPUT	[vector: polygon]	The output buffer layer

Python 代码

Algorithm ID: `gdal:buffervectors`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Clip vector by extent

Clips any OGR-supported vector file to a given extent.

This algorithm is derived from the GDAL ogr2ogr utility.

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[vector: geometry]	The input vector layer

continues on next page

Table 24.976 – continued from previous page

标注	名称	类型	描述
Clip extent	`EXTENT`	[范围]	Defines the bounding box that should be used for the output vector file. It has to be defined in target CRS coordinates. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Clipped (extent)	`OUTPUT`	[same as input] 默认: [保存到临时文件]	Specify the output (clipped) layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Default: ” (no additional options)	Additional GDAL creation options.

输出

标注	名称	类型	描述
Clipped (extent)	`OUTPUT`	[same as input]	The output (clipped) layer. The default format is ”ESRI Shapefile”.

Python 代码

Algorithm ID: `gdal:clipvectorbyextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，*算法 id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

Clip vector by mask layer

Clips any OGR-supported vector layer by a mask polygon layer.

This algorithm is derived from the GDAL ogr2ogr utility.

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	The input vector layer
Mask layer	`MASK`	[vector: polygon]	Layer to be used as clipping extent for the input vector layer.
Clipped (mask)	`OUTPUT`	[same as input] 默认: [保存到临时文件]	The output (masked) layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Default: ” (no additional options)	Additional GDAL creation options.

输出

标注	名称	类型	描述
Clipped (mask)	`OUTPUT`	[same as input]	The output (masked) layer. The default format is ”ESRI Shapefile”.

Python 代码

Algorithm ID: `gdal:clipvectorbypolygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

融合（dissolve）

Dissolve (combine) geometries that have the same value for a given attribute / field. The output geometries are multipart.

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[vector: geometry]	The input layer to dissolve
Dissolve field 可选	FIELD	[tablefield: any]	The field of the input layer to use for dissolving
Geometry column name	GEOMETRY	[string] Default: 'geometry'	The name of the input layer geometry column to use for dissolving.
Dissolved	OUTPUT	[same as input] 默认：[保存到临时文件]	Specify the output layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Produce one feature for each geometry in any kind of geometry collection in the source file	EXPLODE_COLLEC	[boolean] Default: False	Produce one feature for each geometry in any kind of geometry collection in the source file
Keep input attributes	KEEP_ATTRIBUTE	[boolean] Default: False	Keep all attributes from the input layer
Count dissolved features	COUNT_FEATURES	[boolean] Default: False	Count the dissolved features and include it in the output layer.
Compute area and perimeter of dissolved features	COMPUTE_AREA	[boolean] Default: False	Compute the area and perimeter of dissolved features and include them in the output layer
Compute min/max/sum/mean for attribute	COMPUTE_STATIS	[boolean] Default: False	Calculate statistics (min, max, sum and mean) for the numeric attribute specified and include them in the output layer
Numeric attribute to calculate statistics on 可选	STATISTICS_ATT	[tablefield: numeric]	The numeric attribute to calculate statistics on
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: '' (no additional options)	Additional GDAL creation options.

输出

标注	名称	类型	描述
Dissolved	OUTPUT	[same as input]	The output multipart geometry layer (with dissolved geometries)

Python 代码

Algorithm ID: `gdal:dissolve`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Offset curve

Offsets lines by a specified distance. Positive distances will offset lines to the left, and negative distances will offset them to the right.

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[vector: line]	The input line layer
Geometry column name	GEOMETRY	[string] Default: 'geometry'	The name of the input layer geometry column to use
Offset distance (left-sided: positive, right-sided: negative)	DISTANCE	[numeric: double] Default: 10.0	
Offset curve	OUTPUT	[vector: line] 默认: [保存到临时文件]	Specify the output line layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: '' (no additional options)	Additional GDAL creation options.

输出

标注	名称	类型	描述
Offset curve	`OUTPUT`	[vector: line]	The output offset curve layer

Python 代码

Algorithm ID: `gdal:offsetcurve`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

One side buffer

Creates a buffer on one side (right or left) of the lines in a line vector layer.

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[vector: line]	The input line layer
Geometry column name	`GEOMETRY`	[string] Default: 'geometry'	The name of the input layer geometry column to use
Buffer distance	`DISTANCE`	[numeric: double] Default: 10.0	
Buffer side	`BUFFER_SIDE`	[enumeration] Default: 0	One of: • 0 --- Right • 1 --- Left
Dissolve by attribute 可选	`FIELD`	[tablefield: any] Default: Not set	Field to use for dissolving
Dissolve all results	`DISSOLVE`	[boolean] Default: False	**If set, the result is dissolved.** If no field is set for dissolving, all the buffers are dissolved into one feature.
Produce one feature for each geometry in any kind of geometry collection in the source file	`EXPLODE_COLLEC`	[boolean] Default: False	

continues on next page

Table 24.988 – continued from previous page

标注	名称	类型	描述
One-sided buffer	OUTPUT	[vector: polygon] 默认: [保存到临时文件]	Specify the output buffer layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ” (no additional options)	Additional GDAL creation options.

输出

标注	名称	类型	描述
One-sided buffer	OUTPUT	[vector: polygon]	The output buffer layer

Python 代码

Algorithm ID: `gdal:onesidebuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

Points along lines

Generates a point on each line of a line vector layer at a distance from start. The distance is provided as a fraction of the line length.

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[vector: line]	The input line layer
Geometry column name	GEOMETRY	[string] Default: ’geometry’	The name of the input layer geometry column to use
Distance from line start represented as a fraction of line length	DISTANCE	[numeric: double] Default: 0.5	

continues on next page

Table 24.991 – continued from previous page

标注	名称	类型	描述
Points along lines	OUTPUT	[vector: point] 默认：[保存到临时文件]	Specify the output point layer. *One of*: • 保存到临时文件 • 保存到文件…

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ” (no additional options)	Additional GDAL creation options.

输出

标注	名称	类型	描述
Points along line	OUTPUT	[vector: point]	The output point layer

Python 代码

Algorithm ID: `gdal:pointsalonglines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

24.2.8 Vector miscellaneous

Build virtual vector

Creates a virtual vector layer that contains a set of vector layers.

This algorithm is especially useful in case another algorithm needs multiple layers but accept only one `vrt` in which the layers are specified.

参数

标注	名称	类型	描述
Input datasources	`INPUT`	[vector: any] [list]	Select the vector layers you want to use to build the virtual vector
Create "unioned" VRT	`UNIONED`	[boolean] Default: False	Check if you want to unite all the vectors in a single `vrt` file
Virtual vector	`OUTPUT`	[same as input] 默认：[保存到临时文件]	Specify the output layer containing only the duplicates. *One of*: • 保存到临时文件 • 保存到文件…

输出

标注	名称	类型	描述
Virtual vector	`OUTPUT`	[vector: any]	The output virtual vector made from the chosen sources

Python 代码

Algorithm ID: `gdal:buildvirtualvector`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Execute SQL

Runs a simple or complex query with SQL syntax on the source layer. The result of the query will be added as a new layer.

This algorithm is derived from the GDAL ogr2ogr utility.

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	OGR-supported input vector layer
SQL expression	`SQL`	[string]	Defines the SQL query, for example `SELECT * FROM my_table WHERE name is not null`.

continues on next page

Table 24.996 – continued from previous page

标注	名称	类型	描述
SQL dialect	DIALECT	[enumeration] Default: 0	SQL dialect to use. One of: • 0 --- None • 1 --- OGR SQL • 2 --- SQLite
SQL result	OUTPUT	[vector: any]	Specification of the output layer. *One of*: • 保存到临时文件 • 保存到文件… For Save to File, the output format has to be specified. All GDAL vector formats are supported. For Save to a Temporary File the default output vector layer format will be used.

Advanced parameters

标注	名称	类型	描述
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: ” (no additional options)	Additional GDAL creation options.

输出

标注	名称	类型	描述
SQL result	OUTPUT	[vector: any]	Vector layer created by the query

Python 代码

Algorithm ID: gdal:executesql

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

Export to PostgreSQL (available connections)

Imports vector layers inside a PostgreSQL database on the basis of an available connection. The connection has to *be defined properly* beforehand. Be aware that the checkboxes ’Save Username’ and ’Save Password’ are activated. Then you can use the algorithm.

This algorithm is derived from the GDAL ogr2ogr utility.

参数

标注	名称	类型	描述
Database (connection name)	DATABASE	[string]	The PostgreSQL database to connect to
输入图层	INPUT	[vector: any]	OGR-supported vector layer to export to the database
Shape encoding 可选	SHAPE_ENCODING	[string] Default: ”	Sets the encoding to apply to the data
Output geometry type	GTYPE	[enumeration] Default: 0	Defines the output geometry type. One of: • 0 --- • 1 --- NONE • 2 --- GEOMETRY • 3 --- POINT • 4 --- LINESTRING • 5 --- POLYGON • 6 --- GEOMETRYCOLLECTION • 7 --- MULTIPOINT • 8 --- MULTIPOLYGON • 9 --- MULTILINESTRING • 10 --- CIRCULARSTRING • 11 --- COUMPOUNDCURVE • 12 --- CURVEPOLYGON • 13 --- MULTICURVE • 14 --- MULTISURFACE • 15 --- CONVERT_TO_LINEAR • 16 --- CONVERT_TO_CURVE
Assign an output CRS 可选	A_SRS	[crs] Default: Not set	Defines the output CRS of the database table
Reproject to this CRS on output 可选	T_SRS	[crs] Default: Not set	Reprojects/transforms to this CRS on output
Override source CRS 可选	S_SRS	[crs] Default: Not set	Overrides the input layer CRS
Schema (schema name) 可选	SCHEMA	[string] Default: ’public’	Defines the schema for the database table
Table to export to (leave blank to use layer name) 可选	TABLE	[string] Default: ”	Defines a name for the table that will be imported into the database. By default the table name is the name of the input vector file.
Primary Key (new field) 可选	PK	[string] Default: ’id’	Defines which attribute field will be the primary key of the database table
Primary Key (existing field, used if the above option is left empty) 可选	PRIMARY_KEY	[tablefield: any] Default: Not set	Defines which attribute field in the exported layer will be the primary key of the database table
Geometry column name 可选	GEOCOLUMN	[string] Default: ’geom’	Defines in which attribute field of the database there will be the geometry information

continues on next page

Table 24.999 – continued from previous page

标注	名称	类型	描述
Vector dimensions 可选	`DIM`	[enumeration] Default: 0 (2D)	Defines the coordinate dimensions of the imported vector data. One of: • 0 --- 2 (`XY`) • 1 --- 3 (`XYZ`) • 2 --- 4 (`XYZM`)
Distance tolerance for simplification 可选	`SIMPLIFY`	[string] Default: ”	Defines a distance tolerance for the simplification of the vector geometries to be imported. By default there is no simplification.
Maximum distance between 2 nodes (densification) 可选	`SEGMENTIZE`	[string] Default: ”	The maximum distance between two nodes. Used to create intermediate points. By default there is no densification.
Select features by extent (defined in input layer CRS) 可选	`SPAT`	[范围] Default: Not set	You can select features from a given extent that will be in the output table. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as `xmin, xmax, ymin, ymax`
Clip the input layer using the above (rectangle) extent	`CLIP`	[boolean] Default: False	The input layer will be clipped by the extent you defined before
Select features using a SQL ”WHERE” statement (Ex: column=”value”) 可选	`WHERE`	[string] Default: ”	Defines with a SQL ”WHERE” statement which features should be selected from the input layer
Group N features per transaction (Default: 2000) 可选	`GT`	[string] Default: ”	You can group the input features in transactions where N defines the size. By default N limits the transaction size to 20000 features.
Overwrite existing table	`OVERWRITE`	[boolean] Default: True	If there is a table with the same name in the database, and if this option is set to True, the table will be overwritten.
Append to existing table	`APPEND`	[boolean] Default: False	If checked / True the vector data will be appended to an existing table. New fields found in the input layer are ignored. By default a new table will be created.

continues on next page

Table 24.999 – continued from previous page

标注	名称	类型	描述
Append and add new fields to existing table	ADDFIELDS	[boolean] Default: False	If activated the vector data will be appended to an existing table, there won't be a new table created. New fields found in input layer are added to the table. By default a new table will be created.
Do not launder columns/table names	LAUNDER	[boolean] Default: False	With this option checked you can prevent the default behaviour (converting column names to lowercase, removing spaces and other invalid characters).
Do not create Spatial Index	INDEX	[boolean] Default: False	Prevents a spatial index for the output table from being created. By default, a spatial index is added.
Continue after a failure, skipping the failed feature	SKIPFAILURES	[boolean] Default: False	Continue after a failure, skipping the failed feature.
Validate geometries based on Simple Features specification	MAKEVALID	[boolean] Default: False	Applies cleanup operations on geometries to ensure they are valid regarding the rules of the Simple Features specification.
Promote to Multipart	PROMOTETOMULTI	[boolean] Default: True	Casts features geometry type to multipart in the output table
Keep width and precision of input attributes	PRECISION	[boolean] Default: True	Avoids modifying column attributes to comply with input data
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: '' (no additional options)	Additional GDAL creation options.

输出

This algorithm has no output.

Python 代码

Algorithm ID: `gdal:importvectorintopostgisdatabaseavailableconnections`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Export to PostgreSQL (new connection)

Imports vector layers inside a PostgreSQL database. A new connection to the PostgreSQL database must be created.

This algorithm is derived from the GDAL ogr2ogr utility.

参数

标注	名称	类型	描述
输入图层	`INPUT`	[vector: any]	OGR-supported vector layer to export to the database
Shape encoding 可选	`SHAPE_ENCODING`	[string] Default: ”	Sets the encoding to apply to the data
Output geometry type	`GTYPE`	[enumeration] Default: 0	Defines the output geometry type. One of: • 0 --- • 1 --- NONE • 2 --- GEOMETRY • 3 --- POINT • 4 --- LINESTRING • 5 --- POLYGON • 6 --- GEOMETRYCOLLECTION • 7 --- MULTIPOINT • 8 --- MULTIPOLYGON • 9 --- MULTILINESTRING • 10 --- CIRCULARSTRING • 11 --- COUMPOUNDCURVE • 12 --- CURVEPOLYGON • 13 --- MULTICURVE • 14 --- MULTISURFACE • 15 --- CONVERT_TO_LINEAR • 16 --- CONVERT_TO_CURVE
Assign an output CRS 可选	`A_SRS`	[crs] Default: Not set	Defines the output CRS of the database table
Reproject to this CRS on output 可选	`T_SRS`	[crs] Default: Not set	Reprojects/transforms to this CRS on output
Override source CRS 可选	`S_SRS`	[crs] Default: Not set	Overrides the input layer CRS
Host 可选	`HOST`	[string] Default: ’localhost’	Name of the database host
Port 可选	`PORT`	[string] Default: ’5432’	Port number the PostgreSQL database server listens on
Username 可选	`USER`	[string] Default: ”	User name used to log in to the database
Database name 可选	`DBNAME`	[string] Default: ”	Name of the database
Password 可选	`PASSWORD`	[string] Default: ”	Password used with Username to connect to the database
Schema (schema name) 可选	`SCHEMA`	[string] Default: ’public’	Defines the schema for the database table

continues on next page

Table 24.1000 – continued from previous page

标注	名称	类型	描述
Table name, leave blank to use input name 可选	TABLE	[string] Default: ”	Defines a name for the table that will be imported into the database. By default the table name is the name of the input vector file.
Primary Key (new field) 可选	PK	[string] Default: ’id’	Defines which attribute field will be the primary key of the database table
Primary Key (existing field, used if the above option is left empty) 可选	PRIMARY_KEY	[tablefield: any] Default: Not set	Defines which attribute field in the exported layer will be the primary key of the database table
Geometry column name 可选	GEOCOLUMN	[string] Default: ’geom’	Defines in which attribute field to store the geometry information
Vector dimensions 可选	DIM	[enumeration] Default: 0 (2D)	Defines the coordinate dimensions of the imported vector data. One of: • 0 --- 2 (XY) • 1 --- 3 (XYZ) • 2 --- 4 (XYZM)
Distance tolerance for simplification 可选	SIMPLIFY	[string] Default: ”	Defines a distance tolerance for the simplification of the vector geometries to be imported. By default there is no simplification.
Maximum distance between 2 nodes (densification) 可选	SEGMENTIZE	[string] Default: ”	The maximum distance between two nodes. Used to create intermediate points. By default there is no densification.
Select features by extent (defined in input layer CRS) 可选	SPAT	[范围] Default: Not set	You can select features from a given extent that will be in the output table. Available methods are: • Calculate from layer…: uses extent of a layer loaded in the current project • Calculate from layout map…: uses extent of a *layout map item* in the active project • Calculate from bookmark…: uses extent of a saved *bookmark* • Use map canvas extent • Draw on canvas: click and drag a rectangle delimiting the area to take into account • Enter the coordinates as xmin, xmax, ymin, ymax
Clip the input layer using the above (rectangle) extent	CLIP	[boolean] Default: False	The input layer will be clipped by the extent you defined before
Fields to include (leave empty to use all fields) 可选	FIELDS	[string] [list] Default: []	Defines fields to keep from the imported vector file. If none is selected, all the fields are imported.

continues on next page

Table 24.1000 – continued from previous page

标注	名称	类型	描述
Select features using a SQL "WHERE" statement (Ex: column="value") 可选	WHERE	[string] Default: "	Defines with a SQL "WHERE" statement which features should be selected for the output table
Group N features per transaction (Default: 2000) 可选	GT	[string] Default: "	You can group the input features in transactions where N defines the size. By default N limits the transaction size to 20000 features.
Overwrite existing table	OVERWRITE	[boolean] Default: True	If there is a table with the same name in the database, and if this option is set to True, the table will be overwritten.
Append to existing table	APPEND	[boolean] Default: False	If checked / True the vector data will be appended to an existing table. New fields found in the input layer are ignored. By default a new table will be created.
Append and add new fields to existing table	ADDFIELDS	[boolean] Default: False	If activated the vector data will be appended to an existing table, there won't be created a new table. New fields found in input layer are added to the table. By default a new table will be created.
Do not launder columns/table names	LAUNDER	[boolean] Default: False	With this option checked you can prevent the default behaviour (converting column names to lowercase, removing spaces and other invalid characters).
Do not create Spatial Index	INDEX	[boolean] Default: False	Prevents a spatial index for the output table from being created. By default, a spatial index is added.
Continue after a failure, skipping the failed feature	SKIPFAILURES	[boolean] Default: False	Continue after a failure, skipping the failed feature.
Validate geometries based on Simple Features specification	MAKEVALID	[boolean] Default: False	Applies cleanup operations on geometries to ensure they are valid regarding the rules of the Simple Features specification.
Promote to Multipart	PROMOTETOMULTI	[boolean] Default: True	Casts features geometry type to multipart in the output table
Keep width and precision of input attributes	PRECISION	[boolean] Default: True	Avoids modifying column attributes to comply with input data
Additional creation options 可选	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[string] Default: " (no additional options)	Additional GDAL creation options.

输出

This algorithm has no output.

Python 代码

Algorithm ID: `gdal:importvectorintopostgisdatabasenewconnection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 '数据处理 _ 控制台 '以获取如何从 Python 控制台中运行数据处理算法。

Vector Information

Creates an information file that lists information about an OGR-supported data source. The output will be shown in a 'Result' window and can be written into a HTML-file. The information includes the geometry type, feature count, the spatial extent, the projection information, the list of attributes and related type, list of relations and field domains, and many more.

This algorithm is derived from the GDAL ogrinfo utility.

参数

Basic parameters

标注	名称	类型	描述
输入图层	`INPUT`	[vector: geometry]	输入矢量图层
Enable listing of all layers in the dataset	`ALL_LAYERS`	[boolean] Default: False	If checked, QGIS will output the information of all the layers in the input dataset instead of information of the first layer only. Convenient when a multi layer format (GeoPackage, GML, KML, SpatiaLite, SQLite, ...) is used as input.
Summary output only	`SUMMARY_ONLY`	[boolean] Default: True	Suppress listing of individual features and show only summary information like projection, schema, feature count and extents.
Suppress metadata info	`NO_METADATA`	[boolean] Default: False	Suppress metadata printing. Some datasets may contain a lot of metadata strings.
Layer information	`OUTPUT`	[html] 默认：[保存到临时文件]	Specify the output HTML file that includes the file information. *One of*: • 保存到临时文件 • 保存到文件… If no HTML-file is defined the output will be written to a temporary file

Advanced parameters

标注	名称	类型	描述
Additional command line parameters 可选	EXTRA	[string] Default: ” (no additional options)	Add extra GDAL command line options

输出

标注	名称	类型	描述
Layer information	OUTPUT	[html]	The output HTML-file that includes the file information.

Python 代码

Algorithm ID: `gdal:ogrinfo`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

Vector Information (JSON)

Creates an information file that lists information about an OGR-supported data source. The output will be shown in a ’Result’ window and can be written into a `.JSON` file. The information includes the geometry type, feature count, the spatial extent, the projection information, the list of attributes and related type, list of relations and field domains, and many more.

This algorithm is derived from the GDAL ogrinfo utility.

> **Attention**
>
> Running this algorithm requires QGIS installed with GDAL >= 3.7 (see *Help* ► *About* menu).

参数

Basic parameters

标注	名称	类型	描述
输入图层	INPUT	[vector: geometry]	输入矢量图层

continues on next page

Table 24.1004 – continued from previous page

标注	名称	类型	描述
Enable listing of all layers in the dataset	ALL_LAYERS	[boolean] Default: False	If checked, QGIS will output the information of all the layers in the input dataset instead of information of the first layer only. Convenient when a multi layer format (GeoPackage, GML, KML, SpatiaLite, SQLite, ...) is used as input.
Enable listing of features	FEATURES	[boolean] Default: False	
Suppress metadata info	NO_METADATA	[boolean] Default: False	Suppress metadata printing. Some datasets may contain a lot of metadata strings.
Layer information	OUTPUT	[file] 默认：[保存到临时文件]	Specify the output JSON file that includes the file information. *One of*: • 保存到临时文件 • 保存到文件… If no JSON file is defined the output will be written to a temporary file

Advanced parameters

标注	名称	类型	描述
Additional command line parameters 可选	EXTRA	[string] Default: ” (no additional options)	Add extra GDAL command line options

输出

标注	名称	类型	描述
Layer information	OUTPUT	[file]	The output *.JSON* file that includes the file information.

Python 代码

Algorithm ID: `gdal:ogrinfojson`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

当你在数据处理工具箱某个算法前悬停时，算法 *id* 将会展现。*参数字 * 将会提供参数的名称与数值。参考详见 ‘数据处理 _ 控制台 ‘以获取如何从 Python 控制台中运行数据处理算法。

CHAPTER 25

插件

25.1 QGIS 插件

QGIS has been designed with a plugin architecture. This allows many new features and functions to be easily added to the application. Some of the features in QGIS are actually implemented as plugins.

25.1.1 Core and External plugins

QGIS plugins are implemented either as **Core Plugins** or **External Plugins**.

Core Plugins are maintained by the QGIS Development Team and are automatically part of every QGIS distribution. They are written in one of two languages: **C++** or **Python**.

Most of External Plugins are currently written in Python. They are stored either in the 'Official' QGIS Repository at https://plugins.qgis.org/plugins/ or in external repositories and are maintained by the individual authors. Detailed documentation about the usage, minimum QGIS version, home page, authors, and other important information are provided for the plugins in the Official repository. For other external repositories, documentation might be available with the external plugins themselves. External plugins documentation is not included in this manual.

To install or activate a plugin, go to *Plugins* menu and select *Manage and install plugins...*. Installed external python plugins are placed under the `python/plugins` folder of the active *user profile* path.

Paths to Custom C++ plugins libraries can also be added under *Settings* ► *Options* ► *System*.

25.1.2 The Plugins Dialog

The Settings tab

At the bottom of the left panel, the *Settings* tab is the main place you can configure which plugins can be displayed in your application. You can use the following options:

- *Check for Updates on Startup*. QGIS will automatically check for plugin updates at startup, at most once every 3 days. Whenever an installed plugin has an update available, QGIS will inform you.
- *Show also Experimental Plugins*. QGIS will show you plugins in early stages of development, which are generally unsuitable for production use. For these plugins, you can install either the stable or the experimental version, and at any moment switch from one to the other.
- *Show also Deprecated Plugins*. These plugins are usually unmaintained because they have replacement functions in QGIS, a lack of maintainers, they rely on functions that are no longer available in QGIS... They are generally unsuitable for production use and appear grayed in the plugins list.

By default, in the *Plugin Repositories* section, QGIS provides you with its official plugin repository with the URL `https://plugins.qgis.org/plugins/plugins.xml?qgis=version` (where `<version>` represents the exact QGIS version you are running). To add external author repositories, click *Add...* and fill in the *Repository Details* form with a name and the URL. The URL can be of `http://` or `file://` protocol type.

The default QGIS repository is an open repository and you don't need any authentication to access it. You can however deploy your own plugin repository and require an authentication (basic authentication, PKI). You can get more information on QGIS authentication support in 认证 chapter.

If you do not want one or more of the added repositories, they can be disabled from the Settings tab via the *Edit...* button, or completely removed with the *Delete* button.

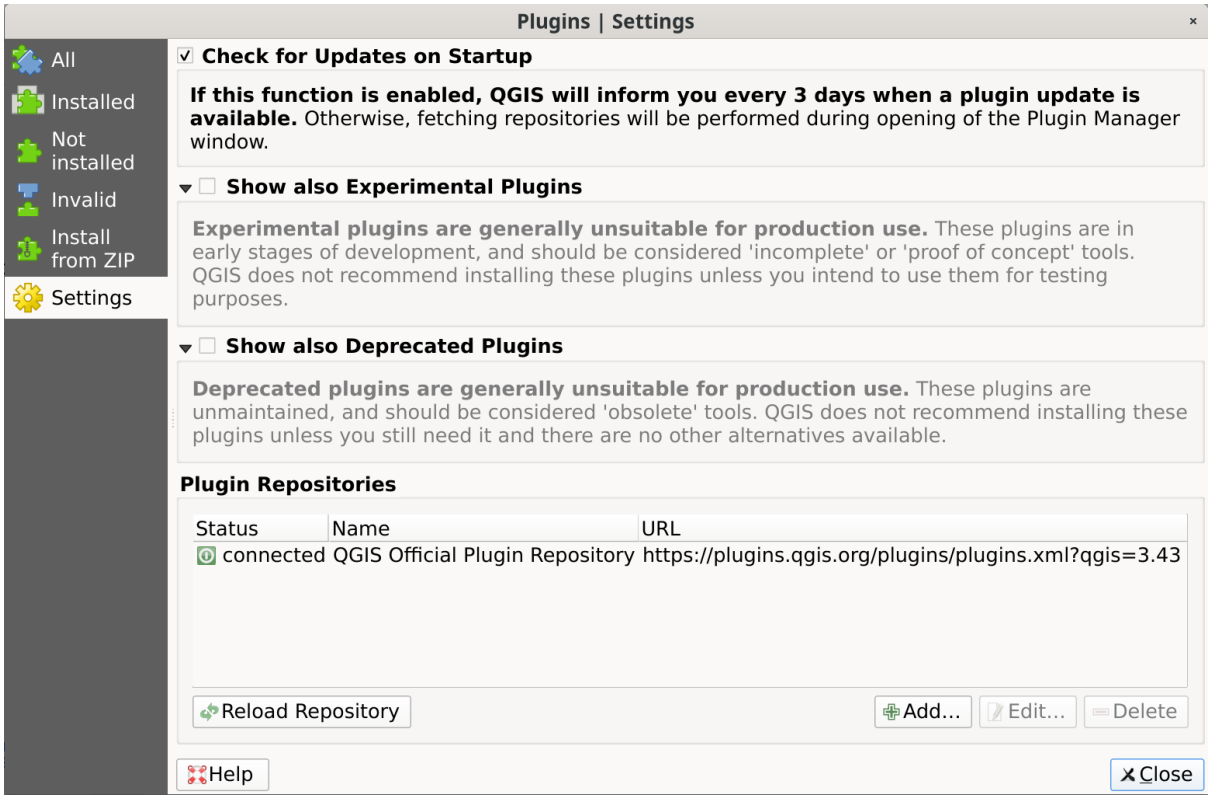


Fig. 25.1: The *Settings* tab

Browsing the plugins

The tabs

The upper tabs in the *Plugins* dialog provide you with lists of plugins based on their install, creation or update status. Depending on the plugins settings, available tabs can be:

- *All*: shows all the available plugins in the enabled repositories
- *Installed*: shows both the plugins you installed and the core plugins that are installed by default and you can not uninstall
- *Not installed*: shows uninstalled or not yet installed plugins in the enabled repositories
- *New*: shows plugins released since the last *Check for Updates on Startup*
- *Upgradeable*: shows installed plugins that have published a more recent version in the repository
- *Invalid*: shows all installed plugins that are currently broken for any reason (missing dependency, errors while loading, incompatible functions with QGIS version...)

At the top of the tabs, a *Search* function helps you find any plugin using metadata information (author, name, description, tag,...).

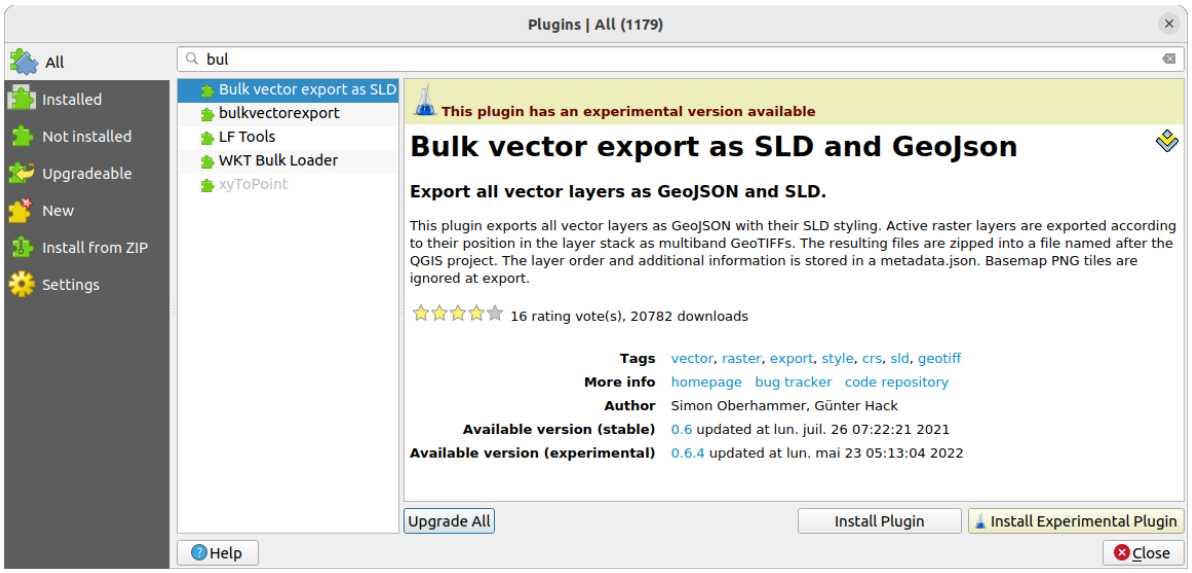


Fig. 25.2: Searching a plugin from the *All* tab

The Plugins

Select a plugin and you will have some metadata displayed in the right panel:

- information on whether the plugin is experimental or has an experimental version available (if *Show also Experimental Plugins* is checked)
- summary and description
- rating vote(s) (you can vote for your preferred plugin!)
- tags
- some useful links to the home page, tracker and code repository

- author(s)
- version(s) available with link to download page in the repository, or path to local folder for installed plugins

The *Plugin Manager* dialog allows you to interact with the latest version of the plugins. When enabled, the experimental version can be shown only if it is more recent than the latest stable version. Depending on the active tab, whether the selected plugin is installed, you will be given some of the following options:

- *Install*: installs the latest stable version of the selected plugin
- *Install Experimental Plugin*: installs the experimental version of the selected plugin
- *Reinstall Plugin*: installs the same stable version of the plugin e.g. after it has failed to load
- *Reinstall Experimental Plugin*: installs the same experimental version of the plugin e.g. after it has failed to load
- *Upgrade Plugin*: upgrades selected plugin to its latest stable version
- *Upgrade Experimental Plugin*: upgrades selected plugin to its experimental version
- *Upgrade All*: upgrades all installed plugins to their more recent stable or experimental version (depending on whether their previously installed version was stable or experimental).
- *Downgrade Plugin*: moves from the experimental version of the plugin to its previous stable version
- *Downgrade Experimental Plugin*: moves from an experimental version of the plugin to its latest published experimental version. This may occur when playing with a not yet published version.
- *Uninstall Plugin*: removes the installed plugin from the user profile

An installed plugin displays a checkbox on its left. Uncheck it to temporarily deactivate the plugin.

Right-click on a plugin in the list and you will be able to sort the plugins list by various metadata. The new order applies to all the tabs. Sort options are:

- *Sort by Name*
- *Sort by Downloads*
- *Sort by Vote*
- *Sort by Status*
- *Sort by Date Created*
- *Sort by Date Updated*

The Install from ZIP tab

The *Install from ZIP* tab provides a file selector widget to import plugins in a zipped format, e.g. plugins downloaded directly from their repository. Encrypted files are supported.

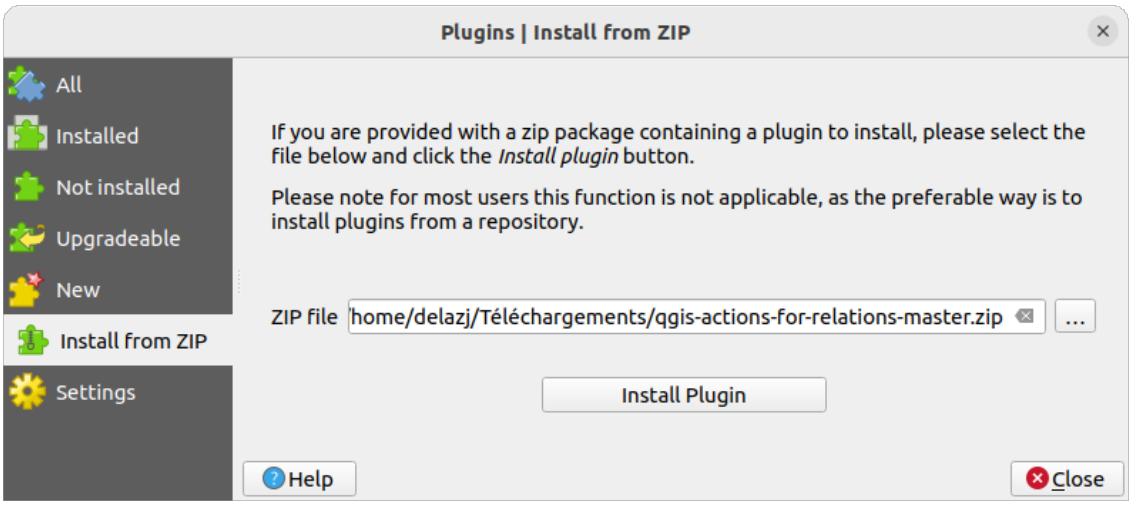


Fig. 25.3: The *Install from zip* tab

25.2 使用 QGIS 核心插件

25.2.1 DB Manager Plugin

The DB Manager Plugin is intended to be the main tool to integrate and manage spatial database formats supported by QGIS (PostgreSQL, SpatiaLite, GeoPackage, Oracle Spatial, Virtual layers) in one user interface. The DB Manager Plugin provides several features. You can drag layers from the QGIS Browser into the DB Manager, and it will import your layer into your spatial database. You can drag and drop tables between spatial databases and they will get imported.

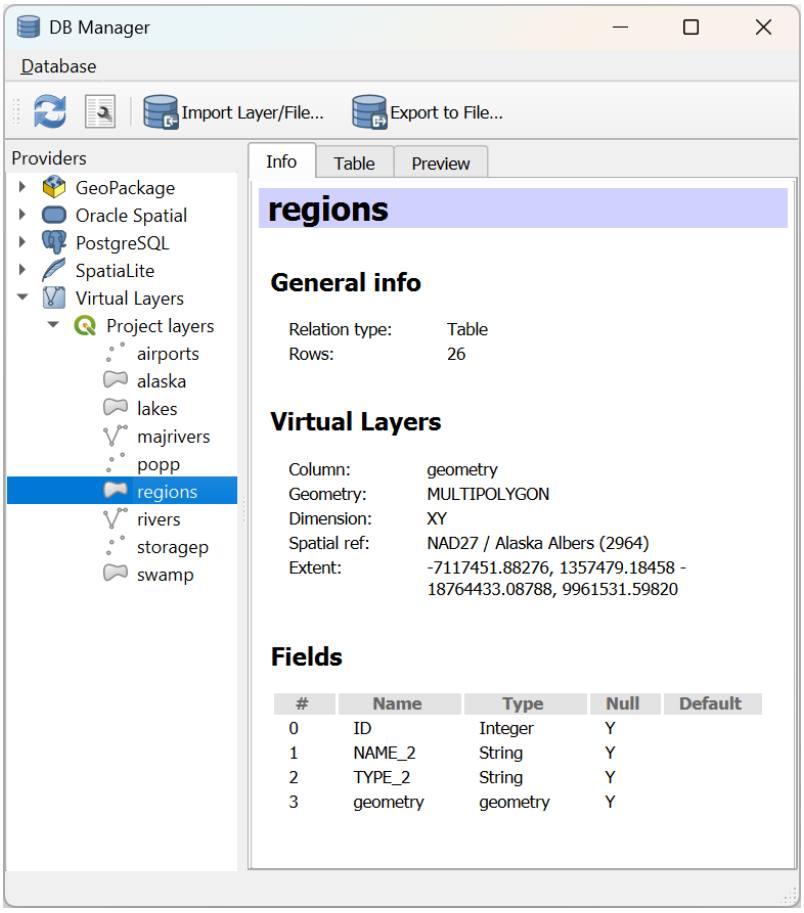


Fig. 25.4: 数据库管理器对话框

The *Database* menu allows you to connect to an existing database, to start the SQL window and to exit the DB Manager Plugin. Once you are connected to an existing database, the menus *Schema* (relevant for DBMSs, such as PostgreSQL) and *Table* will appear.

The *Schema* menu includes tools to create and delete (only if empty) schemas and, if topology is available (e.g. with PostGIS topology), to start a *TopoViewer*.

The *Table* menu allows you to create and edit tables and to delete tables and views. It is also possible to empty tables and to move tables between schemas. You can *Run Vacuum Analyze* for the selected table. *Vacuum* reclaims space and makes it available for reuse, and *analyze* updates statistics that is used to determine the most efficient way to execute a query. *Change Logging...* allows you to add change logging support to a table. Finally, you can *Import Layer/File...* and *Export to File...*.

Note

Using the DB Manager it is possible to add comments for tables and columns of a PostgreSQL Database.

The *Providers* window lists all existing databases supported by QGIS. With a double-click, you can connect to the database. With the right mouse button, you can rename and delete existing schemas and tables. Tables can also be added to the QGIS canvas with the context menu.

If connected to a database, the **main** window of the DB Manager offers four tabs. The *Info* tab provides information about the table and its geometry, as well as about existing fields, constraints and indexes. It allows you to create a spatial index on a the selected table. The *Table* tab shows the table, and the *Preview* tab renders the geometries as preview. When you open an *SQL Window*, it will be placed in a new tab.

Working with the SQL Window

You can use the DB Manager to execute SQL queries against your spatial database. Queries can be saved and loaded, and there the *SQL Query Builder* will help you formulate your queries. You can even view spatial output by checking *Load as new layer* and specifying *Column(s) with unique values* (IDs), *Geometry column* and *Layer name (prefix)*. It is possible to highlight a portion of the SQL to only execute that portion when pressing `Ctrl+R` or clicking the *Execute* button.

QGIS also adds support for the `REGEXP` function in some providers. This allows users to use regular expressions in SQL filters or expressions, for example:

```
SELECT * FROM places WHERE name REGEXP '^A';
```

This returns all features where the `name` field starts with the letter A.

After executing your query, you can select specific cells in the result set. Use the `Ctrl+C` shortcut to copy the selected cells to the clipboard. The copied data is available as a formatted table. This allows you to paste the data into other applications, such as spreadsheet where it will show up as a table.

The *Query History* button stores the last 20 queries of each database and provider.

Double clicking on an entry will add the string to the SQL window.

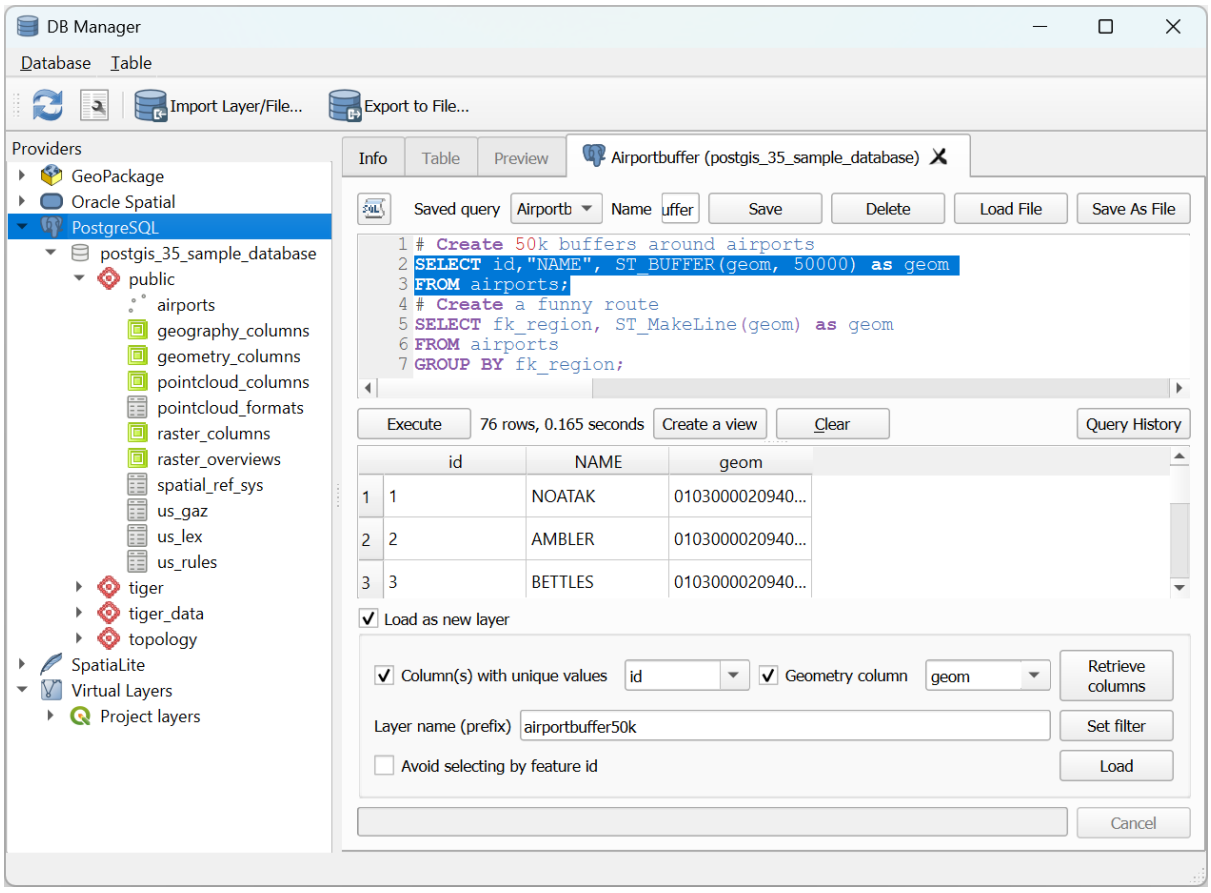


Fig. 25.5: Executing SQL queries in the DB Manager SQL window

Note

The SQL Window can also be used to create Virtual Layers. In that case, instead of selecting a database, select **QGIS Layers** under **Virtual Layers** before opening the SQL Window. See 创建虚拟图层 for instructions on the SQL syntax to use.

Import Vector Layer

You can import layer or file into your database. Here are the parameters you can set for the import process:

- *Input*: Select the layer or file to import. Using the dropdown menu, select from the list of loaded layers in QGIS or click on the ... button to select a file from disk. Check the *Import only selected features* to import only the selected features of the layer.
- *Output table*: Choose the *Schema* and provide a name for the new table.
- *Options*: Here are some options for the import process:
  - *Primary key*: Provide naming for the primary key field. By default, it is named `id`.
  - *Geometry column*: Provide naming for the geometry column. By default, it is named `geom`.
  - *Source SRID*: Define the SRID for the geometry column. By default, it uses the layer's CRS.
  - *Target SRID*: Define the target SRID to reproject the geometries during the import process. By default, it uses the layer's CRS.
  - *Encoding*: Define the encoding of the source data. By default, it uses `Automatic`. It is QGIS's automatic detection mode that attempts to guess the file's character encoding based on available metadata or system locale.
  - *Replace destination table (if exists)*: If a table with the same name already exists in the selected schema, it will be replaced.
  - *Do not promote to multi-part*: Geometries will be imported as single-part geometries.
  - *Convert field names to lower case*: All field names will be converted to lower case.
  - *Create spatial index*: A spatial index will be created on the geometry column after import.
  - *Comment*: Add comments to table. Only available for PostgreSQL databases.

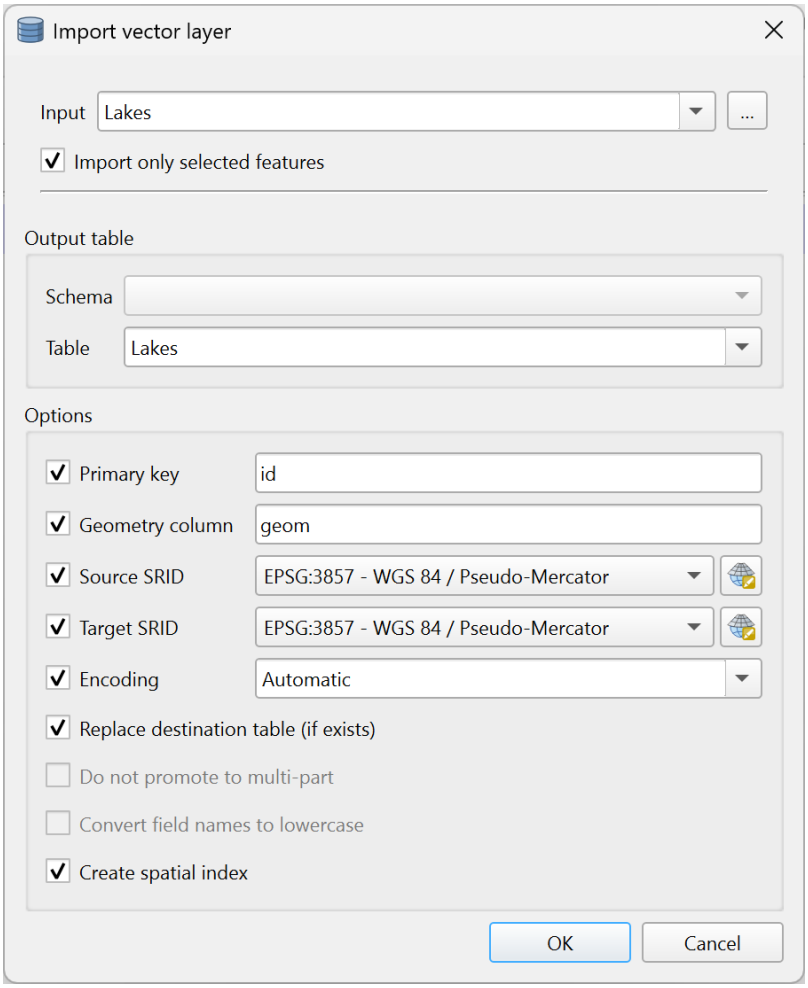


Fig. 25.6: Importing a vector layer into a spatial database using DB Manager

Export to Vector File

To export a table from your database to a vector file, select the desired *Format* and *Save as* location. Under *Options*, you can set the following parameters:

- *Source SRID*: Define the source SRID of the geometry column. By default, it uses the layer's CRS.
- *Target SRID*: Define the target SRID to reproject the geometries during the export process. By default, it uses the layer's CRS.
- *Encoding*: Define the encoding of the output data.
- *Replace destination table (if exists)*: If a table with the same name already exists at the selected location, it will be replaced.

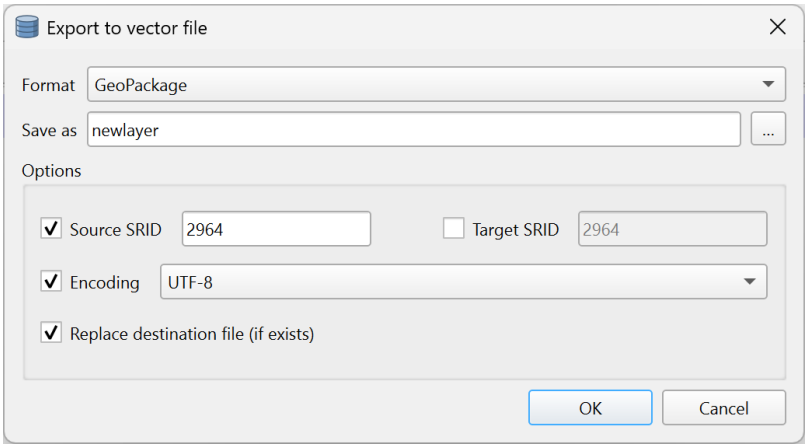


Fig. 25.7: Exporting a table to a vector file using DB Manager

25.2.2 Geometry Checker Plugin

Geometry Checker is a powerful core plugin to check and fix the geometry validity of a layer. It is available from the *Vector* menu (*Check Geometries...*).

Configuring the checks

The *Check Geometries* dialog shows different grouped settings in the first tab (*Setup*):

- *Input vector layers*: to select the layers to check. A *Only selected features* checkbox can be used to restrict the checking to the geometries of the selected features.
- *Allowed geometry types* gives the chance to restrict the geometry type of the input layer(s) to:
  - Point
  - Multipoint
  - Line
  - Multiline
  - Polygon
  - Multipolygon
- *Geometry validity*. Depending on geometry types you can choose between:
  - *Self intersections*
  - *Duplicate nodes*
  - *Self contacts*
  - *Polygon with less than 3 nodes*.
- *Geometry properties*. Depending on geometry types, different options are available:
  - *Polygons and multipolygons may not contain any holes*
  - *Multipart objects must consist of more than one part*

  - *Lines must not have dangles*
- *Geometry conditions*. Allows you to add some condition to validate the geometries with:
  - *Minimal segment length (map units)* 1,00
  - *Minimum angle between segment (deg)* 1,00
  - *Minimal polygon area (map units sqr.)* 1,00
  - *No sliver polygons* with a *Maximum thinness* 1,00 and a *Max. area (map units sqr.)* 1,00
- *Topology checks*. Depending on geometry types, many different options are available:
  - *Checks for duplicates*
  - *Checks for features within other features*
  - *Checks for overlaps smaller than* 1,00
  - *Checks for gaps smaller than* 1,00
  - *Points must be covered by lines*
  - *Points must properly lie inside a polygon*
  - *Lines must not intersect any other lines*
  - *Lines must not intersect with features of layer*
  - *Polygons must follow boundaries of layer*
- *Tolerance*. You can define the tolerance of the check in map layer units.
- *Output vector layer* gives the choice to:
  - *Modify input layer*
  - *Create new layers*

When you are happy with the configuration, you can click on the *Run* button.

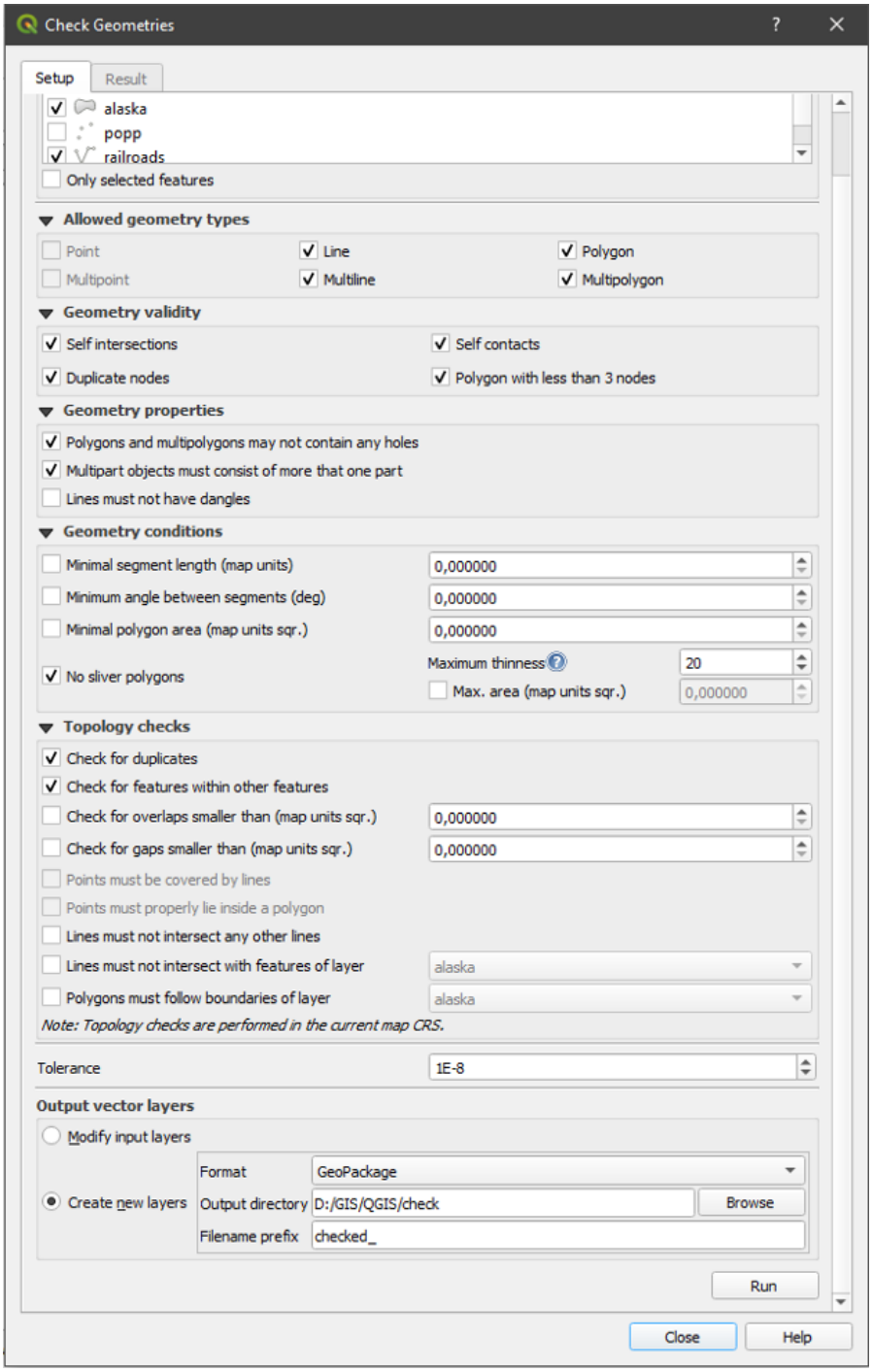


Fig. 25.8: The Geometry Checker Plugin

The *Geometry Checker Plugin* can find the following errors:

- Self intersections: a polygon with a self intersection
- Duplicate nodes: two duplicates nodes in a segment
- Holes: hole in a polygon
- Segment length: a segment length lower than a threshold
- Minimum angle: two segments with an angle lower than a threshold
- Minimum area: polygon area lower than a threshold

- Silver polygon: this error come from very small polygon (with small area) with a large perimeter
- Duplicates features
- Feature within feature
- Overlaps: polygon overlapping
- Gaps: gaps between polygons

The following figure shows the different checks made by the plugin.

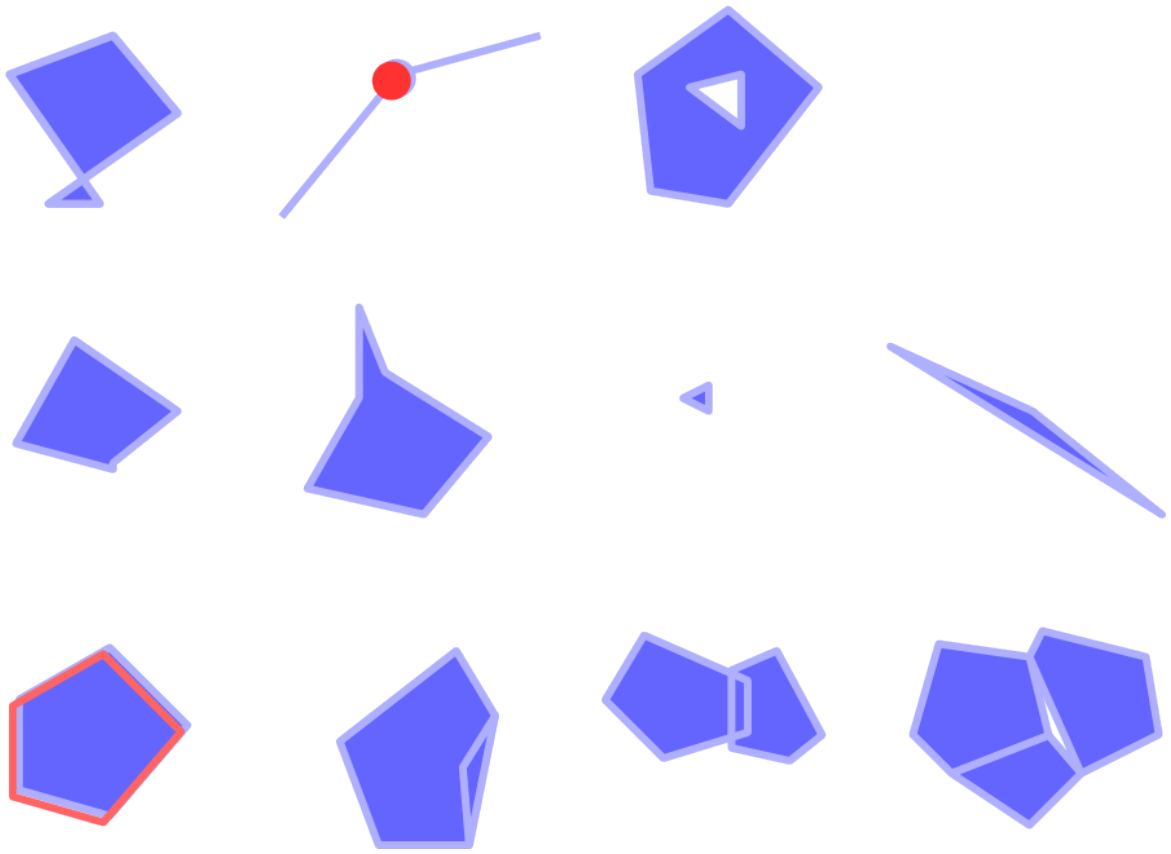


Fig. 25.9: Some checks supported by the plugin

Analysing the results

The results appear in the second tab (*Result*) and as an overview layer of the errors in the canvas (its name has the default prefix `checked_`). A table lists the *Geometry check result* with one error per row and columns containing: the layer name, an ID, the error type, then the coordinates of the error, a value (depending on the type of the error) and finally the resolution column which indicates the resolution of the error. At the bottom of this table, you can *Export* the error into different file formats. You also have a counter with the number of total errors and fixed ones.

You can select a row to see the location of the error. You can change this behavior by selecting another action between *Error* (default), *Feature*, *Don't move*, and *Highlight selected features*.

Below the zoom action when clicking on the table row, you can:

- *Show selected features in attribute table*
- *Fix selected errors using default resolution*
- *Fix selected errors, prompt for resolution method* You will see a window to choose the resolution's method among which:

  - Merge with neighboring polygon with longest shared edge
  - Merge with neighboring polygon with largest area
  - Merge with neighboring polygon with identical attribute value, if any, or leave as is
  - Delete feature
  - No action
- *Error resolution settings* allows you to change the default resolution method depending on the error type

> **Tip**
>
> **Fix multiple errors**
>
> You can fix multiple errors by selecting more than one row in the table with the *CTRL + click* action.

Finally, you can choose which *Attribute to use when merging features by attribute value*.

25.2.3 MetaSearch Catalog Client

简介

MetaSearch is a QGIS plugin to interact with metadata catalog services, supporting both the OGC API - Records and OGC Catalog Service for the Web (CSW) standards.

MetaSearch provides an easy and intuitive approach and user-friendly interface to searching metadata catalogs within QGIS.

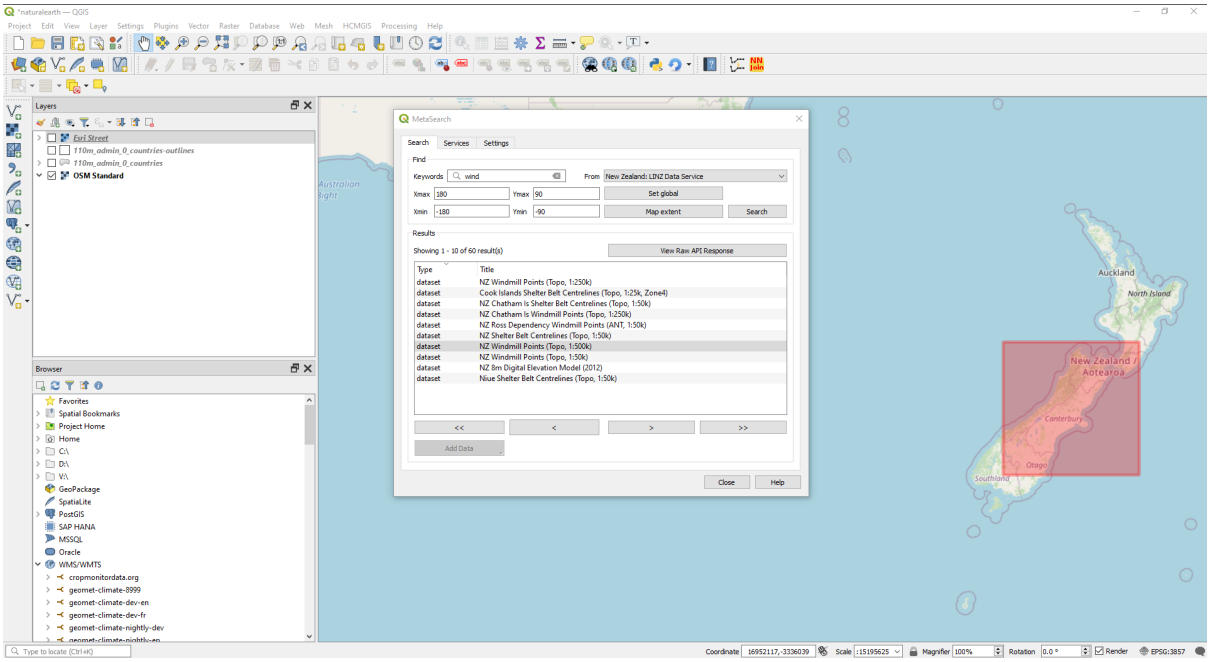


Fig. 25.10: Search and results of Services in MetaSearch

Working with Metadata Catalogs in QGIS

MetaSearch is included by default in QGIS, with all of its dependencies, and can be enabled from the QGIS Plugin Manager.

OGC API - Records

OGC API - Records is an OGC (Open Geospatial Consortium) standard for the discovery of geospatial resources on the Web.

CSW (Catalog Service for the Web)

CSW (Catalog Service for the Web) is an OGC (Open Geospatial Consortium) specification that defines common interfaces to discover, browse and query metadata about data, services, and other potential resources.

Startup

To start MetaSearch, click the icon or select *Web* ► *MetaSearch* ► *MetaSearch* via the QGIS main menu. The MetaSearch dialog will appear. The main GUI consists of three tabs: *Services*, *Search* and *Settings*.

Managing Catalog Services

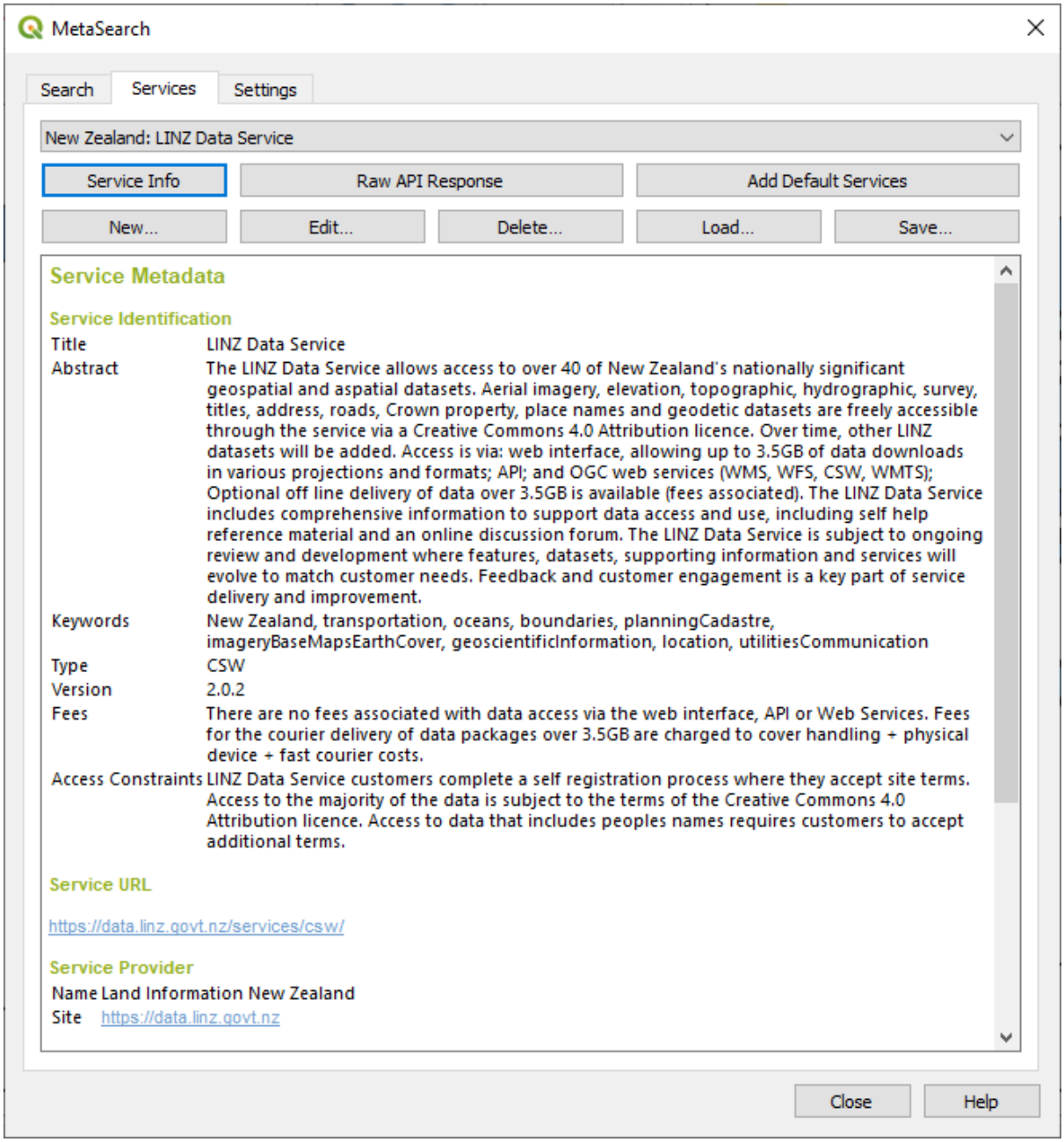


Fig. 25.11: Managing Catalog Services

The *Services* tab allows the user to manage all available catalog services. MetaSearch provides a default list of Catalog Services, which can be added by pressing the *Add Default Services* button.

To find all listed Catalog Service entries, click the dropdown select box.

To add a Catalog Service entry:

1. Click the *New* button
2. Enter a *Name* for the service, as well as the *URL* (endpoint). Note that for OGC CSW 2.0.2 Catalogs, only the base URL is required (not a full GetCapabilities URL). For OGC API - Records Catalogs, the URL should be the path to the collection endpoint
3. If the Catalog requires authentication, enter the appropriate *User name* and *Password* credentials.

4. Click *OK* to add the service to the list of entries.

To edit an existing Catalog Service entry:

1. Select the entry you would like to edit
2. Click the *Edit* button
3. And modify the *Name* or *URL* values
4. 单击 确定。

To delete a Catalog Service entry, select the entry you would like to delete and click the *Delete* button. You will be asked to confirm deleting the entry.

MetaSearch allows loading and saving connections to an XML file. This is useful when you need to share settings between applications. Below is an example of the XML file format.

```
<?xml version="1.0" encoding="UTF-8"?>
<qgsCSWConnections version="1.0">
    <csw type="OGC CSW 2.0.2" name="Data.gov CSW" url="https://catalog.data.gov/
↪csw-all"/>
    <csw type="OGC CSW 2.0.2" name="Geonorge - National CSW service for Norway"␣
↪url="https://www.geonorge.no/geonetwork/srv/eng/csw"/>
    <csw type="OGC CSW 2.0.2" name="Geoportale Nazionale - Servizio di ricerca␣
↪Italiano" url="http://www.pcn.minambiente.it/geoportal/csw"/>
    <csw type="OGC CSW 2.0.2" name="LINZ Data Service" url="http://data.linz.govt.
↪nz/feeds/csw"/>
    <csw type="OGC CSW 2.0.2" name="Nationaal Georegister (Nederland)" url="http://
↪www.nationaalgeoregister.nl/geonetwork/srv/eng/csw"/>
    <csw type="OGC CSW 2.0.2" name="RNDT - Repertorio Nazionale dei Dati␣
↪Territoriali - Servizio di ricerca" url="http://www.rndt.gov.it/RNDT/CSW"/>
    <csw type="OGC CSW 2.0.2" name="UK Location Catalogue Publishing Service" url=
↪"http://csw.data.gov.uk/geonetwork/srv/en/csw"/>
    <csw type="OGC CSW 2.0.2" name="UNEP/GRID-Geneva Metadata Catalog" url="http://
↪metadata.grid.unep.ch:8080/geonetwork/srv/eng/csw"/>
</qgsCSWConnections>
```

To load a list of entries:

1. Click the *Load* button. A new window will appear.
2. Click the *Browse* button and navigate to the XML file of entries you wish to load.
3. Click *Open*. The list of entries will be displayed.
4. Select the entries you wish to add from the list and click *Load*.

Click the *Service Info* button to display information about the selected Catalog Service such as service identification, service provider and contact information. If you would like to view the raw API response, click the *Raw API Response* button. A separate window will open displaying server information in raw JSON or XML format.

Searching Catalog Services

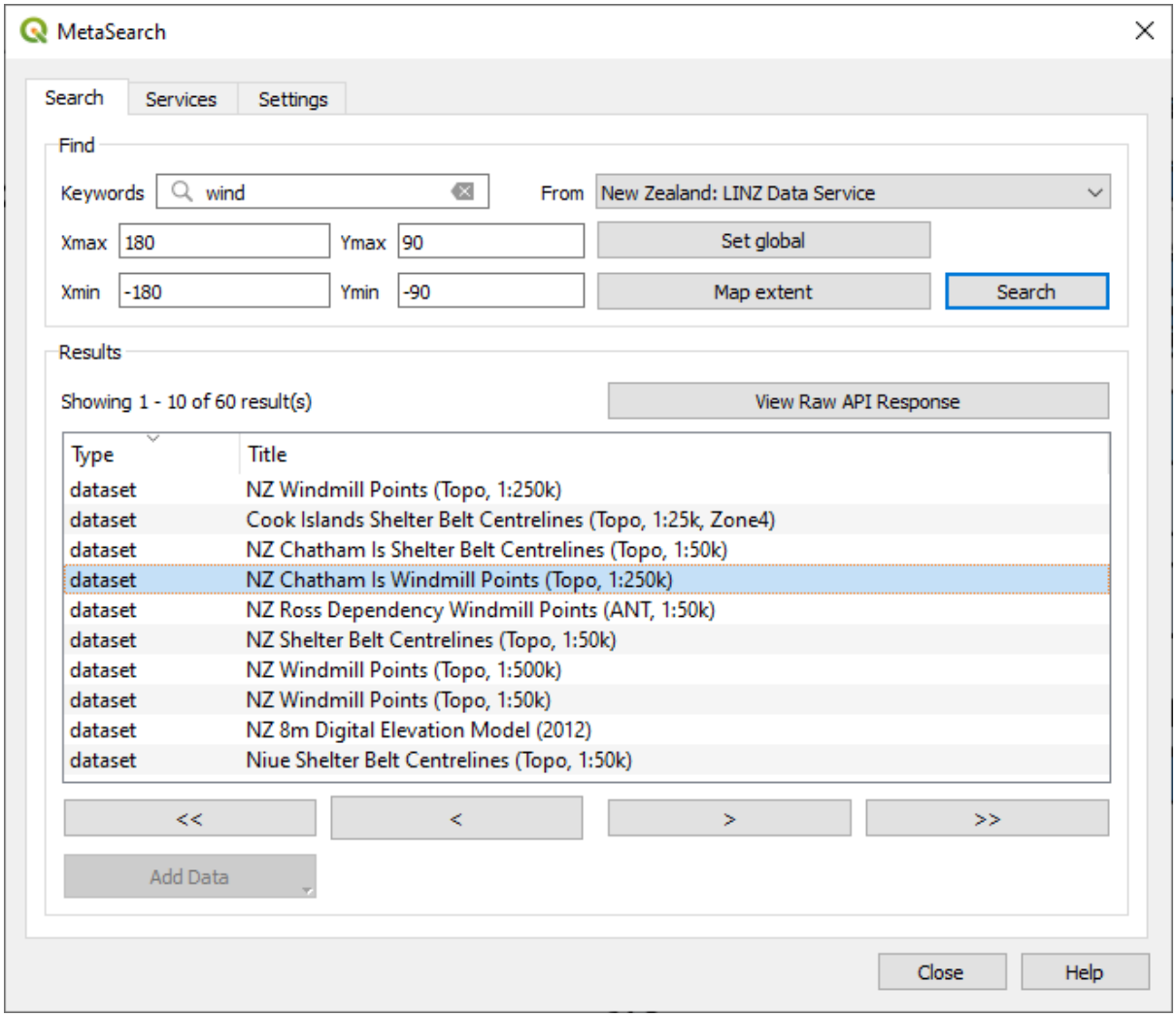


Fig. 25.12: Searching catalog services

The *Search* tab allows the user to query Catalog Services for data and services, set various search parameters and view results.

The following search parameters are available:

- *Keywords*: free text search keywords;
- *From*: the Catalog Service to perform the query against;
- **Bounding box**: the spatial area of interest to filter, defined by *Xmax*, *Xmin*, *Ymax*, and *Ymin*. Click *Set Global* to do a global search, click *Map Extent* to do a search in the visible area, or enter values manually.

Clicking the *Search* button will search the selected Metadata Catalog. Search results are displayed in a list, and can be sorted by clicking on the column header. You can navigate through search results with the directional buttons below the search results.

Select a result and:

- Click the *View Raw API Response* button to open a window with the service response in raw JSON or XML format.
- If the metadata record has an associated bounding box, a footprint of the bounding box will be displayed on the map.

- Double-click the record to display the record metadata with any associated access links. Clicking a link opens the link in the user's web browser.
- If the record is a supported web service (WMS/WMTS, WFS, WCS, ArcGIS REST Service, etc.) or a GIS File, the *Add Data* button will be enabled. When clicking this button, MetaSearch will verify if this is a valid OWS. The service will then be added to the appropriate QGIS connection list, and the appropriate connection dialog will appear. If you choose *Add GIS File* the linked GIS file will be added to the current project.

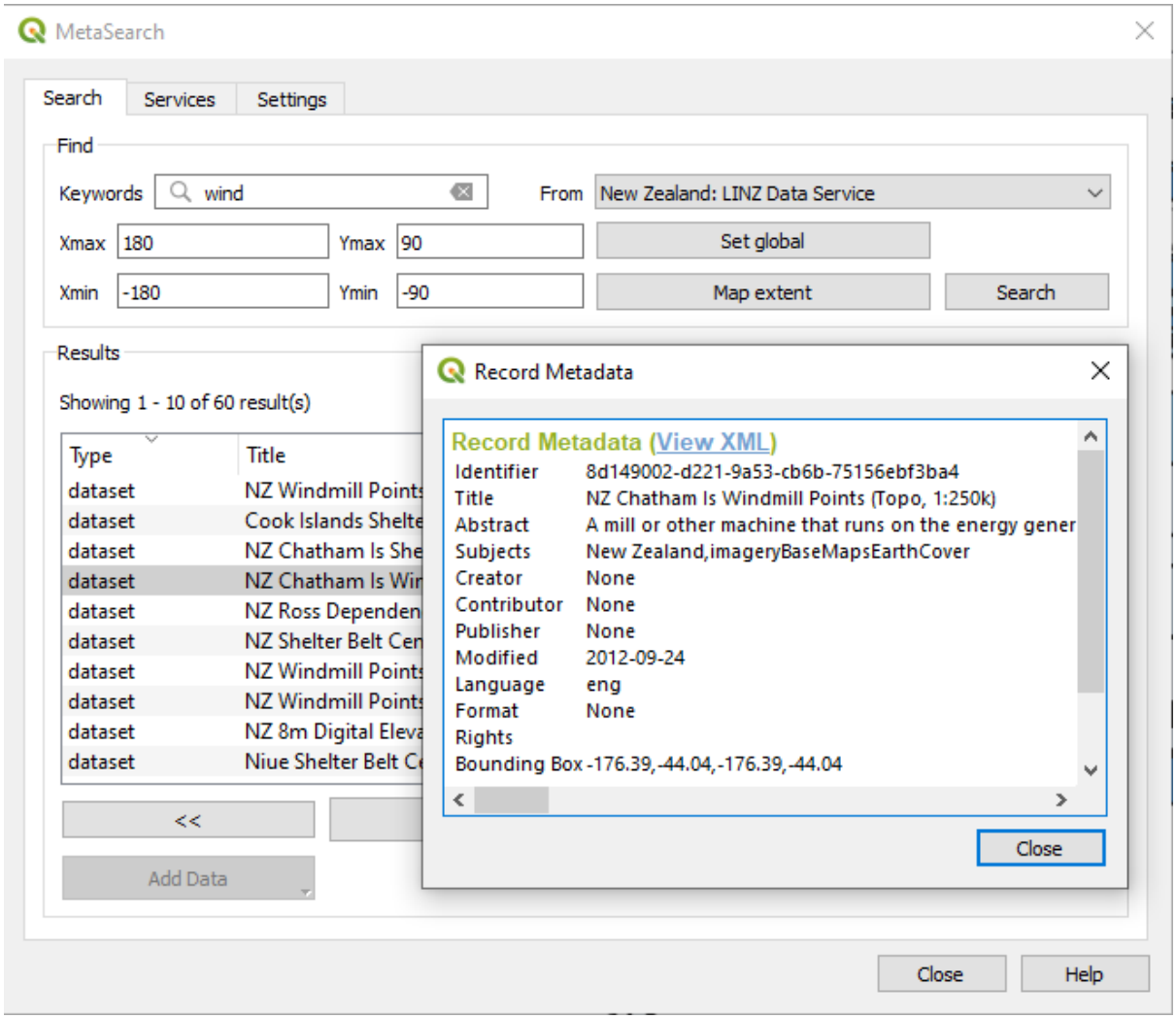


Fig. 25.13: Metadata record display

设置

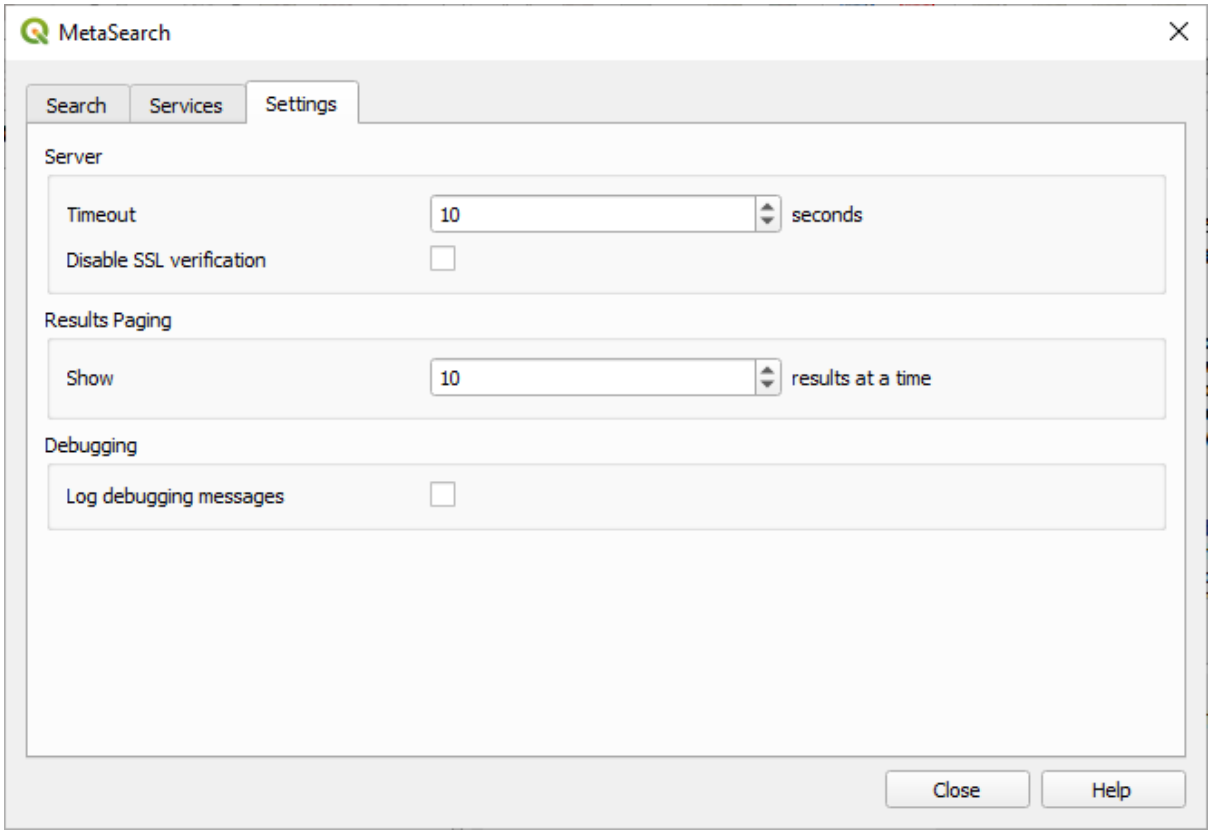


Fig. 25.14: MetaSearch settings

You can fine tune MetaSearch with the following *Settings*:

- *Server Timeout*: when searching metadata catalogs, the number of seconds for blocking connection attempt. Default value is 10.
- *Disable SSL verification*: option to switch off ssl verification.
- *Results paging*: when searching metadata catalogs, the number of results to show per page. Default value is 10.
- *Log debugging messages*: option to enable the logging of debugging messages in the 日志消息面板.

Catalog Server Errors

In some cases, the Catalog will work in a web browser, but not in MetaSearch. This may be due to the Catalog server's configuration/setup. Catalog server providers should ensure URLs are consistent and up to date in their configuration (this is common in HTTP -> HTTPS redirection scenarios). Please see the pycsw FAQ item for a deeper explanation of the issue and fix. Although the FAQ item is pycsw specific it can also apply in general to other Catalog servers.

25.2.4 离线编辑

For data collection, it is a common situation to work with a laptop or a cell phone offline in the field. Upon returning to the network, the changes need to be synchronized with the master datasource (e.g., a PostgreSQL database). If several persons are working simultaneously on the same datasets, it is difficult to merge the edits by hand, even if people don't change the same features.

The Offline Editing Plugin automates the synchronisation by copying the content of the datasource to a SpatiaLite or GeoPackage database and storing the offline edits to dedicated tables. After being connected to the network again, it is possible to apply the offline edits to the master dataset.

To use the plugin:

1. Open a project with some vector layers (e.g., from an Esri Shapefile, PostgreSQL or WFS-T datasource).
2. Assuming you have already enabled the plugin (see *Core and External plugins*) go to *Database* ► *Offline Editing* ► *Convert to offline project*. The eponym dialog opens.
3. Select the *Storage type*. It can be of *GeoPackage* or *SpatiaLite* database type.
4. Use the *Browse* button to indicate the location of the database in which to store the *Offline data*. It can be an existing file or one to create.
5. In the *Select remote layers* section, check the layers you'd like to save. The content of the layers is saved to database tables.

   Note

   Since target database formats do not have native list support, the offline editing plugin transforms {string, number} list fields into string fields where values are separated by commas. This allows reading and edit of the contents of those fields when offline.

   If you would like to handle both the field from the original layer and the offline layer, you can rely on the *try()* and *array* expression functions, e.g.:

   ```
   try(array_contains("field",1),array_contains(string_to_array("field"),1))
   ```

6. You can check *Only synchronize selected features if a selection is present* allowing to only save and work on a subset. It can be invaluable in case of large layers.

   This is all!
7. Save your project and bring it on the field.
8. Edit the layers offline.
9. After being connected again, upload the changes using *Database* ► *Offline Editing* ► *Synchronize*.

Note

Layers that are used offline are marked with the icon in the *Layers* panel.

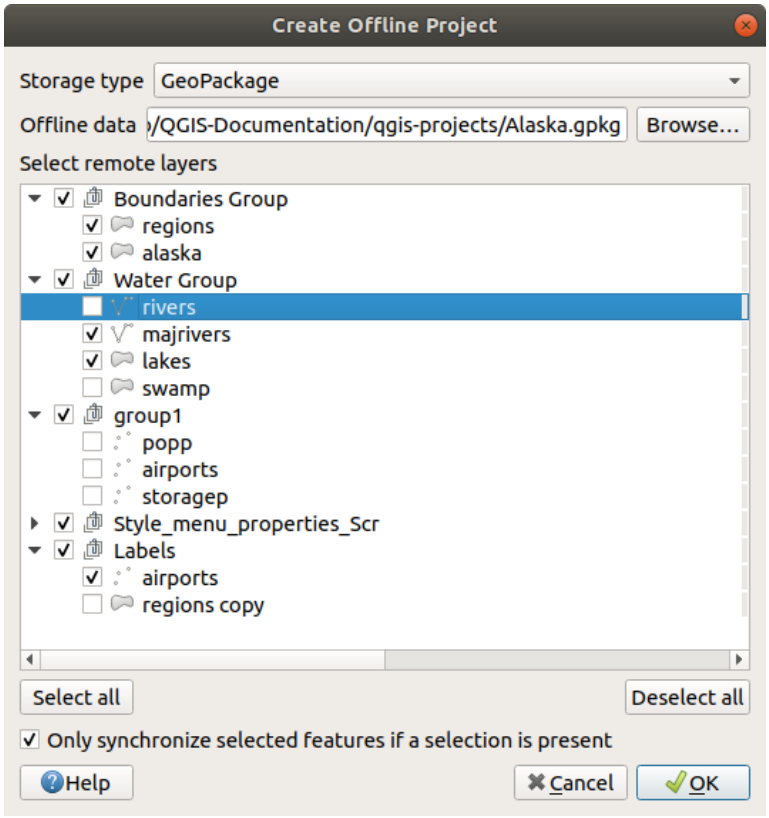


Fig. 25.15: Create an offline project

25.2.5 Topology Checker Plugin

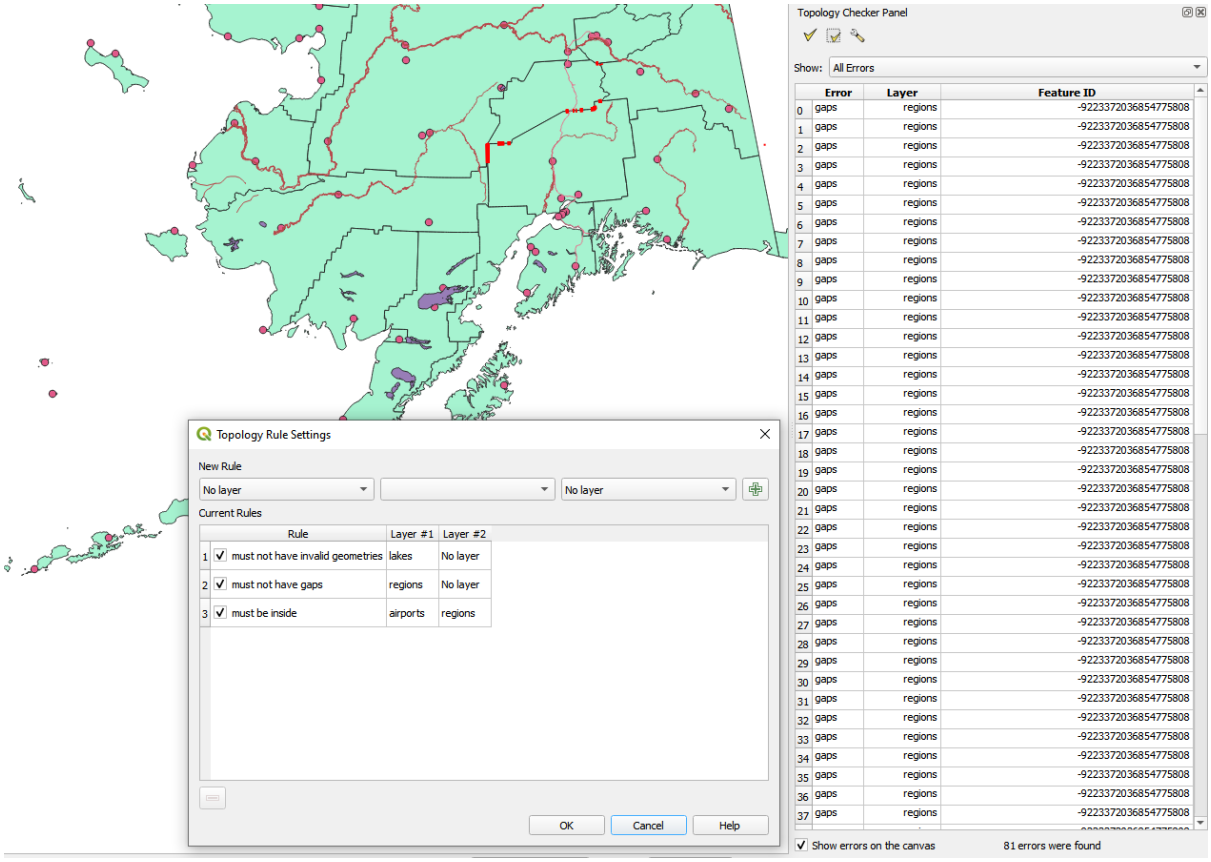


Fig. 25.16: The Topology Checker Plugin

Topology describes the relationships between points, lines and polygons that represent the features of a geographic region. With the Topology Checker plugin, you can look over your vector files and check the topology with several topology rules. These rules check with spatial relations whether your features 'Equal', 'Contain', 'Cover', are 'CoveredBy', 'Cross', are 'Disjoint', 'Intersect', 'Overlap', 'Touch' or are 'Within' each other. It depends on your individual questions which topology rules you apply to your vector data (e.g., normally you won't accept overshoots in line layers, but if they depict dead-end streets you won't remove them from your vector layer).

QGIS has a built-in topological editing feature, which is great for creating new features without errors. But existing data errors and user-induced errors are hard to find. This plugin helps you find such errors through a list of rules.

To enable the *Topology checker* plugin:

1. Go to *Plugins* menu
2. Open *Manage and Install plugins* and choose *Installed*
3. Enable *Topology checker*
4. Close the *Plugin manager* dialog. A *Topology checker* entry is added to the *Vector* menu.

After you enabled *Topology checker* open it and choose *Configure* to create your topology rules.

On **point layers** the following rules are available:

- *Must be covered by*: Here you can choose a vector layer from your project. Points that aren't covered by the given vector layer occur in the 'Error' field.
- *Must be covered by endpoints of*: Here you can choose a line layer from your project.

- *Must be inside*: Here you can choose a polygon layer from your project. The points must be inside a polygon. Otherwise, QGIS writes an 'Error' for the point.
- *Must not have duplicates*: Whenever a point is represented twice or more, it will occur in the 'Error' field.
- *Must not have invalid geometries*: Checks whether the geometries are valid.
- *Must not have multi-part-geometries*: All multi-part points are written into the 'Error' field.

On **line layers**, the following rules are available:

- *End points must be covered by*: Here you can select a point layer from your project.
- *Must not have dangles*: This will show the overshoots in the line layer.
- *Must not have duplicates*: Whenever a line feature is represented twice or more, it will occur in the 'Error' field.
- *Must not have invalid geometries*: Checks whether the geometries are valid.
- *Must not have multi-part geometries*: Sometimes, a geometry is actually a collection of simple (single-part) geometries. Such a geometry is called multi-part geometry. If it contains just one type of simple geometry, we call it multi-point, multi-linestring or multi-polygon. All multi-part lines are written into the 'Error' field.
- *Must not have pseudos*: A line geometry's endpoint should be connected to the endpoints of two other geometries. If the endpoint is connected to only one other geometry's endpoint, the endpoint is called a pseudo node.

On **polygon layers**, the following rules are available:

- *Must contain*: Polygon layer must contain at least one point geometry from the second layer.
- *Must not have duplicates*: Polygons from the same layer must not have identical geometries. Whenever a polygon feature is represented twice or more it will occur in the 'Error' field.
- *Must not have gaps*: Adjacent polygons should not form gaps between them. Administrative boundaries could be mentioned as an example (US state polygons do not have any gaps between them...).
- *Must not have invalid geometries*: Checks whether the geometries are valid. Some of the rules that define a valid geometry are:
  - Polygon rings must close.
  - Rings that define holes should be inside rings that define exterior boundaries.
  - Rings may not self-intersect (they may neither touch nor cross one another).
  - Rings may not touch other rings, except at a point.
- *Must not have multi-part geometries*: Sometimes, a geometry is actually a collection of simple (single-part) geometries. Such a geometry is called multi-part geometry. If it contains just one type of simple geometry, we call it multi-point, multi-linestring or multi-polygon. For example, a country consisting of multiple islands can be represented as a multi-polygon.
- *Must not overlap*: Adjacent polygons should not share common area.
- *Must not overlap with*: Adjacent polygons from one layer should not share common area with polygons from another layer.

When you create a *New rule* click on the Add rule to include it to the *Current rules*. You can enable or disable individual rules by clicking on the checkbox. Right-clicking over a rule provides the following options:

- *Select All* the rules
- *Activate* or *Deactivate* the selected rules
- *Toggle activation* of selected rules
- *Delete* selected rules. This can also be achieved with the Delete selected rules button.

Press *OK* and then choose from the *Topology checker* panel:

- Validate All: applies the active rules to all the features of the involved layer(s)
- or Validate Extent: applies the active rules to the features of the involved layer(s), within the current map canvas. The button is kept pushed and the results will update as the map canvas extent changes.

Errors will show up in the table of results containing type of error, layer and feature ID. Use *Filter errors by rule* menu to filter the errors to a specific error type.

Check *Show errors on the canvas* to show error location on the canvas. Clicking a row in the table will zoom the map canvas to the concerned feature, where you can use *QGIS digitizing tools* to fix the error.

下面是 QGIS 提供的核心插件列表。默认情况下，它们不一定启用。

Icon	插件	描述	手册参考
	数据库管理器	在 QGIS 中管理您的数据库	*DB Manager Plugin*
	几何图形检查器	检查并修复矢量几何图形中的错误	*Geometry Checker Plugin*
	GRASS 7	GRASS 功能	*GRASS GIS Integration*
	GRASS GIS provider	GRASS GIS 数据处理功能	*GRASS GIS Integration*
	MetaSearch Catalog Client	与元数据目录服务交互 (CSW)	*MetaSearch Catalog Client*
	离线编辑	离线编辑并与数据库同步	*离线编辑*
	数据处理	空间数据处理架构	*QGIS 处理架构*
	拓扑检查器	在矢量图层中查找拓扑错误	*Topology Checker Plugin*

Note

To use the Core Plugins GRASS 7, GRASS GIS provider, they have to be configured. Informations can be found *here*.

25.3 QGIS Python console

As you will see later in this chapter, QGIS has been designed with a plugin architecture. Plugins can be written in Python, a very famous language in the geospatial world.

QGIS brings a Python API (see PyQGIS Developer Cookbook for some code sample) to let the user interact with its objects (layers, feature or interface). QGIS also has a Python console.

The QGIS Python Console is an interactive shell for Python command executions. It also has a Python file editor that allows you to edit and save your Python scripts. Both console and editor are based on PyQScintilla2 package. To open the console go to *Plugins* ► *Python Console* (`Ctrl+Alt+P`) or click on the Python Console icon in the *Plugins* toolbar.

25.3.1 The Interactive Console

The console is a Python interpreter that allows you to execute Python commands. Modules from QGIS (analysis, core, gui, server, processing, 3d) and Qt (QtCore, QtGui, QtNetwork, QtWidgets, QtXml) as well as Python's math, os, re and sys modules are already imported and can be used directly.

The interactive console is composed of a toolbar, an input area and an output one.

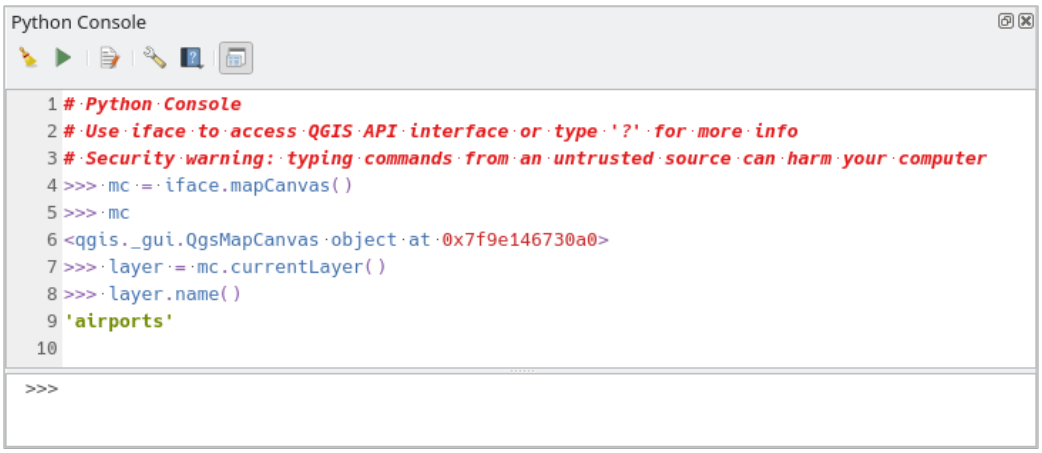


Fig. 25.17: The Python Console

工具栏

The toolbar proposes the following tools:

- Clear Console to wipe the output area;
- Run Command available in the input area: same as pressing `Enter`;
- Show Editor: toggles *The Code Editor* visibility;
- Options...: opens a dialog to configure *console properties*;
- Help... provides a menu to access various documentation:
  - *Python Console Help* (the current page)
  - PyQGIS API documentation
  - PyQGIS Cookbook
- Dock Code Editor to dock or undock the panel in QGIS interface

Input area

The Console input area main features are:

- Code completion, highlighting syntax and calltips for the following APIs:
  - Python
  - PyQGIS
  - PyQt5
  - QScintilla2
  - osgeo-gdal-ogr

  - osgeo-geos
- `Ctrl+Alt+Space` to view the auto-completion list if enabled in the *Python settings*;
- Execute code snippets from the input area by typing and pressing `Enter` or *Run Command*;
- Execute code snippets from the output area using the *Enter Selected* from the contextual menu or pressing `Ctrl+E`;
- Browse the command history from the input area using the `Up` and `Down` arrow keys and execute the command you want;
- `Ctrl+Shift+Space` to view the command history: double-clicking a row will execute the command. The *Command History* dialog can also be accessed from context menu of input area;
- Save and clear the command history. The history will be saved into the `console_history.txt` file under the active *user profile* folder;
- Type the following special commands:
  - `?` to show a help of the Python Console
  - `_api` to open QGIS C++ API documentation or `_api(object)` for a specific object documentation (in QGIS C++ API or Qt API documentation)
  - `_pyqgis` to open QGIS Python API documentation or `_pyqgis(object)` for a specific object documentation (in QGIS Python API or Qt API documentation)
  - `_cookbook` to open PyQGIS Cookbook.
  - `!` followed by a command to execute Shell commands from the Python Console. The console will start a subprocess, and forward its output to the Python Console Output. While the subprocess is running, the Python Console Input switches to STDIN mode and forwards entered character to the child process. This makes it possible to send confirmation when the child program asks for it. When the Console is in STDIN mode, pressing `Ctrl+C` kills the subprocess. It is also possible to affect the result of a command to a variable with the syntax `var = !cmd`

```
>>> !echo QGIS Rocks!
QGIS Rocks

>>> !gdalinfo --version
GDAL 3.10.3, released 2025/04/01

>>> !pip install black
# Install black python formatter using pip (if available)

>>> sql_formats = !ogrinfo --formats | grep SQL
>>> sql_formats
['SQLite -vector- (rw+v): SQLite / Spatialite', '  MSSQLSpatial -vector-␣
↪(rw+): Microsoft SQL Server Spatial Database', '  PostgreSQL -vector-␣
↪(rw+): PostgreSQL/PostGIS', '  MySQL -vector- (rw+): MySQL', '  PGDUMP -
↪vector- (w+v): PostgreSQL SQL dump']
```

Tip

Reuse executed commands from the output panel

You can execute code snippets from the output panel by selecting some text and pressing `Ctrl+E`. No matter if selected text contains the interpreter prompt (`>>>`, `...`).

25.3.2 The Code Editor

Use the Show Editor button in the Interactive Console to enable the editor widget. It allows editing and saving Python scripts and offers advanced functionalities to manage your code. Depending on the *enabled settings*, it provides various capabilities for easier code writing, such as code completion, highlighting syntax and calltips for supported APIs. Automatic indentation, parenthesis insertion, code commenting and syntax checking are also available.

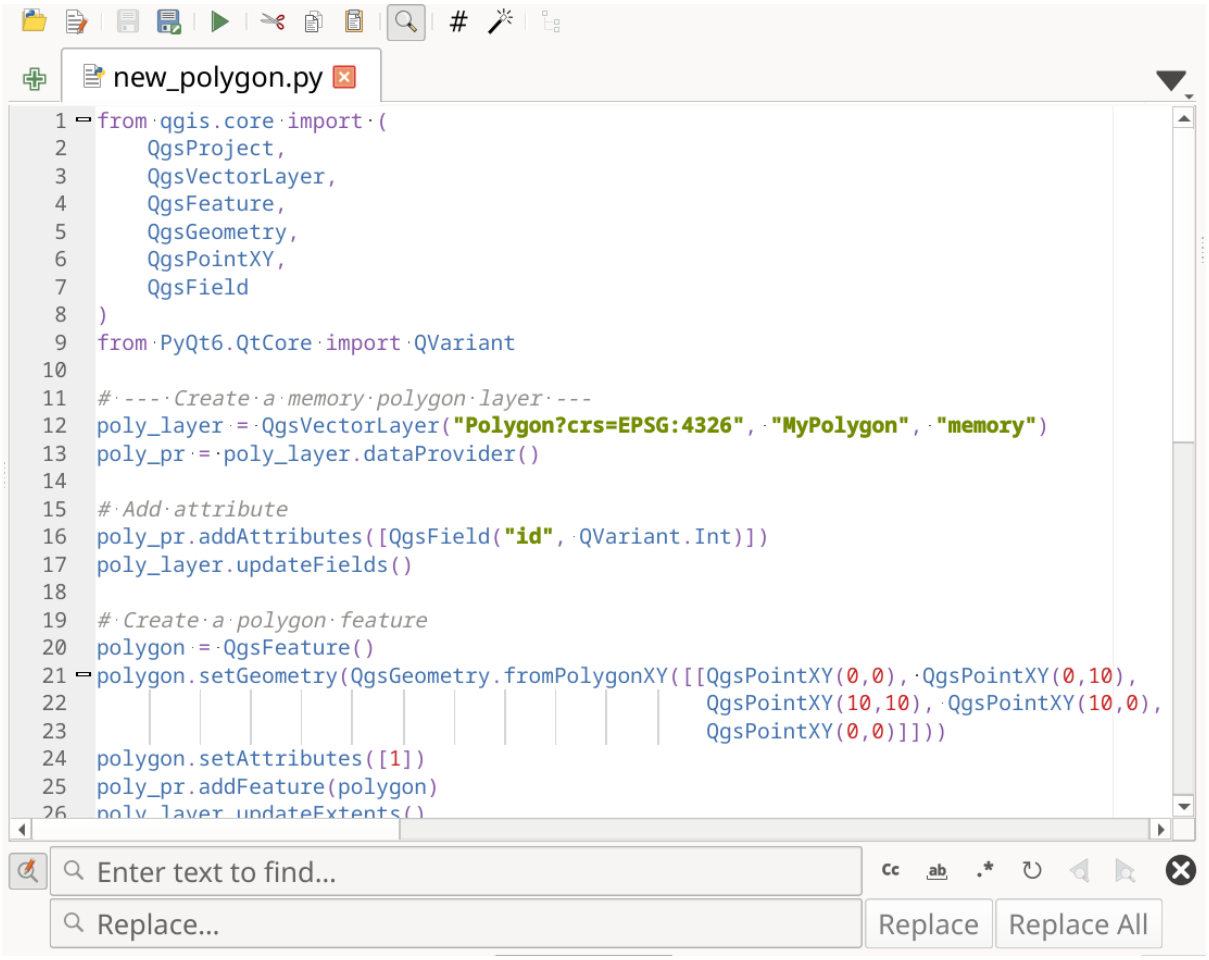


Fig. 25.18: The Python Console editor

The code editor area allows to simultaneously work on different scripts, each in a specific tab. Press New editor to add a new tab. You can run partially or totally a script from the *Code Editor* and output the result in the Interactive Console output area.

> **Tip**
>
> Press `Ctrl+Space` to view the auto-completion list.

At the top of the dialog, a toolbar provides access to a few commands. Right-clicking the editor area provides some more options. All available tools are described in the following table.

Table 25.1: Python Console Editor Tools

Tool name	Function	Location
Open script…	Loads a Python file in the code editor dialog, as a new tab	工具栏
Open in external editor…	Opens a saved Python script in the default external program set for Python file editing	
Save (`Ctrl+S`)	Saves the current script	
Save as… (`Ctrl+Shift+S`)	Saves the current script as a new file	
Run script (`Ctrl+Shift+E`)	Executes the whole script in the Interactive console (this creates a byte-compiled file with the extension `.pyc`)	Toolbar & Contextual menu
Context help (`F1`)	Attempts to display help on the selected string (class, method, object,...) in its corresponding API documentation	Contextual menu
Run selected (`Ctrl+E`)	Executes in the Interactive console the lines selected in the script	Toolbar & Contextual menu
Cut (`Ctrl+X`)	Cuts selected text to the clipboard	
Copy (`Ctrl+C`)	Copies selected text to the clipboard	
Paste (`Ctrl+V`)	Pastes a cut or copied text	
Find text	Allows to search and replace a text in the script. • Use the default Desktop Environment shortcut to find next/previous (`Ctrl+G` and `Shift+Ctrl+G`); • Automatically find first match when typing in find box; • Set initial find string to selection when opening find; • Enable the *Use Regular Expressions* option in the search bar to use regular expressions instead of plain text matching; • Replace the matching text with the text in the replace box. When using a regular expression, you can reference captured groups (e.g. via `\1`) in the replacement text. Use *Replace All* to replace all matches at once. • Pressing `Esc` closes the find bar.	
Toggle comment (`Ctrl+:`)	Comments out or uncomment selected lines, by adding or removing `#` character (potentially followed by a space) at the start of the lines. If the selection contains both comments and actual commands, adds `#` at the start of each non empty line. Pressing again the button reverts to the initial state.	
Reformat code (`Ctrl+Alt+F`)	Allows to manually apply various formatting rules (sort imports, indentation, line length,…) to the code, following *user-defined settings*. This may require installation of additional Python modules.	
Object inspector …	Shows and hides a dedicated browser with a tree structure for classes and functions available in the script. Click an item fo a quick access to its definition. The tool requires an activation from the *Python settings - Run and Debug*.	
Hide editor	Hides the Python code editor block. To make it visible again, press Show editor button from the interactive console toolbar.	Contextual menu

continues on next page

Table 25.1 – continued from previous page

Tool name	Function	Location
Check syntax (`Ctrl+4`)	Browses the code and reports syntax errors, such as missing parenthesis, colons, wrong indentation,⋯	
Undo (`Ctrl+Z`)	Undoes the latest action	
Redo (`Ctrl+Shift+Z`)	Reverts undone actions to a more recent	
Select all (`Ctrl+A`)	Selects the whole script	
Share on GitHub	Shares the script as a *Secret Gist* or *Public Gist* on GitHub, provided a *GitHub access token*.	
Options⋯	Opens the *Python settings* dialog.	

CHAPTER 26

附录

26.1 附录 C：QGIS 文件格式

26.1.1 QGS/QGZ - QGIS 工程文件格式

QGS 是一种用于保存 QGIS 工程的 XML 格式，QGZ 是一种包含 QGS 文件和 QGD 文件的压缩格式（zip）文件，QGD 是一个关联 sqllite 数据库的文件，里面包含了 QGIS 工程的辅助数据，如果一个工程没有辅助数据，QGD 文件将会为空。

QGIS 文件包含了任何存储 QGIS 工程需要的信息，包括

- 工程标题
- 工程参考坐标系
- 图层树
- 捕捉设置
- 关系
- 地图画布范围
- 工程模型
- legend
- 地图视图停靠（二维和三维）
- 与底层数据集（数据源）链接的图层和其他图层属性，包括范围、空间参考系统、连接、样式、渲染器、混合模式、不透明度等。
- 工程属性

下图展示了 QGIS 文件中的顶层标签和展开的 “ProjectLayers“标签。

```
−<qgis version="3.4.13-Madeira" projectname="">
    <homePath path=""/>
    <title/>
    <autotransaction active="0"/>
    <evaluateDefaultValues active="0"/>
    <trust active="0"/>
  +<projectCrs></projectCrs>
  +<layer-tree-group></layer-tree-group>
  +<snapping-settings tolerance="12" unit="1" enabled="0" type="1" mode="2" intersection-snapping="0">
    </snapping-settings>
    <relations/>
  −<mapcanvas name="theMapCanvas" annotationsVisible="1">
      <units>meters</units>
    +<extent></extent>
      <rotation>0</rotation>
    +<destinationsrs></destinationsrs>
      <rendermaptile>0</rendermaptile>
    </mapcanvas>
    <projectModels/>
  +<legend updateDrawingOrder="true"></legend>
    <mapViewDocks/>
    <mapViewDocks3D/>
  +<projectlayers></projectlayers>
  +<layerorder></layerorder>
  +<properties></properties>
    <visibility-presets/>
    <transformContext/>
  +<projectMetadata></projectMetadata>
    <Annotations/>
    <Layouts/>
  </qgis>
```

Fig. 26.1: QGIS 文件中的顶层标签

```
−<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: QGIS 文件中展开的顶层 “ProjectLayers“标签

26.1.2 QLR - QGIS 图层定义文件定义

图层定义文件（QLR）是一种包含图层数据源的指针和次图层的 QGIS 样式信息的 XML 文件

此文件的使用场景很简单：使用单个文件打开数据源并引入所有相关样式信息。QLR 文件还允许您在易于打开的文件中屏蔽底层数据源。

QLR 使用场景的一个示例是打开 MS SQL 图层。您无需进入 MS SQL 连接对话框，连接、选择、加载并最终设置样式。您只需要添加指向正确 MS SQL 图层的.qlr 文件并包含所有必要的样式即可。

未来，一个 qlr 文件将会支持引用多个图层

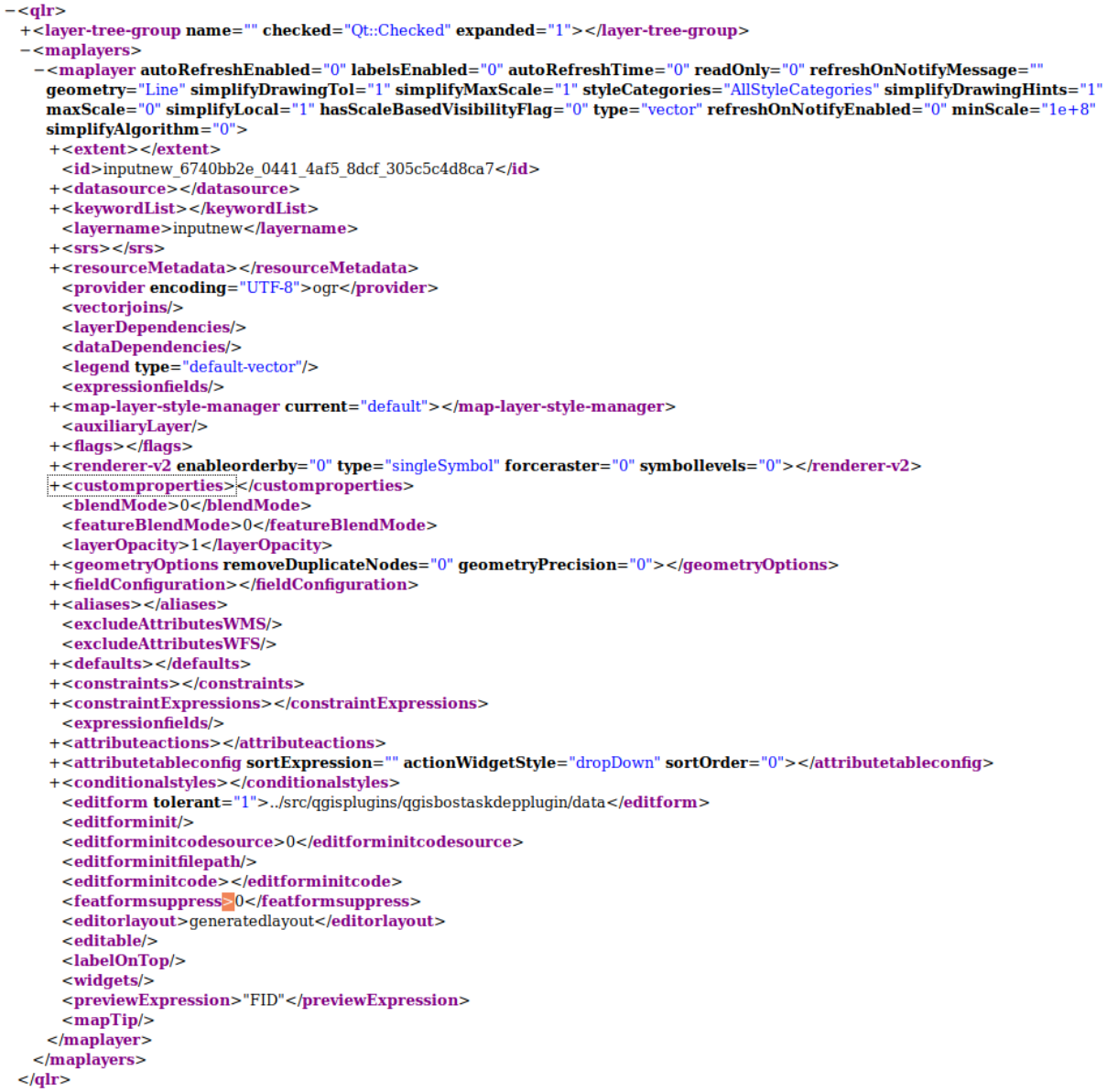


Fig. 26.3: QLR 文件的顶层标签

26.1.3 QML - QGIS 样式文件格式

QML 是一种用于存储图层样式的 XML 文件格式

QML 文件包含了 GIS 能够处理的所有信息 (用于渲染几何要素的信息)，包括符号定义、尺寸、旋转、标注、不透明度、混合模式等

下图展示了 QML 文件中的顶层标签（仅展开了 renderer_v2 标签和它的 symbol 标签）。

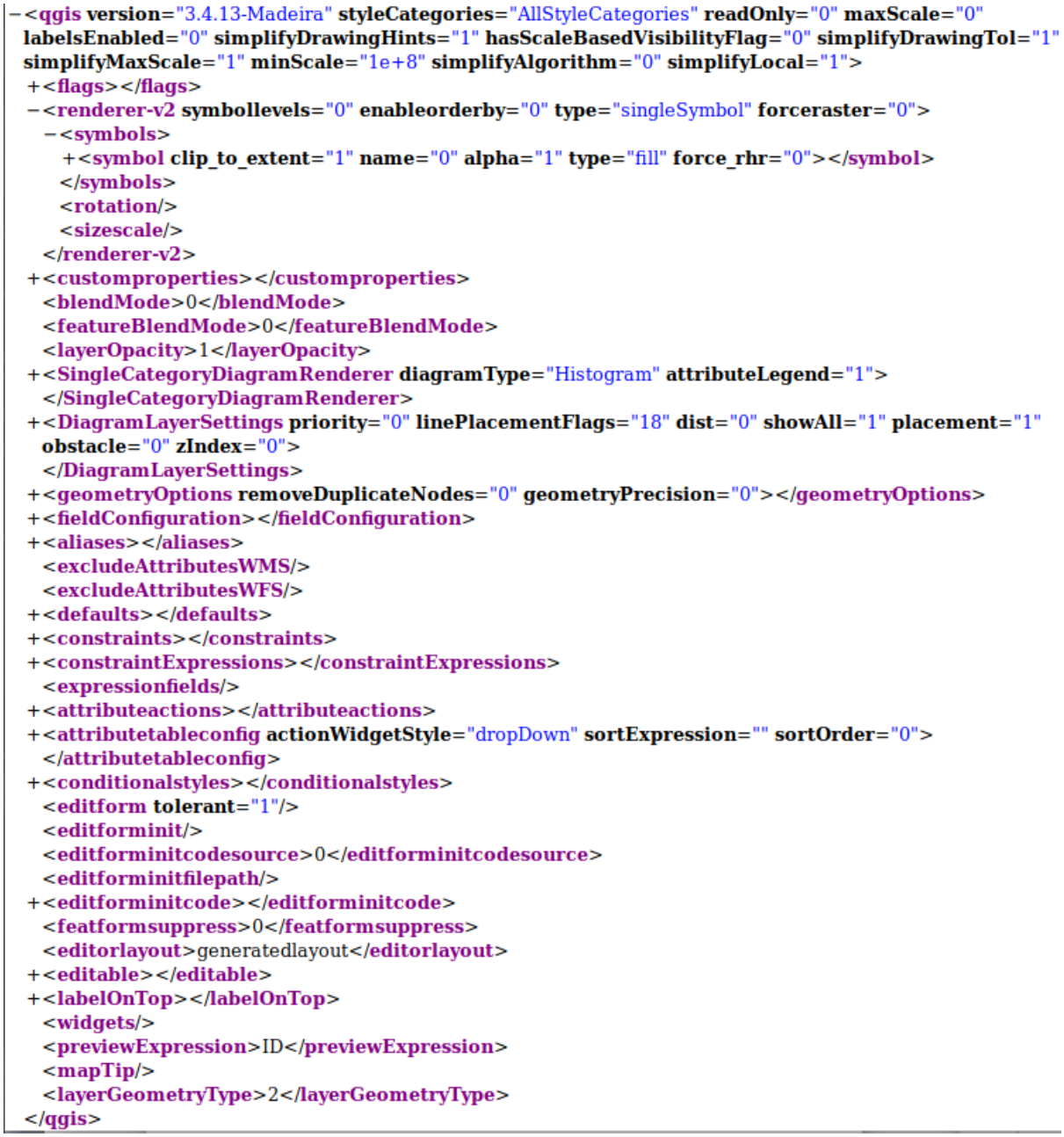


Fig. 26.4: QML 文件的顶层标签（仅展开了 renderer_v2 标签和它的 symbol 标签）

26.2 Appendix D: QGIS R script syntax

Contributed by Matteo Ghetta - funded by Scuola Superiore Sant'Anna

Writing R scripts in Processing is a bit tricky because of the special syntax.

A Processing R script starts with defining its **Inputs** and **Outputs**, each preceded with double hash characters (##).

Before the inputs, the group to place the algoritm in can be specified. If the group already exists, the algorithm will be added to it, if not, the group will be created. In the example below, the name of the group is *My group*:

```
##My Group=group
```

26.2.1 Inputs

All input data and parameters have to be specified. There are several types of inputs:

- vector: `##Layer = vector`
- vector field: `##F = Field Layer` (where *Layer* is the name of an input vector layer the field belongs to)
- raster: `##r = raster`
- table: `##t = table`
- number: `##Num = number`
- string: `##Str = string`
- boolean: `##Bol = boolean`
- elements in a dropdown menu. The items must be separated with semicolons `;`: `##type=selection point;lines;point+lines`

26.2.2 输出

As for the inputs, each output has to be defined at the beginning of the script:

- vector: `##output= output vector`
- raster: `##output= output raster`
- table: `##output= output table`
- plots: `##output_plots_to_html` (##showplots in earlier versions)
- To show R output in the *Result Viewer*, put > in front of the command whose output you would like to show.

26.2.3 Syntax Summary for QGIS R scripts

A number of input and output parameter types are offered.

Input parameter types

Parameter	Syntax example	Returning objects
vector	Layer = vector	sf object (or SpatialDataFrame object, if ##load_vector_using_rgdal is specified)
vector point	Layer = vector point	sf object (or SpatialDataFrame object, if ##load_vector_using_rgdal is specified)
vector line	Layer = vector line	sf object (or SpatialDataFrame object, if ##load_vector_using_rgdal is specified)
vector polygon	Layer = vector polygon	sf object (or SpatialPolygonsDataFrame object, if ##load_vector_using_rgdal is used)
multiple vector	Layer = multiple vector	sf object (or SpatialDataFrame objects if ##load_vector_using_rgdal is specified)
table	Layer = table	dataframe conversion from csv, default object of `read.csv` function
field	Field = Field Layer	name of the Field selected, e.g. `"Area"`
raster	Layer = raster	RasterBrick object, default object of `raster` package
multiple raster	Layer = multiple raster	RasterBrick objects, default object of `raster` package
number	N = number	integer or floating number chosen
string	S = string	string added in the box
longstring	LS = longstring	string added in the box, could be longer then the normal string
selection	S = selection first;second;third	string of the selected item chosen in the dropdown menu
crs	C = crs	string of the resulting CRS chosen, in the format: `"EPSG:4326"`
extent	E = extent	Extent object of the `raster` package, you can extract values as `E@xmin`
point	P = point	when clicked on the map, you have the coordinates of the point
file	F = file	path of the file chosen, e.g. ”/home/matteo/file.txt”
folder	F = folder	path of the folder chosen, e.g. ”/home/matteo/Downloads”

A parameter can be **OPTIONAL**, meaning that it can be ignored.

In order to set an input as optional, you add the string `optional` **before** the input, e.g:

```
##Layer = vector
##Field1 = Field Layer
##Field2 = optional Field Layer
```

Output parameter types

Parameter	Syntax example
vector	Output = output vector
raster	Output = output raster
table	Output = output table
file	Output = output file

Note

You can save plots as `png` from the *Processing Result Viewer*, or you can choose to save the plot directly from the algorithm interface.

脚本主体

The script body follows R syntax and the **Log** panel can help you if there is something wrong with your script.

Remember that you have to load all additional libraries in the script:

```
library(sp)
```

26.2.4 Examples

Example with vector output

Let's take an algorithm from the online collection that creates random points from the extent of an input layer:

```
##Point pattern analysis=group
##Layer=vector polygon
##Size=number 10
##Output=output vector
library(sp)
spatpoly = as(Layer, "Spatial")
pts=spsample(spatpoly,Size,type="random")
spdf=SpatialPointsDataFrame(pts, as.data.frame(pts))
Output=st_as_sf(spdf)
```

Explanation (per line in the script):

1. `Point pattern analysis` is the group of the algorithm
2. `Layer` is the input **vector** layer
3. `Size` is a **numerical** parameter with a default value of 10
4. `Output` is the **vector** layer that will be created by the algorithm
5. `library(sp)` loads the **sp** library
6. `spatpoly = as(Layer, "Spatial")` translate to an sp object
7. Call the `spsample` function of the `sp` library and run it using the input defined above (`Layer` and `Size`)
8. Create a *SpatialPointsDataFrame* object using the `SpatialPointsDataFrame` function
9. Create the output vector layer using the `st_as_sf` function

That's it! Just run the algorithm with a vector layer you have in the QGIS Legend, choose the number of random point. The resulting layer will be added to your map.

Example with raster output

The following script will perform basic ordinary kriging to create a raster map of interpolated values from a specified field of the input point vector layer by using the `autoKrige` function of the `automap` R package. It will first calculate the kriging model and then create a raster. The raster is created with the `raster` function of the raster R package:

```
##Basic statistics=group
##Layer=vector point
##Field=Field Layer
##Output=output raster
##load_vector_using_rgdal
require("automap")
require("sp")
require("raster")
```

(continues on next page)

(continued from previous page)

```
table=as.data.frame(Layer)
coordinates(table)= ~coords.x1+coords.x2
c = Layer[[Field]]
kriging_result = autoKrige(c~1, table)
prediction = raster(kriging_result$krige_output)
Output<-prediction
```

By using `##load_vector_using_rgdal`, the input vector layer will be made available as a `SpatialDataFrame` objects, so we avoid having to translate it from an `sf` object.

Example with table output

Let's edit the `Summary Statistics` algorithm so that the output is a table file (csv).

The script body is the following:

```
##Basic statistics=group
##Layer=vector
##Field=Field Layer
##Stat=Output table
Summary_statistics<-data.frame(rbind(
    sum(Layer[[Field]]),
    length(Layer[[Field]]),
    length(unique(Layer[[Field]])),
    min(Layer[[Field]]),
    max(Layer[[Field]]),
    max(Layer[[Field]])-min(Layer[[Field]]),
    mean(Layer[[Field]]),
    median(Layer[[Field]]),
    sd(Layer[[Field]]))),
  row.names=c("Sum:","Count:","Unique values:","Minimum value:","Maximum value:",
↪"Range:","Mean value:","Median value:","Standard deviation:"))
colnames(Summary_statistics)<-c(Field)
Stat<-Summary_statistics
```

The third line specifies the **Vector Field** in input and the fourth line tells the algorithm that the output should be a table.

The last line will take the `Stat` object created in the script and convert it into a `csv` table.

Example with console output

We can use the previous example and instead of creating a table, print the result in the **Result Viewer**:

```
##Basic statistics=group
##Layer=vector
##Field=Field Layer
Summary_statistics<-data.frame(rbind(
sum(Layer[[Field]]),
length(Layer[[Field]]),
length(unique(Layer[[Field]])),
min(Layer[[Field]]),
max(Layer[[Field]]),
max(Layer[[Field]])-min(Layer[[Field]]),
mean(Layer[[Field]]),
median(Layer[[Field]]),
sd(Layer[[Field]])),row.names=c("Sum:","Count:","Unique values:","Minimum value:",
↪"Maximum value:","Range:","Mean value:","Median value:","Standard deviation:"))
colnames(Summary_statistics)<-c(Field)
>Summary_statistics
```

The script is exactly the same as the one above except for two edits:

1. no output specified (the fourth line has been removed)
2. the last line begins with >, telling Processing to make the object available through the result viewer

Example with plot

To create plots, you have to use the `##output_plots_to_html` parameter as in the following script:

```
##Basic statistics=group
##Layer=vector
##Field=Field Layer
##output_plots_to_html
####output_plots_to_html
qqnorm(Layer[[Field]])
qqline(Layer[[Field]])
```

The script uses a field (`Field`) of a vector layer (`Layer`) as input, and creates a *QQ Plot* (to test the normality of the distribution).

The plot is automatically added to the Processing *Result Viewer*.

26.3 附录 E：QGIS 应用程序网络连接

这是 QGIS 进行的预配置/自动网络连接的列表。其中一些连接是用户发起的，因为它们在生效之前需要用户进行操作，而其他连接是自动发生的。

名称	目标	UX mode	Server	Information sent to server	Information stored on server
qgis.org					
Python API help	Browser PyQGIS documentation	User initiated	https://qgis.org/pyqgis/3.44/search	IP, QGIS version, OS	IP in server log
version.qgi					
New version check	Notify the user on new QGIS versions availability	Automatic	https://version.qgis.org	IP, QGIS version, OS	IP in server log
feed.qgi					
QGIS Feed	Retrieve QGIS news from the feed	Automatic	https://feed.qgis.org	IP, QGIS version, language code, last download timestamp, OS	IP in server log; QGIS version, OS and IP are aggregated and used to collect some statistics
plugins.qgis.					
Check for plugin updates	Notify the user about plugin updates	User initiated/automa (configurable)	https://plugins.qgis.org	IP, QGIS version, OS	IP in server log
Plugins list	Retrieve the list of plugins	User initiated	https://plugins.qgis.org	IP, QGIS version, OS	IP in server log
Plugin installation	Download and install a plugin package	User initiated	https://plugins.qgis.org	IP, QGIS version, OS	Increase plugin download counter by one
Styles	List user contributed styles	User initiated	https://hub.qgis.org/styles	IP, QGIS version, OS	Increase download counter by one
3rd party					
Terrain data	Produce a DEM for 3D views	User initiated	https://s3.amazonaws.com/elevation-tiles-prod/terrarium/{z}/{x}/{y}	IP, QGIS version, OS	see Amazon TOS
Google Map Geocode	Geocoding services	User initiated	https://maps.googleapis.com/maps/api/geocode/json	IP, QGIS version, OS	See google maps API TOS
Nominatim Geocode	Geocoding services	User initiated	https://nominatim.qgis.org/ui/search.html	IP, QGIS version, OS	
Geodetic grid	Add a new PROJ grid	User initiated	https://cdn.proj.org	IP, PROJ version	Access logs are permanently deleted after one day

CHAPTER 27

文献和网络参考

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.