Boundless Desktop Documentation

Release 1.1.0

Boundless

Sep 20, 2017

Contents

What is Boundless Desktop?

Boundless Desktop is a fully supported software package prepared by Boundless and available for Windows and Mac OS X platforms. It's composed of open source geospatial client-side tools and is capable of addressing the most demanding desktop GIS users needs, including spatial analysis, data management, cartography and web publishing.

Boundless Desktop 1.1 is composed of the following set of open source tools:

- QGIS 2.18 by http://www.qgis.org

- Connect plugin for QGIS by Boundless

- PgAdmin 4 1.5 by http://www.pgadmin.org

- Qt Designer 4.8 for QGIS by http://www.qt.io

- GDAL/OGR 2.2 by http://www.gdal.org

- Master Password Helper plugin by Boundless

- OAuth2 authentication plugin by Boundless

- Reporting tool plugin by Boundless

(See *Boundless Desktop components* section for more details on each component)

Boundless Desktop is also enhanced by a set of Boundless-supported plugins for QGIS, those are aimed at improving QGIS functionality, but also a better integration with the other two Boundless Products: Boundless Suite and Boundless Exchange. (See more in *Boundless plugins for QGIS*)

The central element of our QGIS installation is the Boundless connect plugin, which acts as a single entry point to Boundless technology and content for QGIS. This provides access to Boundless Connect content, which currently includes Boundless-supported plugins, basemaps, and knowledge-based content, like documentation, tutorials and lessons for lessons plugins.

Boundless Desktop is not composed solely of zeros and ones! It also comes with the deep technical knowledge and immediate readiness of Boundless experts. From product maintenance and day-to-day online support, whatever are your needs, there is a Boundless support plan suited for you (see more about it in the Boundless Connect page).
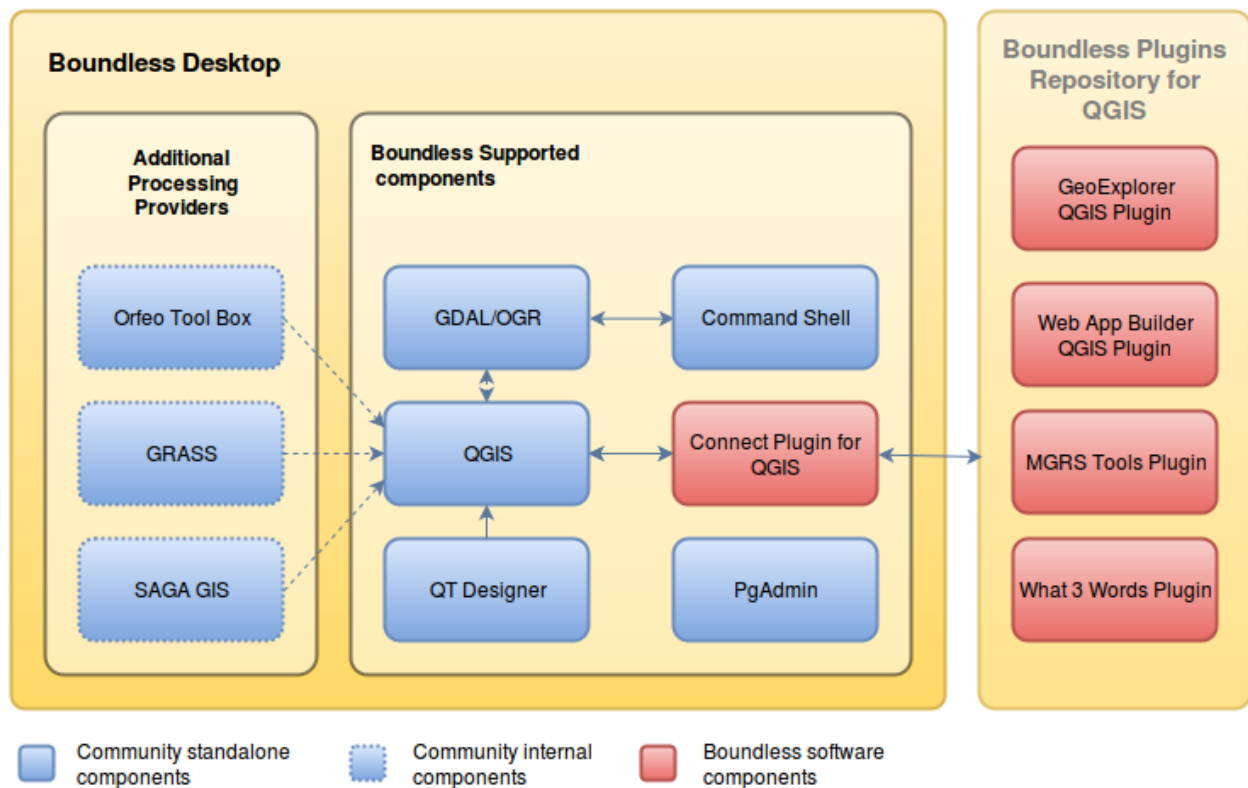
Boundless Desktop

Boundless Plugins Repository for QGIS

Additional Processing Providers

Boundless Supported components

Orfeo Tool Box

GDAL/OGR

Command Shell

GeoExplorer QGIS Plugin

GRASS

QGIS

Connect Plugin for QGIS

Web App Builder QGIS Plugin

SAGA GIS

QT Designer

PgAdmin

MGRS Tools Plugin

What 3 Words Plugin

Community standalone components

Community internal components

Boundless software components

Fig. 1.1: Boundless Desktop ecosystem

What's new in 1.1.0

This version contains numerous component upgrades and bug fixes. Highlights include:

- GDAL/OGR updated to 2.2.0 (see GDAL 2.2 changelog).

- pgAdmin III replaced by pgAdmin 4 (code-signed).

- QGIS updated to 2.18.10 which includes lots of new features and improvements (see QGIS 2.18 visual changelog), including:

    - Native support for XYZ tile layers;

    - Native support for reading Map and Feature Services published by ArcGIS Server in Esri REST format;

    - New styling panel.

- New customization support (see *For System Administrators*):

    - QgsSettings support allowing to define global defaults and settings in a `qgis_global_setting.ini` file (future QGIS 3.0 feature, currently available only on QGIS 2.18 for Boundless Desktop);

    - `init_scripts` support for running customization scripts on QGIS launch.

- Several improvements in *Boundless Connect Plugin*, including:

    - Search for basemaps from partners providers;

    - Use Mapbox Streets basemap in QGIS default project;

    - Search and install lessons for lessons plugin.

- New *Master Password Helper* plugin (C++ core plugin).

- New *Support Tool* plugin (core plugin) and `createreport` command line script (see *How to ask for support* page for instructions).

- New *Terrain Analysis* plugin*

- New *Image Discovery* plugin*

- Several improvements to *Web App Builder* plugin*, including:

    - Updates WebSDK;

- New WebSDK compiler service;

- Enhanced support to advanced layer styling.

- Several enhancements to PKI authentication method support.

- New support for OAuth2 authentication method. (See *OAuth2 Plugin*)

- Improved symbology integration with other products from Boundless ecosystem using Mapbox GL Style library.

- New 'Boundless Documentation' URL link in Start menu and shortcuts on user's desktop

- "OpenGeo QGIS" logo replaced by the official QGIS 2.x logo

* - Available in Boundless QGIS Plugin repository (see *Boundless plugins for QGIS*).

System requirements

Boundless Desktop should run well on most recent hardware systems. Nevertheless, the performance may vary depending on the data type, data size, and the type of operation/analysis performed. The following system requirements are the minimum and recommended settings for reliable average use.

Note: Currently, Boundless does not provide Boundless Desktop installers for Linux. Read more about it in the *Installing Boundless Desktop* section.

Windows system requirements

Minimum:

- OS: 64-bit Windows 7 (SP1) or Server 2008 R2
- CPU: Hyperthreaded dual core
- RAM for Desktop: 1 GB
- Screen resolution: 1280x800 at standard dpi
- Screen color depth: 32-bit
- Video RAM for Desktop: 256 MB or higher
- Graphics card: DirectX 9 (OpenGL 2.0)
- Disk space for Desktop (and data): 3 (10) GB

Recommended:

- OS: 64-bit Windows 10 or higher
- CPU: Quad core or better
- RAM for Desktop: 2 GB or higher

- Screen resolution: 1440x900 or higher at standard dpi
- Screen color depth: 32-bit
- Video RAM for Desktop: 512 MB or higher
- Graphics card: DirectX 11 (OpenGL 3.2) or better
- Disk space for Desktop (and data): 3 (40) GB or higher

Mac system requirements

Minimum:

- Hardware: Apple
- OS: OS X 10.9 Mavericks
- CPU: Intel Core 2 Duo
- RAM for Desktop: 1 GB
- Screen resolution: 1280x800 at standard dpi
- Screen color depth: 32-bit
- Video RAM for Desktop: 256 MB or higher
- Disk space for Desktop (and data): 3 (10) GB

Recommended:

- OS: OS X 10.12 Sierra
- CPU: Intel Core 2 Quad, i5 or better
- RAM for Desktop: 2 GB or higher
- Screen resolution: 1440x900 or higher at standard dpi
- Screen color depth: 32-bit
- Video RAM for Desktop: 512 MB or higher
- Disk space for Desktop (and data): 3 (40) GB or higher

Installing Boundless Desktop

Boundless provides packages for Boundless Desktop on both Windows and OS X.

Installing Boundless Desktop on Windows

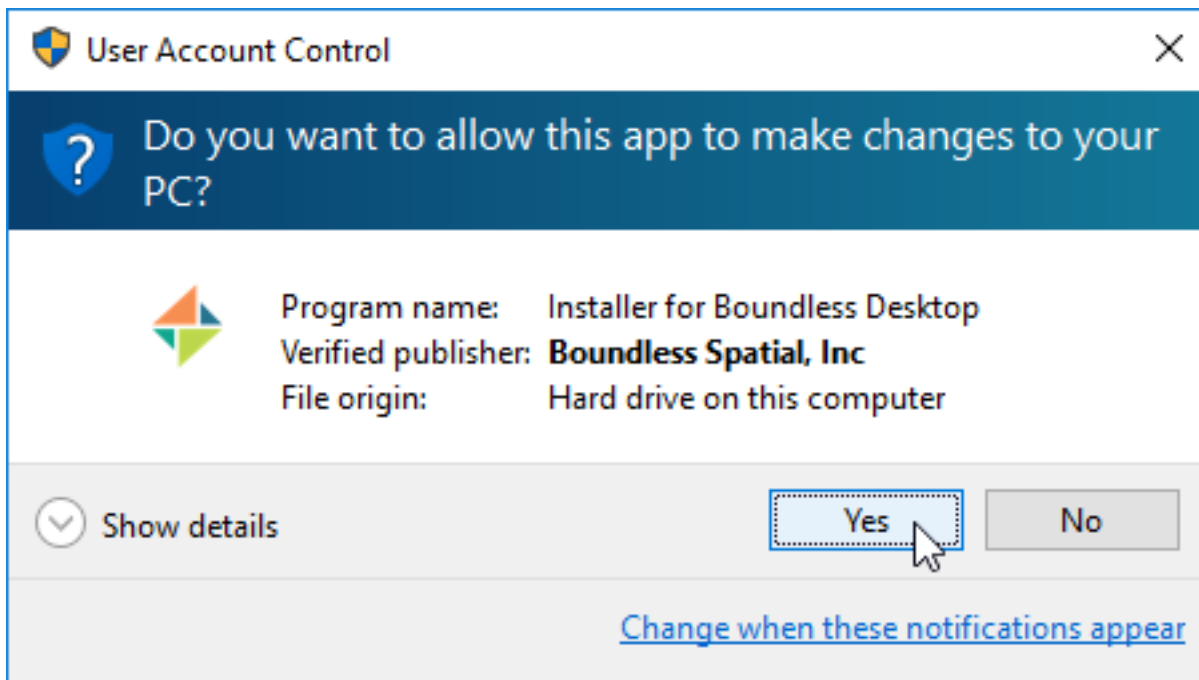This section will describe how to install Boundless Desktop on Windows.

Prerequisites

Boundless Desktop can run on any recent hardware/software combination. See *Windows system requirements* for more details.

Install

To install Boundless Desktop on windows:

1. Double-click the downloaded installer file (Administrator permissions needed). If Windows' User Account Control asks you permission to allow the app to make changes, click *YES*.

2. In the *Welcome* screen, click *Next*.

3. Read the *License Agreement*. Then, click *I Agree*.

4. Read the *Read Me*. Then, click *Next*.

5. Select the Destination folder where you would like to install Boundless Desktop, and click *Next*.

6. Select the name and location of the Start Menu folder to be created, and click *Next*.

7. Check the components you would like to install. You can hover the mouse over a component's name to see a full description of it. Click *Next*.

8. You are ready to install the software. Click *Install* to start the installation.

9. Please wait while the installation proceeds.

10. After installation, click *Finish*.

After a successful install, Boundless Desktop components, such as QGIS, can be run from shortcuts in the Start Menu (see *Quick start guide* for more information).

Installing Boundless Desktop on Mac OS X

This section will describe how to install Boundless Desktop on Mac OS X.

Prerequisites

Boundless Desktop can run on any recent hardware/software combination. See *Mac system requirements* for more details.

Note: On Mac OS, installing Boundless Desktop 1.1.0 will remove any previously installed version. To avoid the removal, before installing Boundless Desktop 1.1.0, you can run the following command in a terminal (you will need administrator permissions).

```
for pkg in $(pkgutil --pkgs='com\.boundlessgeo\.pkg*'); do sudo pkgutil --volume '/' -
↪-forget "${pkg}"; done
```

If you have accidentally removed a 1.0.x version and want to re-install it without being blocked by a warning that states "a newer version is already installed," you can also use the same command before installation.
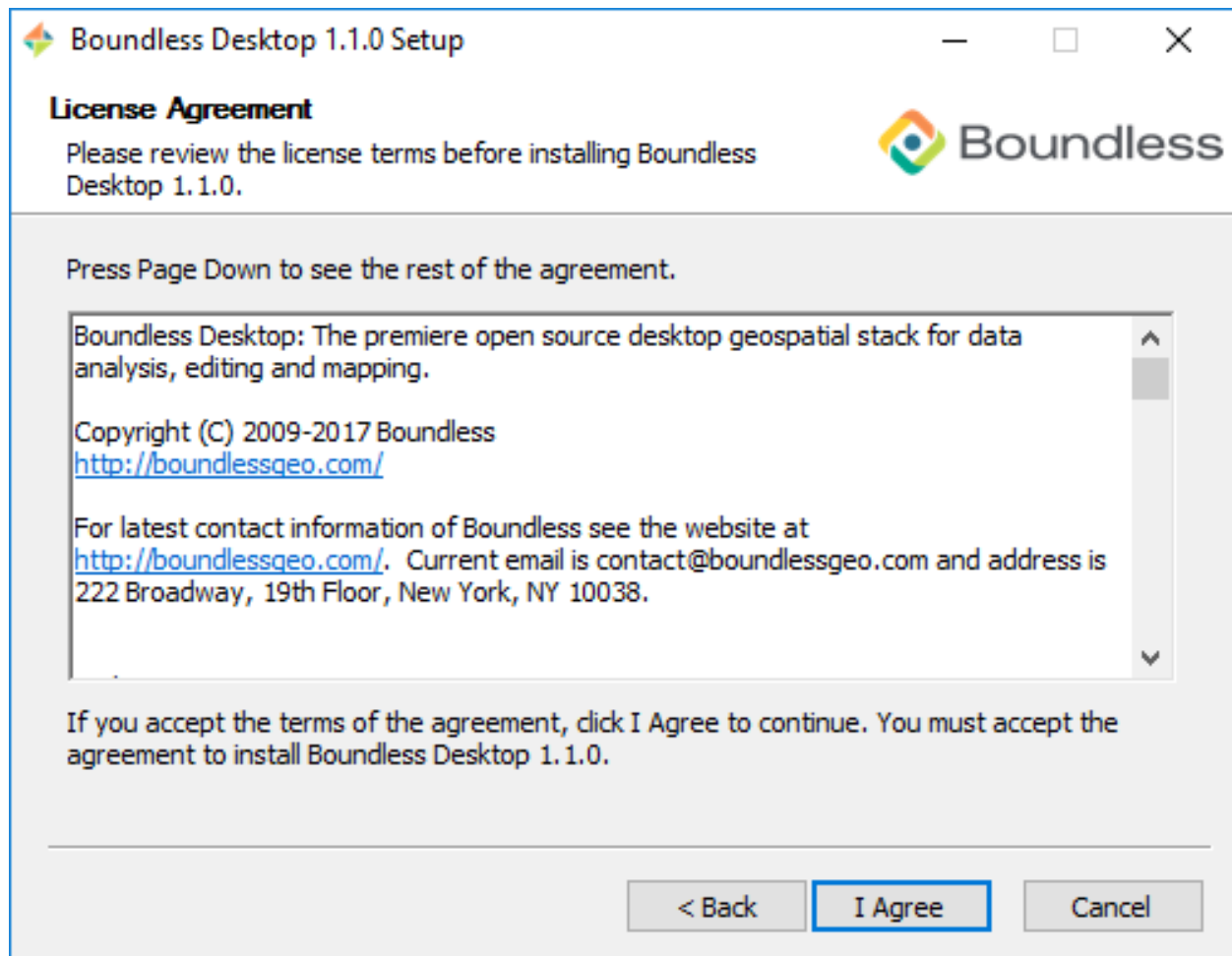
Fig. 4.1: Welcome screen
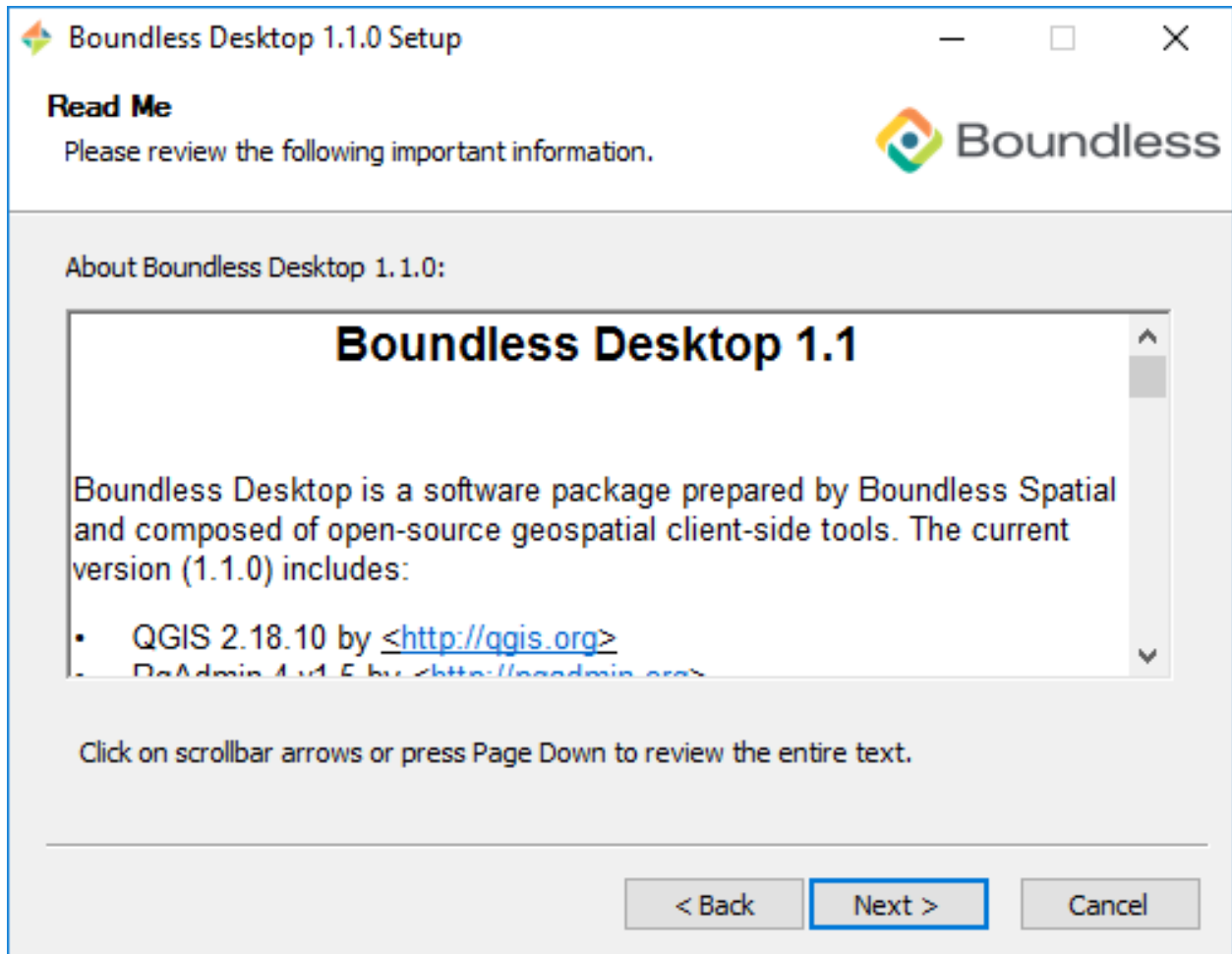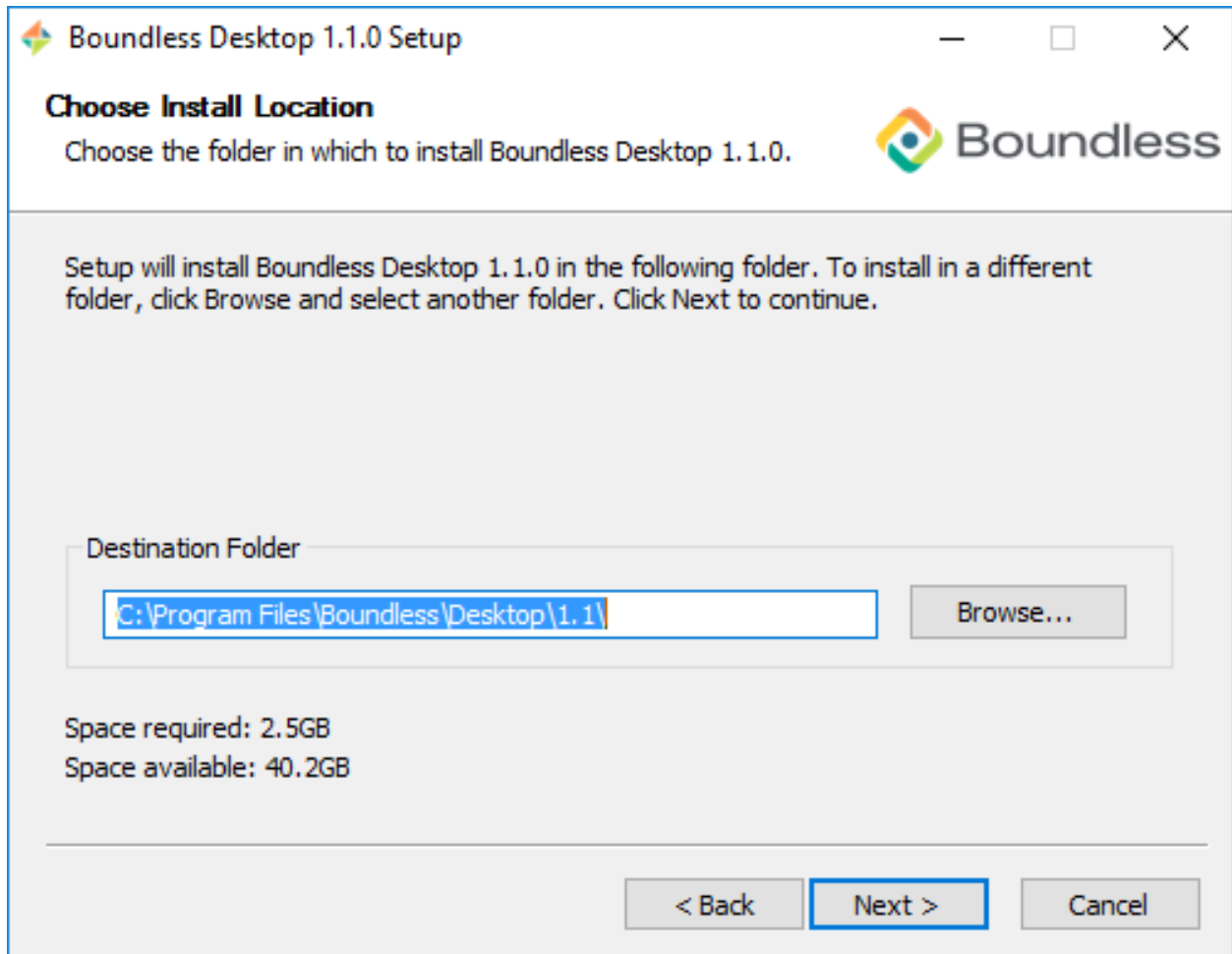
Fig. 4.2: License agreement

Fig. 4.3: Read me file

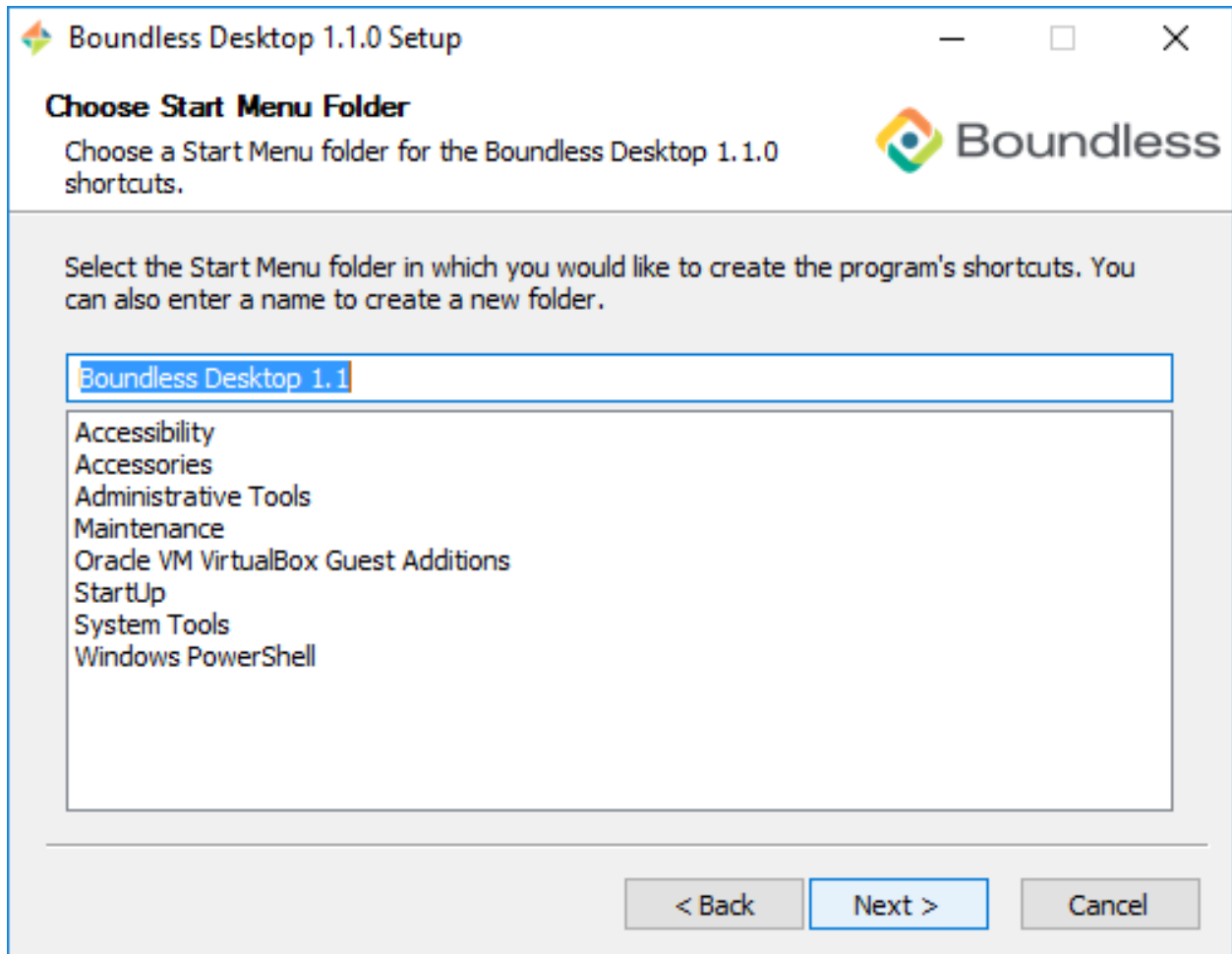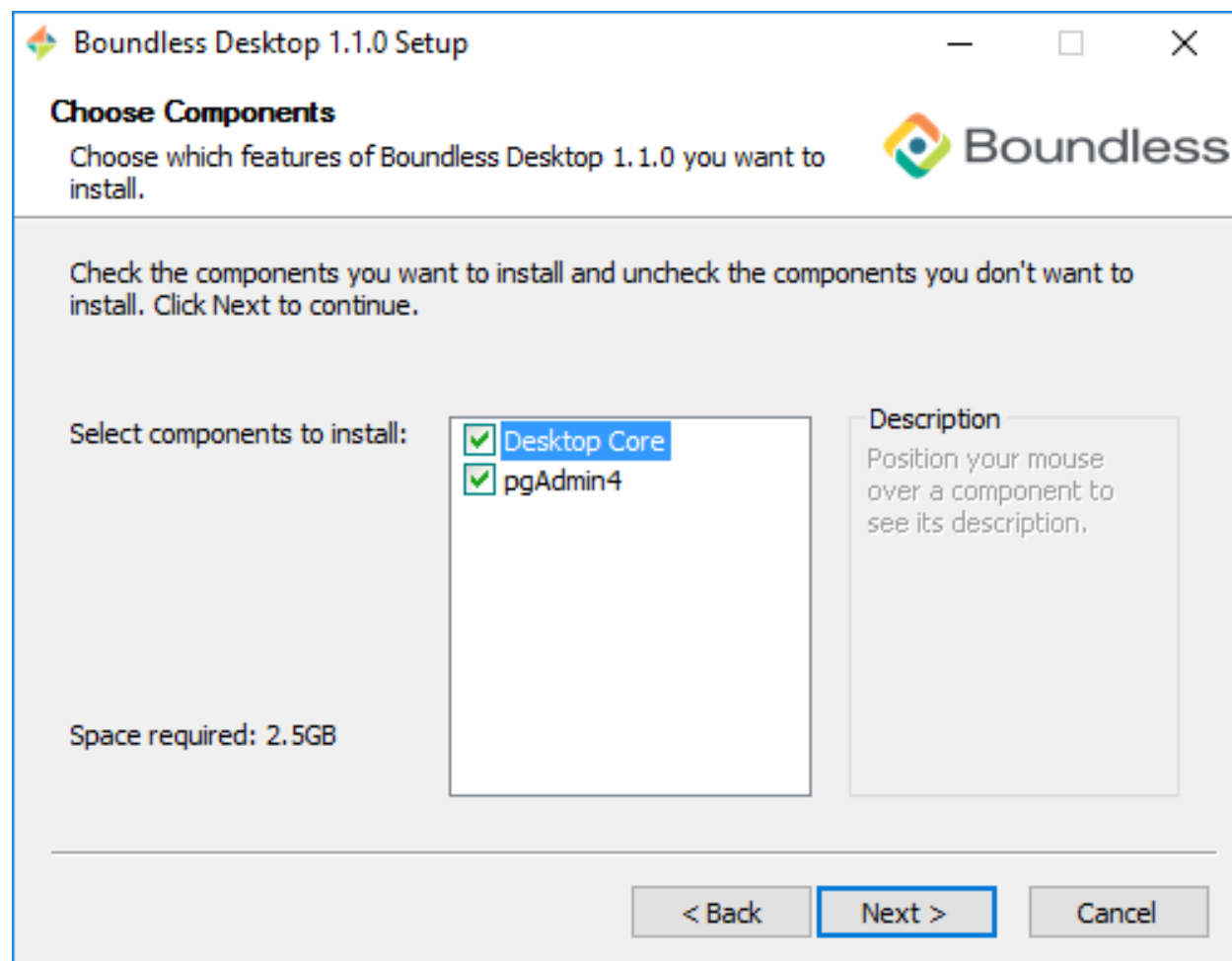Fig. 4.4: Destination folder for the installation
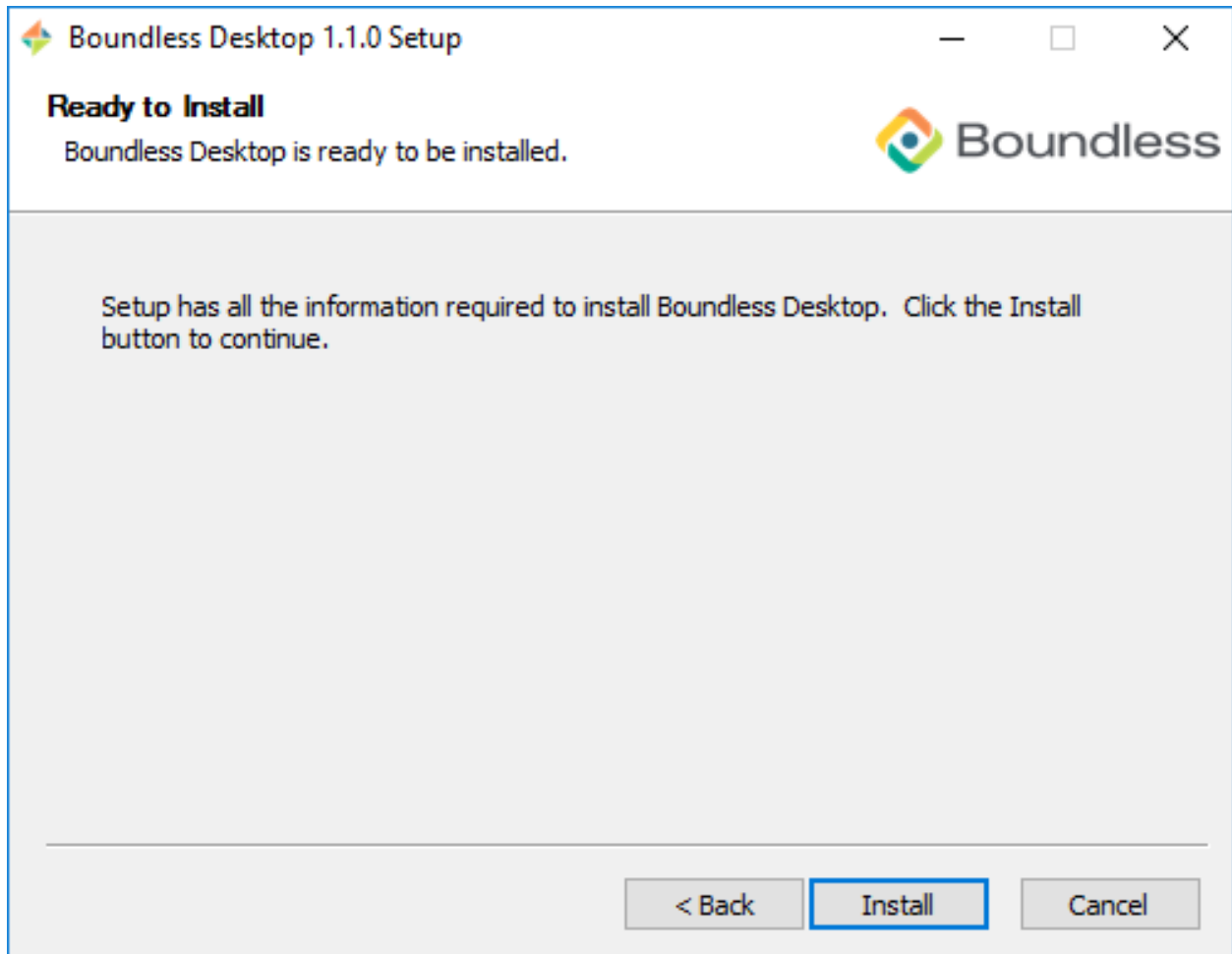
Fig. 4.5: Start Menu folder to be created
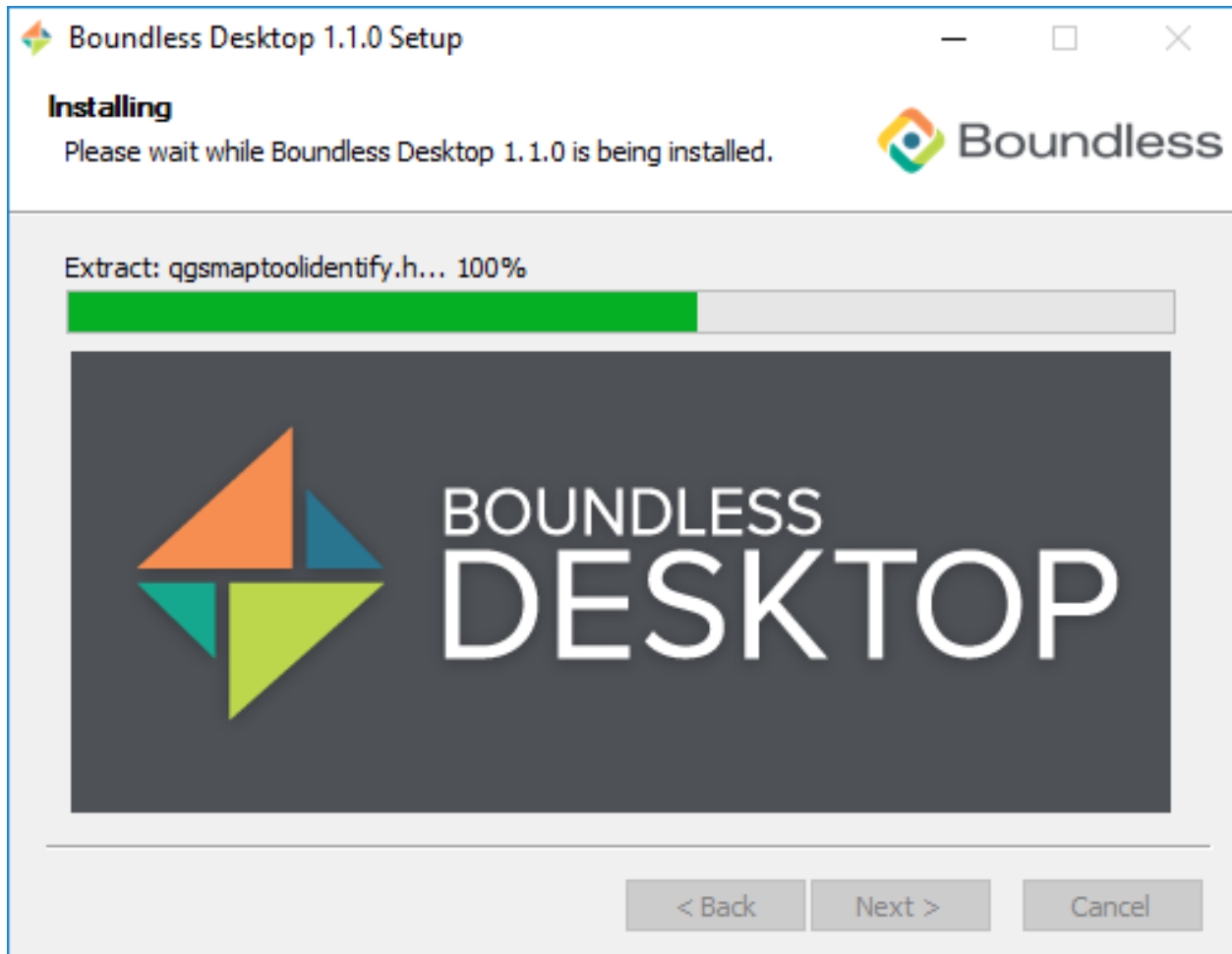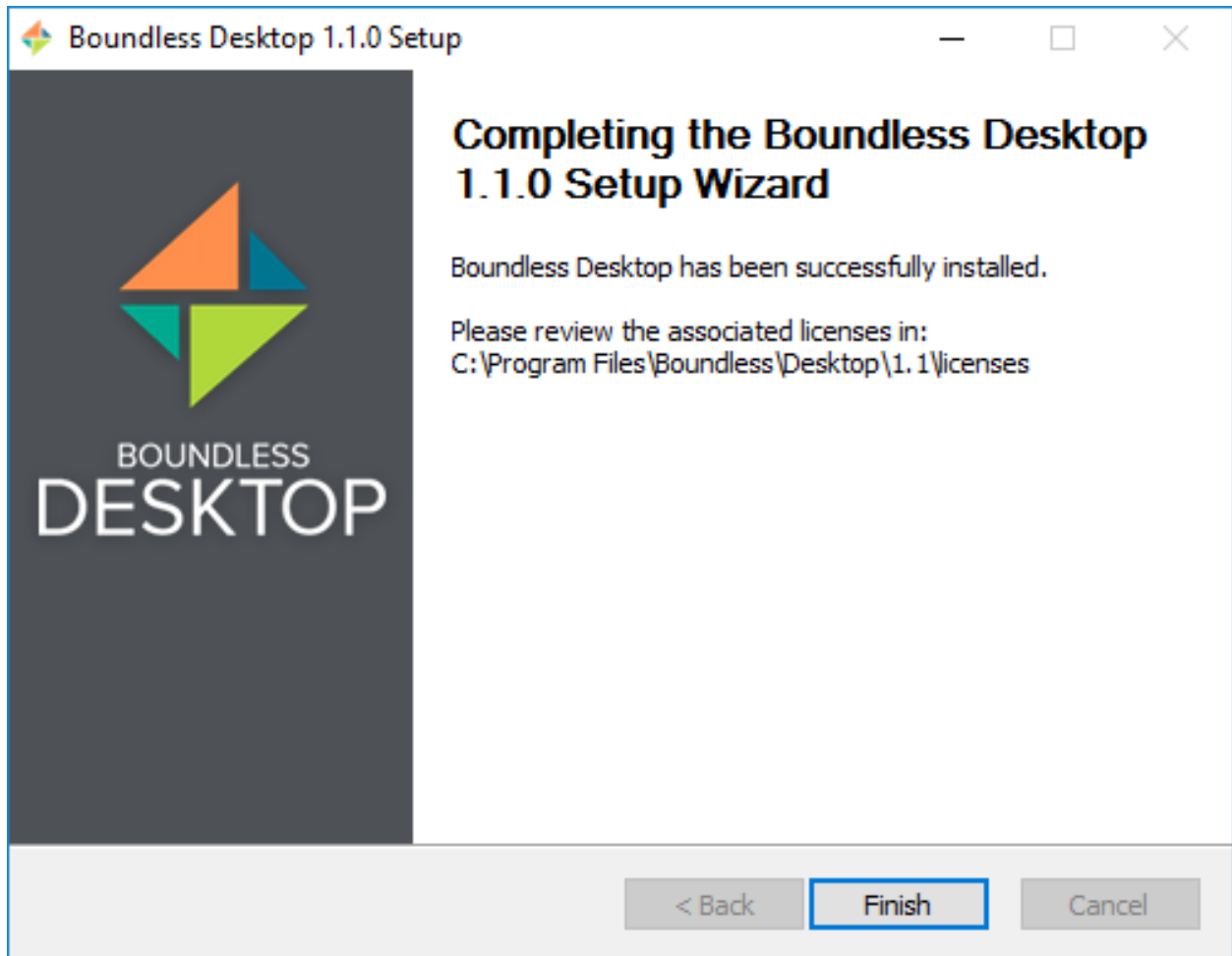
Fig. 4.6: Ready to install

Fig. 4.7: Installation

Fig. 4.8: Boundless Desktop successfully installed

Install

To install Boundless Desktop:

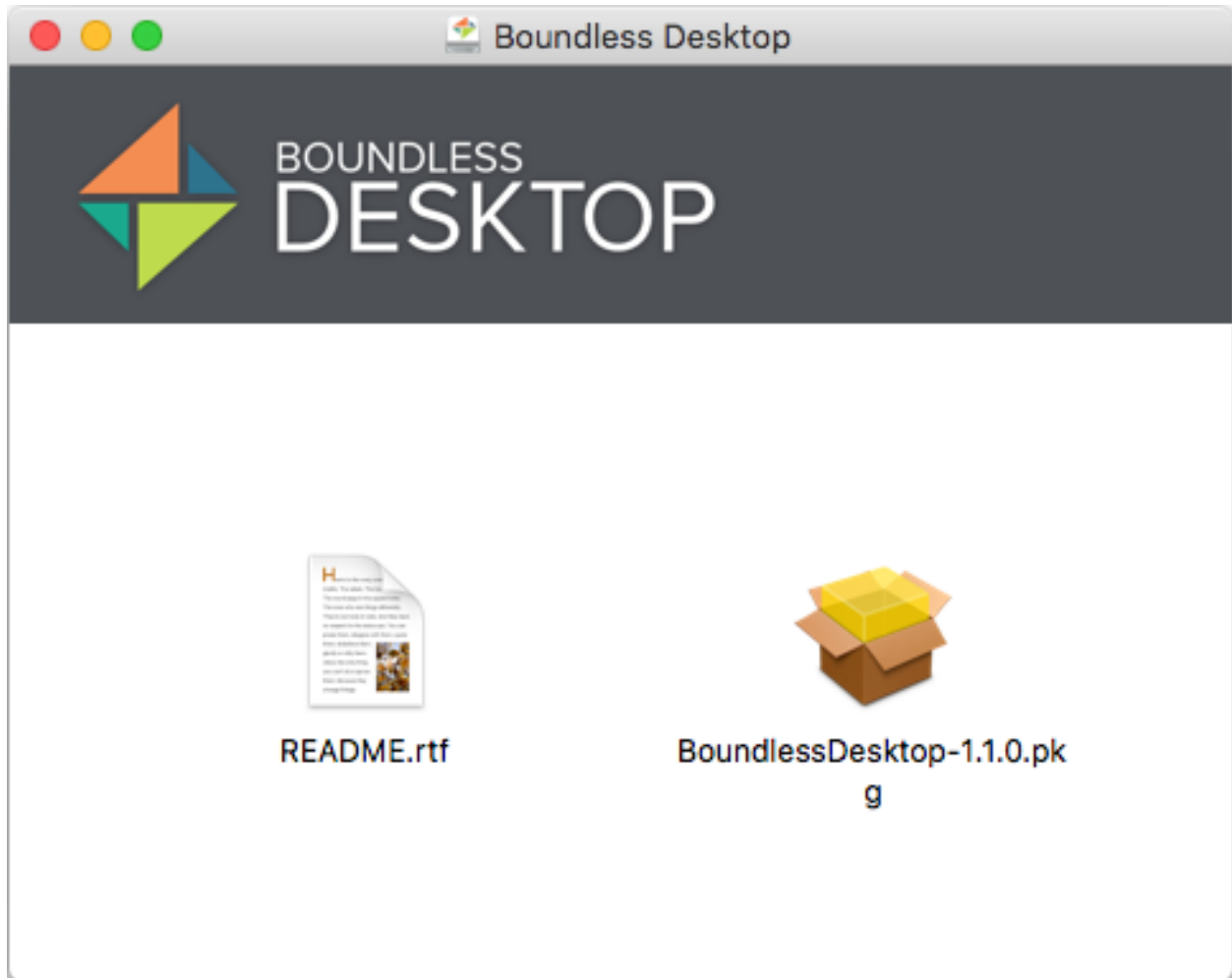1. Open the provided DMG archive. There will be a single `BoundlessDesktop-1.1.0.pkg`

Fig. 4.9: Boundless Desktop package

2. Double-click the installer package to launch the installer.

3. At the *Welcome* screen, click *Continue*.

4. Read the *Read Me* and then click *Continue*.

5. Read the *License Agreement* and click *Continue*.

6. When prompted about the license agreement, click *Agree* to continue the installation.

7. In the *Installation type* dialog check the components you would like to install and click *Continue*.

8. You are ready to install the software. Click *Install* to start the installation.

9. When prompted, enter your username and password. Click *Install Software* to continue.

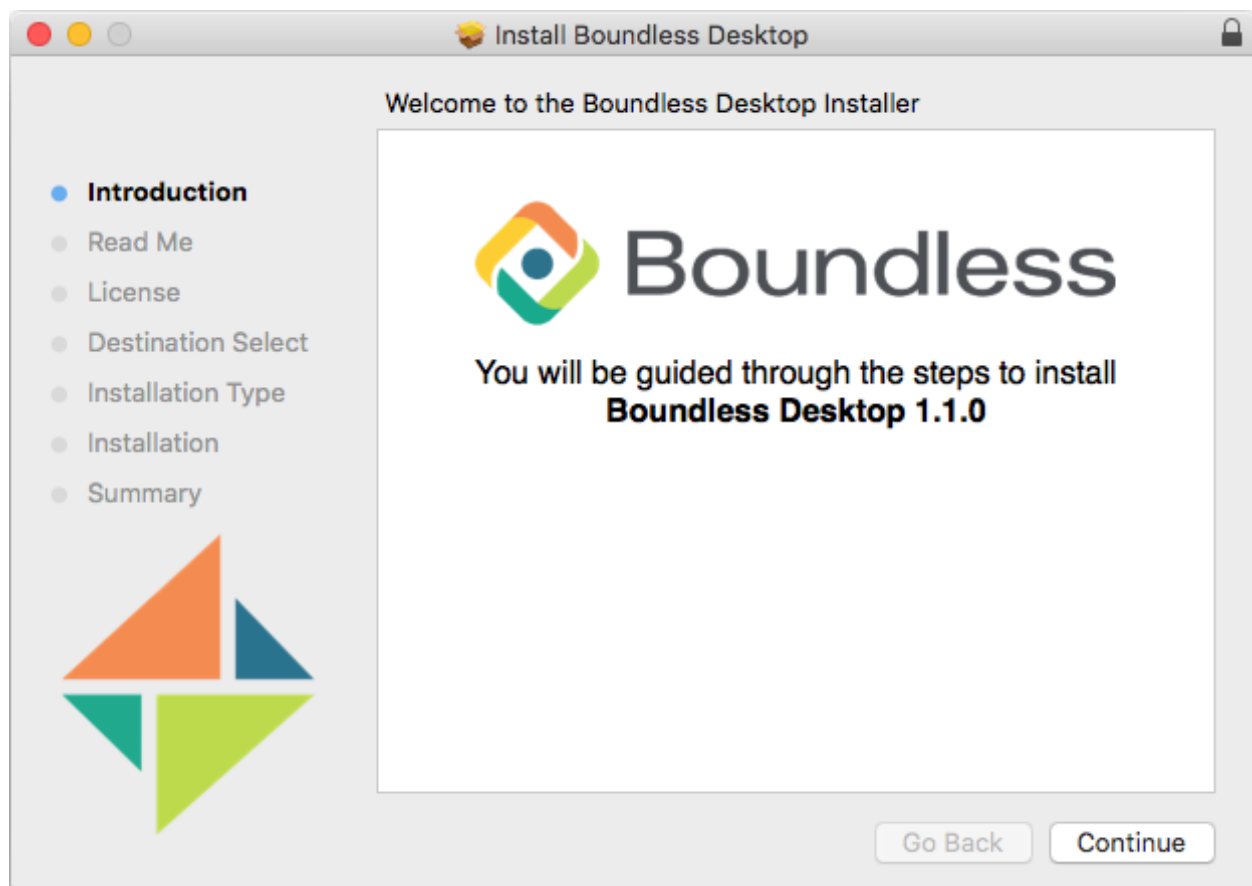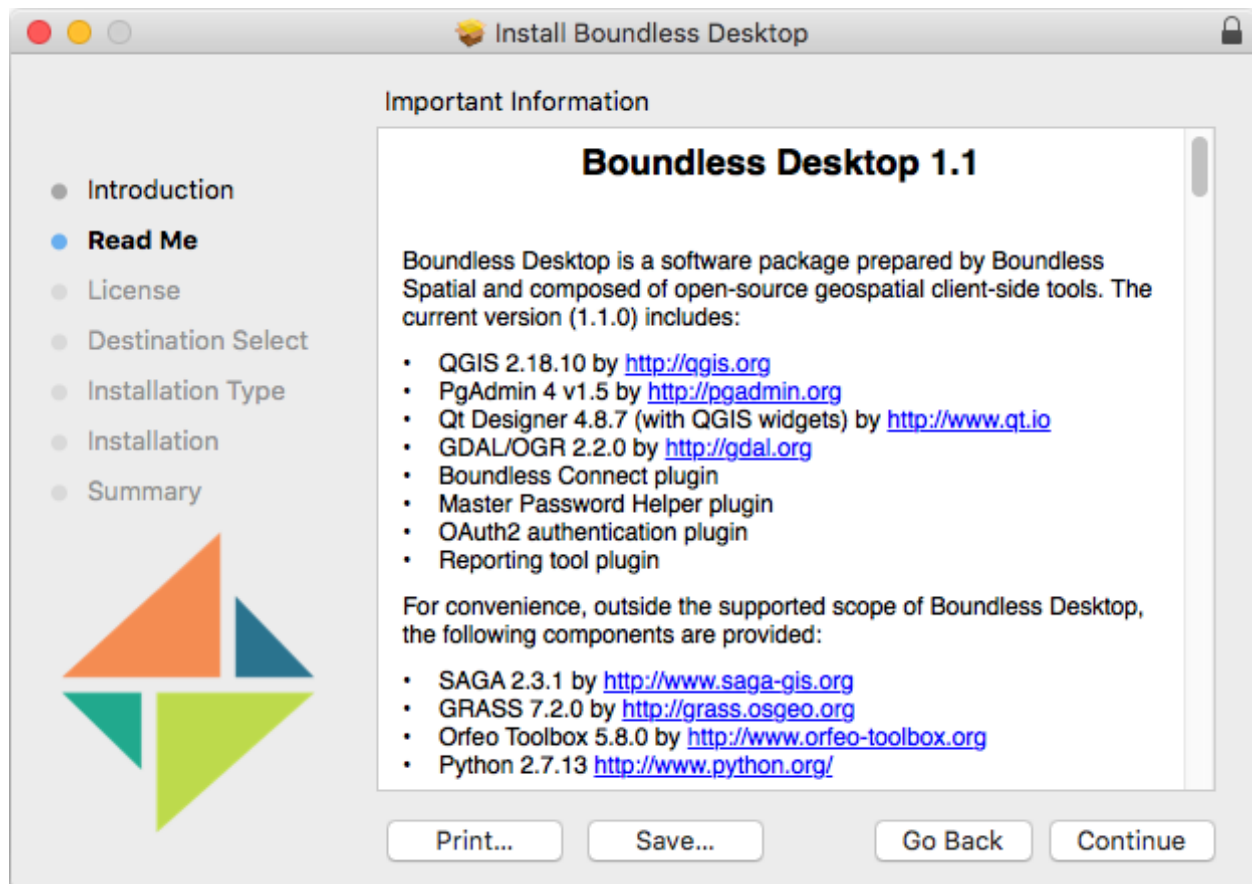10. Please wait while the installation proceeds.

Fig. 4.10: Welcome screen
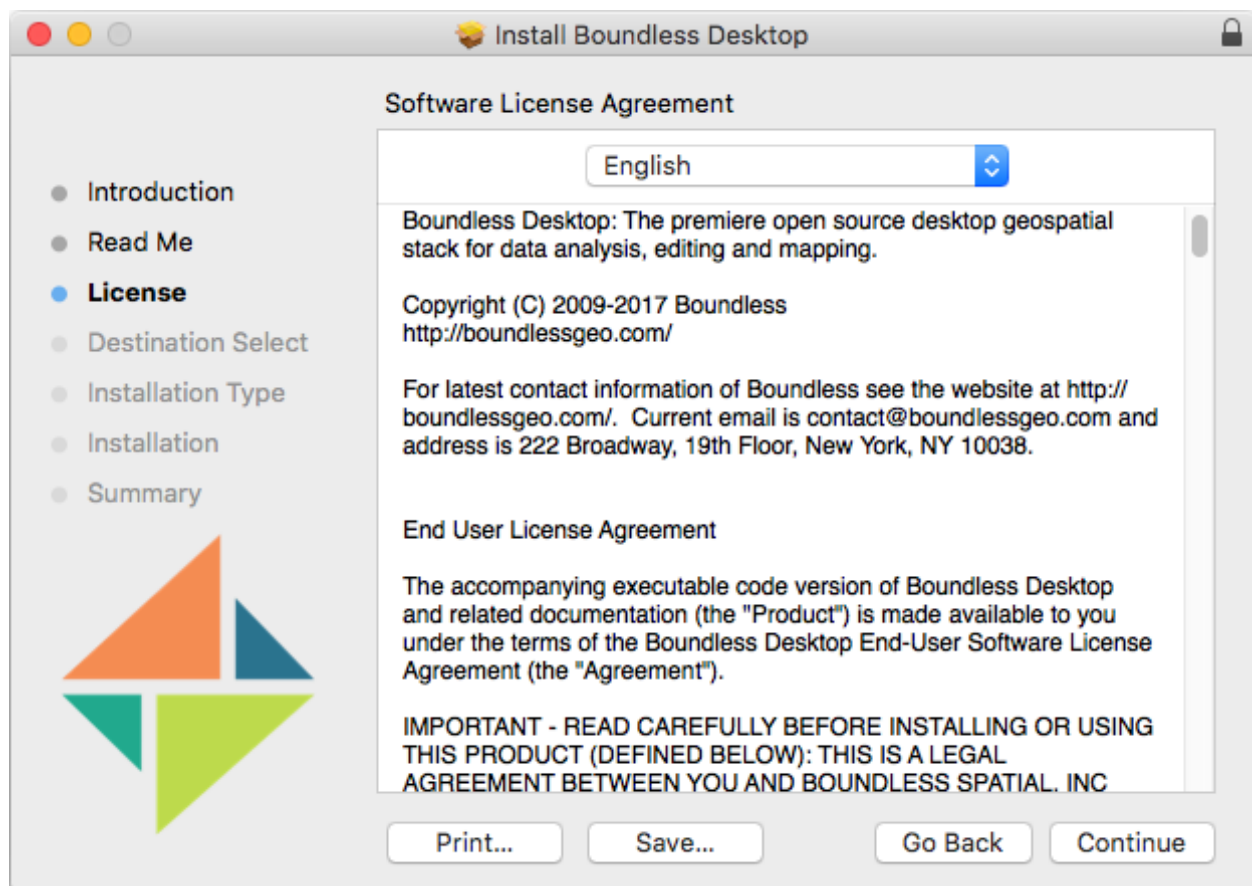
Fig. 4.11: Read me file
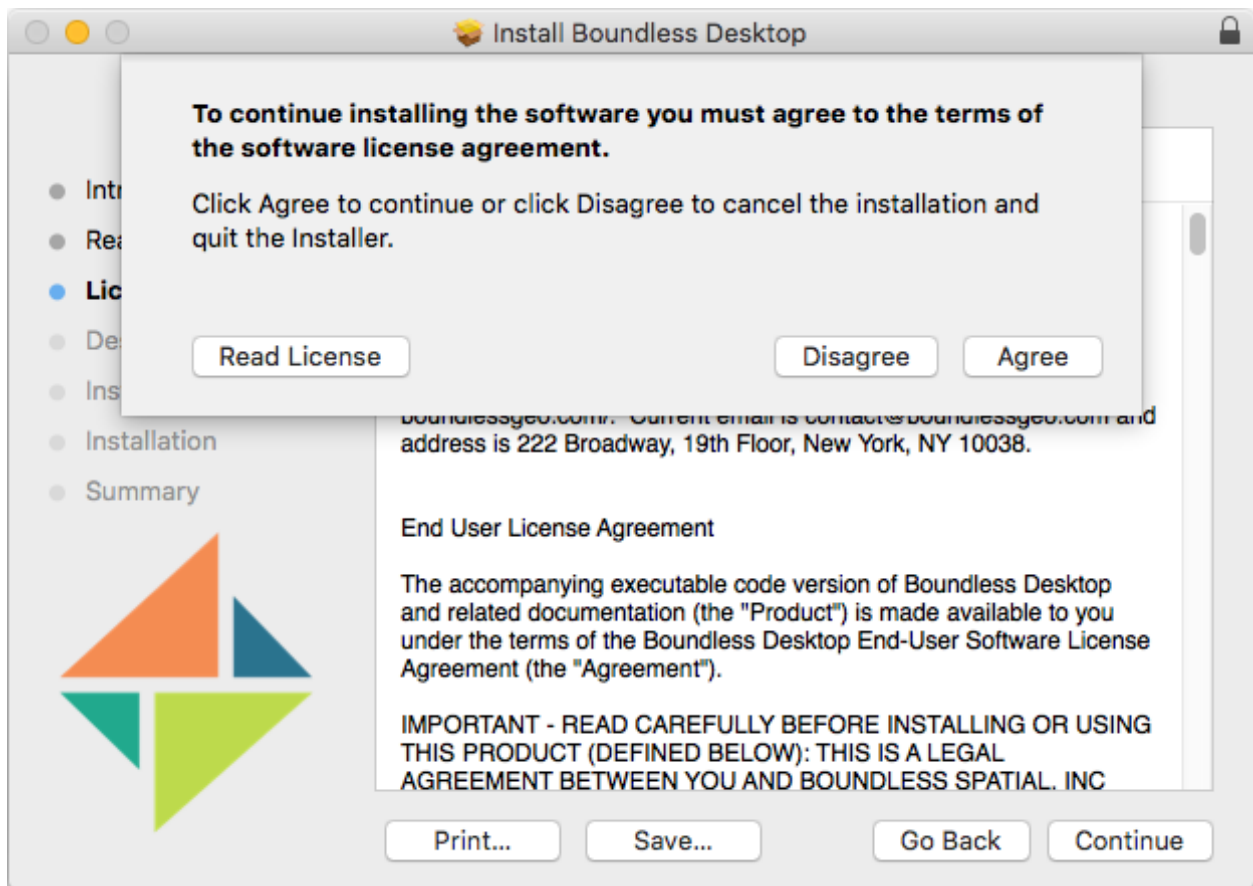
Fig. 4.12: License agreement screen

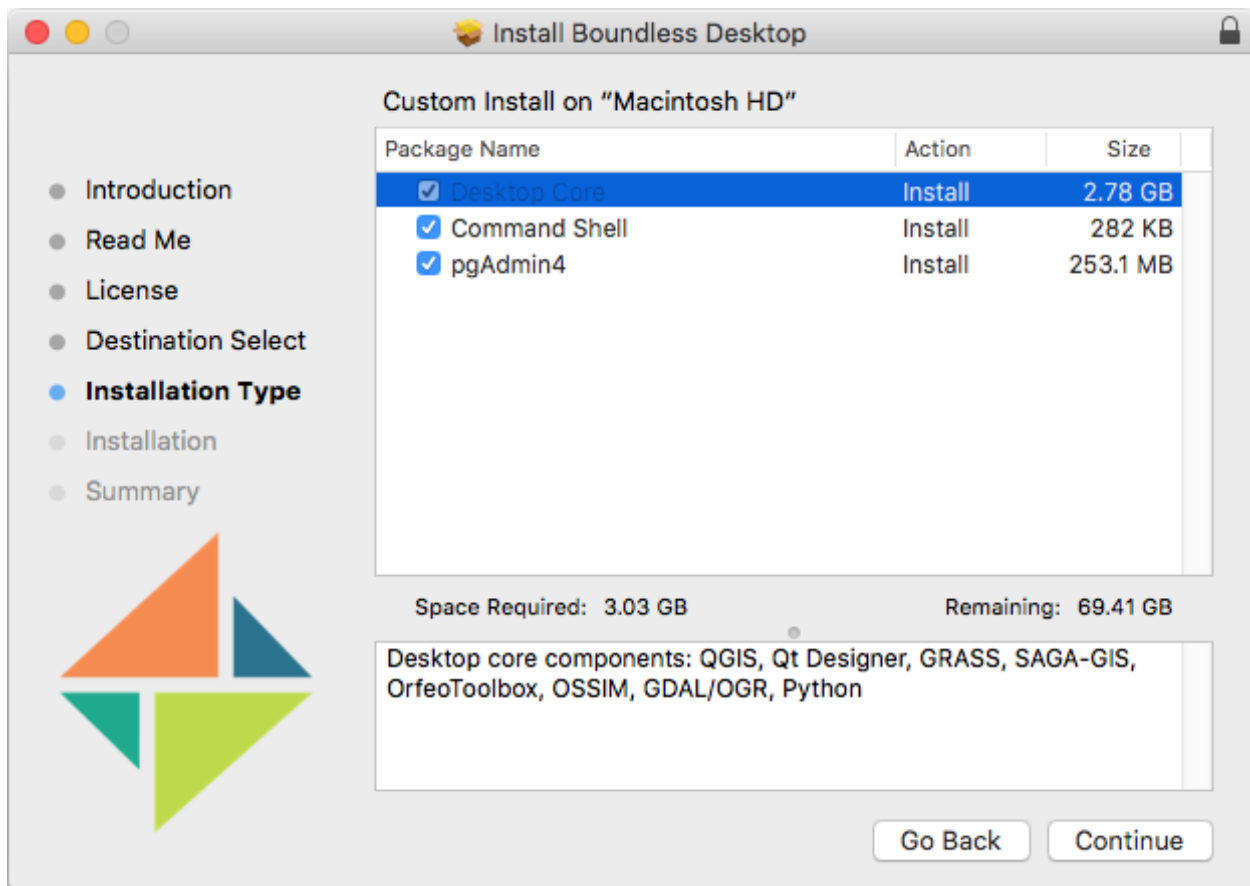Fig. 4.13: License agreement confirmation

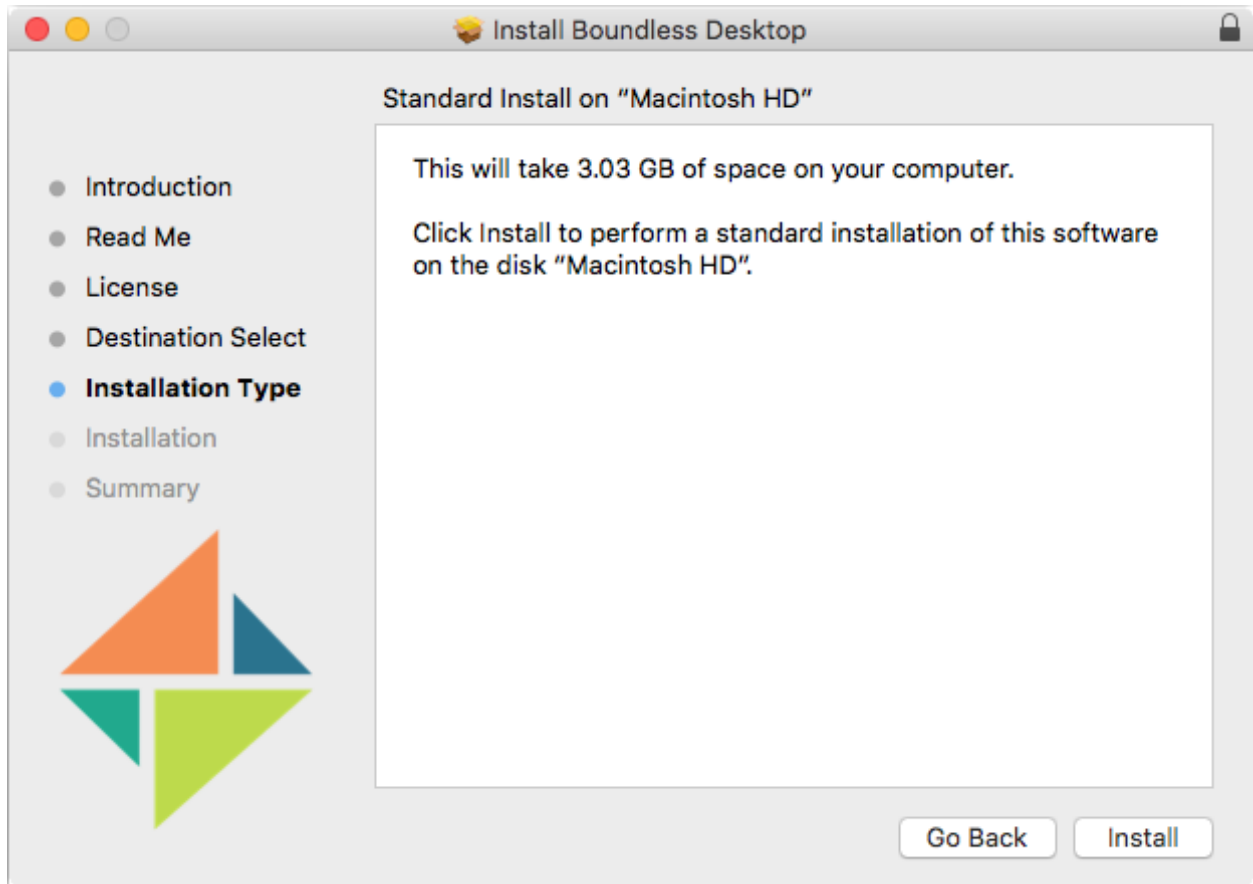Fig. 4.14: Installation Type screen
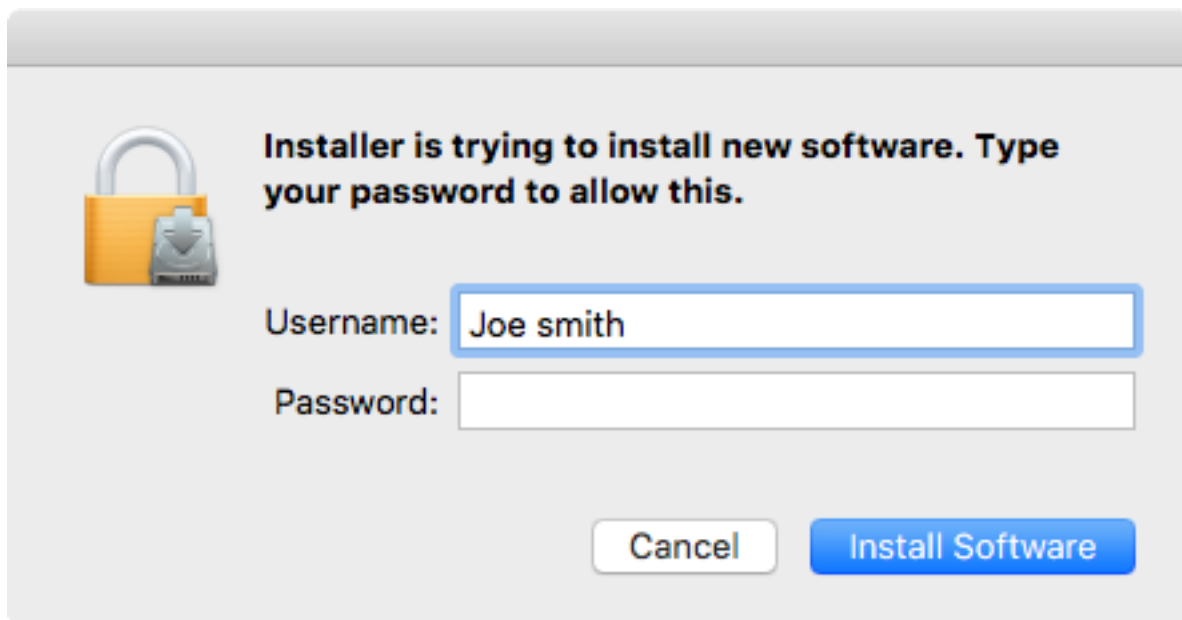
Fig. 4.15: Installation Type screen

Fig. 4.16: Insert your administrator username and password
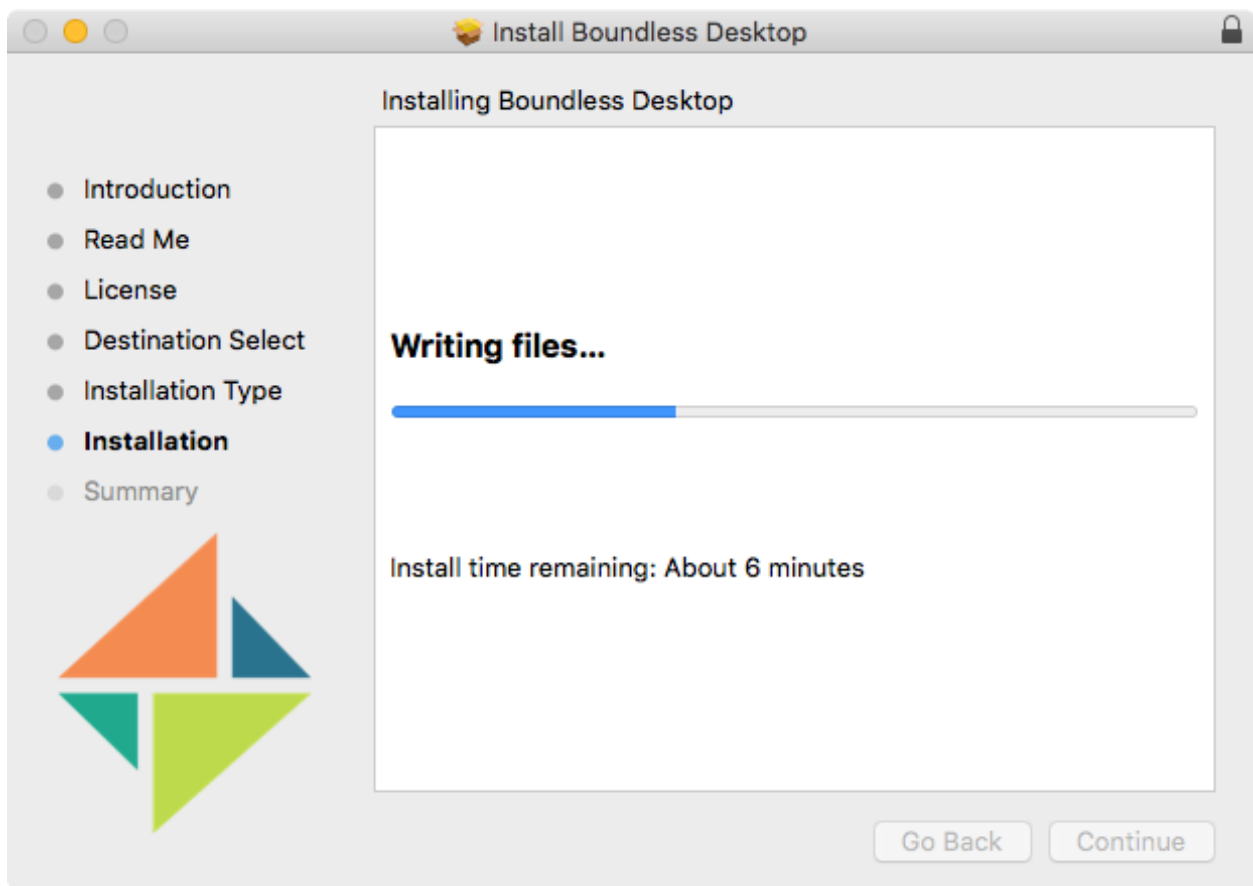
Fig. 4.17: Installation

11. After installation, click *Close*.
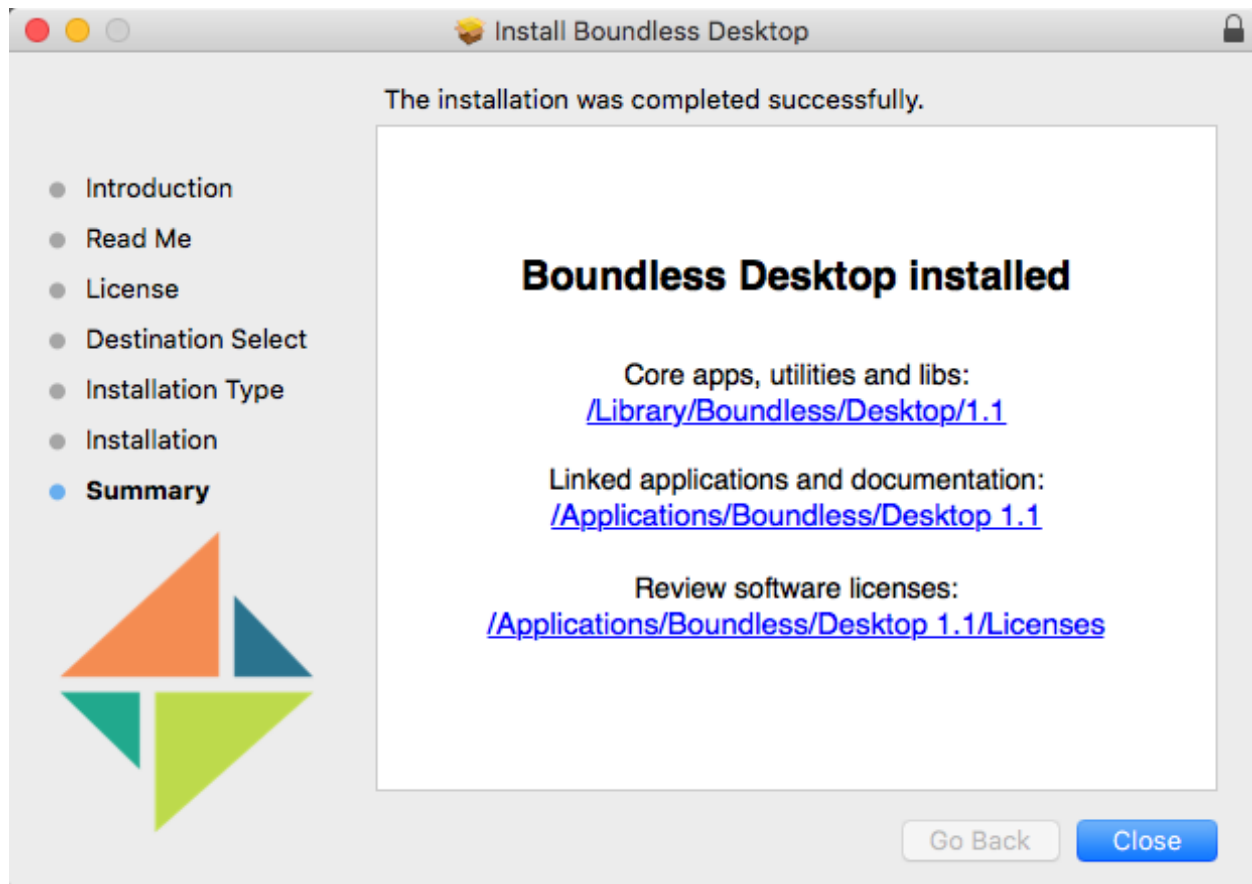
Fig. 4.18: Boundless Desktop successfully installed

After a successful install, Boundless Desktop components, such as QGIS, can be run from shortcuts in the *Applications* Folder (see *Quick start guide* for more information).

Uninstalling Boundless Desktop

Windows

In windows, to uninstall Boundless Desktop 1.1 core apps, libs and utilities and the symbolically linked applications and documentation, use the *Uninstall Desktop* shortcut available in the Boundless Desktop shortcut folder or from the start menu.

1. Double-click the shortcut to run the uninstaller (administrator permissions are required).

2. In the *User Account Control* dialog, click *YES*.

3. In the next dialog, press *Uninstall* to proceed with the uninstallation of Boundless Desktop from your computer.

4. Once the process is finished, click *Close* to close the uninstaller window.
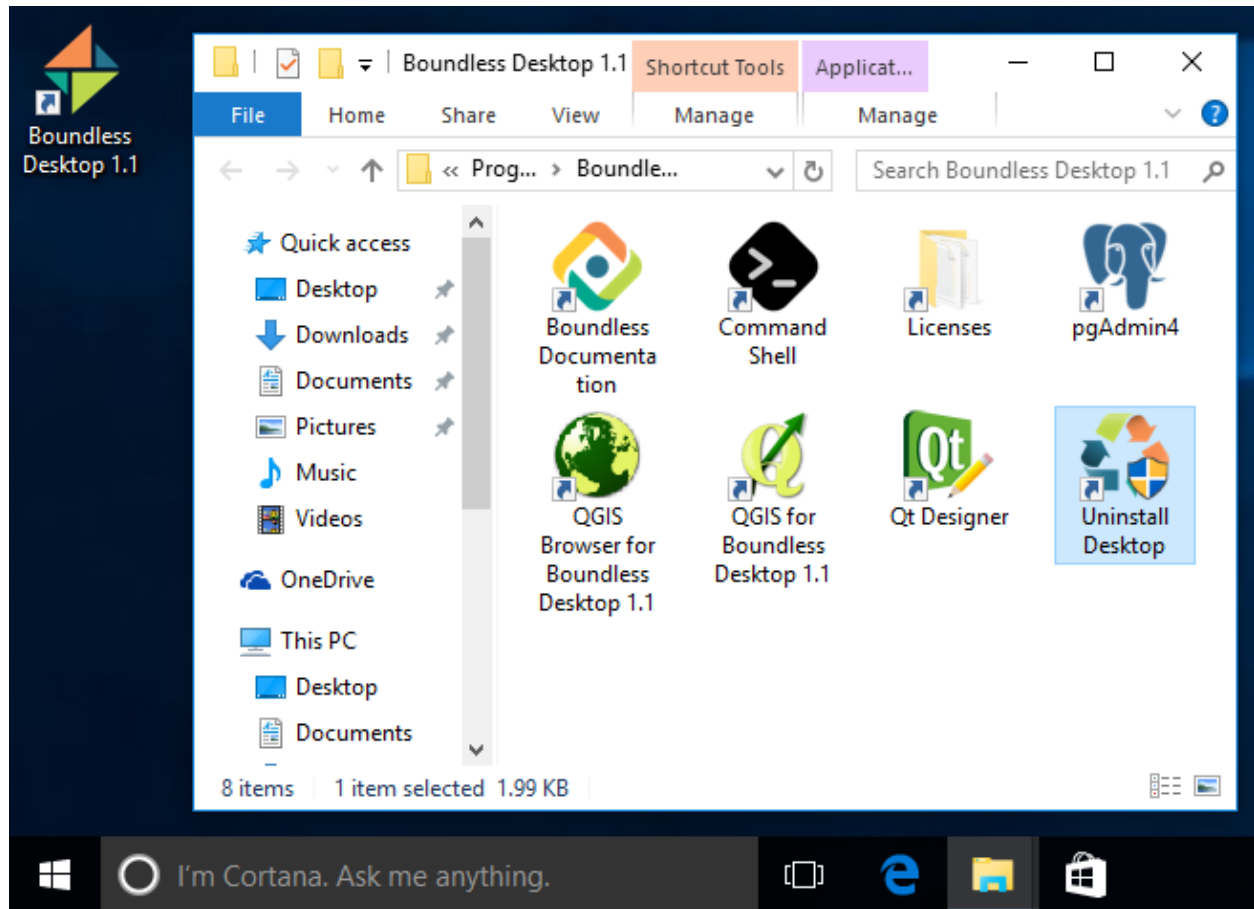
Fig. 4.19: Uninstall Desktop shortcut location
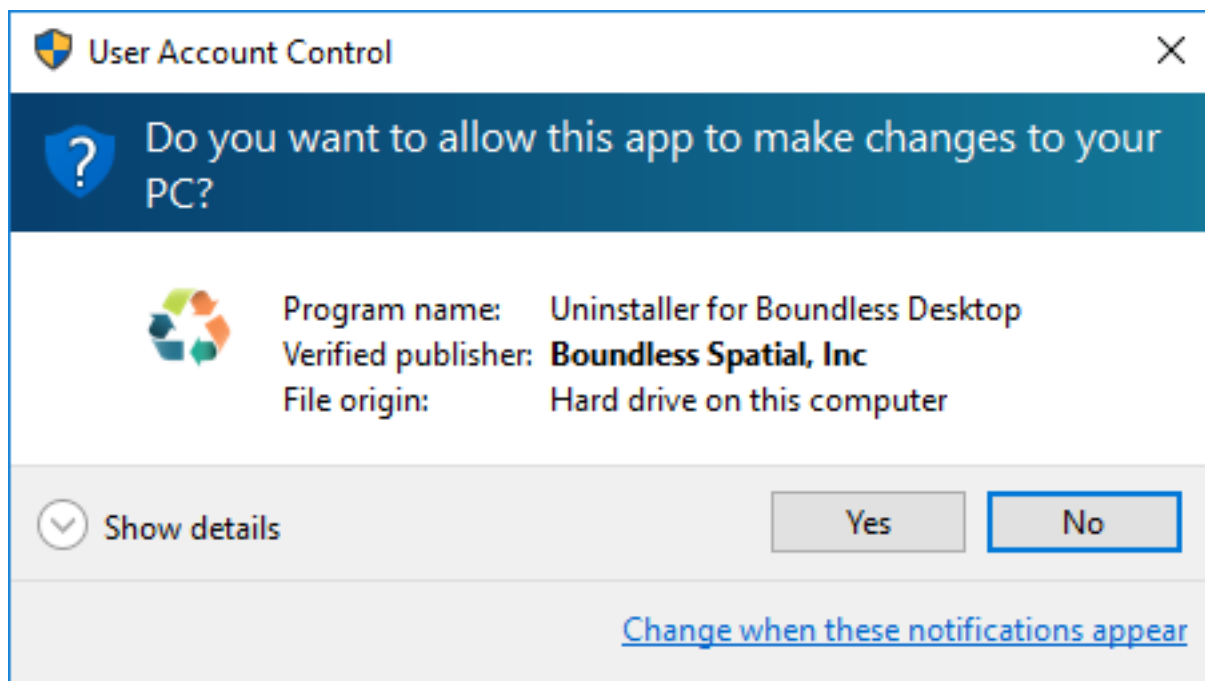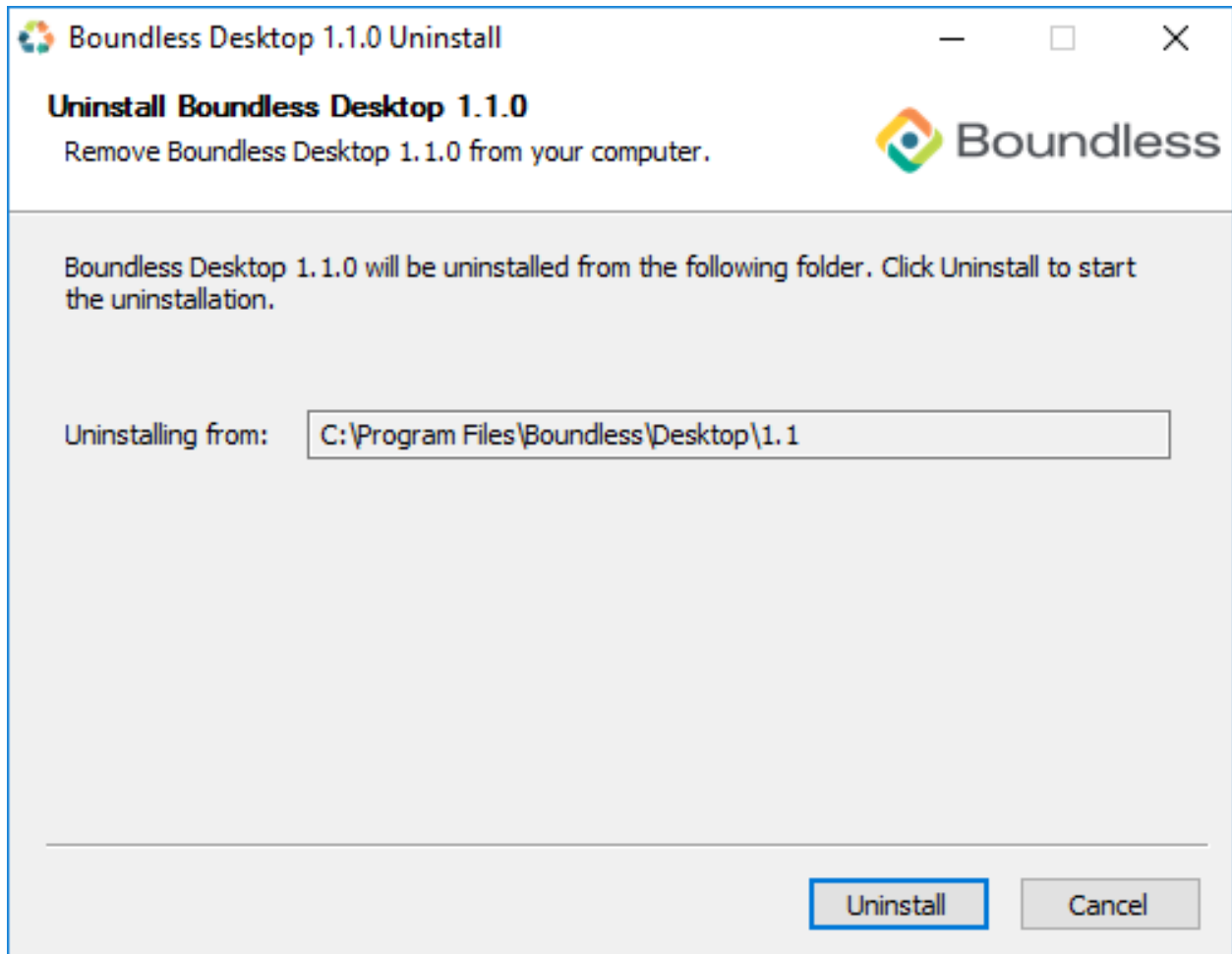
Fig. 4.20: Boundless Desktop Uninstaller dialog

OS X

In Mac OS X, to uninstall the Boundless Desktop core apps, libs and utilities, and the symbolically linked applications and documentation, use the *Uninstall Desktop* shortcut available in the Boundless Desktop's shortcuts folder (*Applications → Boundless → Desktop 1.0*).
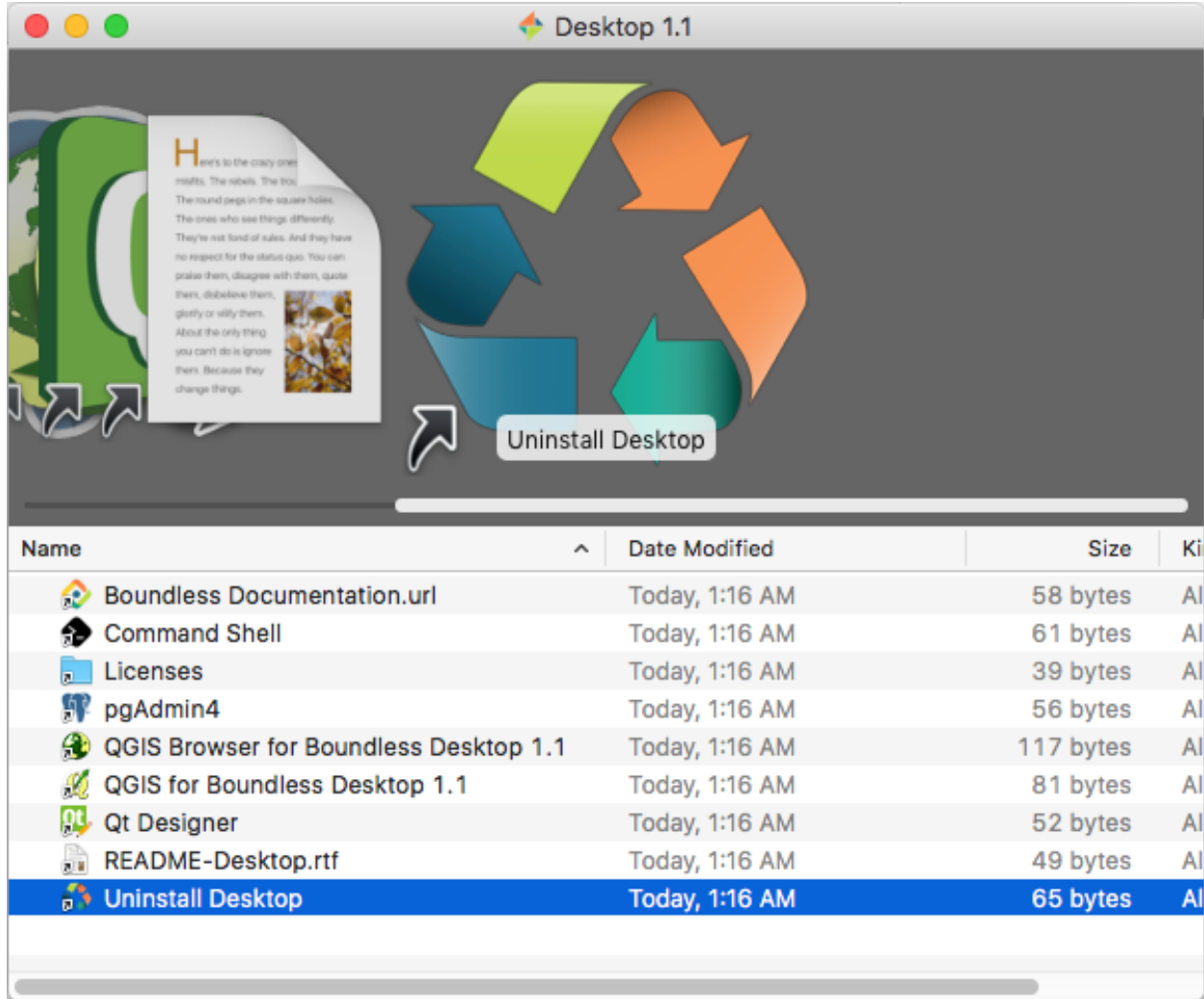
Fig. 4.21: Uninstall Desktop shortcut location

1. Double-click the shortcut to run the uninstaller. Click *Uninstall* to start the uninstall procedure.

2. Administrator permissions are required. Enter your administrator credentials to proceed with the uninstall.

3. Once the uninstall is finished, a dialog will inform the user about what was removed. Click *Quit* to close it.

Note: Both in Windows and Mac OS X, all user's configuration and settings (for example, QGIS settings and plugins) will be preserved for future installations of *Boundless Desktop*.

Note: Although Boundless does not provides installers for Linux, all open source tools that ship with Boundless Desktop are also available for Linux. Please look for instructions in their communities' official documentation. The
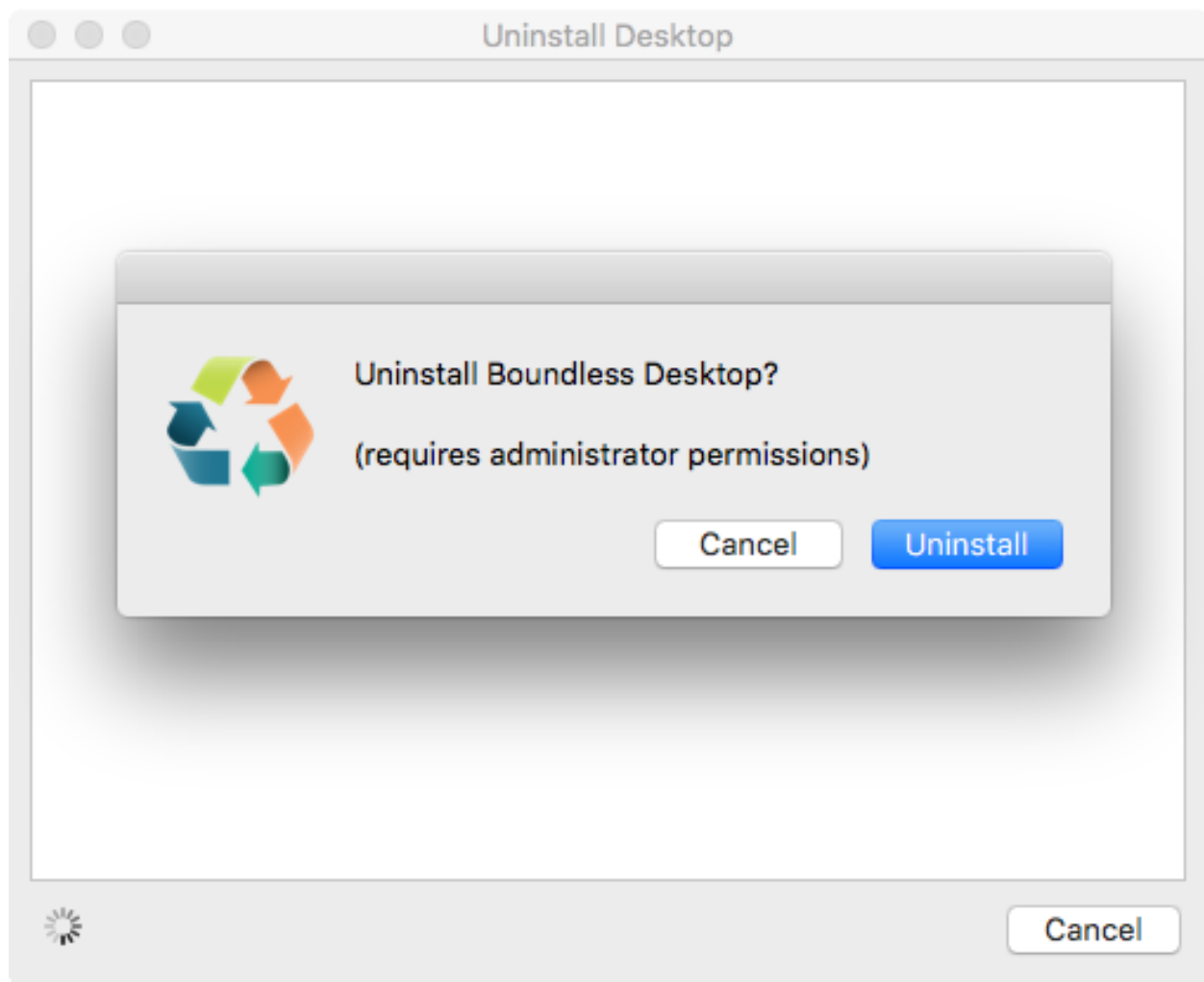
Fig. 4.22: Uninstall Desktop shortcut location
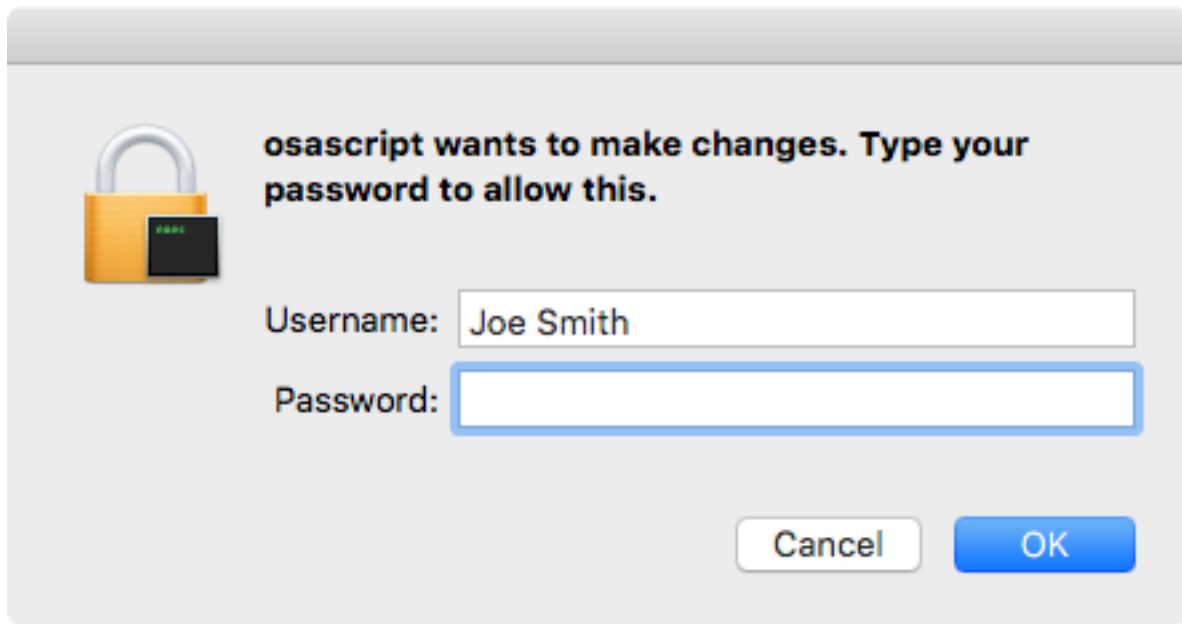
Fig. 4.23: Entering administrator's credentials

Boundless Desktop components section in this documentation provide a list of links to online resources for each tool, which includes installation instructions.

In that case, for installing Boundless Connect plugin or any other Boundless plugin for QGIS, please consult the *Boundless plugins for QGIS* section.
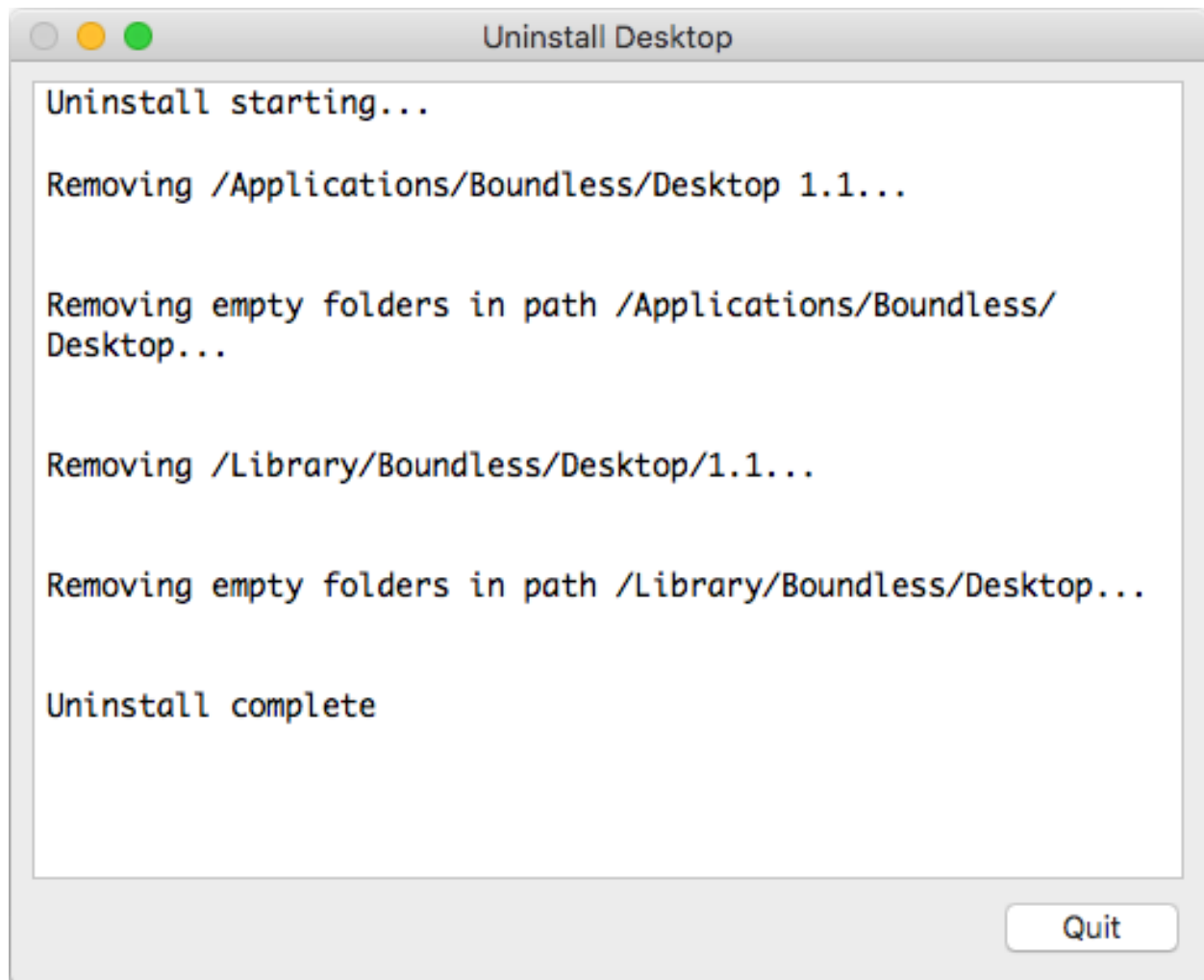
Fig. 4.24: Uninstallation complete

Quick start guide

Windows

After installation, a Boundless Desktop shortcut is added for easy access to all the installed tools. Double-click the icon to open the shortcuts folder. Then, double-click any of the icons to launch the respective tools.

The same shortcuts are also added to the start menu in a group called *Boundless Desktop* 1.1. Click on any of the tools' icons or names to launch it.

There is also an *Uninstall* shortcut to easily remove *Boundless Desktop* from your computer. Please see *Uninstalling Boundless Desktop* for more information about uninstalling procedures.

Note: Users should not 'pin' the launched apps to the Windows taskbar because the pinned icons will not launch the apps correctly next time.

Mac OS X

After installation, you will find a Boundless icon in the *Applications* folder.

Double-clicking the Boundless icon will lead you to a folder with a *Desktop* 1.1 icon. Double-clicking it will open the Boundless Desktop's shortcuts folder. In it, double-click any of the icons to launch the respective tool.

In both Windows and Mac, there's a handy *Licenses* shortcut to access the licenses for each individual tool. Please make sure to consult those licenses, including Boundless EULA for Boundless desktop.

There is also a link to Boundless Desktop's README file, that provide valuable information about it.

For more information on how to use each tool, please see the *Boundless Desktop components* section.

Fig. 5.1: Boundless desktop shortcut and folder in windows
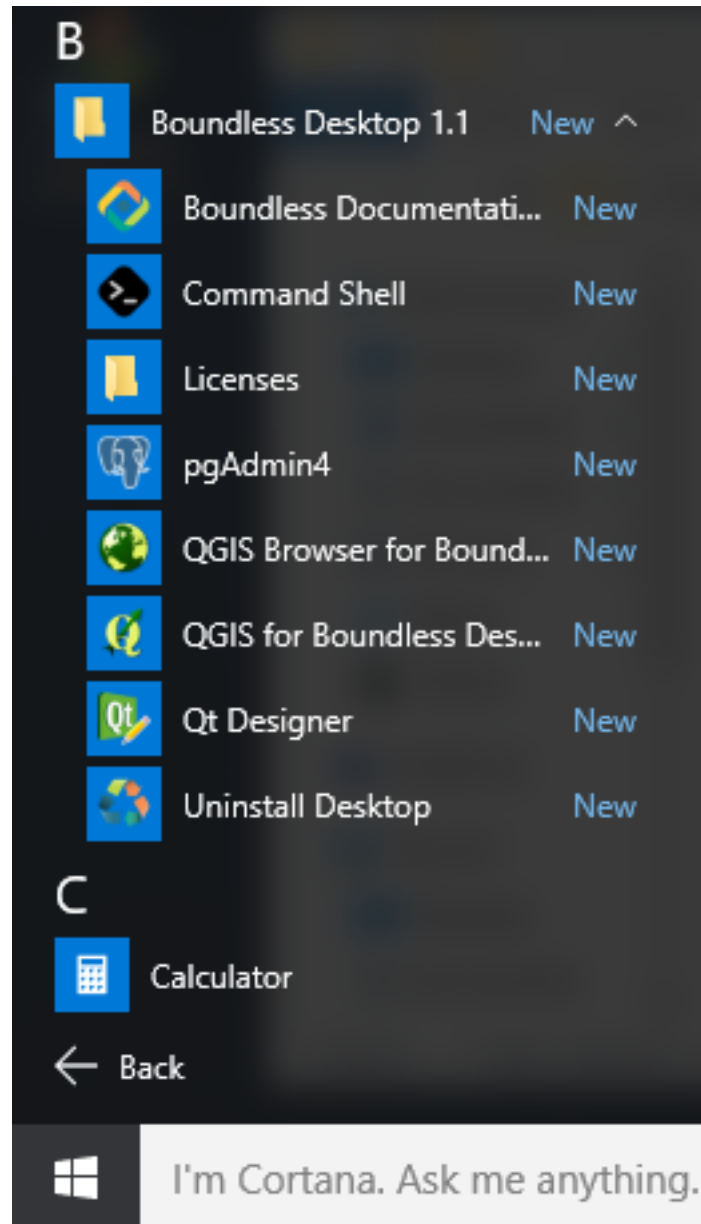
Fig. 5.2: Boundless Desktop shortcuts group in the windows start menu
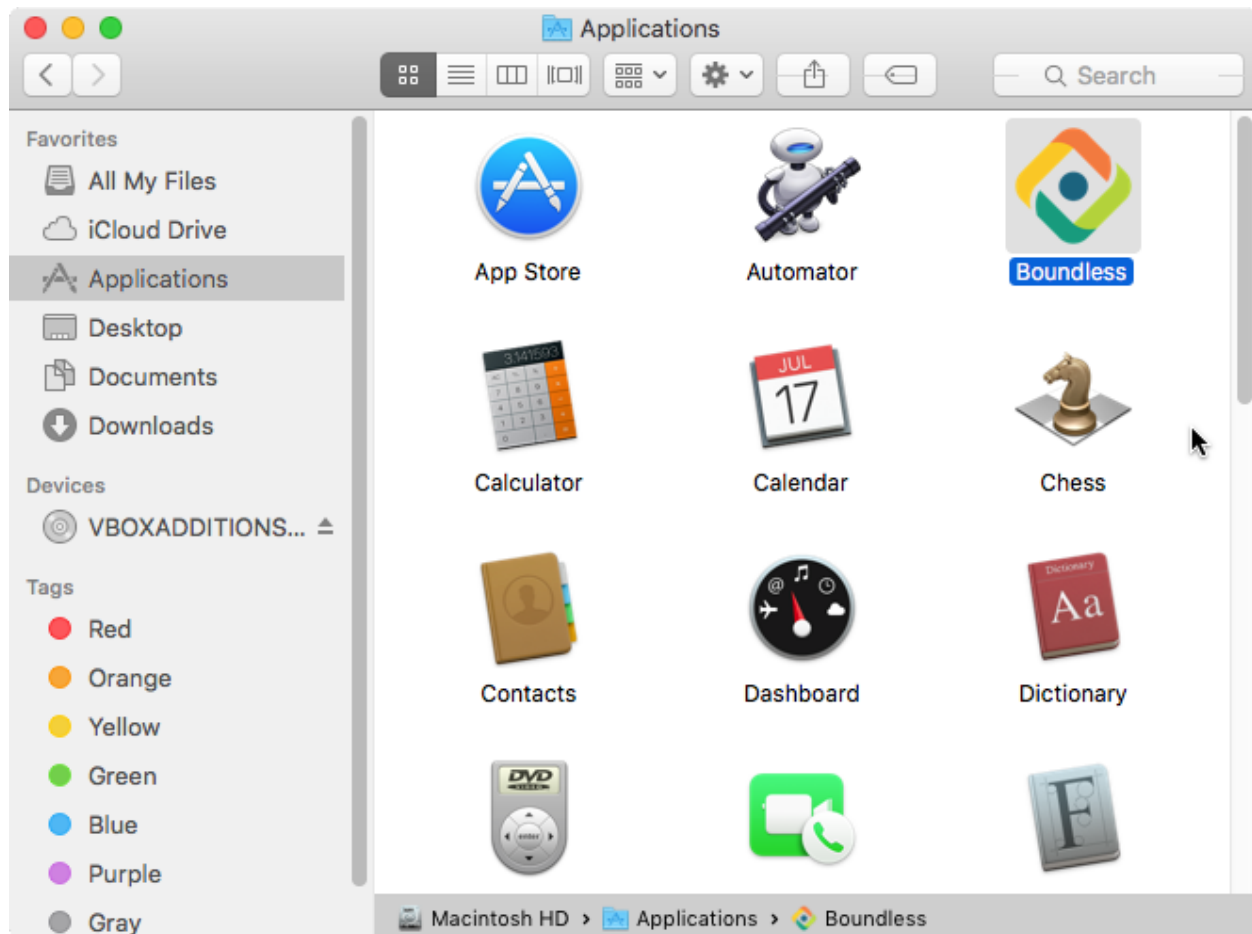
Fig. 5.3: Boundless icon in Mac OS X Applications folder
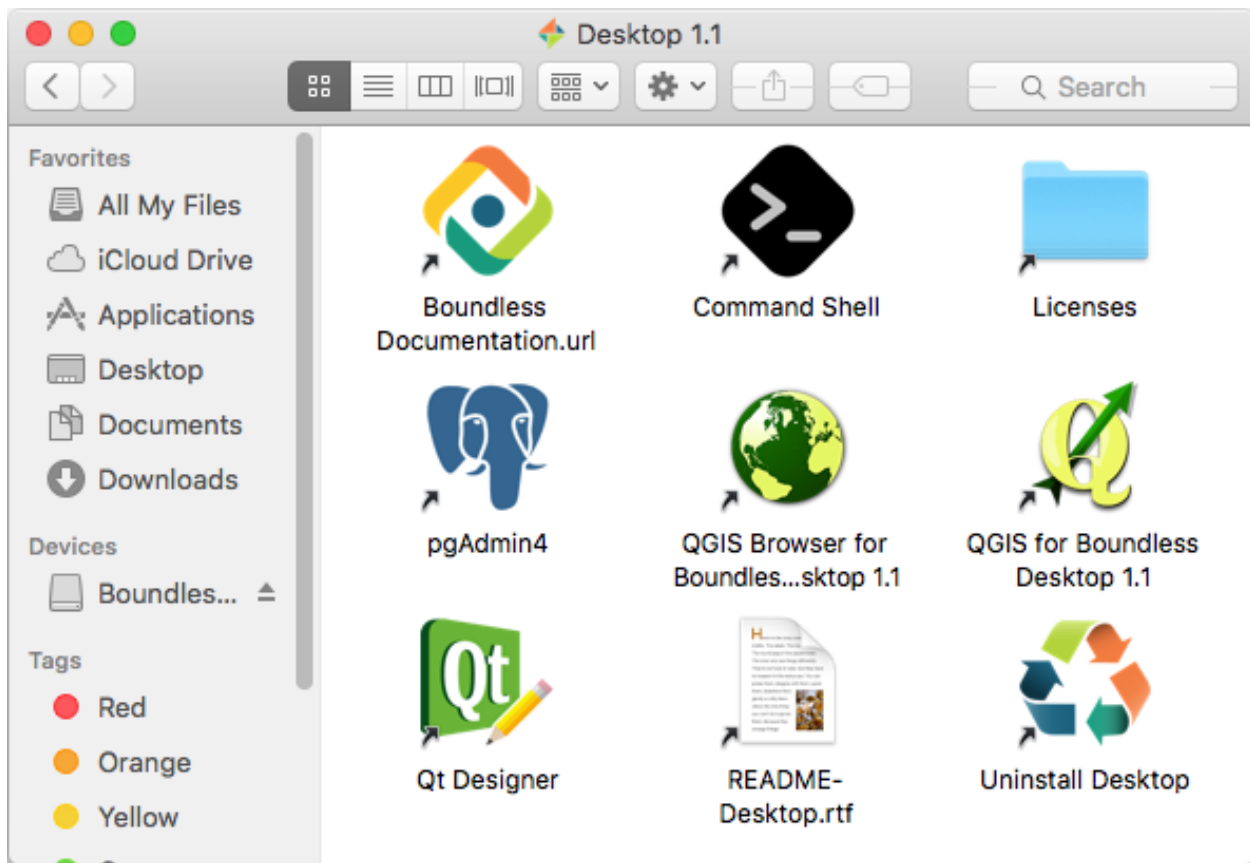
Fig. 5.4: Boundless desktop shortcuts folder in Mac OS X

Boundless Desktop components
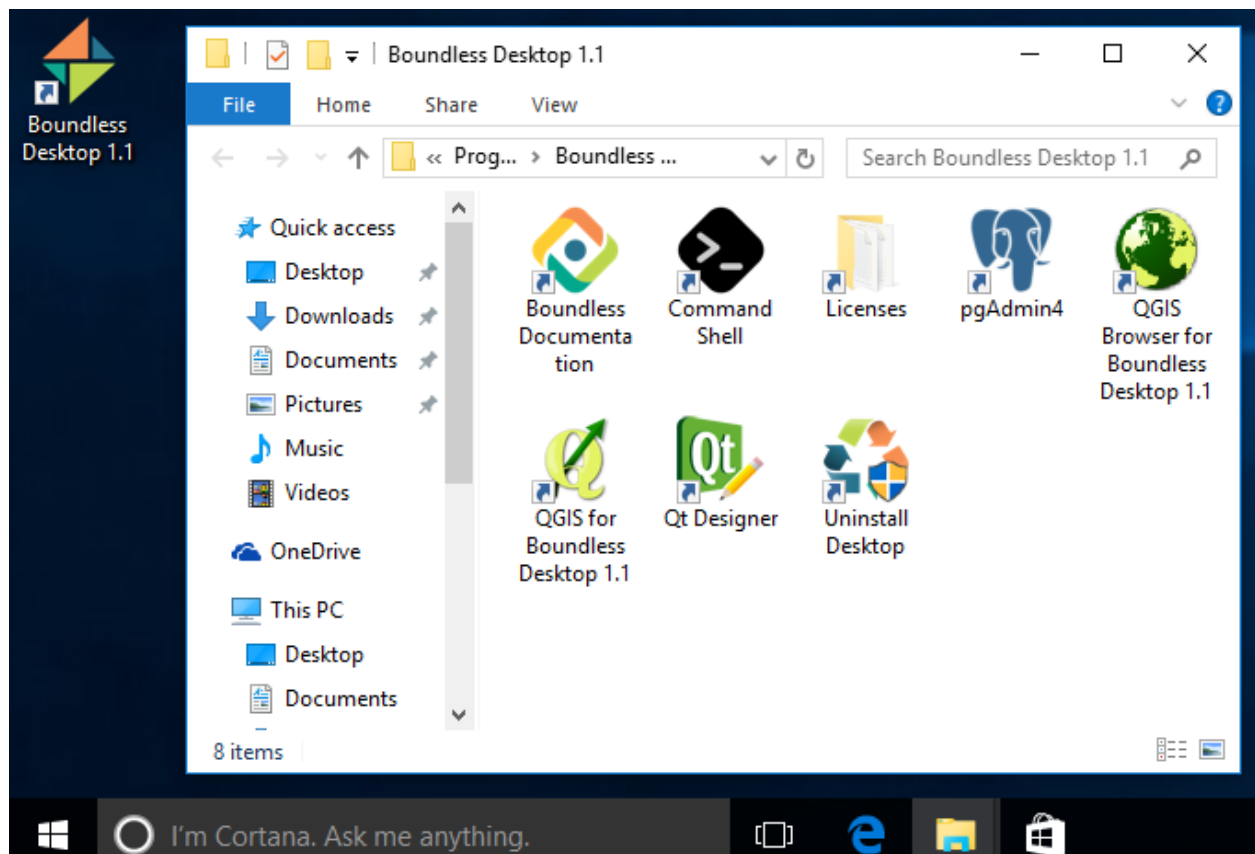
Boundless Desktop is composed of several open source geospatial client-side tools. Each tool have it's own usefulness and facets. In the following section you will find more about each one of them, some quickstarts, and relevant links to their online resources.

QGIS

About

QGIS is the central piece of the Boundless Desktop package installation. The well known leading Open Source GIS for desktop, QGIS, is a cross-platform desktop application for viewing, editing, and analysing geospatial data from a variety of (proprietary and open) vector, raster, and database formats.

QGIS's development is steered by the QGIS Project, which works with hundreds of volunteers and companies (including Boundless) from all over the world, that helps enhance and maintain QGIS.

For convenience, outside the supported scope of Boundless Desktop, the following processing providers are available for QGIS's Processing framework:

- SAGA 2.3 by http://www.saga-gis.org
- GRASS 7.2 by http://grass.osgeo.org
- Orfeo Toolbox 5.0 by https://www.orfeo-toolbox.org

Quick start guide

This section aims to give you a brief overview about QGIS. We will mainly focus on QGIS's graphical user interface (GUI) and some of the most common operations like loading data, managing layers and printing a map. For more detailed information on QGIS usage see the tutorials on our learning centre and/or consult QGIS's official documentation.

For this Quick start, we will be using the Natural Earth data. Please, download Natural Earth Quickstart Kit and unzip it to any folder that you find convenient to access.

QGIS user interface

1. Use any of the available QGIS's shortcuts on your computer to open QGIS.

   By default, the QGIS GUI should resemble the one presented in the next figure.
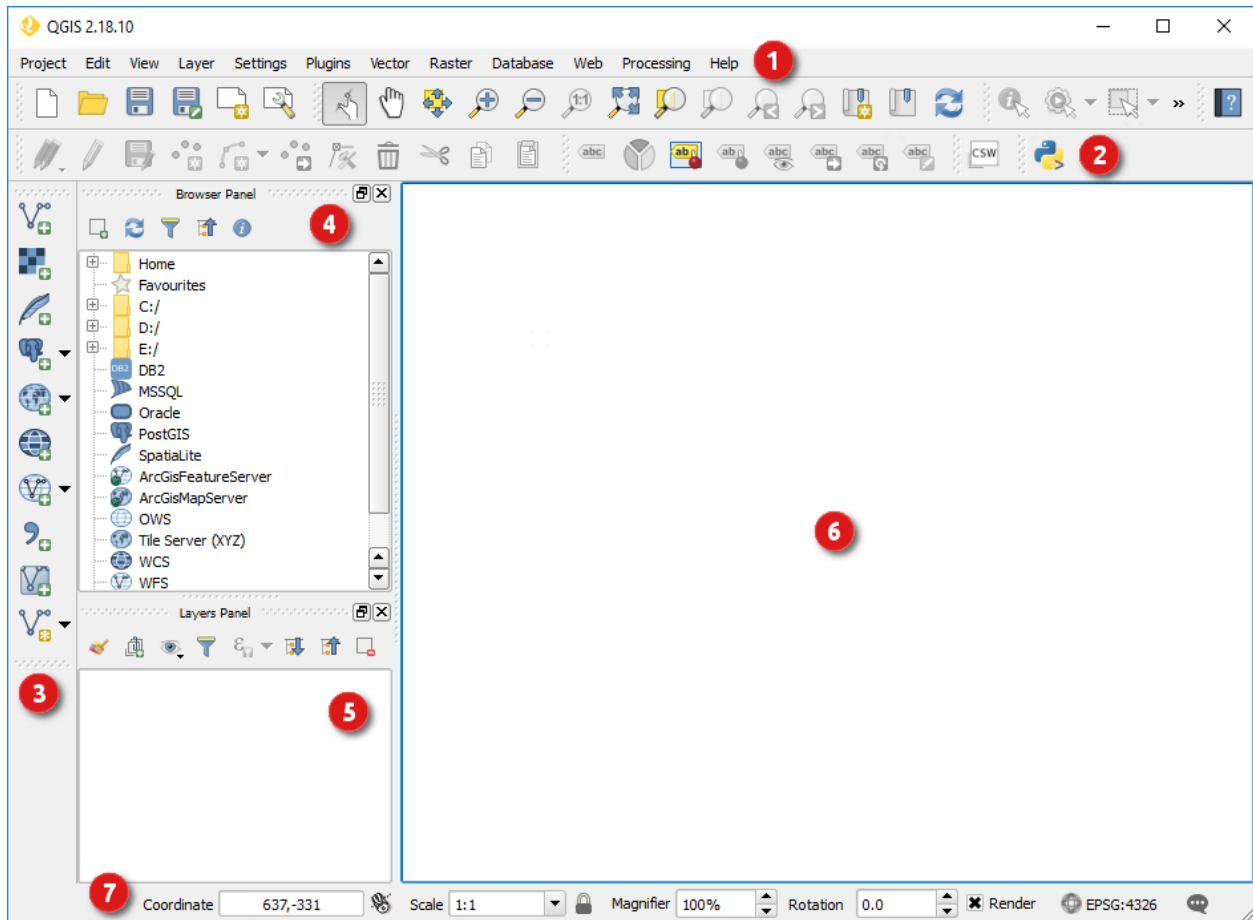
Fig. 6.1: QGIS Graphical User Interface (GUI)

As a very basic overview of the default GUI we have:

- On the top of the screen you will find the menu bar (1). The menu bar provides access to various QGIS features using a standard hierarchical menu.

- Below it, you will find the main toolbar area (2). Toolbars allow access to most of the same functions as the menus, plus additional tools for interacting with the map. Hold your mouse over the item for a short description of the tool's purpose. Toolbars can be moved by dragging and dropping them elsewhere and even hidden using *View → Toolbars*.

- Another toolbar can be found at the left side of the screen (3). By default, the *Manage Layers Toolbar* is placed there. This toolbar can be used to load and create data, as we will see later.

- Next to it, there are two Panels by default: The *Browser Panel* (4), which is used to browse and load data, and the *Layers Panel* (5) which is used to toggle layers visibility, setting their relative order, accessing the

layer's properties and much more. Panels can also be moved by drag and drop, or even hidden using *View –> Panels*.

- To the right and middle of the screen, you find the map canvas (6) where all visible layers' geometries will be displayed.

- At the bottom of the screen you will find the status bar (7) with several useful information and controls for the current map scale, the mouse pointer coordinates, the current Coordinate Reference System's EPSG code in use, and more.

Loading data

In QGIS, to load data, you can either use the *Browser Panel*, the *Manage Layers Panel* or even use your operating system file explorer.

1. Using the *Browser Panel*, browse to the sample data location, more precisely into the `Natural_Earth_quick_start\10m_physical` folder. Double-click the folder's names or click the plus signs next to it to view its contents. Find the `ne_10m_coastline.shp` shapefile and drag and drop it from the browser to the map canvas. It will load the vector layer.
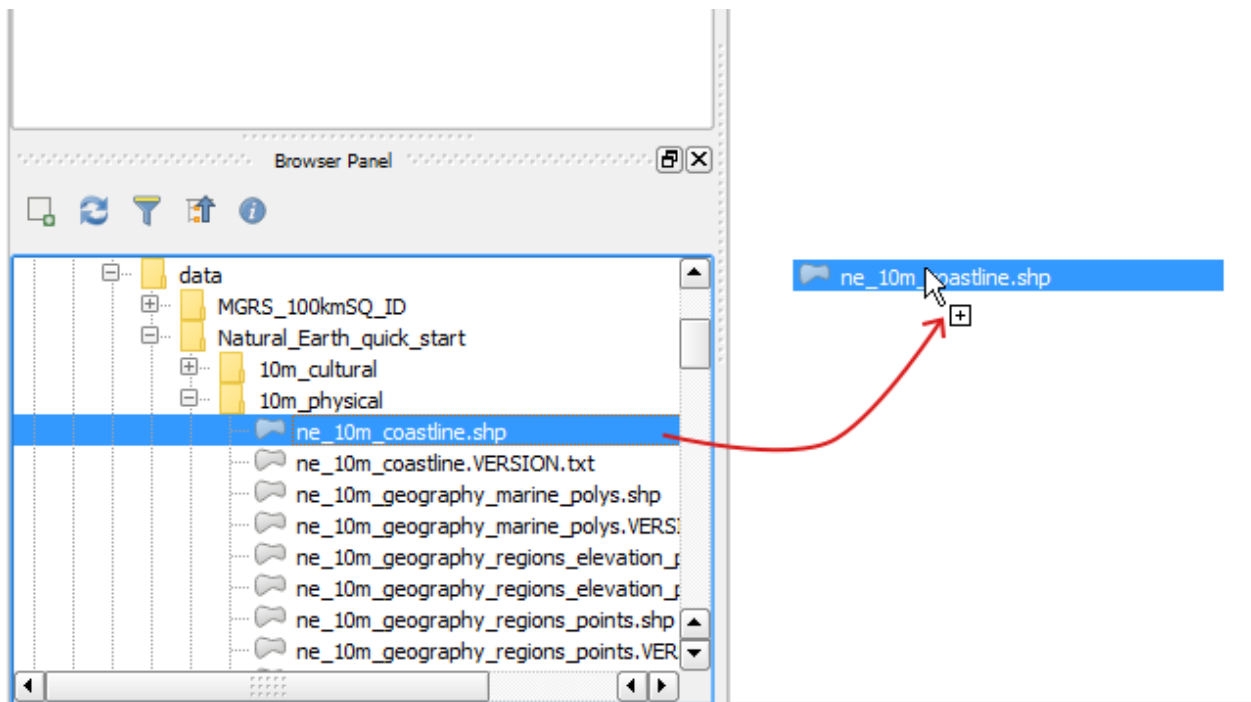
Fig. 6.2: Loading a layer using the Browser Panel

The layer should now be visible on the map canvas, using a random style. It should also be visible in the *Layers Panel* list.

2. Let's open another file, this time using the *Manage Layers toolbar* buttons. Notice that there is one button for each type of dataset, so we should select the most suitable one. Click on the *Add Raster Layer*. Then, navigate ito the folder `Natural_Earth_quick_start\50m_raster\NE1_50M_SR_W`, select the `NE1_50M_SR_W.tif` file and click *Open*.

3. Finally, let's open an ESRI fileGeodatabase, just because it has a small catch. In the guilabel:*Manage Layers toolbar* click the *Add Vector Layer*. In the next dialog, under *Source type*
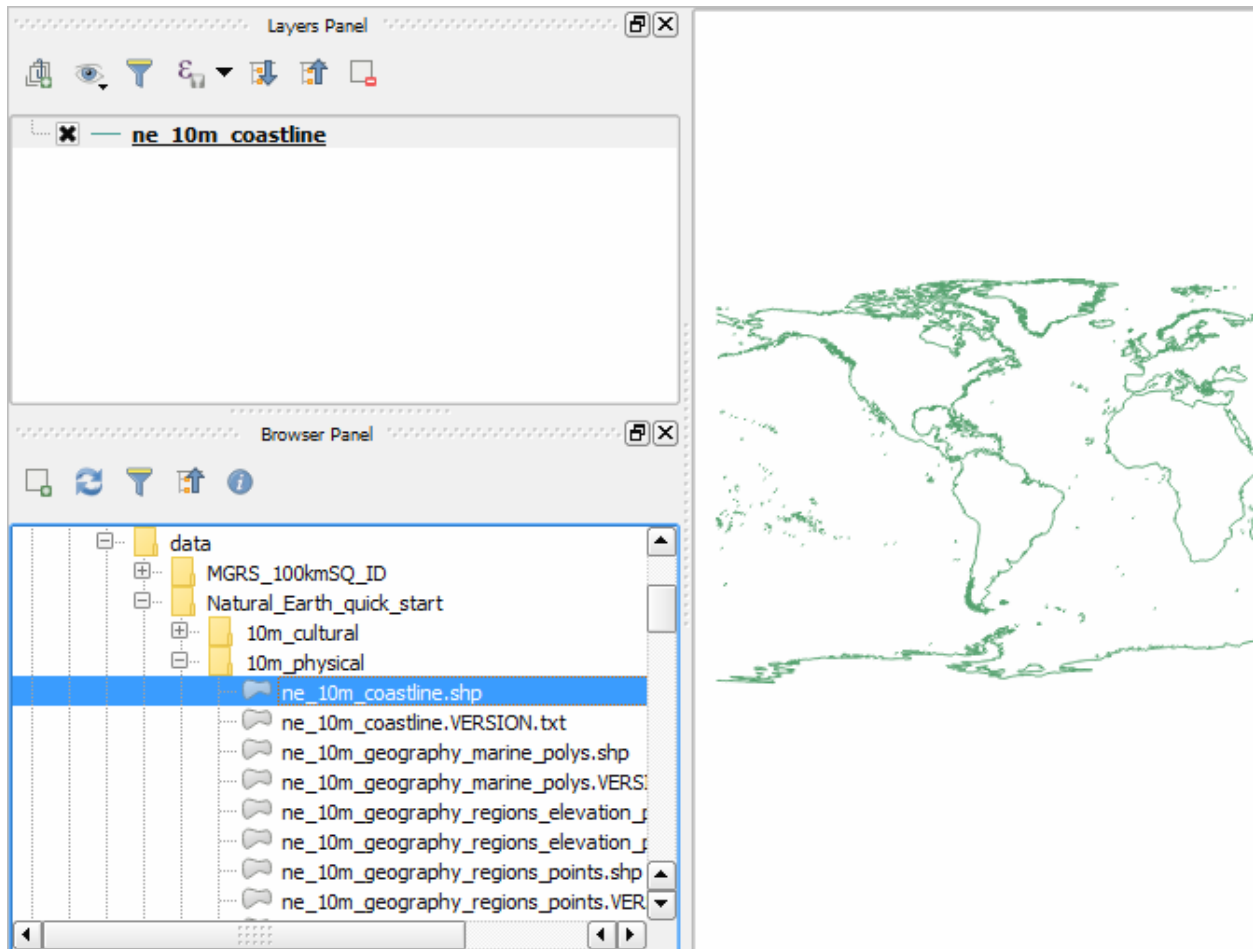
Fig. 6.3: Successfully loaded layer
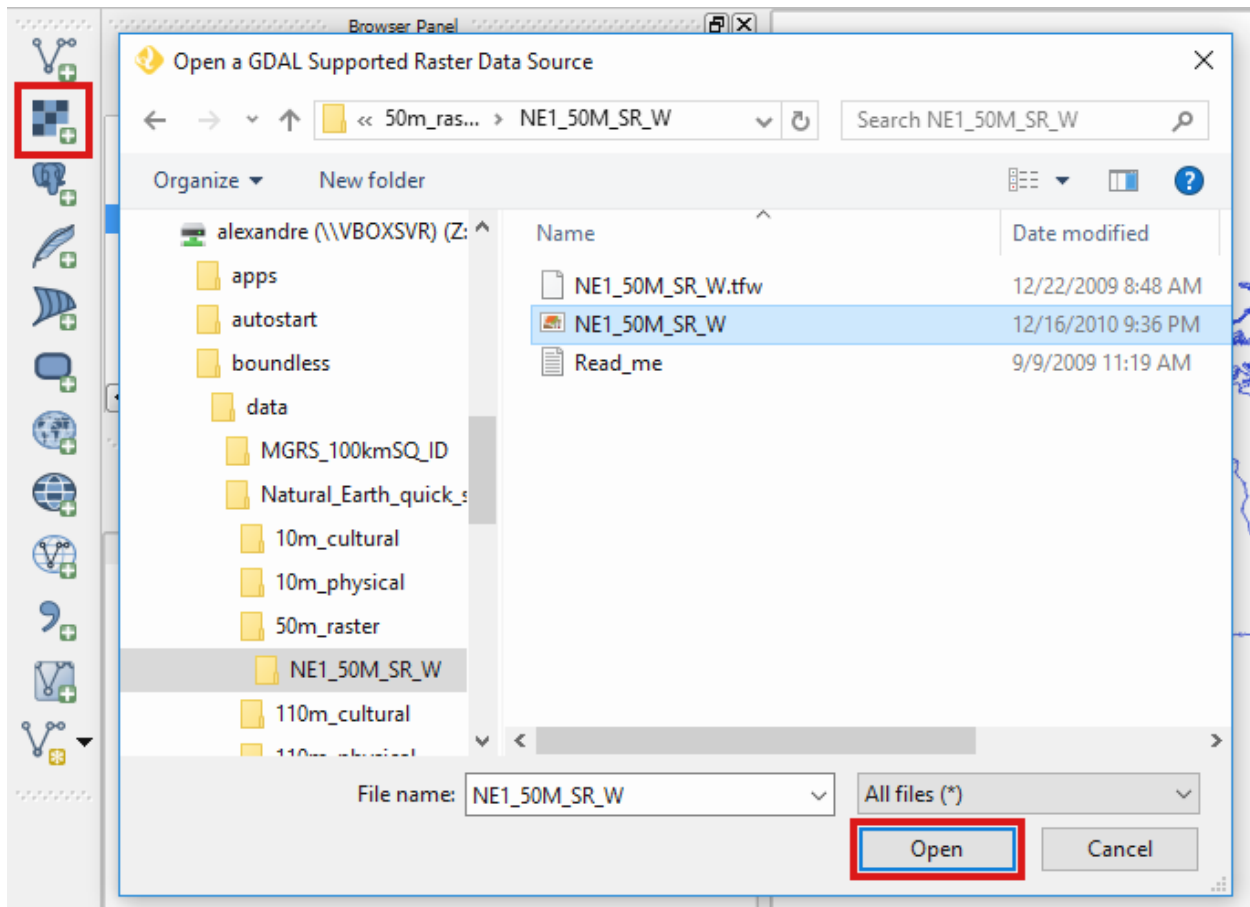
Fig. 6.4: Loading a layer using Add Raster Layer

check the `Directory` option. Then, making sure that *Type* is set to `ESRI FileGDB`, use the *Browse* button to navigate and select the `Natural_Earth_quick_start\10m_cultural` `\ne_10m_admin_1_states_provinces_geodb.gdb` folder. Click *choose*. Finally, back in the dialog window, click *Open* to load the layer.
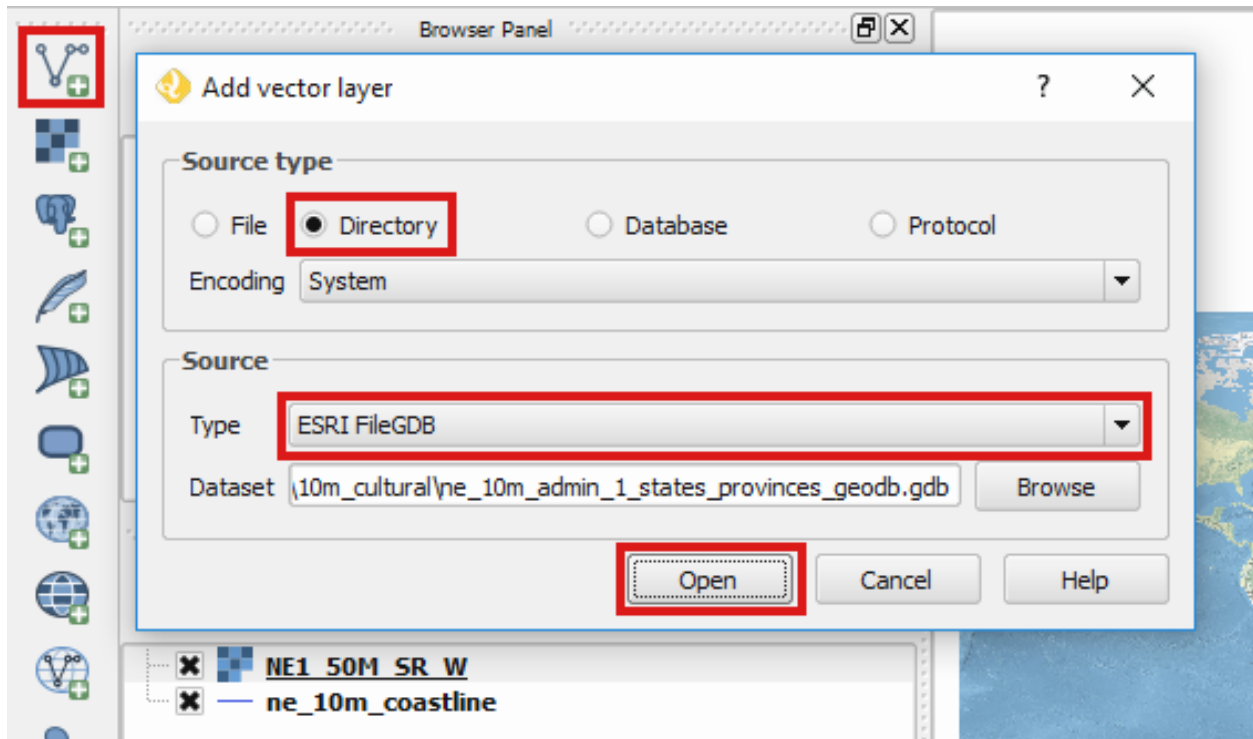
Fig. 6.5: Loading an ESRI FileGeodataBase layer using Add Vector Layer

4. Feel free to add any other data, but bare in mind that you can load several files at once by holding the `Ctrl` key during file selection in any of the two described methods. Also, you can drag and drop files from your operating system's file manager (Windows Explorer in Windows or Finder in Mac OS X) into QGIS Map canvas to load them.

Navigating in the map canvas

To navigate the map canvas you can primarily use your mouse wheel. For more precise control over the map canvas, you can also use the *Map Navigation Toolbar* tools.

1. Position your mouse pointer in an area you that you want to have a closer look, and spin your mouse wheel up to Zoom In. Spin the mouse wheel in the opposite direction to Zoom Out.

2. To pan, just press the mouse wheel down and hold it, move the pointer around and release the wheel once satisfied.

3. As stated above, the *Map Navigation Toolbar* provides more precise ways to navigate the map.

   Press the *Zoom Full* button to show the full extent of your data. Now select the *Zoom In* tool and draw a rectangle around an area of interest using by clicking and dragging the left-mouse-button on the map canvas.

4. Notice you can use the *Zoom last* and *Zoom last* to undo and redo changes to the map canvas extent
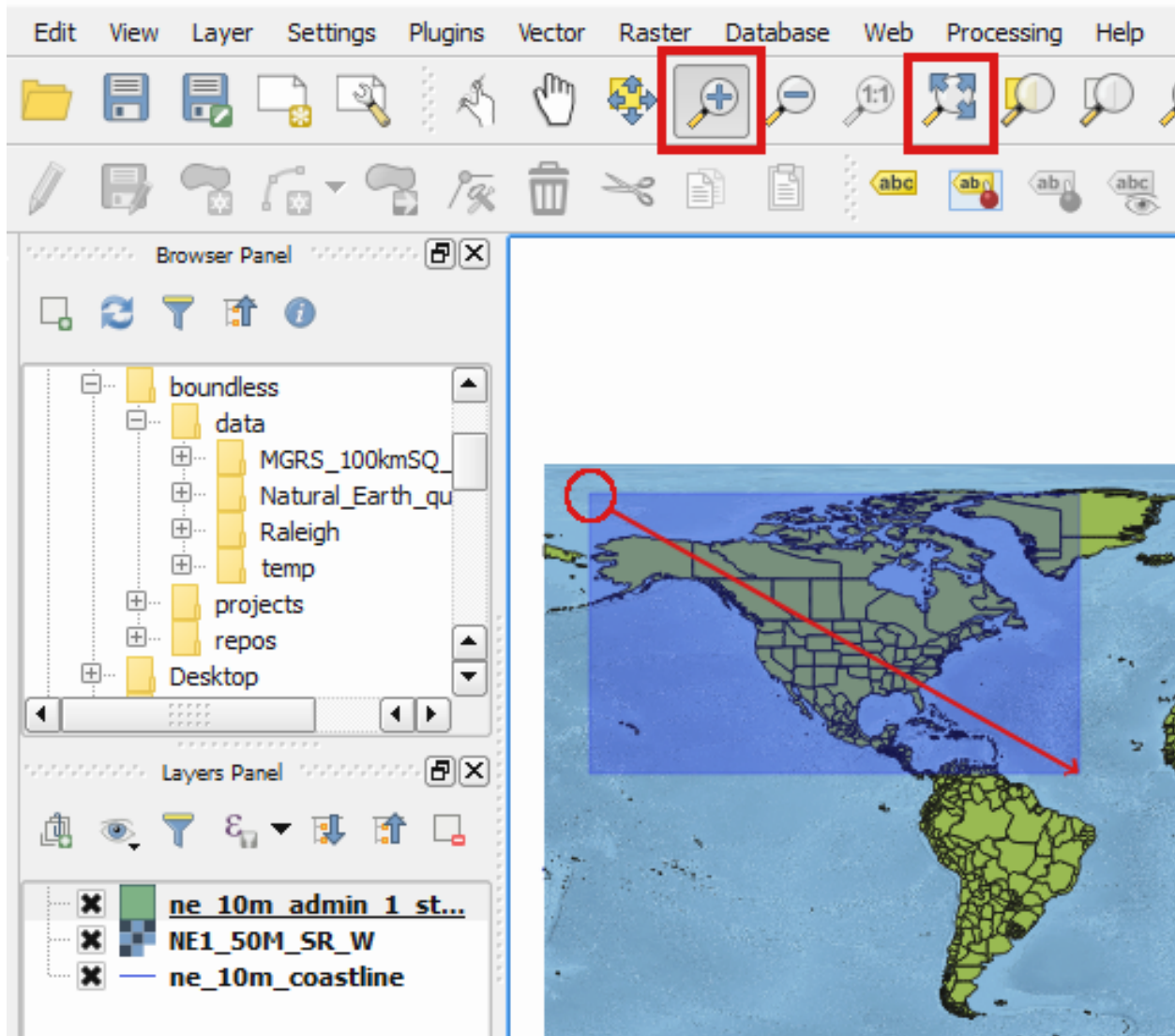
Fig. 6.6: Loading an ESRI FileGeodataBase layer using Add Vector Layer

Managing Layers

We have been using the *Layers Panel* already, but let's have a deeper look into it's potential.

1. Select a layer by clicking on its name on the layers list/legend. The layer will become the *active layer*, meaning that many layer specific tools and actions will apply to that layer in particular. For example, select the `ne_10_coastline` layer and, in the *Map Navigation Toolbar*, click the *Zoom to Layer* button. This will zoom the map canvas to the full extent of a particular layer.
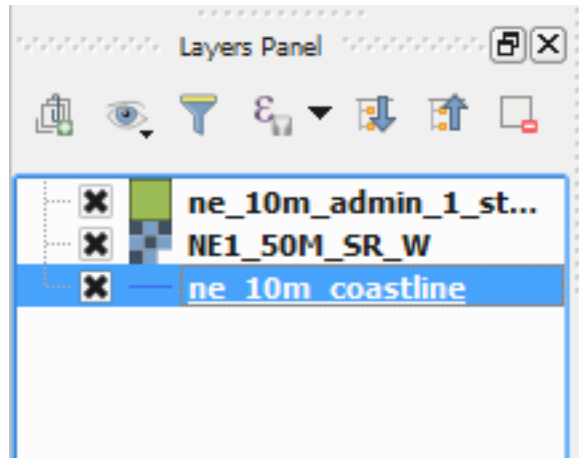
Fig. 6.7: Layer active in the Layers Panel

2. You can change the order of the layers (and consequently their rendering order) by dragging them up and down in the *Layers*. Do this making sure to put the raster layer at the bottom, the polygons layer above it, and the line layer at the top.
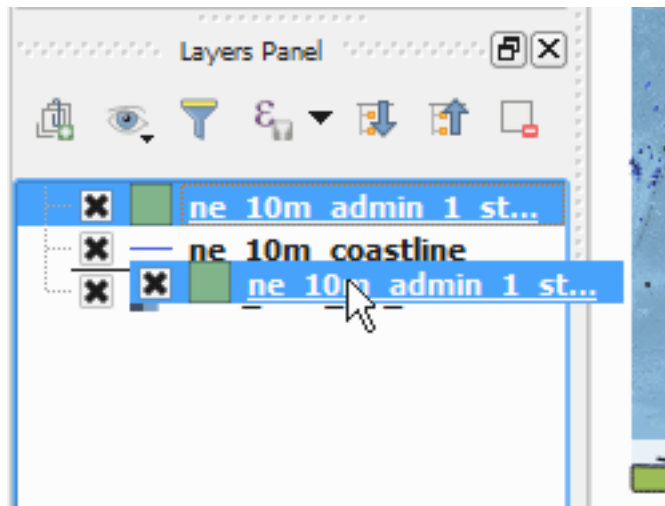
Fig. 6.8: Changing the order of the layers

3. You can change the visibility status of the layers by (un)checking the small checkbox next to its name. Give it a try and see the result in the map canvas. (Make sure to keep all layers visible in the end)

4. By double-clicking the layer name in the *Layers Panel*, or right-clicking and selecting *properties*, you will open the layer's properties. Double-click the `ne_10m_admin_1_states_provinces` layer, navigate to the
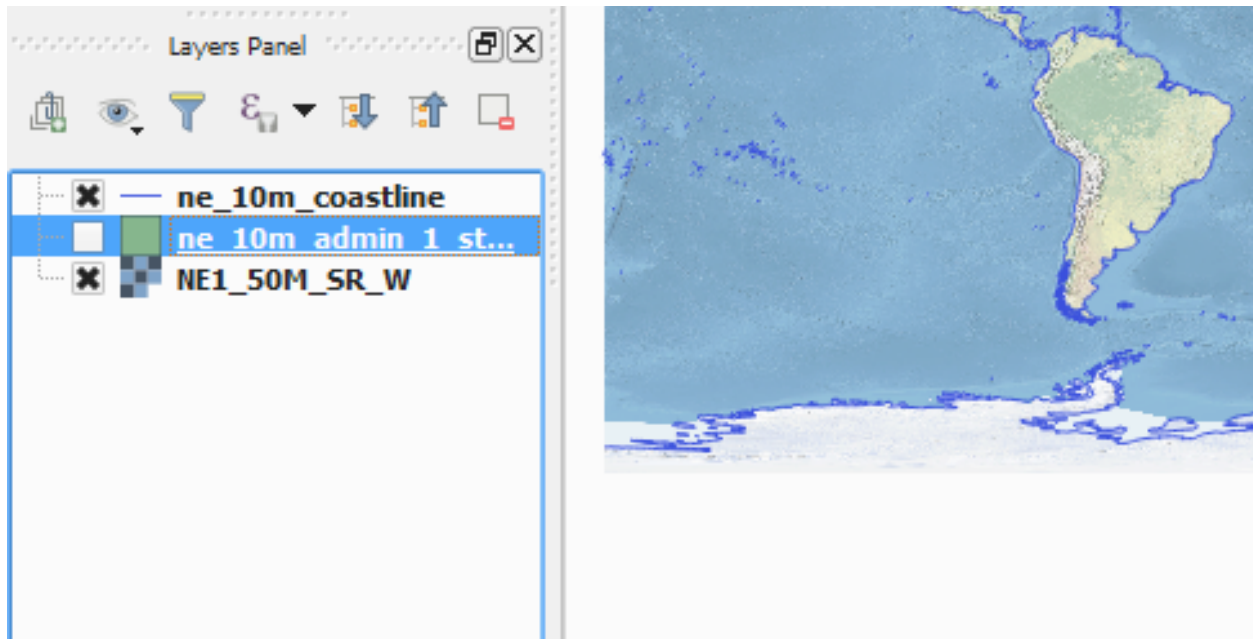
Fig. 6.9: Changing the layers' visibility

Style tab. There you can change how the layer will be displayed in the map. Click the *Simple fill* in the symbols layers list, and in the Fill *Fill type* select `No brush`. Press *Ok* to apply the changes and close the properties dialog.

At this time you might want to save your project.

5. Go to *Project → Save* or hit Ctrl+S. Choose the destination folder where your project will be saved, type in a sugestive name and click *Save*.

Exploring data's attributes

To make proper use of the dataset, one should know its attributes. Let's see how to retrieve the attributes of our layers.

1. Make sure the `ne_10m_admin_1_states_provinces` layer is still active and in the *Attributes toolbar* (if not visible, go to *View → Toolbars*), select the *Identify tool*. Then, click the map over one of the geometries of the layer. The *Identify Results* Panel will show up, where you can see the feature's fields and respective values. (You may need to expand the panel a bit to see it all).

Seeing layer's attributes using the identify tool in a feature

2. You can also see all attributes of your layer in its attributes table. Having the `ne_10m_admin_1_states_provinces` layer selected, click the *Open Attributes table* in the *Attributes toolbar* (or right-click the layer's name in the *Layers Panel* and choose *Open Attribute Table*). The layer's attribute table will show up.

3. In the attribute table, use the mouse wheel to quickly scroll up and down the attributes, or the scroll bar to move horizontally.

4. Select one feature by clicking its id number at the left side of the feature's row of attributes. Then, use the *Zoom to Selected Rows* tool at the top of the attribute table to zoom the map to that particular layer.

5. Repeat step 4 selecting several rows by holding the `Ctrl` key while clicking the id numbers. In the end, make sure to deselect all features using the *Deselect All* button in the attribute table.
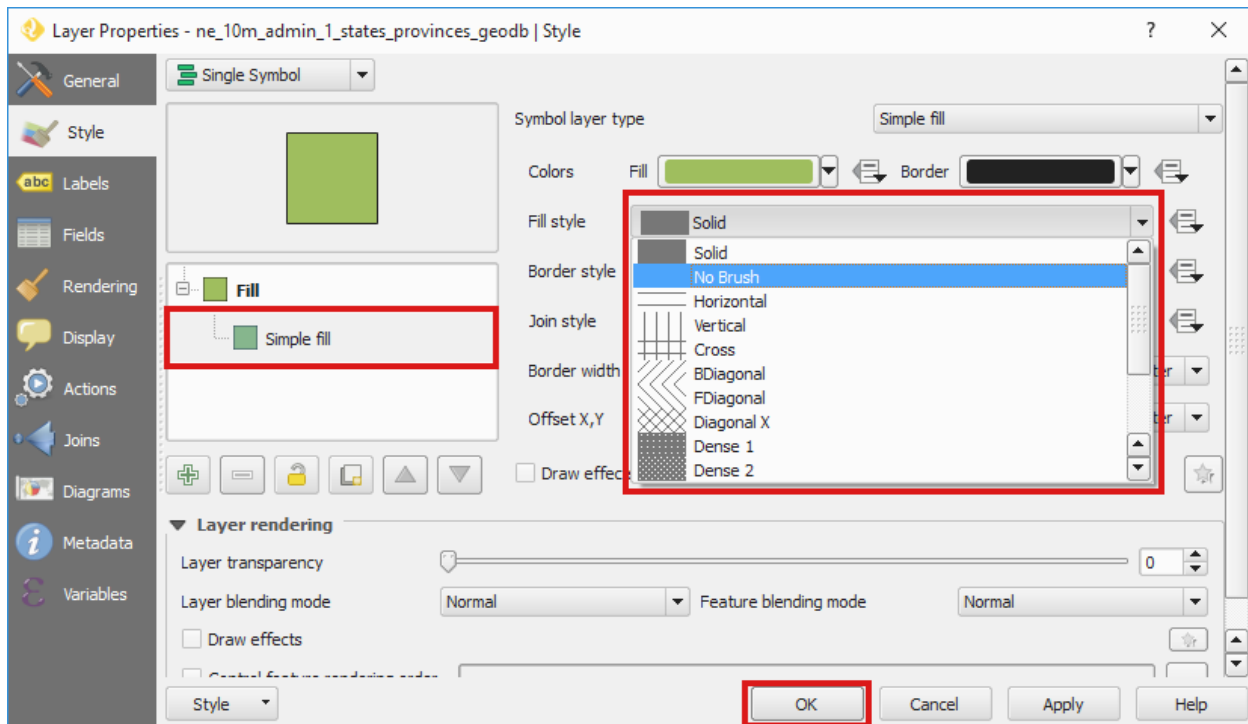
Fig. 6.10: Changing the layers' style in the properties dialog

Add simple labels

Now that we already know our data attributes, let's use one as a label for our geometries.

1. Go back to the `ne_10m_admin_1_states_provinces` properties menu by double-clicking its name in the *Layers panel*. Go to the Labels tab, and select `Show labels for this layer`. Then, in the *Label with* combobox select the `abbrev` field. Press *Ok* to apply the changes, close the properties dialog and see how it looks.

Print a simple map

Now let's see how to print a very simple map with the layers that we have loaded. In QGIS, you can have as many map layouts (a.k.a. print compositions) as you like, and you can manage them in the *Print Composer Manager*.

1. Once you are satisfied with the map's looks, click the *New Print Composer* button in the *File toolbar*, type a representative name for the composer and click *Ok*.

2. The print composer will open with an empty page. To add a map item, click the *Add Map* in the *Toolbox* toolbar and draw a rectangle covering most of the page by clicking and dragging over it. The map content should appear.

3. You can adjust the map item position and size by clicking and dragging the corner and side handles.

4. You can also adjust the map extent using the *Move item content* tool. While this tool is selected, you can pan the map content clicking and dragging inside of it, and change its scale using the mouse wheel. More precise controls to set the map item position, size, scale and extent can be found in the *Item properties tab/panel*.

5. Now that we are satisfied with our very minimalist map, let's export it. In the *Composer* toolbar, click *Export to PDF*. Choose a location and name for your PDF file and click *Ok*.
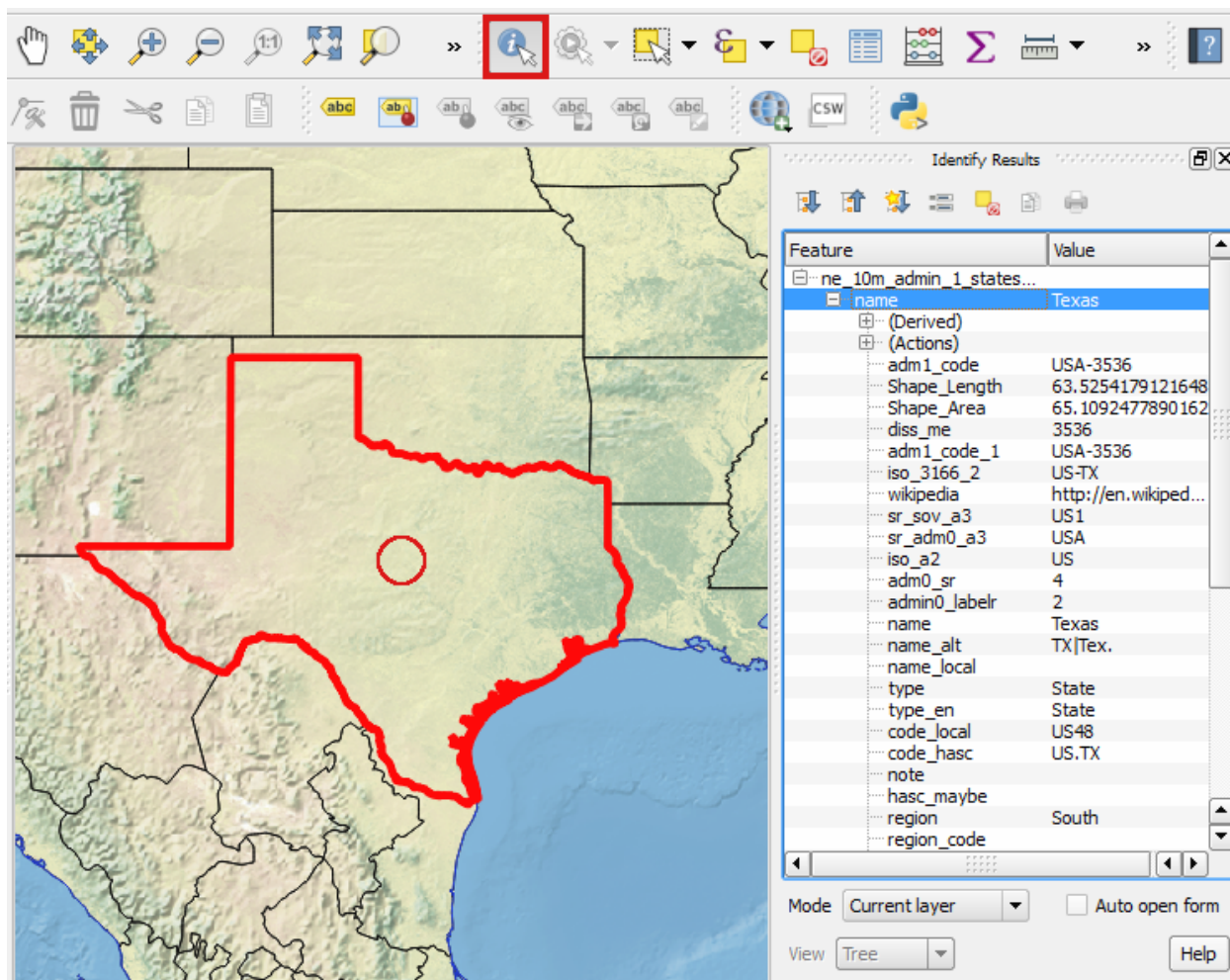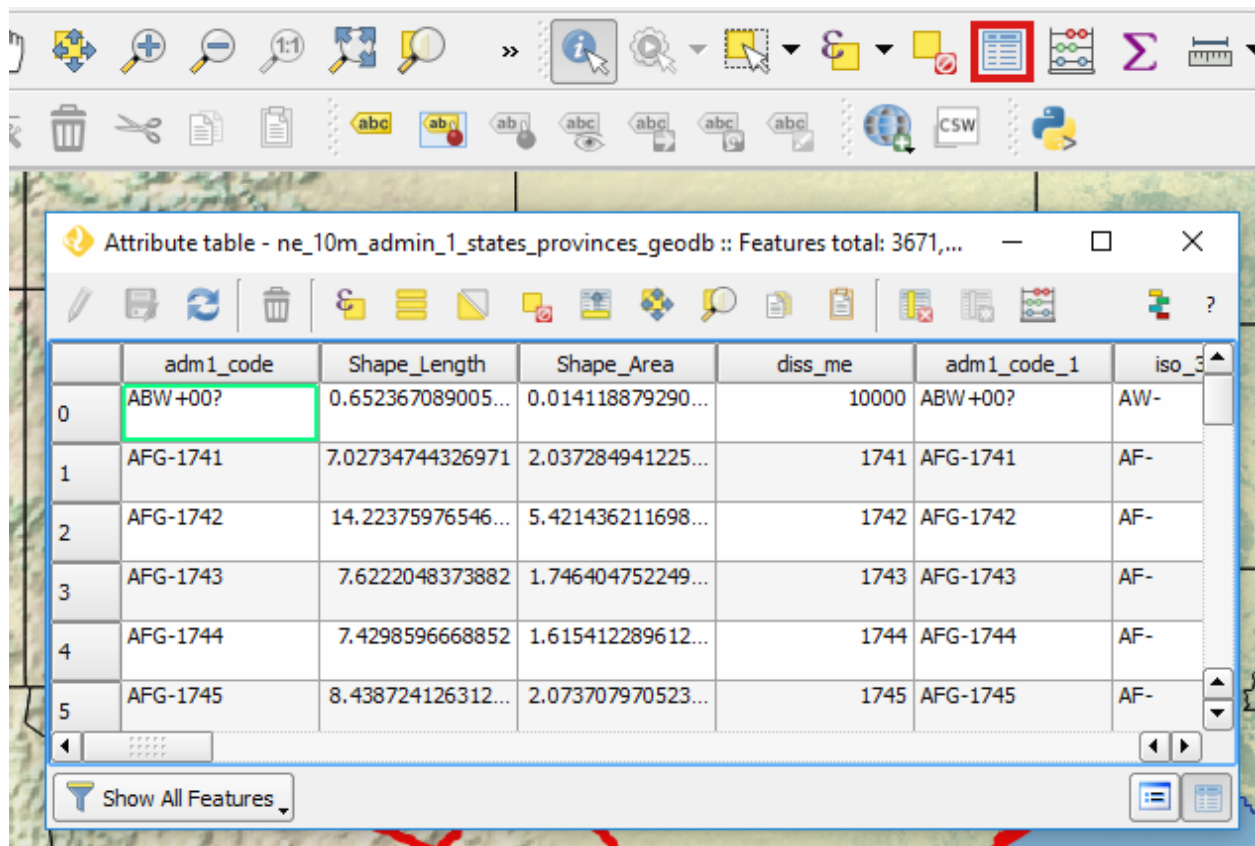
Fig. 6.11: Seeing layer's full attributes using the attribute table
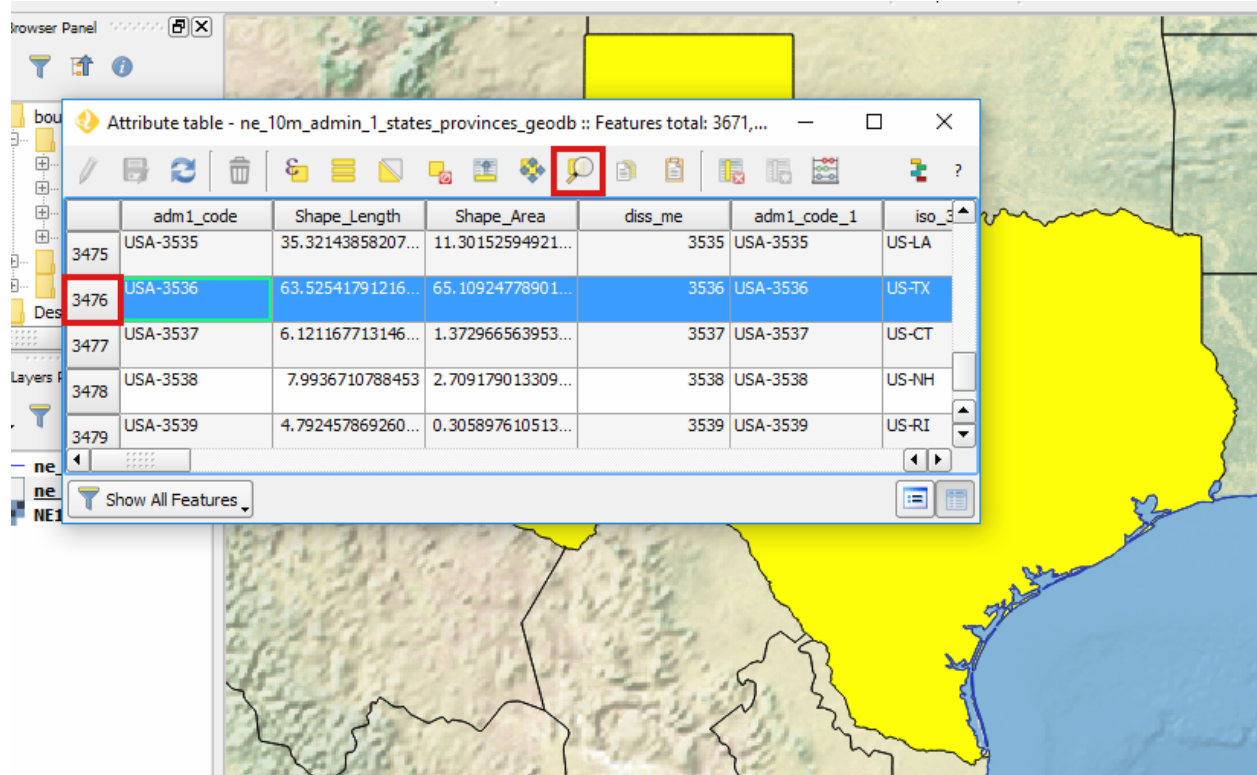
Fig. 6.12: Selecting a row in the attribute table and zooming to it's feature
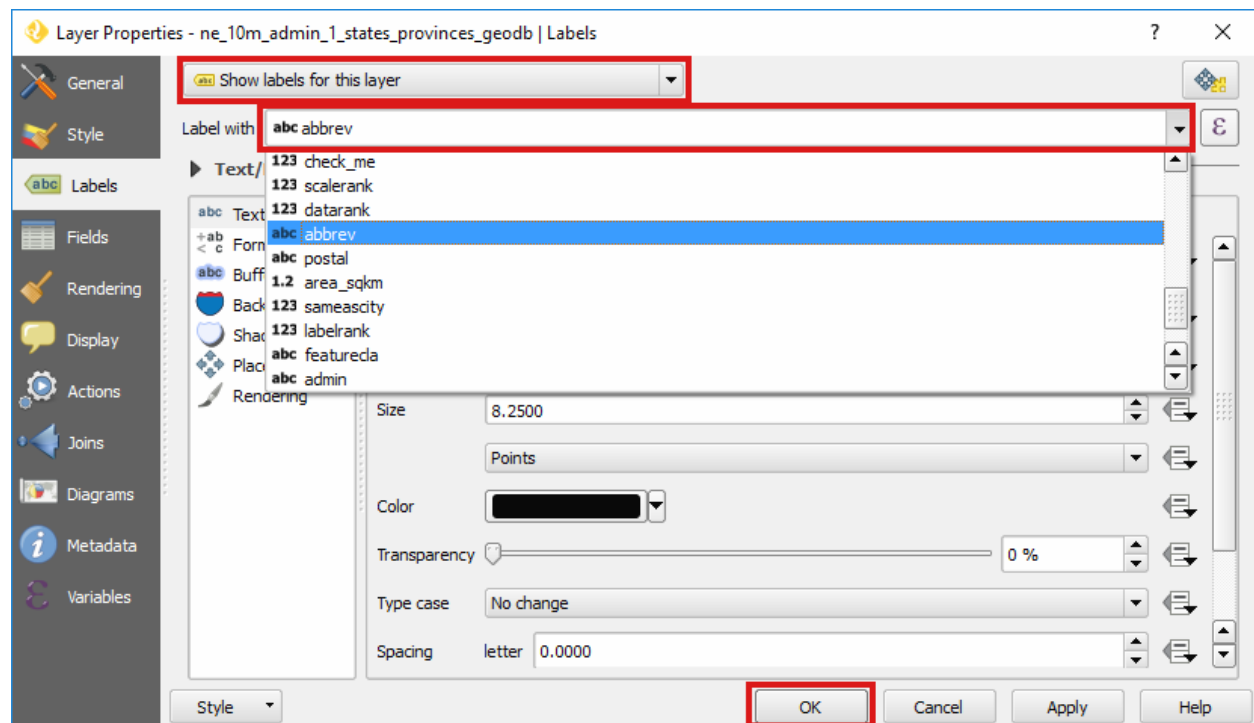
Fig. 6.13: Layer's properties Label tab
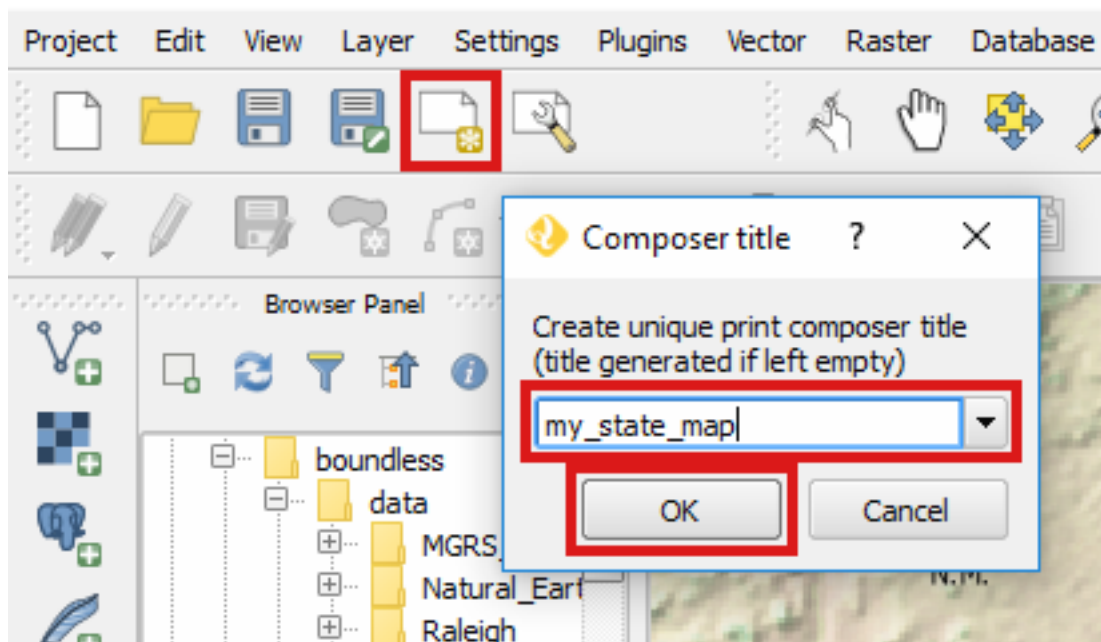
Fig. 6.14: Creating a new composer and choosing a name
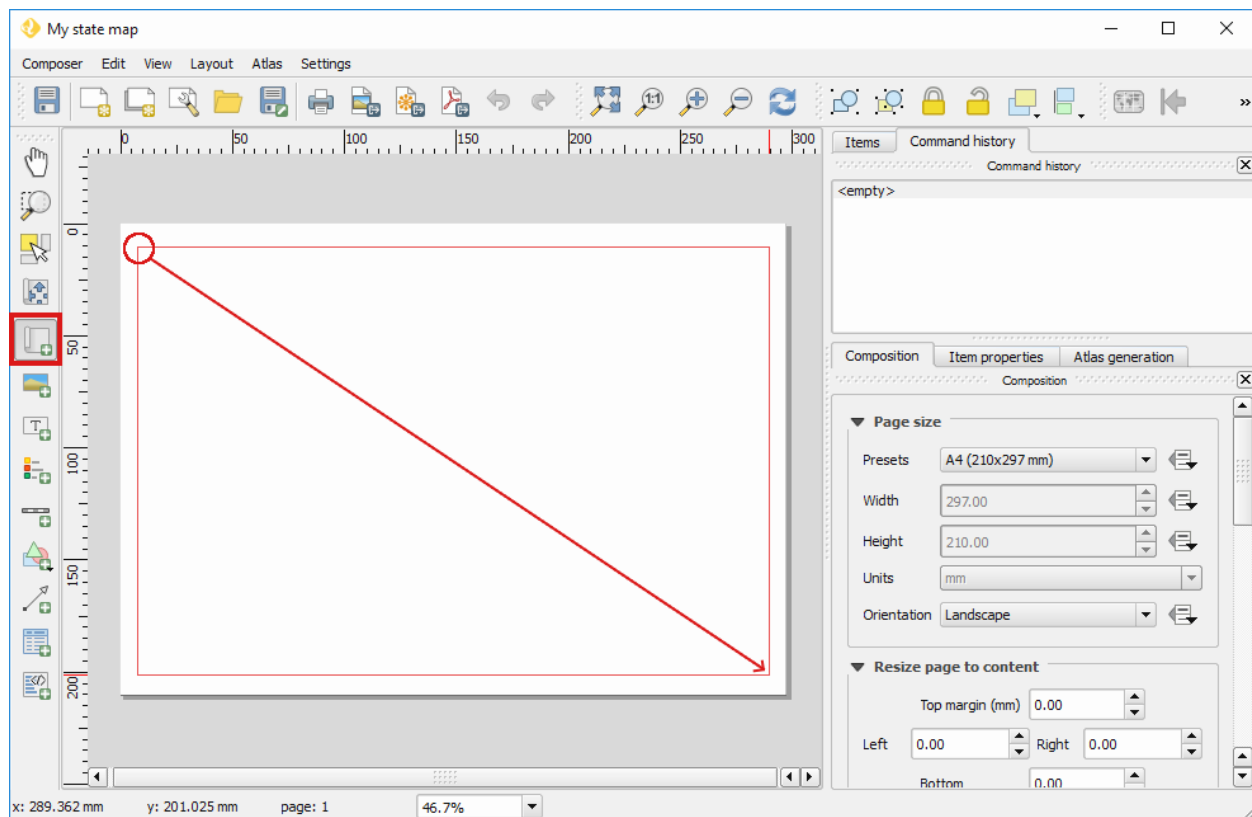
Fig. 6.15: Adding a map item to the print composer page
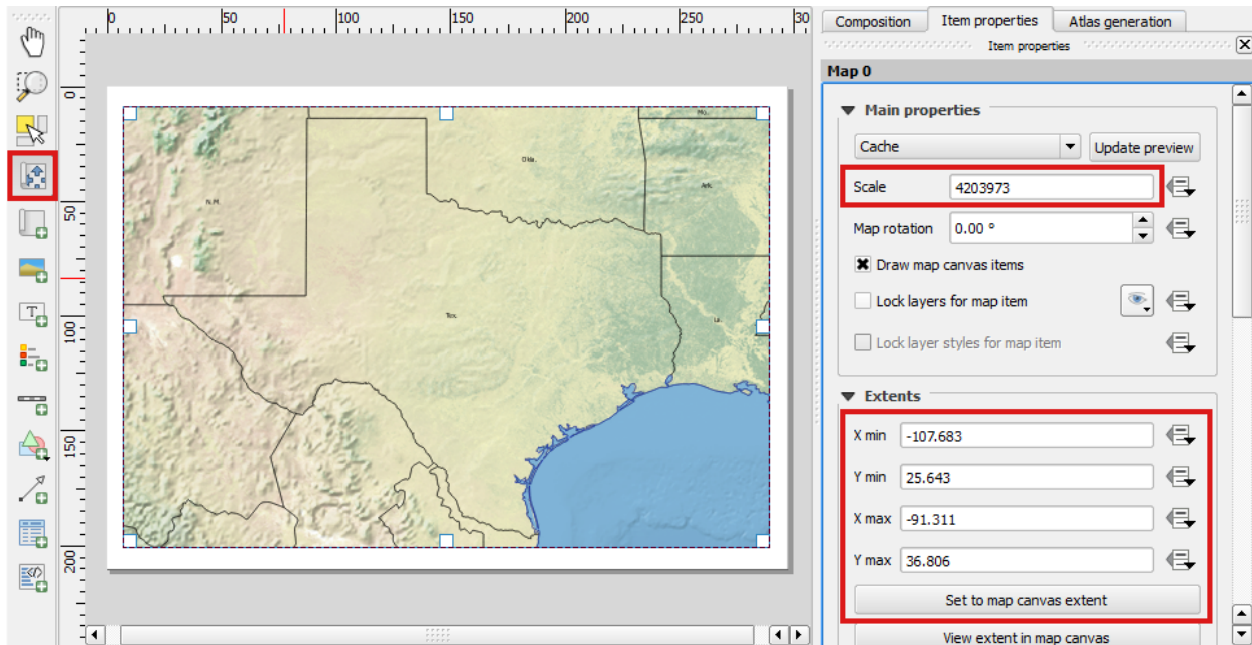
Fig. 6.16: Adjusting map item's scale and extent

Obviously, we could do more complex maps by adding other items like legends, labels and images. Please see our learning centre to learn how to work with them. Also, if you have interested, have a look into this QGIS Map Gallery.

QGIS Browser

Alongside with QGIS you will find QGIS Browser, another QGIS standalone application in the Boundless Desktop folder. QGIS Browser can be used to browse the datasets quickly on your local computer, network or remote services. You can see its metadata, preview its geometries and see the attribute table.

Online resources

- Official Site: http://www.qgis.org
- Documentation: http://docs.qgis.org/2.18/en/docs/index.html
- Official Plugins Repository: http://plugins.qgis.org/plugins/

pgAdmin

About

pgAdmin is a feature rich Open Source administration and development platform for PostgreSQL, the most advanced Open Source database in the world, which includes, among others, the Postgis spatial extention.

pgAdmin is developed by a community of PostgreSQL experts around the world and is available in more than a dozen languages. It is Free Software released under the PostgreSQL License.

pgAdmin is aimed to answer the needs of all users, from writing simple SQL queries to developing complex databases. The graphical interface supports all PostgreSQL features and makes its administration easy. The application includes,
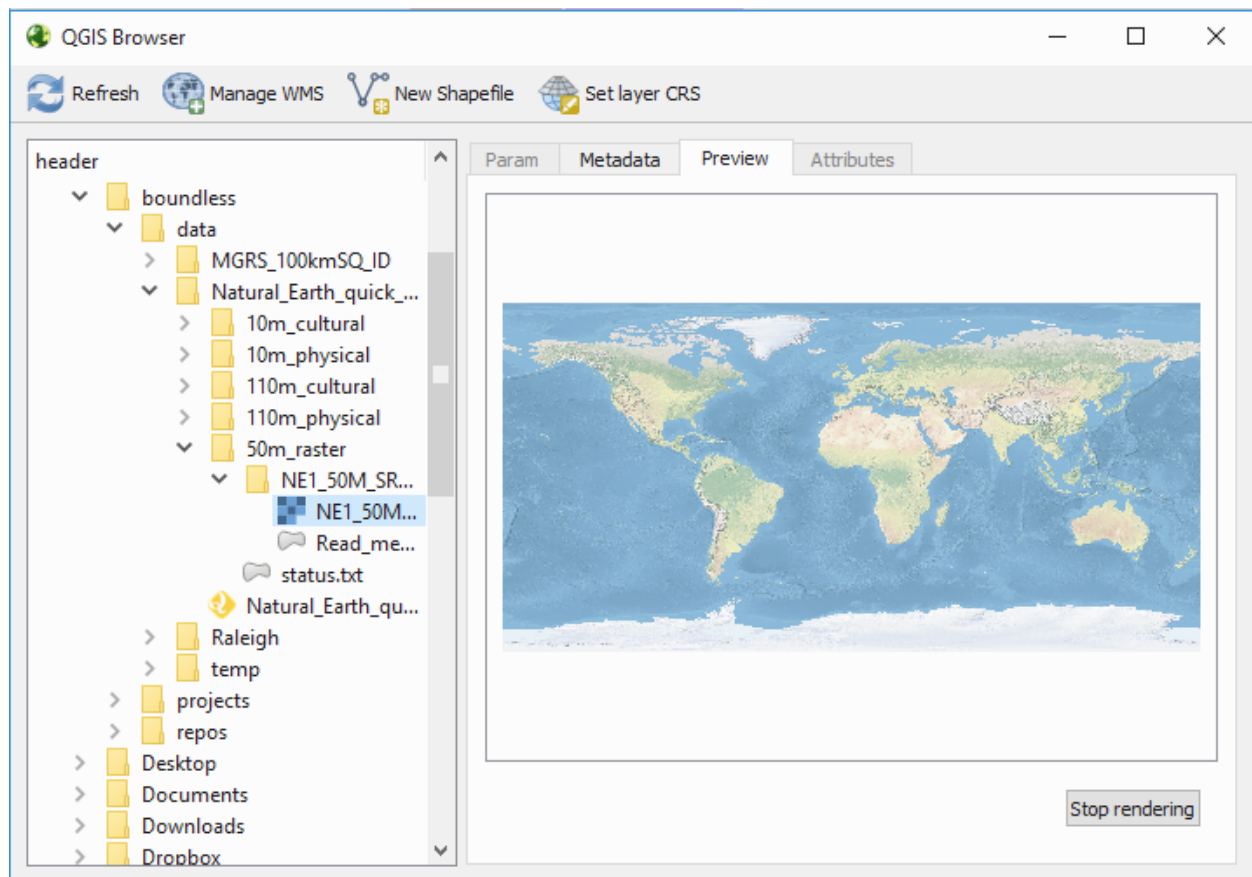
Fig. 6.17: Standalone QGIS browser GUI

among other things: syntax highlighting SQL editor, a server-side code editor, an SQL/batch/shell job scheduling agent, and much more. See more about it in the official website.

Quick start guide

Note: To execute this Quick Start Guide you will need a running instance of **PostgreSQL** database with the spatial extension **PostGIS** installed, please ask your system administrator for the connection parameters. For further information about **PostgreSQL** or **PostGIS** see *Online resources*.

The following quick start guide will introduce you to **pgAdmin** basics. It will show how to access to a spatially enabled database and run some simple queries on its tables.

1. Download the data for this tutorial `here` and unzip it anywhere you think it's convenient.

2. Import the geodata from the shapefile into the database, if you have **shp2pgsql** command line utility at hand you can import the data with:

   ```
   shp2pgsql events.shp | psql -U postgres pg_test
   ```

   If you don't have this command available you can still use **QGIS** *db manager* to import you data into the database.

#. Open **pgAdmin4** using any of the available shortcuts in your computer.

   By default, **pgAdmin4** will show the following window on opening:

1. On the *Browser* panel, right-click on the *Servers*, and navigate to *Create → Server...* to create a new database connection.

2. In the *Create - Server* dialog, enter the connection *Name*. Click the *Connection* tab, enter the connection parameters in the dialog and click *Save*.. figure:: img/

   A database connection is usually identified by:

   - host

   - port

   - username

   - password

3. Click on the newly added server to open its objects tree and select the *events* table or any other table you want to query. Then, click the *Properties* tab.

4. Perform operations on selected tables with `Right-Click`

5. Preview data with *View Data → View First 100 Rows*

6. In the menus, click *Tools → Query*

7. In the query editor, type the following query:

   ```
   SELECT name, St_AsText( geom ) FROM events;
   ```

8. Then, click the *Execute/Refresh* button or press `F5`

Online resources

- Official Site: https://www.pgadmin.org/

- Documentation: https://www.pgadmin.org/docs/pgadmin4/1.x/

- **PostgreSQL** https://www.postgresql.org

- **PostGIS** http://postgis.net
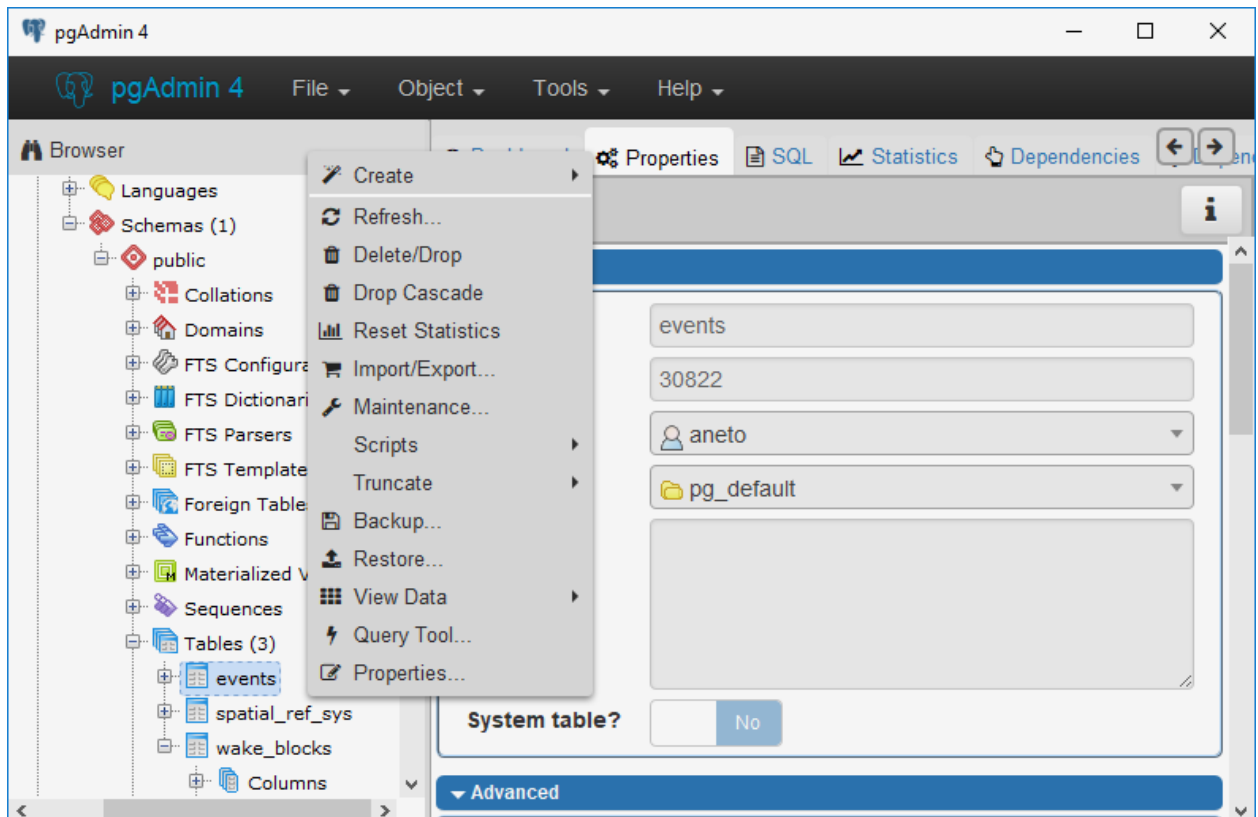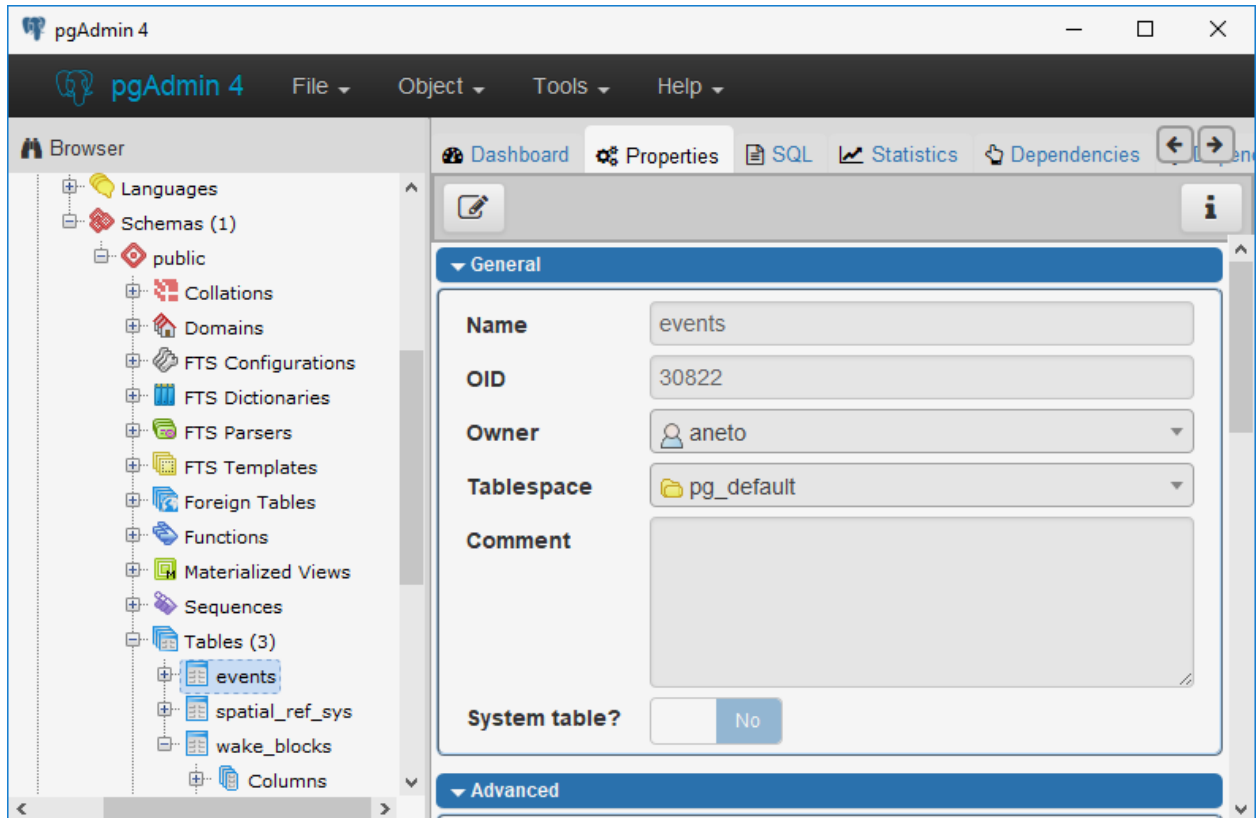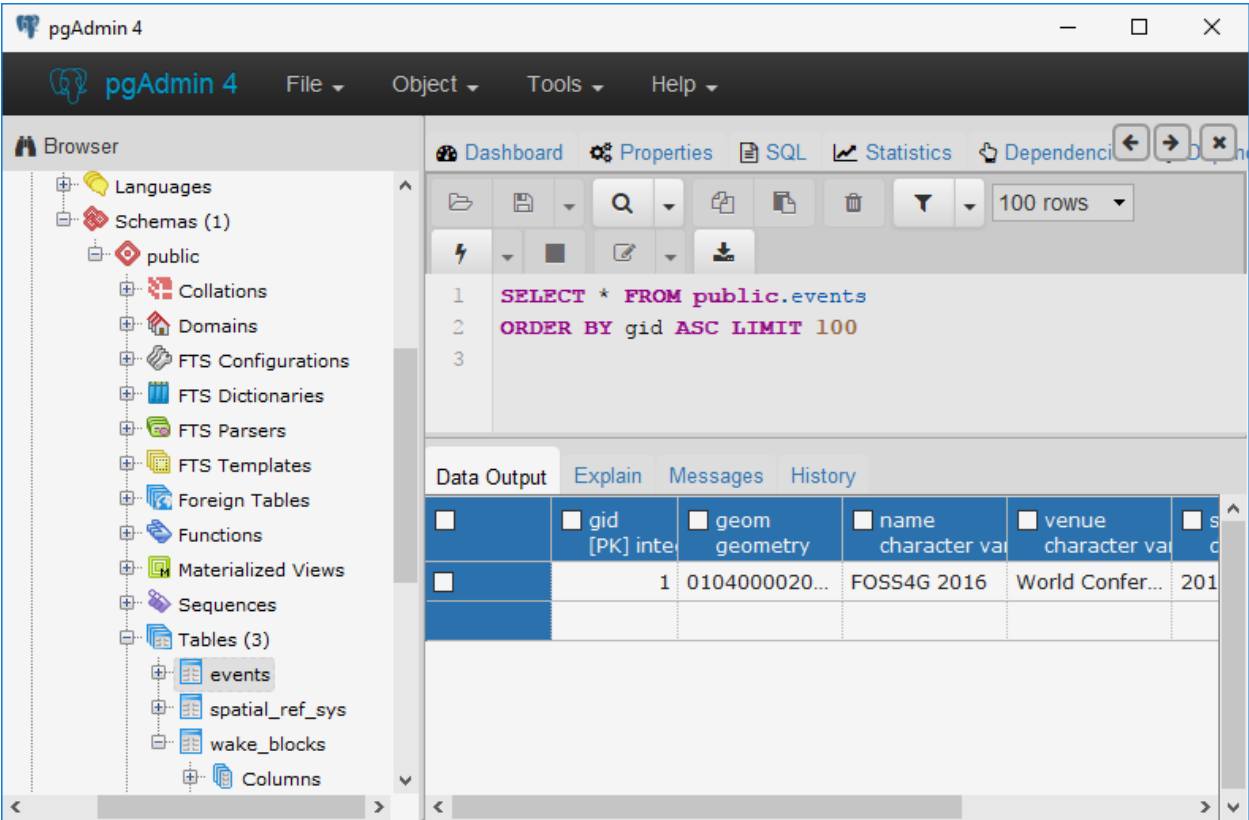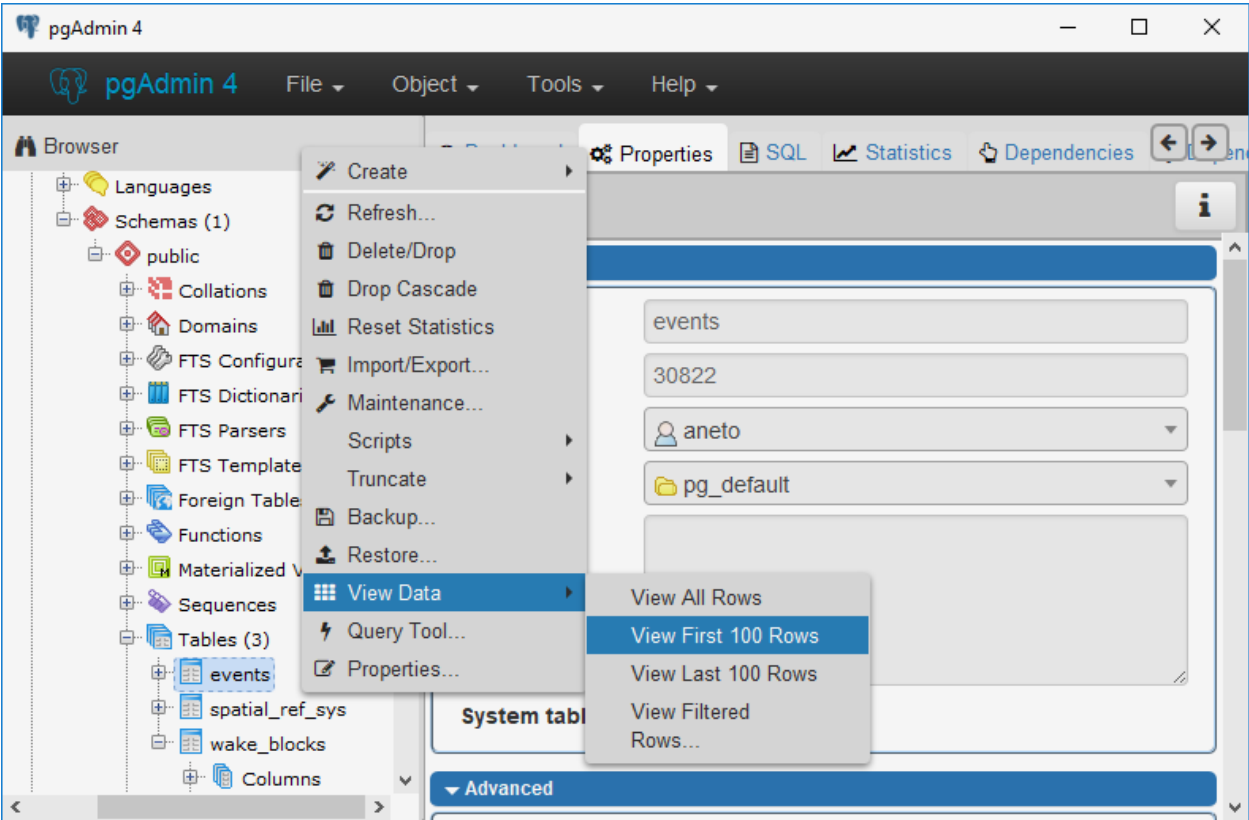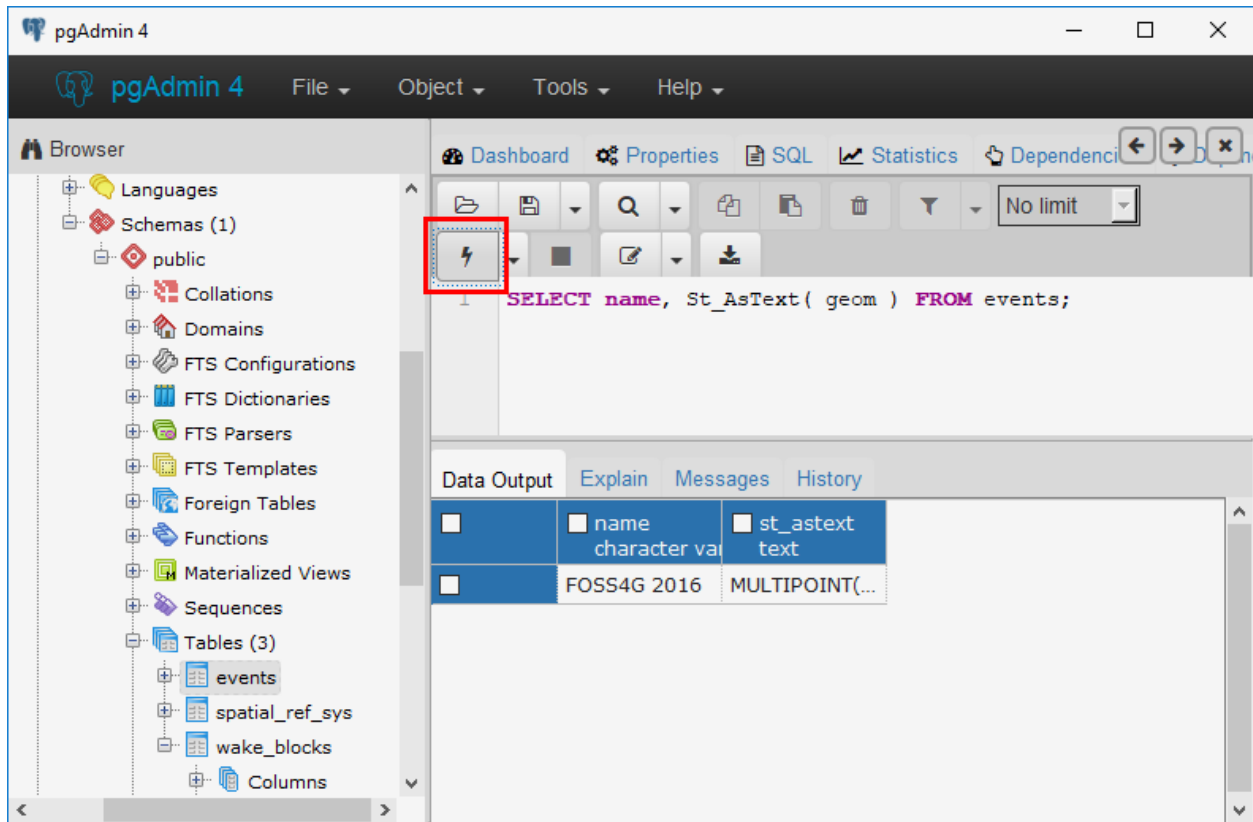
Qt Designer

About

Qt Designer is a Qt's Framework tool for designing and building graphical user interfaces (GUIs) from Qt components. With it, you can compose and customise widgets or dialogs in a what-you-see-is-what-you-get (WYSIWYG) manner, and also test them using different styles and resolutions.

The reason why *Boundless Desktop* includes *Qt Designer* is because it's extremely useful to create QGIS plugins' interface dialogs, but also to create highly customised feature forms for editing vector layers' attributes in QGIS projects. With that in mind, the shipped version brings a set of very convenient special widgets for QGIS (as an example, a combobox with all project's loaded layers).

Qt Designer is created by the same team that delivers the Qt Framework. See more information about it in Qt official website

Quick start guide

The following quick start guide will introduce you to Qt Designer basics. It will show how to create a simple custom feature form for a vector layer in QGIS.

1. Download the data for this tutorial `here`. Unzip it to whatever folder you think is most convenient.

2. Open *Qt Designer* using any of the available shortcuts in your computer.

   By default, Qt Designer will show the *New Form* dialog on opening.

3. Choose `Dialog with Buttons Bottom` from the *template\forms* list and press *Create*.
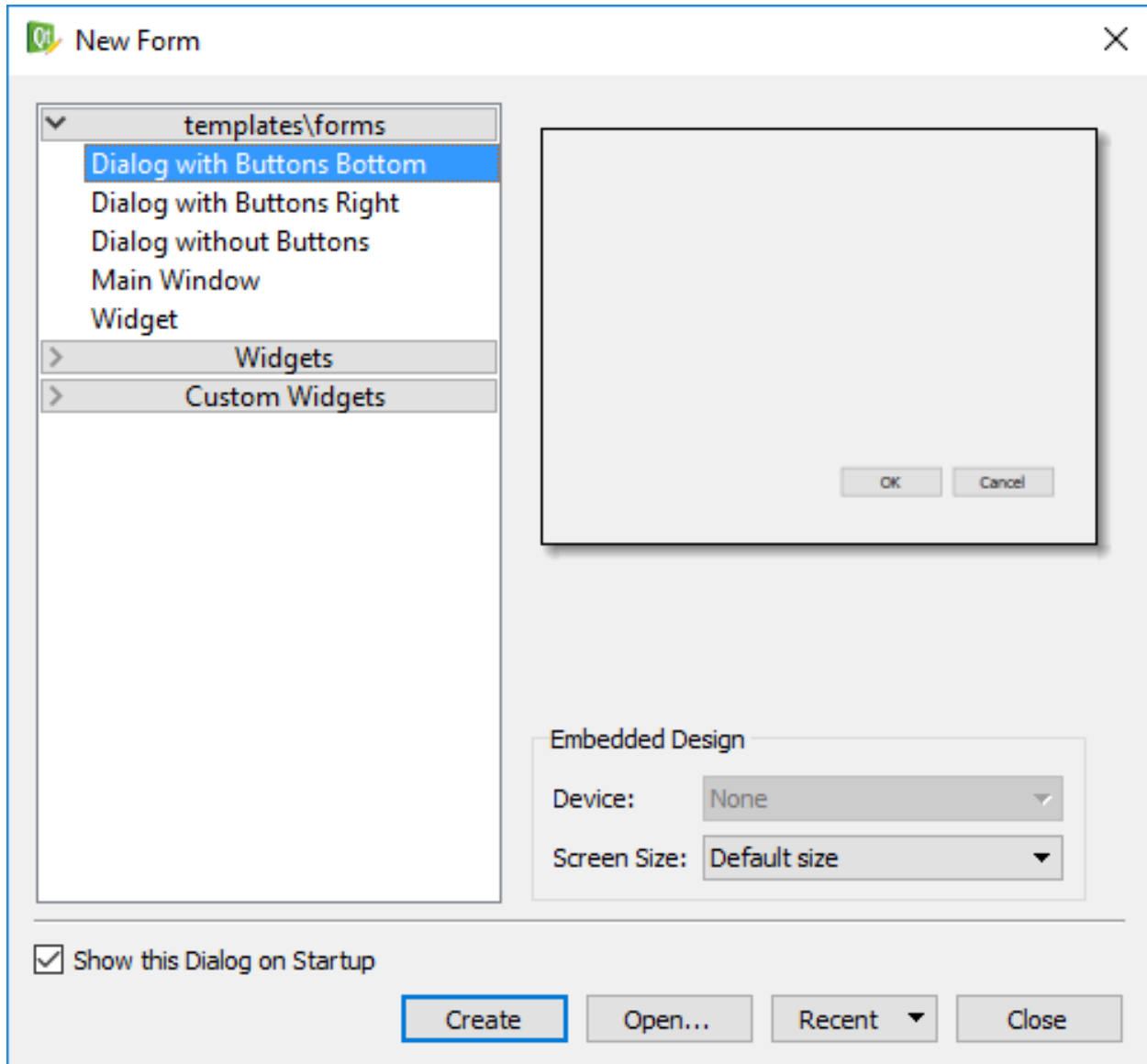
Fig. 6.18: New form dialog window

4. Have a quick look into the *Qt Designer*'s graphical user interface (GUI):

   - In the middle of the screen, you will find the recently created new dialog, a blank form with only a pair of ok-cancel buttons (1).
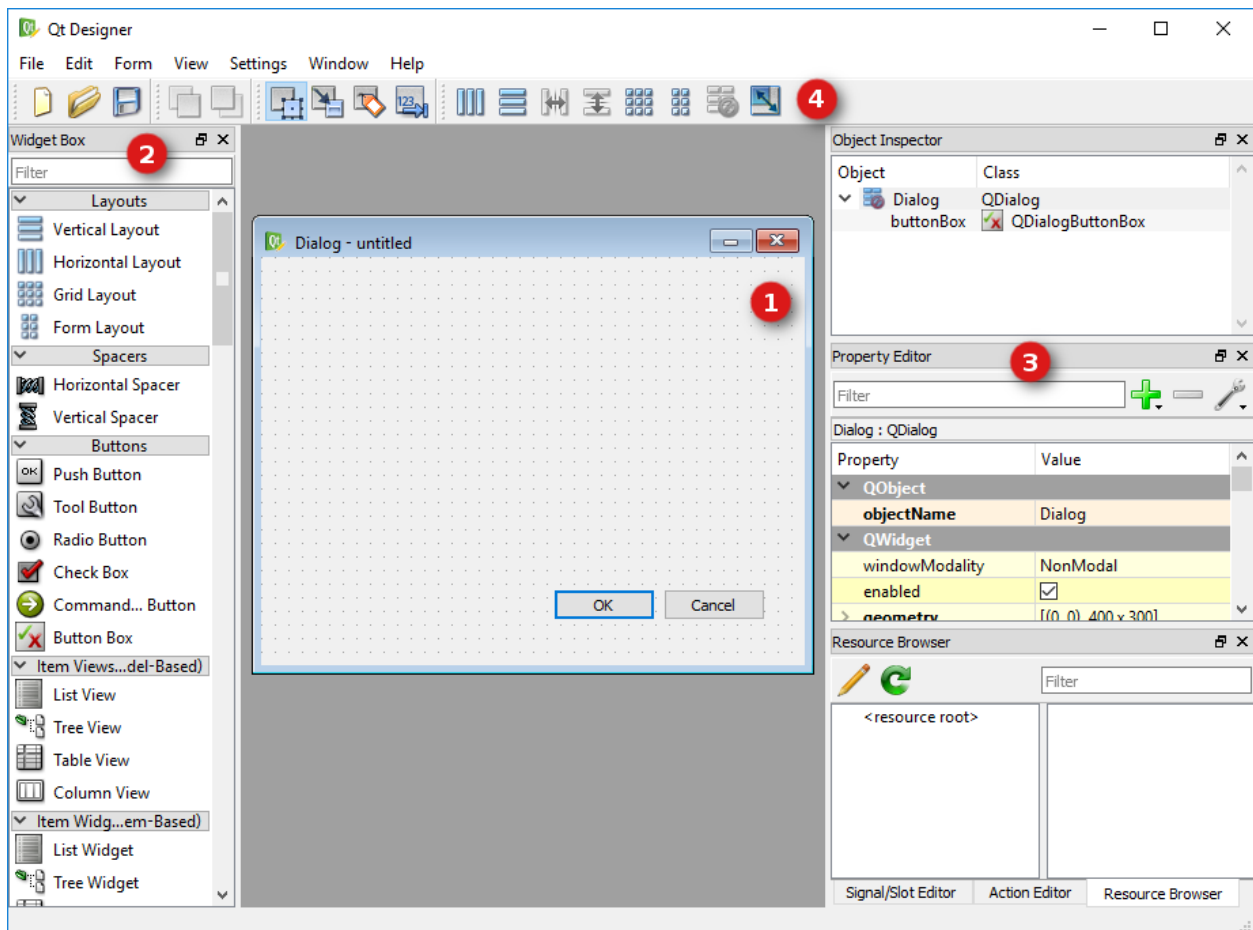
- On the right-hand side of the screen you will find the *Widget box* which lists all available widgets (2). This list can be used to insert widgets by simply dragging and dropping them on top of the dialog.

- In the left-hand side (among others) there's the *Property Editor* (3), that can be used to edit the properties of the dialog itself or of each inserted widgets.

- Finally, in the top, near the menus, you will find the *Forms* toolbar(4), which we will use to impose some alignment layout for consistency purposes.

Note: In Mac OS X, Qt Designer starts up in *Multiple Top-Level Windows mode* by default, looking very different from what is presented above. To change it, you should go to *Designer → Preferences...* and choose `Docked Window` in the *User Interface Mode*.

To see what widgets we want to use on the new form, we will need to inspect the data's attributes and the layer's current feature form.

5. Open *QGIS* by using any of the available shortcuts in your computer.

6. Click the *Add Vector layers* button in the *Manage layers Toolbar* (or use the *Layer → Add Layer → Add Vector Layer...*) to browse to the `events.shp` shapefile location. (see QGIS *Quick start guide* if you need help with this task)
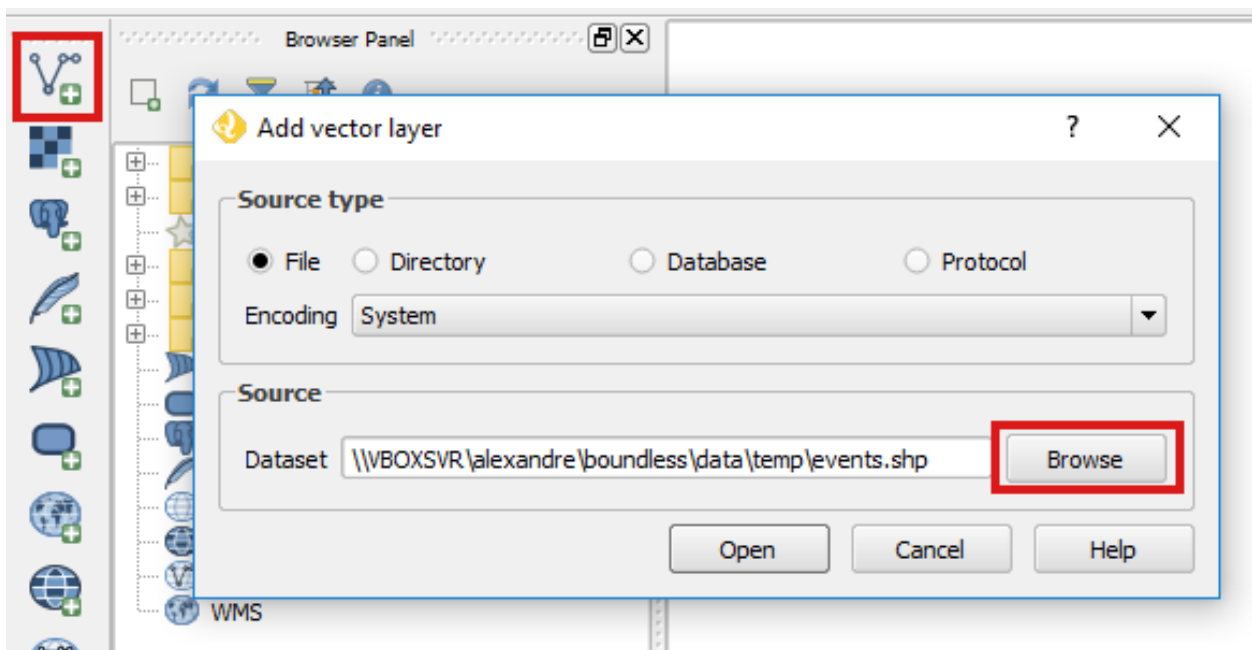
7. Right-click the recently loaded layer's name in the *Layers panel* and choose the option *Open attribute table*. Then, take some time noticing the layer's field names and types.

8. Let's now have a look at the layer's current feature form. Go back to QGIS main window, select the *events* layer and click *Toggle editing* in the *digitising toolbar* (if not visible check *View → Toolbars → Digitizing toolbar*). In the same toolbar, select the *Add Feature* tool and click anywhere in the map canvas to add a new point to your layer. This step should insert a point in the map canvas and show the default layer's feature form. Take some time analysing the feature form and then press *Cancel* to discard any changes.

Let's go back to *Qt Designer* and add the necessary widgets for the first field: *name*.

9. We will start by adding a text label to our form. Scroll down the *Widget box* to locate the *Label* widget in the *Display widgets* group. Now, drag and drop it on the form. You can change the label's text either by double-

---
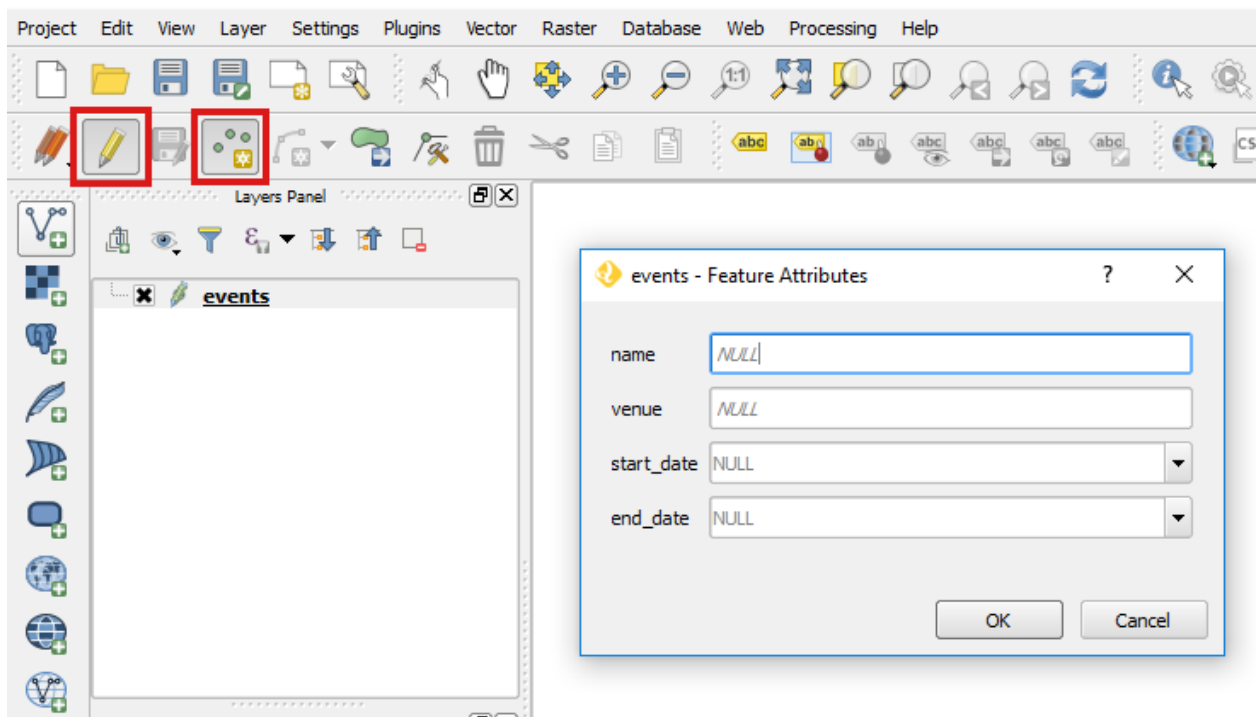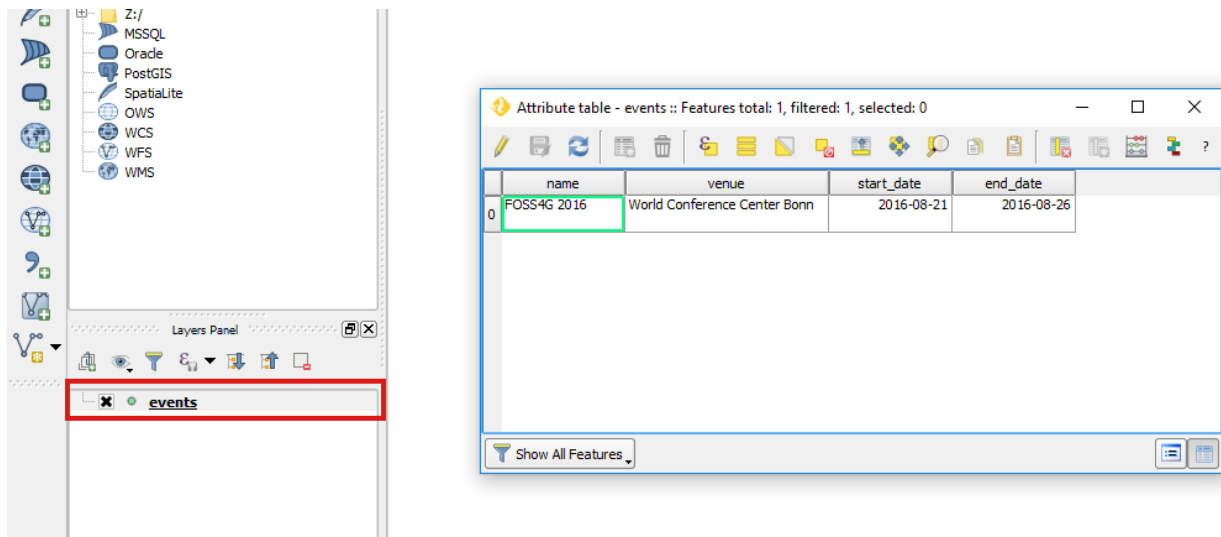
clicking label itself and typing or using the *Property Editor*, more precisely at the *Text* property. Set the value to `Event's name`. If necessary, expand the label box using the small squares in its border.
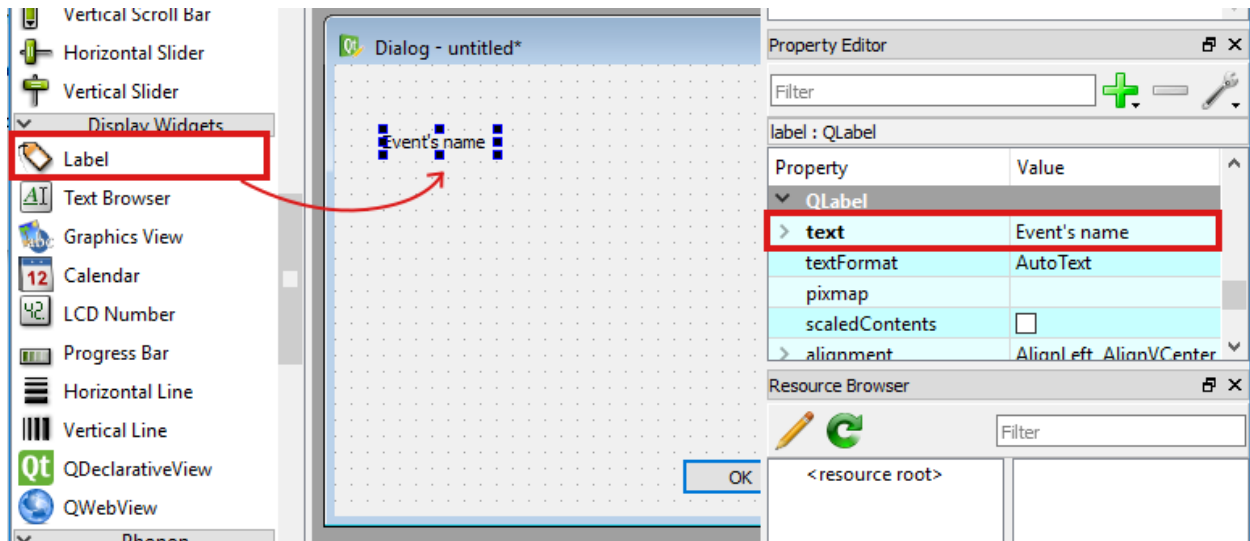
10. Now let's add an input widget for the *name* field. This time, instead of scrolling down the *Widget box* to locate the widget, try typing `line edit` on the filter bar at the top of it. Drag and drop a *Line edit* widget to be right below of the label.
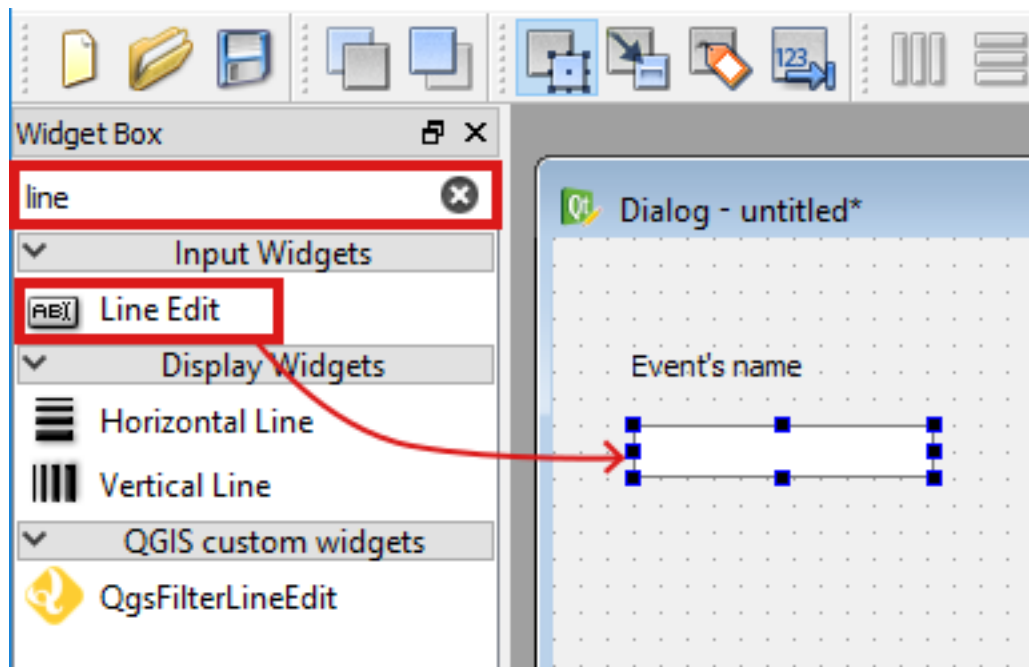
11. Repeat steps 7 and 8, for the *venue* field (or just select both existing widgets pressing the `Ctrl` key, and copy (`Ctrl+c`) and paste (`Ctrl+v`). Use `Event's venue` for the label, and put both *Label* and *Line edit* widgets below the other two.

12. Finally, let's add some special QGIS input widgets and respective labels to control the date fields. Using any of the two options described in steps 7 and 8, find and insert two *QgsDateTimeEdit* widgets. Put them side by

side, and above them add the labels *Start date* and *End date*. Your form should look similar to the one in the following figure:
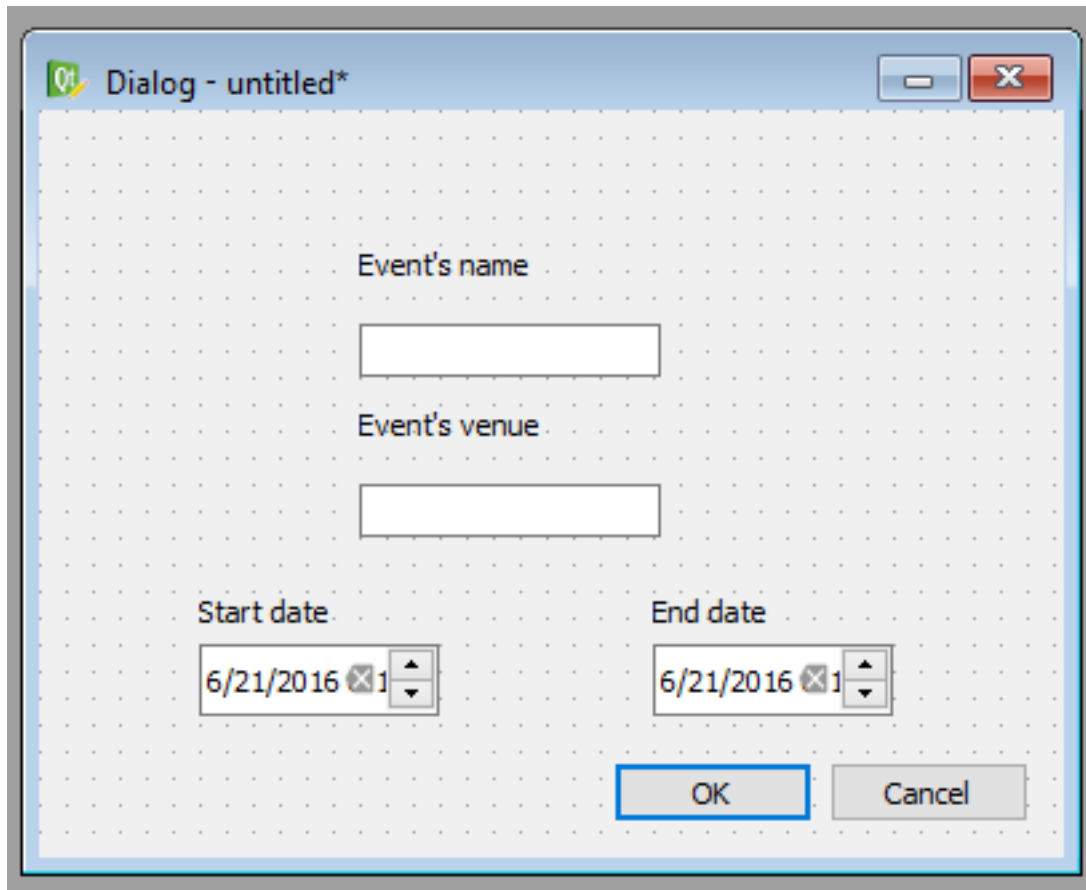
To connect this dialog's input widgets to the QGIS layer's fields, we need to set their *objectName* property's value with the same name as the target field. As you will see, this can be done in several different ways.

1. Select the *Name of event Line edit* and in the *Property Editor* set the value of the *objectName* property to `name`.

2. Right-click the *Venue of the event Line edit*, and choose *Change objectName*. Set the value to have the name of the second field: `venue`.

3. For *QgsDateTimeEdit* widgets simply use the *Object Inspector* located above the *Property Editor*. Double-click on their current names and change them to `start_date` and `end_date`

   By now, all your input widgets should have the name of the corresponding fields, and in the *Object Inspector* you should have something similar to the next figure.

   We are almost finished. All we need to do now is align all widgets so that they look good together. For that, we will use a grid layout.

4. Right-click anywhere on the dialog's empty space (this will switch any focus over the widgets to the dialog instead) and choose *layout → Layout in a grid* (or use the toolbar). After that, all widgets should be aligned according to their relative place to each others.

   After you have applied the grid layout, you can now resize the form to make it more compact, by click and move the bottom-left-corner of the form.

5. Finally, using *File → Save as..* let's save the form using a suggestive name, for instance `events_form.ui`. For easy discovery, save it in the same folder that the data is located.
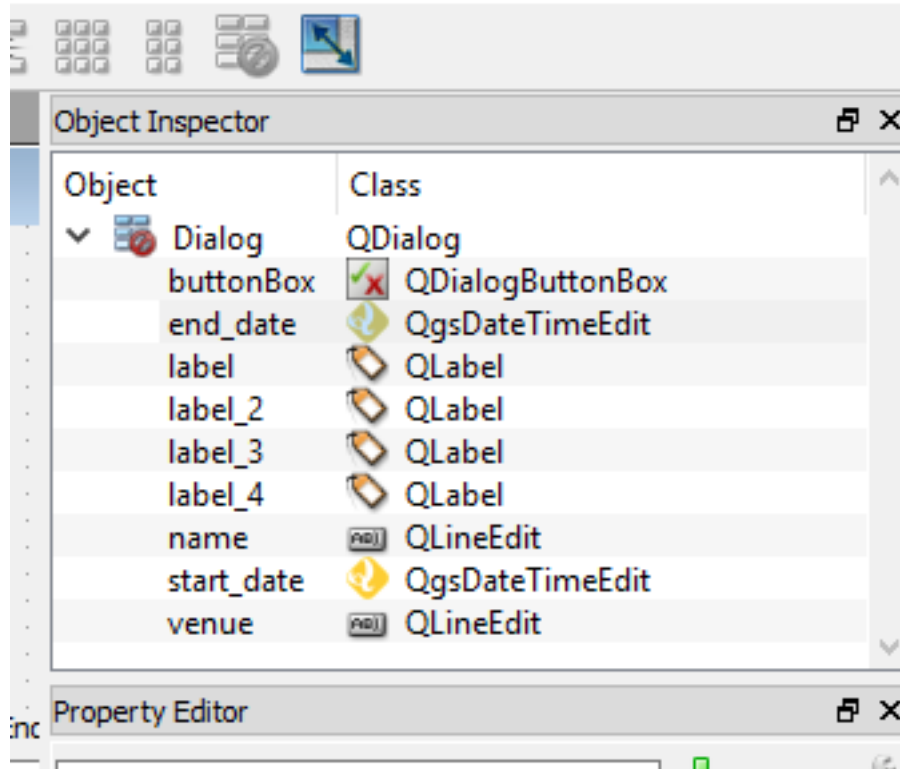
Now, let's go back to QGIS and test the feature form in action.

6. In QGIS, right-click the layer's name in the *Layers panel* and choose *Properties*. Go to the *Fields* tab and in the *Attribute editor layout* select the `Provide ui-file` option. Then, in the *Edit UI* browse to the *events_form.ui* location. Finally, press *Ok* to apply the changes and close the layer's properties dialog.

7. Assuming that the *events* layer is still in editing mode, let's use the *Add feature* again to create a new point in the map canvas. This time, it should open the customised feature form created in *Qt designer* with all its widgets.

   To finish our feature form, let's make a small fix in the date fields otherwise the values won't be saved in the table because of the use of different date formats in Qt form and QGIS.

8. Back in *Qt Designer*, select the two *QgsDateTimeEdit* widgets (Hold the `Ctrl` key while clicking) and, in the *Properties Editor* set the *Display format* value to `yyyy-M-dd` and check the *calendarPopup* property. The changes will be applied on both widgets. Save the form again.

9. Back in QGIS, go to the fields tab in the Properties menu and for each date field change the widget to date/time. Make sure to use the same date format value in the *Widget display* as the one used in the form (`yyyy-MM-dd`). Also select the *calendar popup* and *allow NULL* options. Press *Ok* when you have finished.

10. Now, repeat step #17 to see the feature form changes take effect. The date values should be represented differently and a new icon will alow to choose the date from a popup calendar. Besides, saving the input to the table will now work fine.

A lot more form customizations can be done using *Qt Designer*. You can use many different widgets, organise the widgets in groups or tabs, and you can even add some Python logic to make your forms responsive.

Online resources

- Official Site: http://www.qt.io

- Documentation: http://doc.qt.io/qt-4.8/designer-manual.html
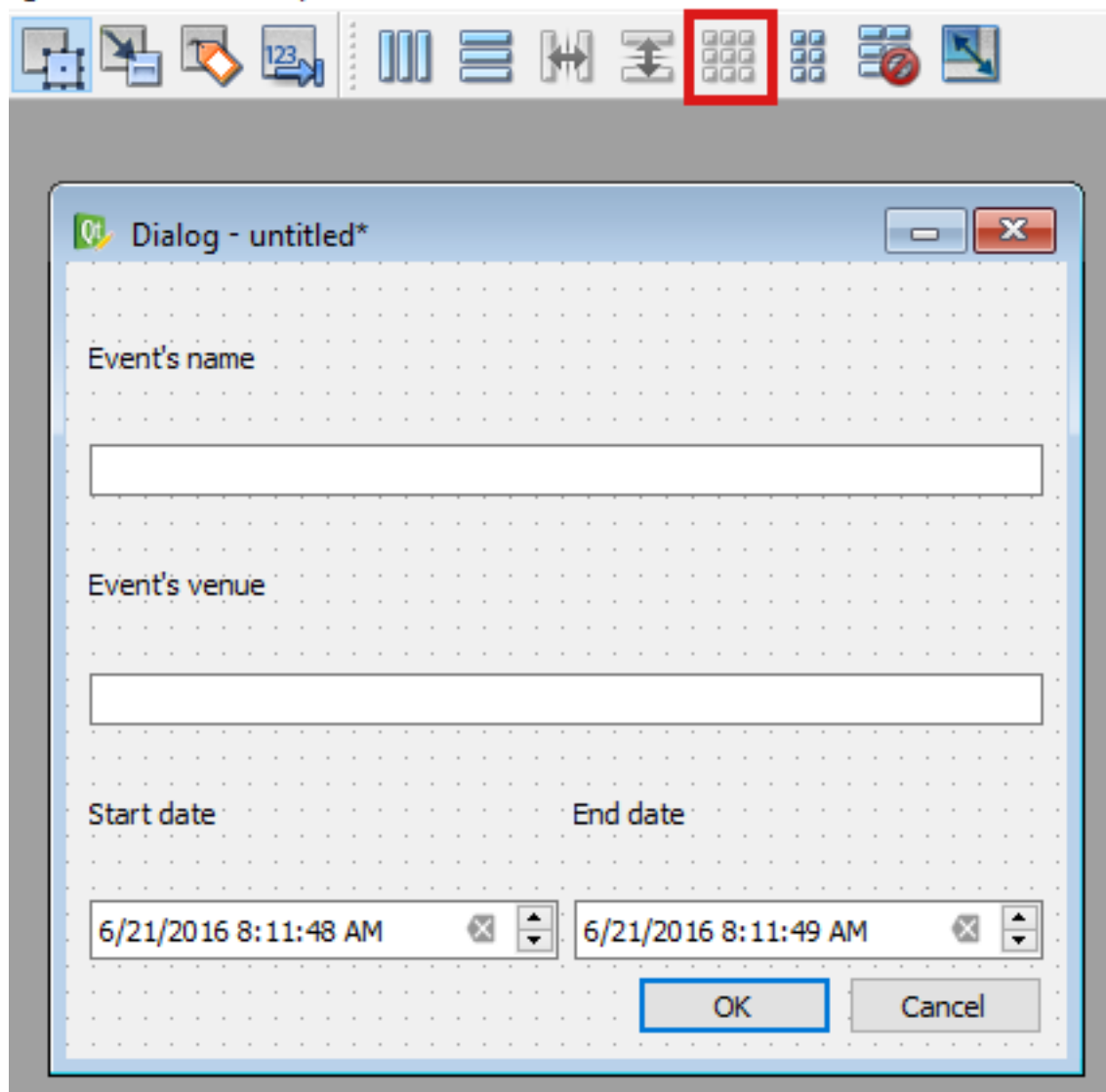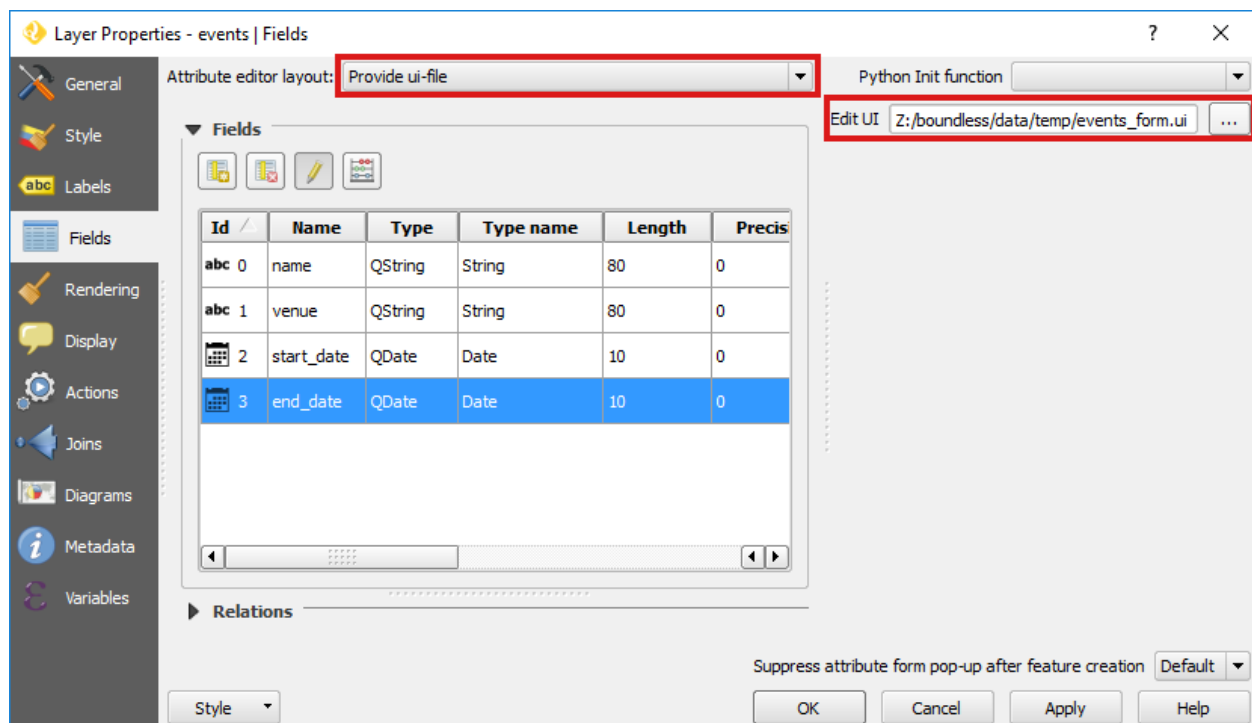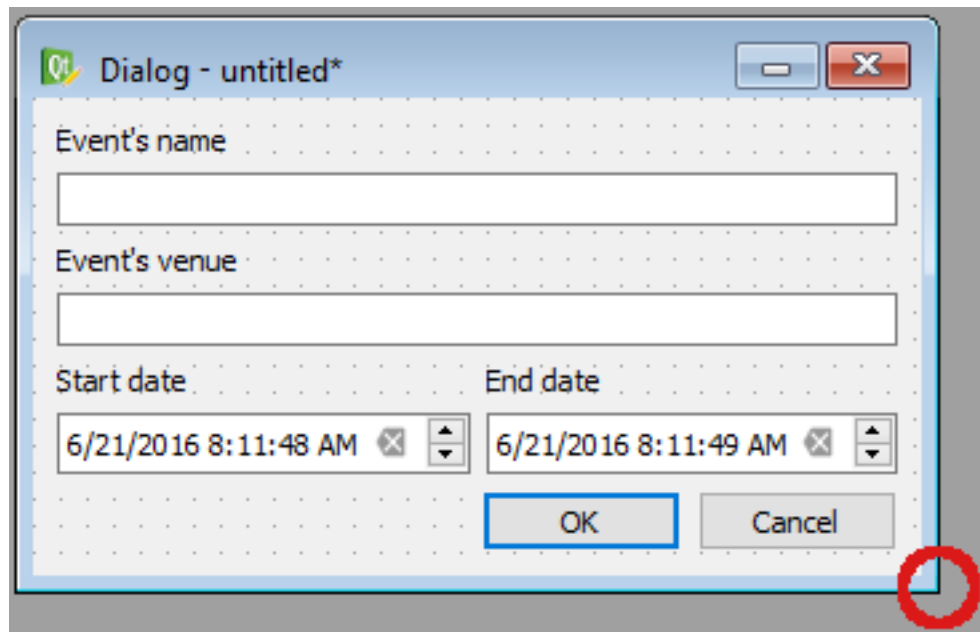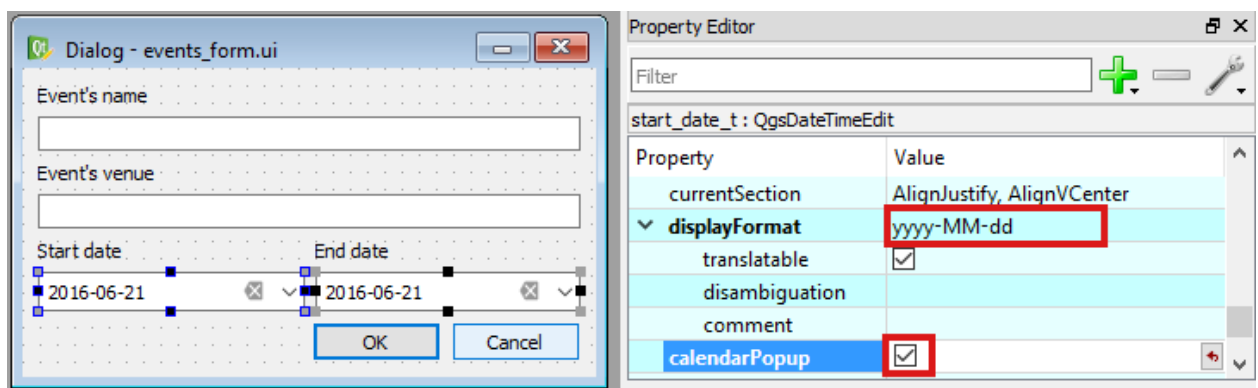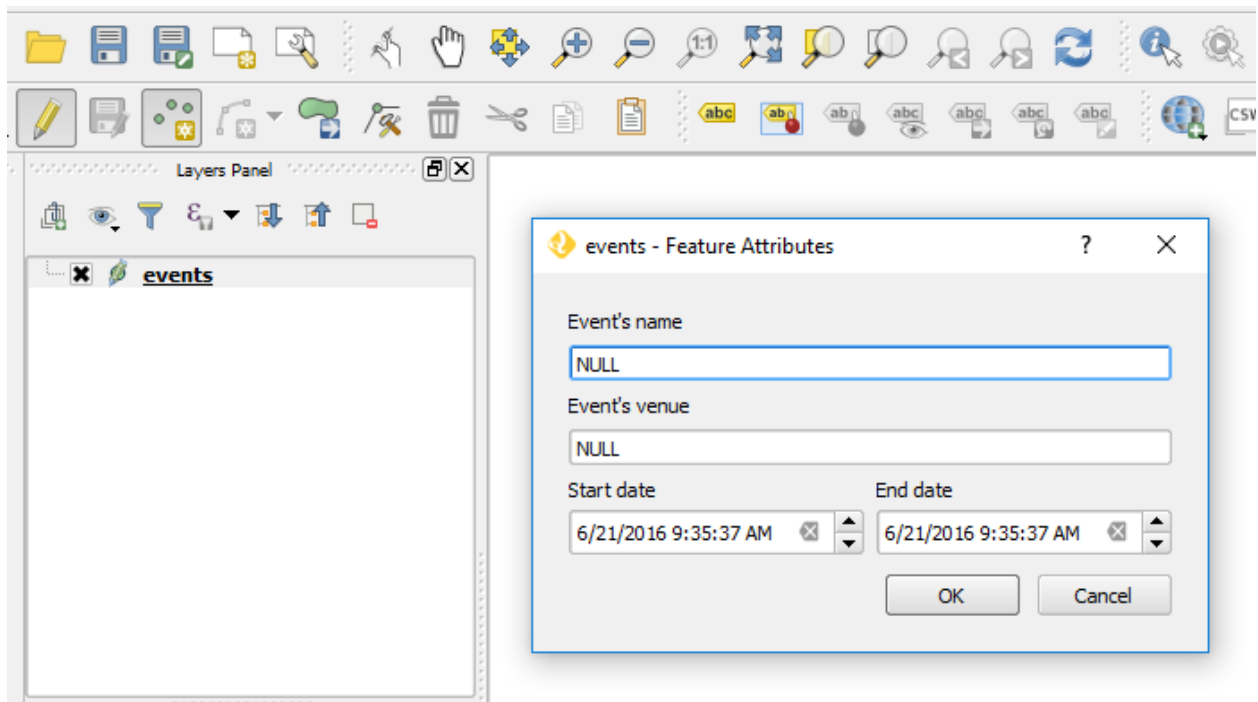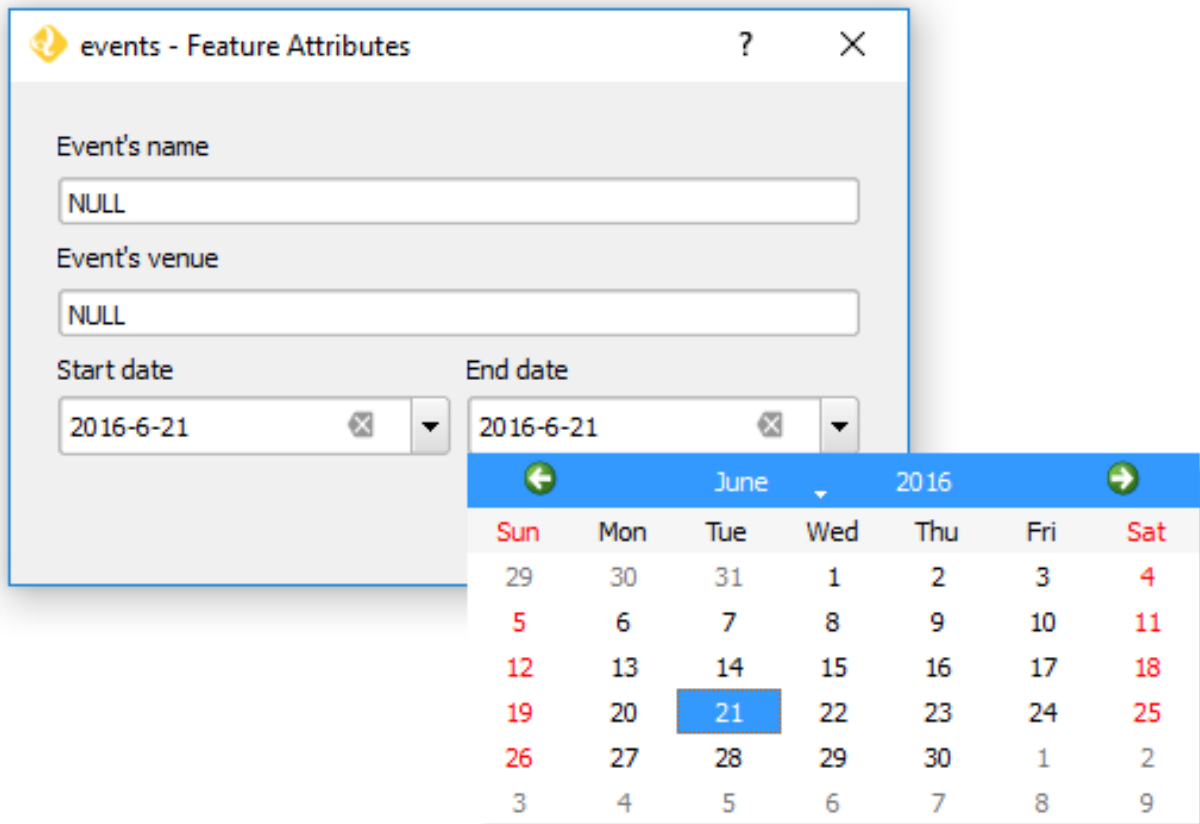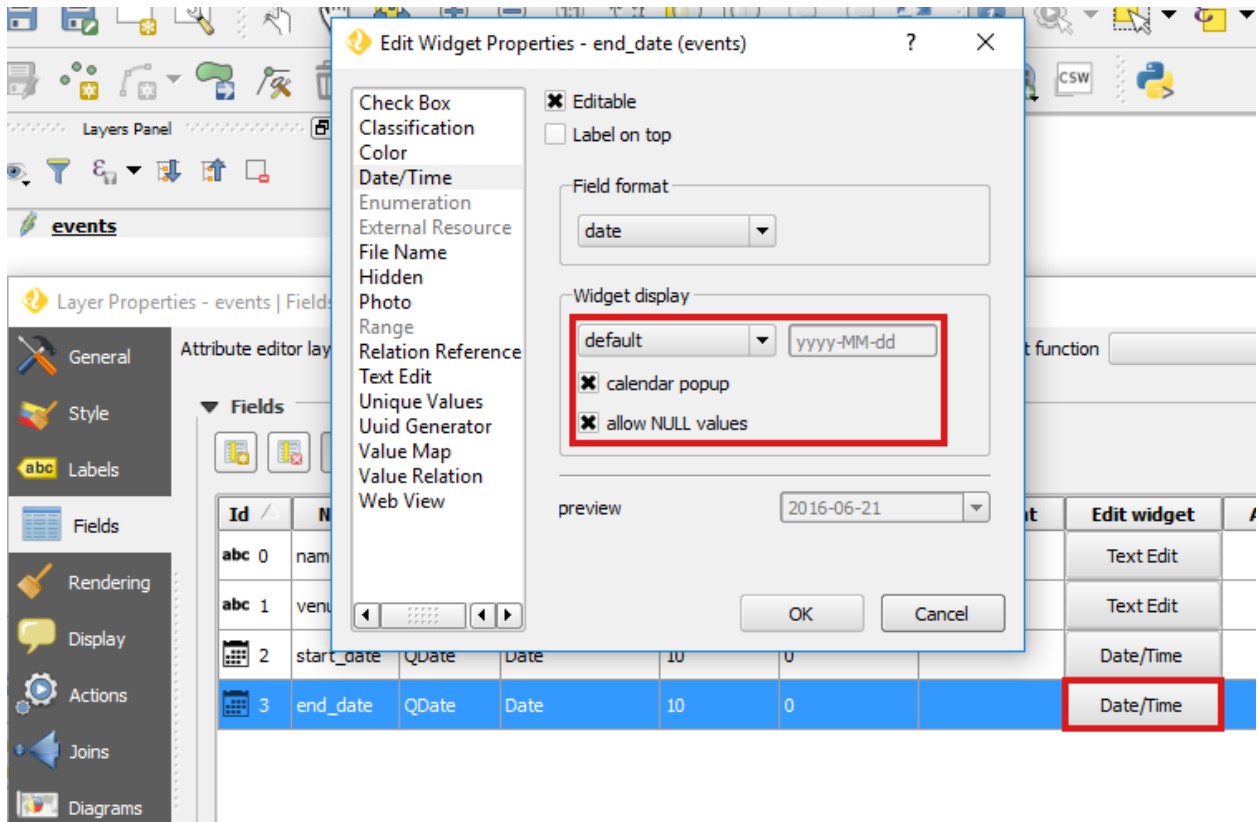
GDAL/OGR

About

Geospatial Data Abstraction Library (GDAL/OGR) is a cross-platform C++ translator library for raster and vector geospatial data formats that is released under an X/MIT style Open Source license by the Open Source Geospatial Foundation. As a library, it presents a single abstract data model to the calling application for all supported formats. It also comes with a variety of useful command line utilities for data translation and processing. GDAL supports over 50 raster formats, and OGR over 20 vector formats.

In Boundless Desktop installation, GDAL/OGR utilities are available through QGIS or the installed Command Shell. The shipped version of GDAL/OGR includes two extra libraries, LibKML and OpenJPEG2, for .kmz and JPEG2000 support, respectively.
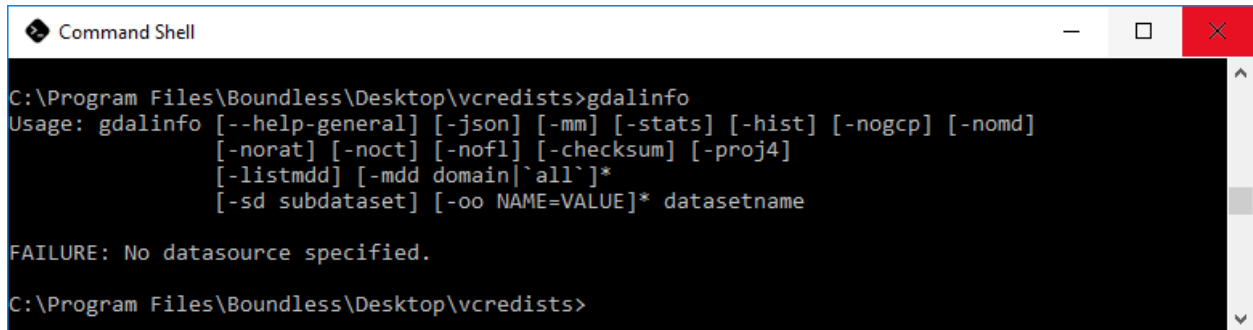
Quick start guide

GDAL/OGR is a suite of command line programs that can be launched through the **Command Shell**. **Command Shell** can be found in the *Boundless Desktop* folder.
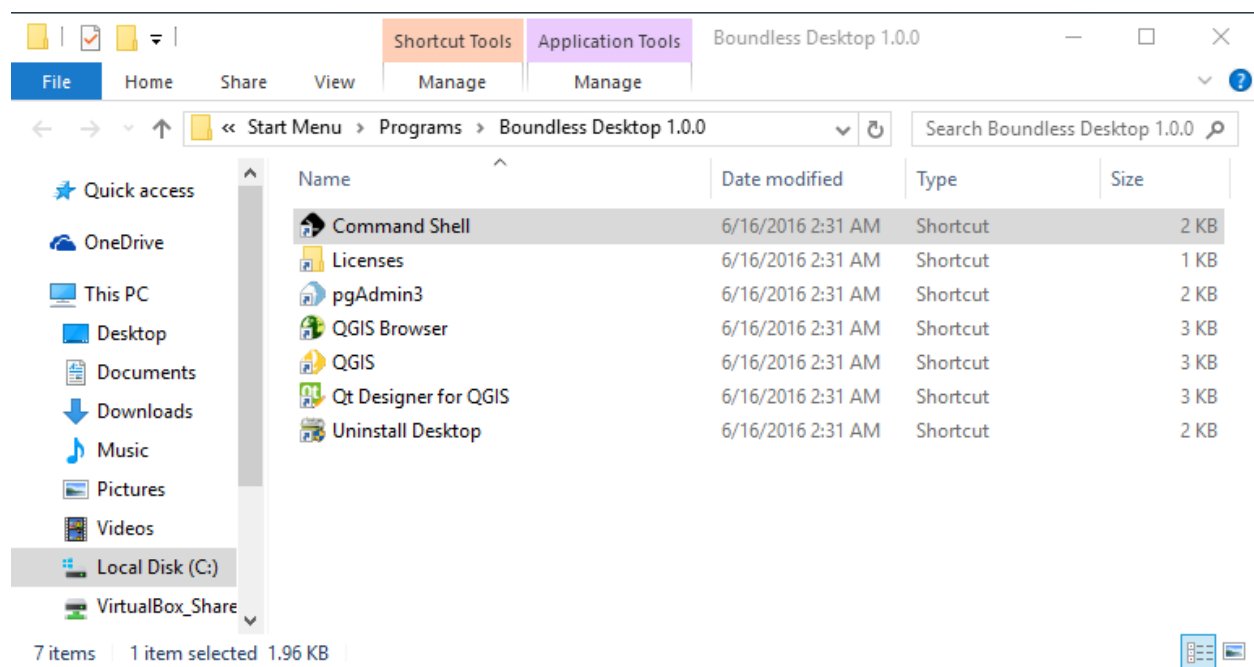
1. Download the data for this quick start `here` and unzip it anywhere you think it's convenient.

2. Open **Command Shell** using any of the available shortcuts in your computer.

3. If you are on Windows, you will immediately see the list of available commands. To list them again, type `o-help` and press `Enter`. To print the list of available commands on Mac, type `list-cmds` and press `Enter`.

4. To execute a **GDAL/OGR** command, type its name an press enter. If executed without arguments **GDAL/OGR** commands normally print a short help text that can also be accessed with *–help* option, for example:

```
gdalinfo --help
```

```
Command Shell                                            —    □    ✕

C:\Program Files\Boundless\Desktop\vcredists>gdalinfo
Usage: gdalinfo [--help-general] [-json] [-mm] [-stats] [-hist] [-nogcp] [-nomd]
                [-norat] [-noct] [-nofl] [-checksum] [-proj4]
                [-listmdd] [-mdd domain|`all`]*
                [-sd subdataset] [-oo NAME=VALUE]* datasetname

FAILURE: No datasource specified.

C:\Program Files\Boundless\Desktop\vcredists>
```

5. To print all information on the shapefile *events.shp* contained in the sample data, navigate to the location of the unzipped quick start data and run the following:

```
ogrinfo -al events.shp
```

The output of the command is:

```
INFO: Open of `events.shp'
      using driver `ESRI Shapefile' successful.

Layer name: events
Geometry: Point
Feature Count: 1
Extent: (7.123115, 50.718396) - (7.123115, 50.718396)
Layer SRS WKT:
GEOGCS["GCS_WGS_1984",
    DATUM["WGS_1984",
        SPHEROID["WGS_84",6378137,298.257223563]],
    PRIMEM["Greenwich",0],
    UNIT["Degree",0.017453292519943295]]
name: String (80.0)
venue: String (80.0)
start_date: Date (10.0)
end_date: Date (10.0)
OGRFeature(events):0
  name (String) = FOSS4G 2016
  venue (String) = World Conference Center Bonn
  start_date (Date) = 2016/08/21
  end_date (Date) = 2016/08/26
  POINT (7.12311523114681 50.718396029588526)
```

6. To convert the shapefile *events.shp* to a different coordinate reference system (UTM 33N) and save it to a different format (MapInfo) you can use **ogr2ogr**:

```
ogr2ogr -t_srs EPSG:32633 -f "MapInfo File" events_32633.tab events.shp
```

7. To check the transformed data you can use **ogrinfo**:

```
ogrinfo -al events_32633.tab
```

The output of the command is:

```
Had to open data source read-only.
INFO: Open of `events_32633.tab'
      using driver `MapInfo File' successful.

Layer name: events_32633
Geometry: Point
Feature Count: 1
Extent: (-55706.115697, 5648162.052998) - (-55706.115697, 5648162.052998)
Layer SRS WKT:
PROJCS["unnamed",
    GEOGCS["unnamed",
        DATUM["WGS_1984",
            SPHEROID["WGS 84",6378137,298.257223563],
            TOWGS84[0,0,0,0,0,0,0]],
        PRIMEM["Greenwich",0],
        UNIT["degree",0.0174532925199433]],
    PROJECTION["Transverse_Mercator"],
    PARAMETER["latitude_of_origin",0],
    PARAMETER["central_meridian",15],
    PARAMETER["scale_factor",0.9996],
    PARAMETER["false_easting",500000],
    PARAMETER["false_northing",0],
    UNIT["Meter",1.0]]
name: String (80.0)
venue: String (80.0)
start_date: Date (10.0)
end_date: Date (10.0)
OGRFeature(events_32633):1
  name (String) = FOSS4G 2016
  venue (String) = World Conference Center Bonn
  start_date (Date) = 2016/08/21
  end_date (Date) = 2016/08/26
  Style = SYMBOL(a:0,c:#000000,s:12pt,id:"mapinfo-sym-35,ogr-sym-10")
  POINT (-55706.115697181601718 5648162.052997644990683)
```

Online resources

- Official Site: http://www.gdal.org/
- GDAL Raster Utilities: http://www.gdal.org/gdal_utilities.html
- OGR Vector Utilities: http://www.gdal.org/ogr_utilities.html

Python interpreter

A full Python interpreter install is embedded within Boundless Desktop. Desktop does not use any Python interpreter or modules that you have installed on your system.

To manage packages available to the embedded Python install, open the included Command Shell application (as an administrator) and use `pip`:

```
pip --help
```

Please see pip documentation for more details.

Pre-installed Packages

The `IPython` and `Jupyter` packages are installed with Desktop. This allows for easy installation of the IPyConsole plugin for QGIS, an excellent additional console to QGIS's embedded Python console.

Plugins

Boundless Desktop's functionality can be remarkably extended by the use of plugins for QGIS.

Boundless plugins for QGIS

QGIS for Boundless Desktop includes all the core plugins of a standard QGIS installation, plus the *Boundless Connect plugin*, *Master Password Helper C++ plugin*, *OAuth2 authentication C++ plugin*, and *Reporting tool plugin*.

The following list comprises the currently available Boundless plugins for QGIS:

- **Boundless Connect Plugin** *Updated*

  Core plugin shipped with Boundless Desktop that acts as a single entry point to Boundless technology and content for QGIS. The plugin provides Boundless Desktop integration with Boundless Connect Portal, making it easier access online resources like training materials, documentation, lessons, basemaps, data, etc. The plugin can also be used to install other Boundless plugins for QGIS.

  (See Boundless Connect plugin Documentation for details)

- **Geoserver Explorer** *Updated*

  Plugin for configuring GeoServer through QGIS. It allows you to easily set up your GeoServer instance, from preparing data and styling to publishing directly to a catalog, all through the QGIS interface.

  (See Geoserver Explorer Documentation for details)

- **Image Discovery** *NEW*

  Plugin for allowing you to efficiently search, browse, preview and download imagery from imagery provider's catalogs in QGIS.

  (See Image Discovery Documentation for details)

- **Lessons** *Updated*

  Plugin for creating and following step-by-step lessons and tutorials within QGIS.

  (See Lessons Plugin Documentation for details)

- **Master Password Helper** *NEW*

  Core C++ plugin shipped with Boundless Desktop that allows you to store and synchronize QGIS's master password in the user's operating system's Wallet/keyChain/Password Manager. Using this plugin, you no longer need to repeatedly type the master password inside QGIS, as the system stores it and uses it whenever is required.

  (See Master Password helper for details)

- **MIL-STD2525**

  Plugin for facilitating the use of the MIL-STD-2525 military standard for symbology in QGIS.

  (See MIL-STD2525 Plugin Documentation for details)

- **MGRS Tools**

  A plugin that enables you to work with the Military Grid Reference System (MGRS) coordinates in QGIS.

  (See MGRS Tools Documentation for details)

- **OAuth2 Plugin**

  Core C++ plugin shipped with Boundless Desktop for allowing you to use OAuth2 authentication method to access online/remote resources. Different grant flows are available.

- **Support Tool** *NEW*

  Core plugin shipped with Boundless Desktop for collecting information from the user's QGIS installation and operating system to help in debugging issues. The information can be saved as a file or copied to the clipboard.

  (See Support Tool Documentation for details)

- **Terrain Analysis** *NEW*

  Plugin for facilitating the access to several raster analysis tools: Slope, Hillshade, Clip, Aspect, Viewshed, and Profile.

  (See Terrain Analysis Documentation for details)

- **Web App Builder** *Updated*

  A plugin that allows the easy creation of web applications based on layers, map compositions and bookmarks, as configured within a QGIS project. The resulting web applications can also include additional web services, various controls, and other interactivity.

  (See Web App Builder Documentation for details)

- **What3words plugin**

  A plugin that enables you to use What3words three word addresses in QGIS.

  (See What3words Plugin Documentation for details)

Boundless QGIS Plugin Repository

QGIS's functionality can be remarkably extended by the use of plugins.

This is the QGIS plugin repository for Boundless plugins that are not in the official QGIS repository.

Using the repository in QGIS

If you are using *Boundless Desktop*, this repository should be already available in its QGIS along with the Boundless Connect plugin. Otherwise, if you are using QGIS from one of the community's installers, you can setup this repository in the *Plugin manager* by going to *Plugins → Manage and install plugins* and, in the *Settings* tab, add a new repository with the following URL:

- https://qgis.boundlessgeo.com/plugins/plugins.xml
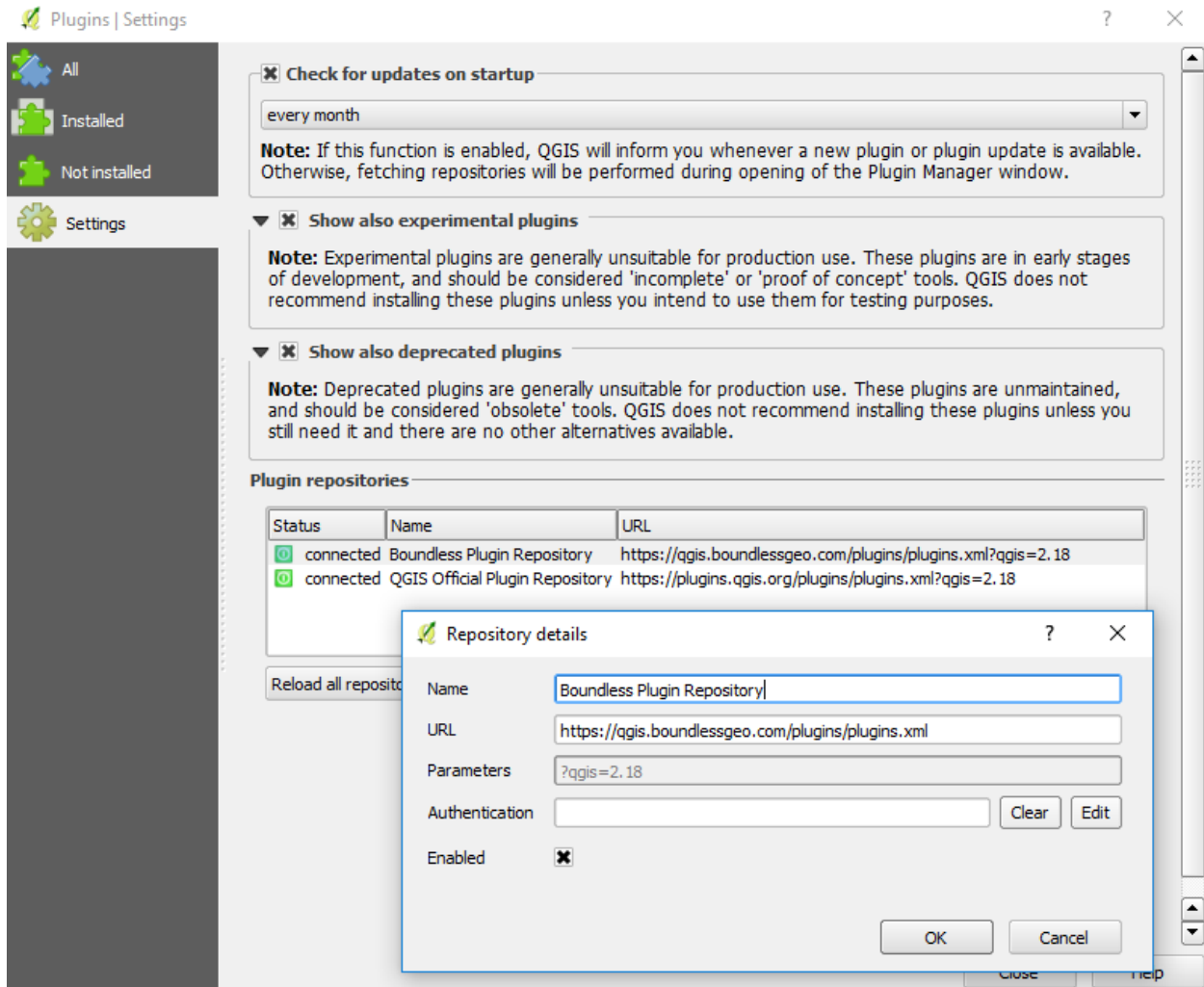
Fig. 7.1: Adding Boundless QGIS plugin repository

Note: You may wish to check **Show also experimental plugins** as well as **Show also deprecated plugins** options. The latter is required to install the updated, though deprecated, version of the OpenGeo Explorer plugin.

Warning: On Windows operating systems, while trying to connect to the repository, it's possible that you get a **Unable To Get Local Issuer Certificate** error message, and you won't be able to access the repository. To solve the issue, try using your browser (Chrome, Edge or Internet Explorer) to open the repository's URL. This operation should add the needed certificate to your Windows system.

Browsing the repository

- Available plugins

- Archives of released versions

Beta Repository

If you have signed up for access to the **Beta Plugin Repository**, you must set it up to be able to download and install the latest beta versions of our plugins.

To set the BETA repository, follow the instructions *above* giving the repository another name (e.g. `Boundless Plugin Beta Repository`) and using the following URL:

- https://qgis-beta.boundlessgeo.com/plugins/plugins.xml

Looking for something else?

- Boundless plugins in the official QGIS plugin repository

- Boundless Plugins documentation

- Older OpenGeo Explorer documentation

Warning: OpenGeo Explorer is deprecated and users are recommended to migrate to the new GeoServer Explorer plugin.

How to ask for support

At Boundless we have prepared two levels of support for Boundless Desktop:

- **Questions and Answers**: a crowdsourced support system available to any subscription level. You can ask questions and receive answers from a community of users and Boundless staff. Go to Questions and Answers.

- **Support tickets**: available only for Enterprise users, you will get direct help from Boundless dedicated support staff that, together with the development team, will search for a solution to your issue. Go to Support tickets.

Support tool

In either case, before you reach out for support, it's crucial that you can provide enough information about the problem that you are facing and about your system setup.

For that reason, we have created the *Support tool* library and QGIS plugin that helps you with the task of collecting information about your QGIS configuration and Operating System.

Usage

If your problem does not affect QGIS launching, then the most convenient way of using the Report tool is using it from QGIS.

1. From Boundless Desktop folder, open the `QGIS`.

2. In QGIS menus, click *Plugins → Boundless Support Tool*.

A new dialog will open with the full report visible, and the location of the saved copy.

If the issue you are facing is not letting QGIS launch correctly, then you may need to run the Reporting tool from the command line.

1. From Boundless Desktop folder, open the `Command Shell`.

2. In the command line type the following command and press ENTER.
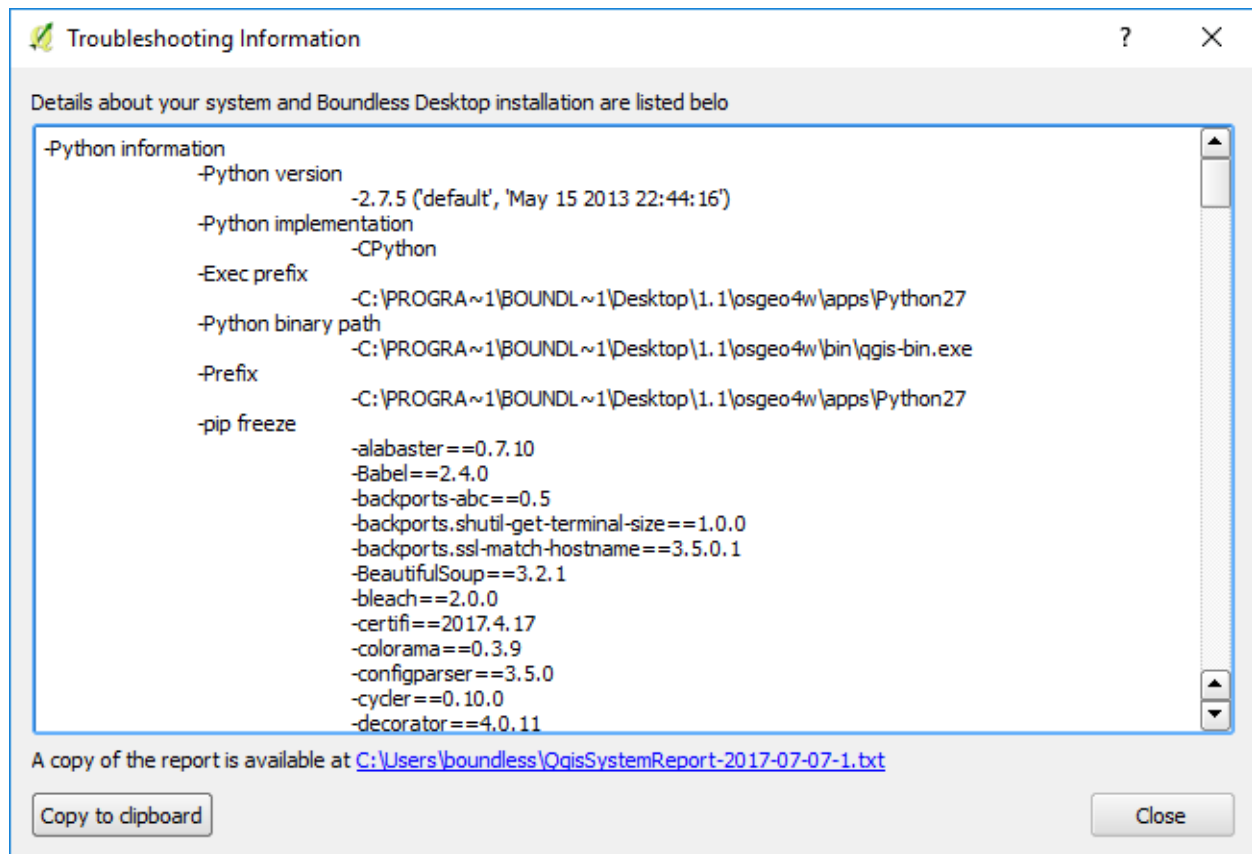
```
createreport
```

The tool will save a report and inform you where it was saved.

Note: We advice you to review the report before sending it to someone else, as the report may contain confidential information that you are not willing to share.
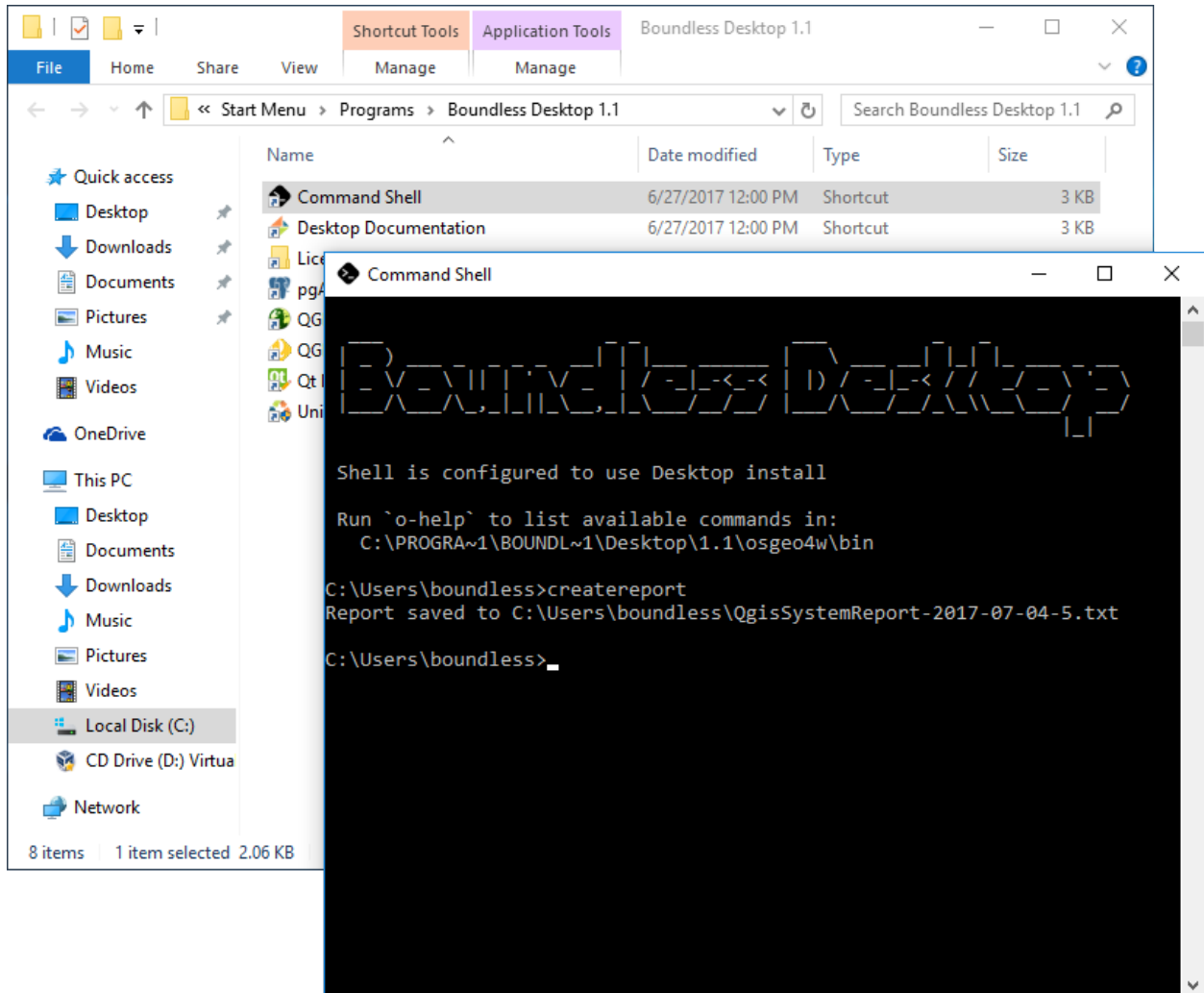
For System Administrators

Starting from Boundless Desktop 1.1, there are available tools to help system administrators deploy and manage Boundless Desktop in their organization machines.

Currently, the tools are solely focused on QGIS, the main component of Boundless Desktop, but it's planned to provide similar tools for the other applications in the future.

QGIS Initialization scripts

QGIS has the native ability to run a specified Python script when it starts. By loading an init script that will in turn load and run all the scripts in a given directory, this system allows Boundless Desktop's QGIS to be configured every time it starts.

Setting the scripts path to a network shared folder, allows the system administrator to perform QGIS configuration tasks in several machines by simply adding or changing the scripts in that folder.

Usage

Set the `QGIS_INIT_SCRIPTS_DIRECTORY` environment variable to a directory path containing Python scripts to run at QGIS start, if the variable is not set, QGIS will look in a default local folder determined by `QgsApplication.pkgDataPath() + /init_scripts`.

On Windows, the default local folder is:

```
C:\Program Files\Boundless\Desktop\1.1\osgeo4w\apps\qgis\init_scripts
```

On Mac OS, the default local folder is:

```
/Library/Boundless/Desktop/1.1/Cellar/qgis2-bdesk/2.18.10/QGIS for Boundless Desktop␣
↪1.1.app/Contents/MacOS/../Resources/init_scripts
```

Tip: Setting the `QGIS_INIT_SCRIPTS_DIRECTORY` environment variable to a network shared folder, allows the system administrator to perform configuration tasks in several machines by simply adding or changing the scripts in that folder.

Note: Scripts are alphabetically sorted

Scripts run in alphabetically order, so if necessary prefix them with with some integers.

> **Warning:** In Windows, if the `QGIS_INIT_SCRIPTS_DIRECTORY` environment variable is set to a folder other than the default one, the `000_win_load_authorities.py` script must be copied from the default *init_scripts* folder to the new one. Otherwise, when trying to connect to some services endpoints, users may receive SSL Errors warnings claiming that a valid certificate for that URL is not available.

Scripts examples

Inside the `init_scripts` default folder there is an `examples` folder with some example scripts to illustrate a few common customization tasks, which are transcript below.

Tip: In order to run the examples, you can set `QGIS_INIT_SCRIPTS_DIRECTORY` environment variable to point to the examples directory before launching QGIS as specified above.

Checking for a setting value

```
# Sample script for QGIS Init Script
# This script will check for a setting value in user
# settings and set it to 'A Value' if not found

from qgis.PyQt.QtCore import QSettings
from qgis.core import QgsMessageLog
QgsMessageLog.logMessage("Init script: %s" % __file__, tag="Init script",
→level=QgsMessageLog.INFO)

if not QSettings().value("InitScript/MyTestSetting"):
    QgsMessageLog.logMessage("Setting 'InitScript/MyTestSetting' to 'A Value'", tag=
→"Init script", level=QgsMessageLog.INFO)
    QSettings().setValue("InitScript/MyTestSetting", "A Value")
else:
    QgsMessageLog.logMessage("'InitScript/MyTestSetting' already set to '%s'" %
→QSettings().value("InitScript/MyTestSetting"), tag="Init script",
→level=QgsMessageLog.INFO)
```

Single run script

```
# Sample script for QGIS Init Script
# This script will check for a setting value to determine
# if it needs to be run.
```

```
# When run, it will set the setting value and log a line.


from qgis.PyQt.QtCore import QSettings
from qgis.core import QgsMessageLog
QgsMessageLog.logMessage("Init script: %s" % __file__, tag="Init script",
→level=QgsMessageLog.INFO)


if not QSettings().value("InitScript/RunOnceHasRun"):
    QgsMessageLog.logMessage("Setting 'InitScript/RunOnceHasRun' to 'true'", tag=
→"Init script", level=QgsMessageLog.INFO)
    QSettings().setValue("InitScript/MyTestSetting", True)
else:
    QgsMessageLog.logMessage("'InitScript/MyTestSetting' already set to '%s'" %
→QSettings().value("InitScript/MyTestSetting"), tag="Init script",
→level=QgsMessageLog.INFO)
```

Install a plugin from a zip file

```
# Sample script for QGIS Init Script
# This script will install the HelloWorld plugin into user space (if not yet
→installed)
# The plugin will also be enabled.

import os
import zipfile

from qgis.PyQt.QtCore import QSettings
from qgis.utils import home_plugin_path, loadPlugin, startPlugin, plugins
from qgis.core import QgsMessageLog
QgsMessageLog.logMessage("Init script: %s" % __file__, tag="Init script",
→level=QgsMessageLog.INFO)


if "HelloWorld" not in plugins:
    # Installing
    zip_ref = zipfile.ZipFile(os.path.join(os.path.dirname(os.path.realpath(__file__
→)), 'test_data', 'helloworld.zip'), 'r')
    zip_ref.extractall(home_plugin_path)
    zip_ref.close()
    loadPlugin('HelloWorld')
    startPlugin('HelloWorld')
    QgsMessageLog.logMessage("Plugin HelloWorld has been successfully installed", tag=
→"Init script", level=QgsMessageLog.INFO)
else:
    QgsMessageLog.logMessage("Plugin HelloWorld has been already installed", tag=
→"Init script", level=QgsMessageLog.INFO)
```

Add a WMS connection

```
# Sample script for QGIS Init Script
# This script will add a demo WMS endpoint
```

```python
from qgis.PyQt.QtCore import QSettings
from qgis.utils import home_plugin_path, loadPlugin, startPlugin, plugins
from qgis.core import QgsMessageLog
QgsMessageLog.logMessage("Init script: %s" % __file__, tag="Init script",
→level=QgsMessageLog.INFO)


WMS_URL="http://demo.boundlessgeo.com/geoserver/wms"
WMS_NAME="Boundlessgeo Geoserver Demo"

settings = QSettings()

if "Qgis/WMS/%s/authcfg" % WMS_NAME not in settings.allKeys():
    settings.setValue("Qgis/WMS/%s/authcfg" % WMS_NAME, "")
    settings.setValue("Qgis/WMS/%s/username" % WMS_NAME, "")
    settings.setValue("Qgis/WMS/%s/password" % WMS_NAME, "")
    settings.setValue("Qgis/connections-wms/%s/dpiMode" % WMS_NAME, 7)
    settings.setValue("Qgis/connections-wms/%s/ignoreAxisOrientation" % WMS_NAME,
→False)
    settings.setValue("Qgis/connections-wms/%s/ignoreGetFeatureInfoURI" % WMS_NAME,
→False)
    settings.setValue("Qgis/connections-wms/%s/ignoreGetMapURI" % WMS_NAME, False)
    settings.setValue("Qgis/connections-wms/%s/invertAxisOrientation" % WMS_NAME,
→False)
    settings.setValue("Qgis/connections-wms/%s/referer" % WMS_NAME, "")
    settings.setValue("Qgis/connections-wms/%s/smoothPixmapTransform" % WMS_NAME, "")
    settings.setValue("Qgis/connections-wms/%s/url" % WMS_NAME, WMS_URL)
    QgsMessageLog.logMessage("WMS %s has been successfully installed" % WMS_NAME, tag=
→"Init script", level=QgsMessageLog.INFO)
else:
    QgsMessageLog.logMessage("WMS %s was already installed" % WMS_NAME,  tag="Init
→script", level=QgsMessageLog.INFO)
```

Set a default project

```python
# Sample script for QGIS Init Script
# This script will add default project

import os
from shutil import copyfile
from qgis.PyQt.QtCore import QSettings
from qgis.core import QgsMessageLog, QgsApplication
QgsMessageLog.logMessage("Init script: %s" % __file__, tag="Init script",
→level=QgsMessageLog.INFO)


settings = QSettings()
DEFAULT_PROJECT_PATH=os.path.join(QgsApplication.qgisSettingsDirPath(), "project_
→default.qgs")

if not os.path.exists(DEFAULT_PROJECT_PATH):
    try:
        copyfile(os.path.join(os.path.dirname(os.path.realpath(__file__)), 'test_data
→', 'project_default.qgs'), DEFAULT_PROJECT_PATH)
        QgsMessageLog.logMessage("Default project has been successfully installed",
→tag="Init script", level=QgsMessageLog.INFO)
```

```python
        # Set the settings to use the default project
        settings.setValue("qgis/newProjectDefault", True)
    except Exception as ex:
        QgsMessageLog.logMessage("Error installing default project: %s" % ex, tag=
→"Init script", level=QgsMessageLog.CRITICAL)
else:
    QgsMessageLog.logMessage("Default project was already installed",  tag="Init␣
→script", level=QgsMessageLog.INFO)
```

Add an authentication configuration

```python
# Sample script for QGIS Init Script
# This script will initialize the authentication DB and
# add and HTTP Basic authentication configuration.
# The script will also set the master password to `password`
#
# Note: this script can only run if the authentication DB was not
# already created by the user, this means that it must be executed
# only once on a fresh QGIS installation before the user
# started QGIS for the first time.


from qgis.PyQt.QtCore import QSettings
from qgis.core import QgsAuthManager, QgsAuthMethodConfig, QgsMessageLog

AUTHDB_MASTERPWD = 'password'

QgsMessageLog.logMessage("Init script: %s" % __file__, tag="Init script",␣
→level=QgsMessageLog.INFO)

# Do not run twice!
if not QSettings().value("InitScript/AuthCfgCreated", type=bool):
    QSettings().setValue("InitScript/AuthCfgCreated", True)
    # Check if authdb master password is set
    am = QgsAuthManager.instance()
    if not am.masterPasswordHashInDb():
        # Set it!
        am.setMasterPassword(AUTHDB_MASTERPWD, True)
        # Create config
        am.authenticationDbPath()
        am.masterPasswordIsSet()
        cfg = QgsAuthMethodConfig()
        cfg.setId('myauth1') # Optional, useful for plugins to retrieve an authcfg
        cfg.setName('Example Auth Config HTTP Basic')
        cfg.setMethod('Basic')
        cfg.setConfig('username', 'username')
        cfg.setConfig('password', 'password')
        am.storeAuthenticationConfig(cfg)
    else:
        QgsMessageLog.logMessage("Master password was already set: aborting", tag=
→"Init script", level=QgsMessageLog.INFO)

else:
    QgsMessageLog.logMessage("AuthCfg was already created: aborting", tag="Init script
→", level=QgsMessageLog.INFO)
```

Show a message on start up

```
# Sample script for QGIS Init Script
# This script will open an info dialog


from qgis.PyQt.QtGui import QMessageBox

QMessageBox.information(None, "Hello", "Hello World!")
```

QGIS Global Settings File

Starting from Boundless Desktop 1.1, QGIS supports global settings: a feature that is currently only available in QGIS 3.x series.

Almost all settings for QGIS application are stored using the *QSettings* framework that is provided by the *Qt* library upon which QGIS is built.

For every setting value, an optional default value can be specified inline in the application code, the global settings implementation allows the inline default values to be overridden by an external and totally optional global settings file (in *.ini* format).

Therefore, by providing a global settings file, the order for a setting value lookup becomes:

- user settings file
- global settings file
- inline default

This global setting file can be used to provide pre-configuration and/or custom default values for all settings used inside QGIS.

QGIS Global settings and defaults can be set by editing the `qgis_global_setting.ini` file. Moreover, it is possible to overwrite the path for the `qgis_global_setting.ini`, by using one of the following:

- Specifying the file's path using the –globalsettings option when running QGIS. For instance:

```
$ qgis --globalsettings /home/user/qgis_global_setting.ini
```

- Setting the `QGIS_GLOBAL_SETTINGS_FILE` environment variable with the file location.

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

If none of the two above methods is used, QGIS will use the file's default location. On Windows, the default location folder is:

```
C:\Program Files\Boundless\Desktop\1.1\osgeo4w\apps\qgis\qgis_global_setting.ini
```

While on Mac OS, the default location is:

```
/Library/Boundless/Desktop/1.1/Cellar/qgis2-bdesk/2.18.10/QGIS for Boundless Desktop␣
→1.1.app/Contents/MacOS/../Resources/qgis_global_setting.ini
```

Exporting QGIS settings to INI format

Since Windows and macOS versions of QGIS don't store settings in INI format, having to create a `qgis_global_setting.ini` file from scratch can be a laborious task.

For that purpose, a script is available to dump QGIS QgsSettings to a `.ini` file. The resultant file can then be used to help populate a `qgis_global_settings.ini` file.

1. Download `qgis-settings-to-ini.py.zip` file to your machine and extract it.

2. From within QGIS, open the *Python Console* (*Plugins → Python Console*).

3. In the *Python Console*, click the *Show Editor* button.

4. In the Editor toolbar, click the *Load Script*, browse to the `qgis-settings-to-ini.py` file and click *Open*.

5. Finally, click the *Run script* button.

A message in the *Python Console* will inform you of the path to the output file, or any runtime errors..