

Making QGIS plugins

GEO1005 (2017-18 Q2)

Spatial Decision Support for Planning and
Crisis Management

Jorge Gil

Vector, Raster, Database, Web, Processing

<https://plugins.qgis.org/plugins/popular/>

- Mmqgis
- Processing
- Table Manager
- OpenLayers
- Street View
- GeoSearch
- Nearest Neighbour
- Point Sampling tool
- Contour
- Concave Hull
- Heatmap
- RT Qspider
- osmSearch
- OpenRouteService
- Quick OSM
- PDOK BAG geocoder
- PDOK services
- Time Manager
- InaSAFE

- Windows: C:\Users\user\.qgis2/python/plugins
- UNIX \ Mac: ~/.qgis2/python/plugins

Note:

The .qgis2 folder is invisible. On a Mac the best way to find it is using Finder menu ‘Go > Go To Folder...’ and type the above path.

Once there, make a shortcut to the folder or add it to the Sidebar.

- QGIS plug-ins are written in pure Python code and are open source.
- Existing plug-ins are a great resource for learning how to make new plugins, and learn how to implement specific features.

How to make QGIS plugins

Workflow:

1. Create basic plug-in using “Plugin Builder”
2. Open the plugin as a new project in PyCharm
3. Modify user interface (UI) in QtDesigner
4. Modify python code to add functionality
5. *Deploy* (compiles and creates plugin in QGIS plugins folder)
6. *Reload* the plugin using “Plug-in Reloader”
7. Test the plugin in QGIS
8. Check for errors in the Python Stacktrace or the Python console
9. Fix the code in Pycharm
10. *Compile* resources (only if icons or UI are modified)
11. *Quick deploy* (updates plugin in plugins folder)
12. → Go back to step 6 (until it works!)
13. *Commit* changes to Github repository
14. → Go back to step 3 (until the plugin is complete!)
Many iterations later...
 13. Create a *zip* file of the plugin
 14. Publish in QGIS plugins repository

Plugins | Installed (260)

All
Installed
Not installed
Upgradeable
New
Invalid
Settings

Search plugin

MetaSearch Catalogue Client
mmqgis
MultiEdit
multiPrint
norGIS ALKIS-Einbindung
NTv2 Datum Transformations
Open Aerial Map (OAM)
OpenLayers Plugin
OSMDownloader
Oursins
PDOK BAG Geocoder
PDOK services plugin
Plugin Builder
Plugin Reloader
PostGIS geoprocessing tools
pyUPVBib
QGIS Cloud Plugin
QuickMultiAttributeEdit
QuickWKT
Raster Terrain Analysis plugin
Remote Debug
Road graph plugin
Search & format EPSG CRS Plugin
SLD4raster
Spatial Query Plugin

Plugin Builder

Creates a QGIS plugin template for use as a starting point in plugin development

Create a template for a QGIS plugin

★★★★★ 48 rating vote(s), 45411 downloads

Tags: development
More info: [homepage](#) [tracker](#) [code repository](#)

Author: [GeoApt LLC](#)

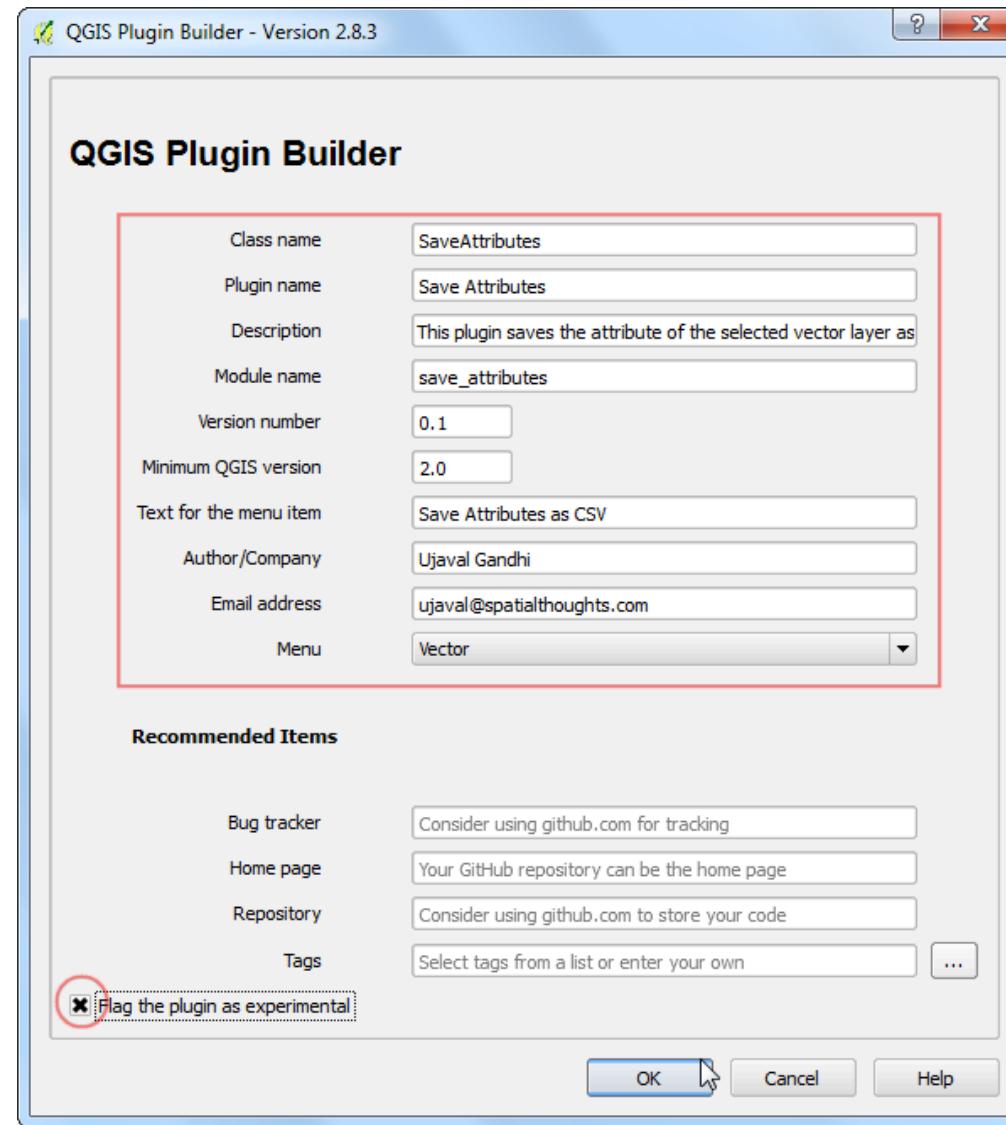
Installed version: 2.10.1 (in /Users/jorge/.qgis2/python/plugins/pluginbuilder)
Available version: 2.10.1 (in QGIS Official Plugin Repository)

changelog:
2015-08-21 New docked widget template (courtesy worthlutz)

Upgrade all Uninstall plugin Reinstall plugin

Close Help

QGIS Plugin Builder settings



QGIS Plugin Builder output

- Save the plugin to your local drive.
- This is the development folder and the source code will be synchronised with the Github repository.
- **Do not save it in the QGIS plugins folder!**

Name	Date Modified	Size	Kind
__init__.py	2 Nov 2015 20:43	2 KB	Python script
.DS_Store	Today 13:56	10 KB	The U...ument
.gitignore	21 Jun 2017 12:11	43 bytes	TextEd...ument
.idea	Today 12:50	65 KB	Folder
external	11 Jan 2016 10:34	422 KB	Folder
help	2 Dec 2015 18:37	1.1 MB	Folder
i18n	31 Aug 2015 11:02	333 bytes	Folder
icon.png	3 Dec 2015 23:15	2 KB	PNG image
icons	9 Jan 2016 00:02	61 KB	Folder
Makefile	2 Nov 2015 20:43	7 KB	TextEd...ument
metadata.txt	19 Dec 2015 07:47	1 KB	text
pb_tool.cfg	6 Nov 2016 15:02	3 KB	INI co...tion file
plugin_upload.py	2 Nov 2015 20:43	3 KB	Python script
pylintrc	2 Nov 2015 20:43	9 KB	TextEd...ument
README.html	2 Nov 2015 20:43	2 KB	HTML text
README.txt	2 Nov 2015 20:43	970 bytes	text
resources.py	9 Jan 2016 00:03	229 KB	Python script
resources.qrc	9 Jan 2016 00:03	230 bytes	Qt Res...ce File
scripts	31 Aug 2015 11:02	2 KB	Folder
spatial_decision_dockwidget_base.ui	Today 12:49	12 KB	Qt UI File
spatial_decision_dockwidget.py	2 Dec 2016 15:33	30 KB	Python script
spatial_decision.py	29 Nov 2016 10:57	8 KB	Python script
test	2 Nov 2015 20:43	19 KB	Folder
utility_functions.py	23 Nov 2016 02:50	32 KB	Python script

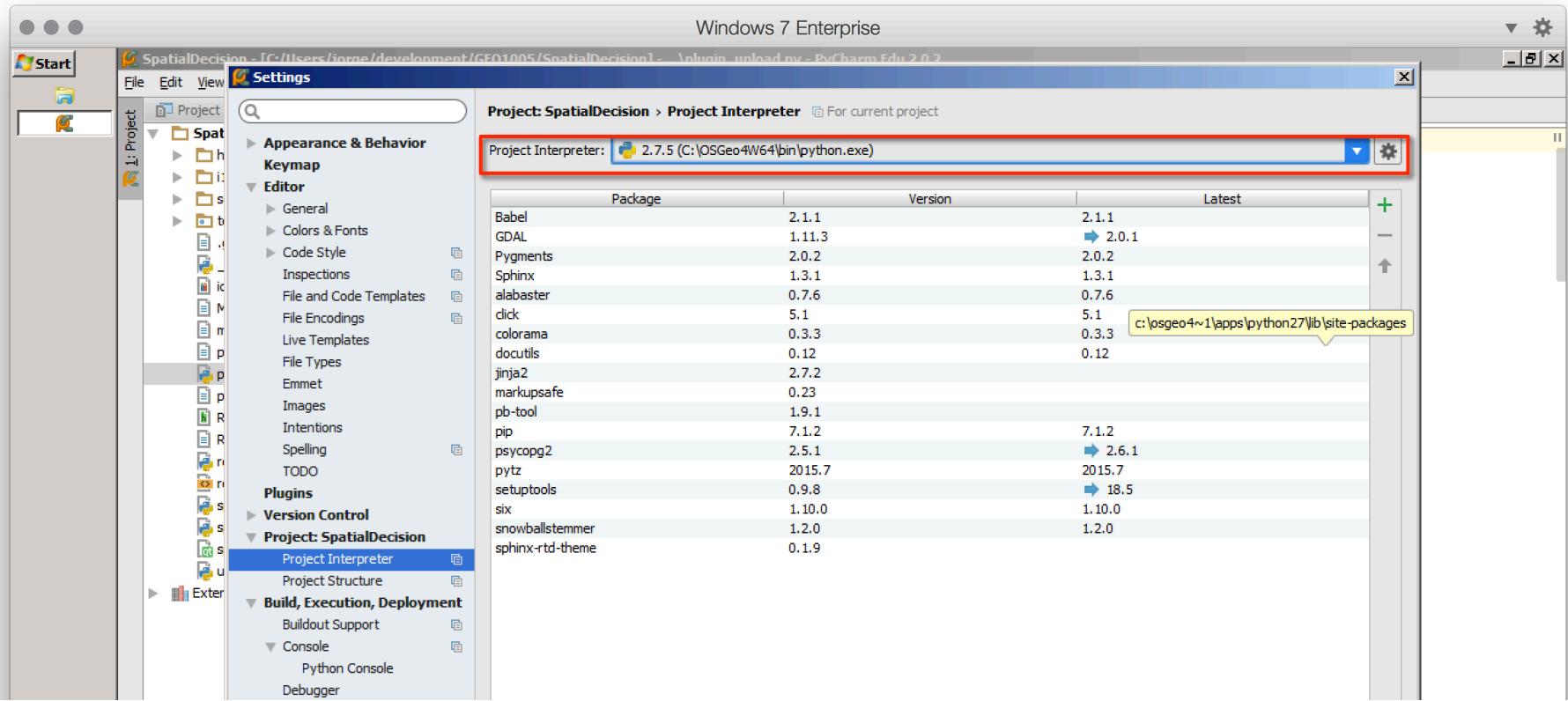
Create project in PyCharm

- Use PyCharm to modify the plugin code and components

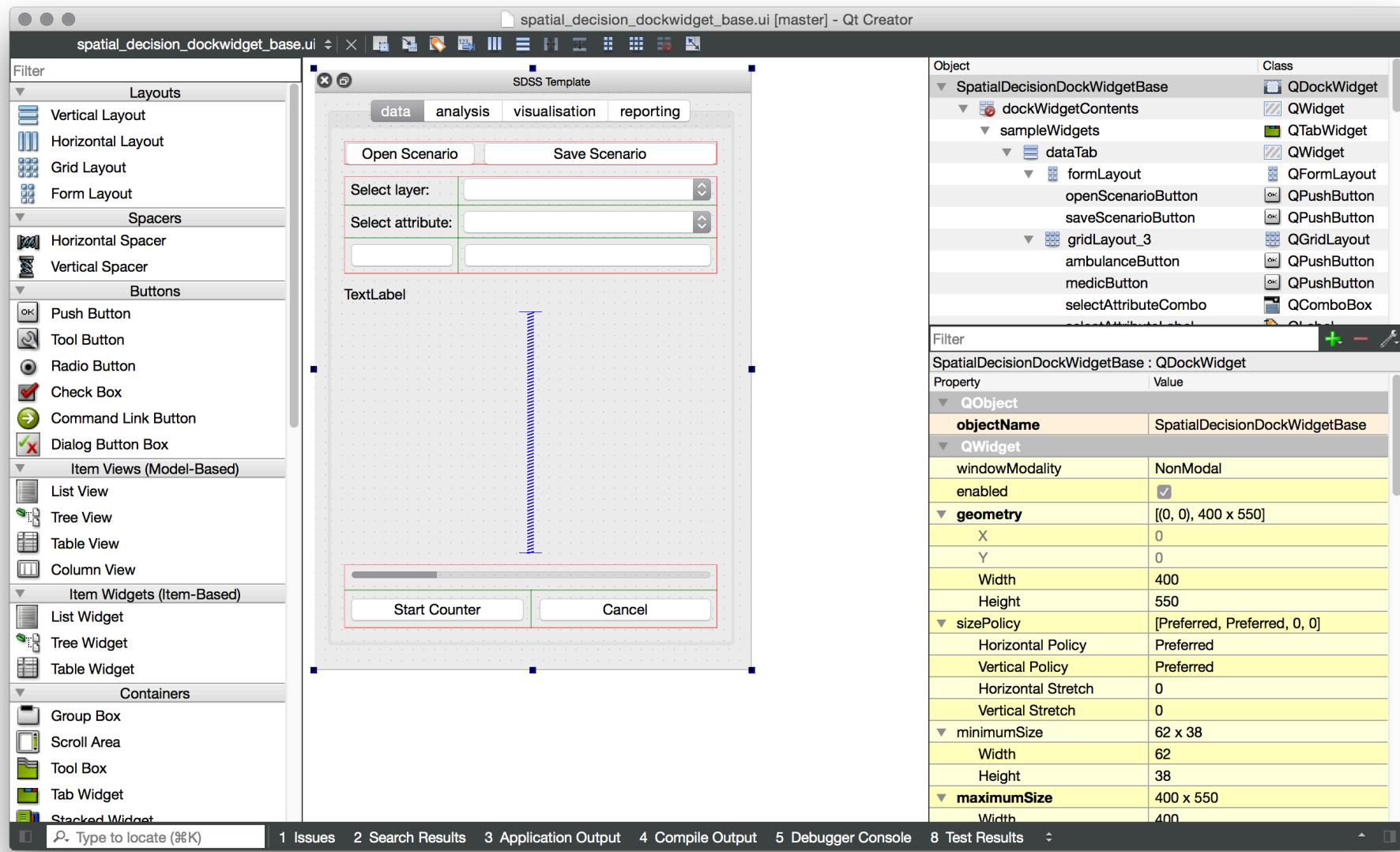
When working on a PyQGIS project you must select the Python Interpreter in the Settings window.

This points to the required QGIS packages and pb_tool.

ALWAYS start PyCharm using the batch shortcut.

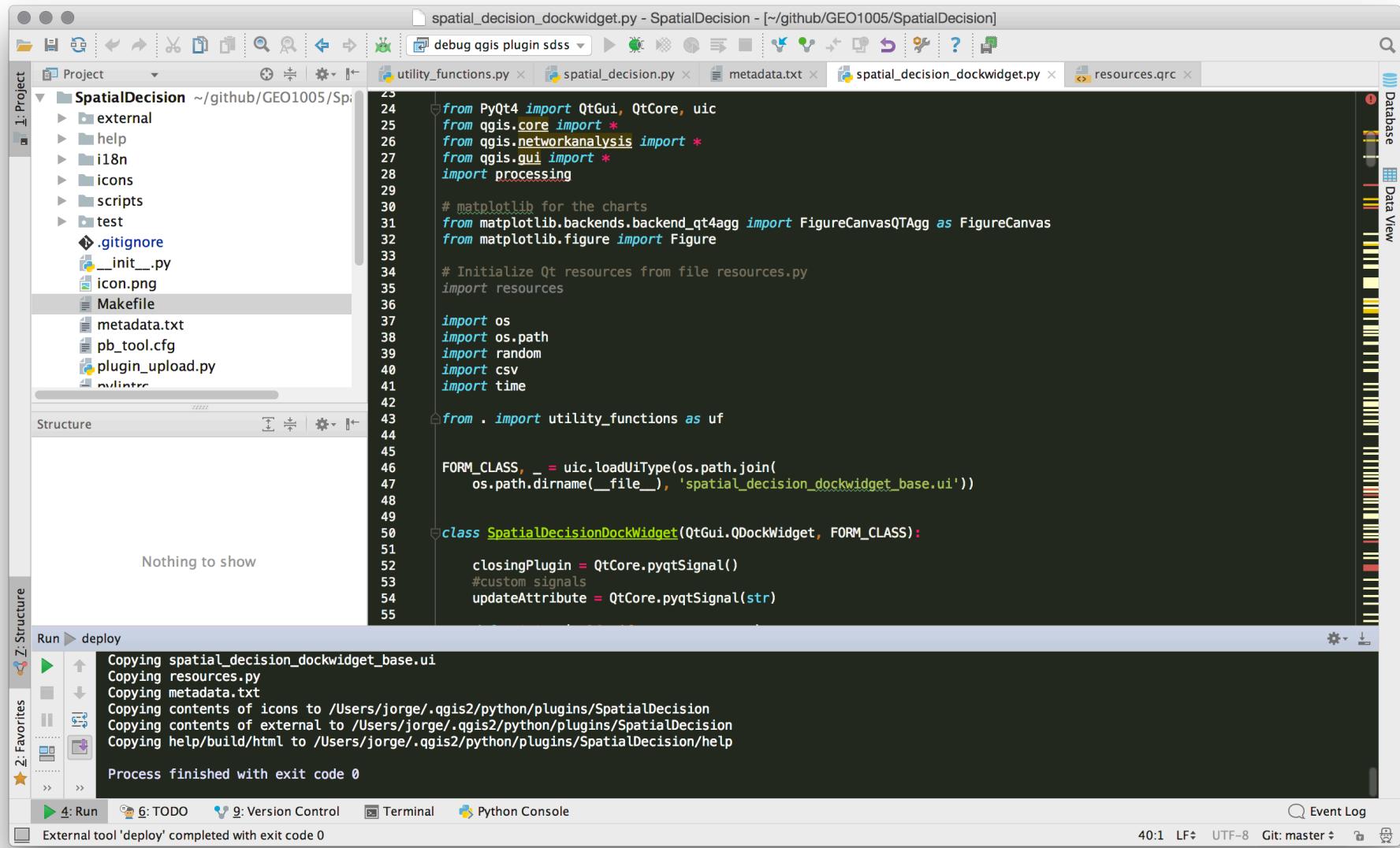


- Use Qt Designer / Creator to modify the UI files



Modify the code and deploy (with pb_tool)

- Deploy the plugin to the QGIS plugins folder for testing



```

23
24     from PyQt4 import QtGui, QtCore, uic
25     from qgis.core import *
26     from qgis.networkanalysis import *
27     from qgis.gui import *
28     import processing
29
30     # matplotlib for the charts
31     from matplotlib.backends.backend_qt4agg import FigureCanvasQTAgg as FigureCanvas
32     from matplotlib.figure import Figure
33
34     # Initialize Qt resources from file resources.py
35     import resources
36
37     import os
38     import os.path
39     import random
40     import csv
41     import time
42
43     from . import utility_functions as uf
44
45
46     FORM_CLASS, _ = uic.loadUiType(os.path.join(
47         os.path.dirname(__file__), 'spatial_decision_dockwidget_base.ui'))
48
49
50     class SpatialDecisionDockWidget(QtGui.QDockWidget, FORM_CLASS):
51
52         closingPlugin = QtCore.pyqtSignal()
53         #custom signals
54         updateAttribute = QtCore.pyqtSignal(str)
55

```

The PyCharm interface shows the project structure on the left, with files like spatial_decision_dockwidget.py, utility_functions.py, spatial_decision.py, metadata.txt, and resources.qrc listed. The bottom status bar indicates the deployment process completed with exit code 0.

ALWAYS USE pb_tool to deploy a plugin for use and testing in QGIS.

Don't copy manually the folder or symlink to the source folder.

- **Compile** – process resources and ui files that need to be converted to python.
- **Deploy** – compile and copy the relevant plugin files into the QGIS plugins directory.
- **Quick deploy** – only updates the plugin files, it's faster than deploy
- **Clean compile** – deletes the compiled files.
- **Clean deploy** – deletes the plugin from the QGIS plugins directory.
- **Zip** – creates a zip file with the relevant plugin files for distribution.

Plugins | Installed (260)

All Search plugin

Installed

Not installed

Upgradeable

New

Invalid

Settings

- MetaSearch Catalogue Client
- mmqgis
- MultiEdit
- multiPrint
- norGIS ALKIS-Einbindung
- NTv2 Datum Transformations
- Open Aerial Map (OAM)
- OpenLayers Plugin
- OSMDownloader
- Oursins
- PDOCK BAG Geocoder
- PDOCK services plugin
- Plugin Builder
- Plugin Reloader**
- PostGIS geoprocessing tools
- pyUPVBib
- QGIS Cloud Plugin
- QuickMultiAttributeEdit
- QuickWKT
- Raster Terrain Analysis plugin
- Remote Debug
- Road graph plugin
- Search & format EPSG CRS Plugin
- SLD4raster
- Spatial Query Plugin

This plugin is experimental

Plugin Reloader

Reloading a chosen plugin in one click (only useful for Python Plugin Developers)

★★★★★ 33 rating vote(s), 12172 downloads

Tags: reloader, reload, python, development, developer
More info: [homepage](#) [tracker](#) [code repository](#)

Author: [Borys Jurgiel](#)

Installed version: 0.6.3 (in /Users/jorge/.qgis2/python/plugins/plugin_reloader)
Available version: 0.6.3 (in QGIS Official Plugin Repository)

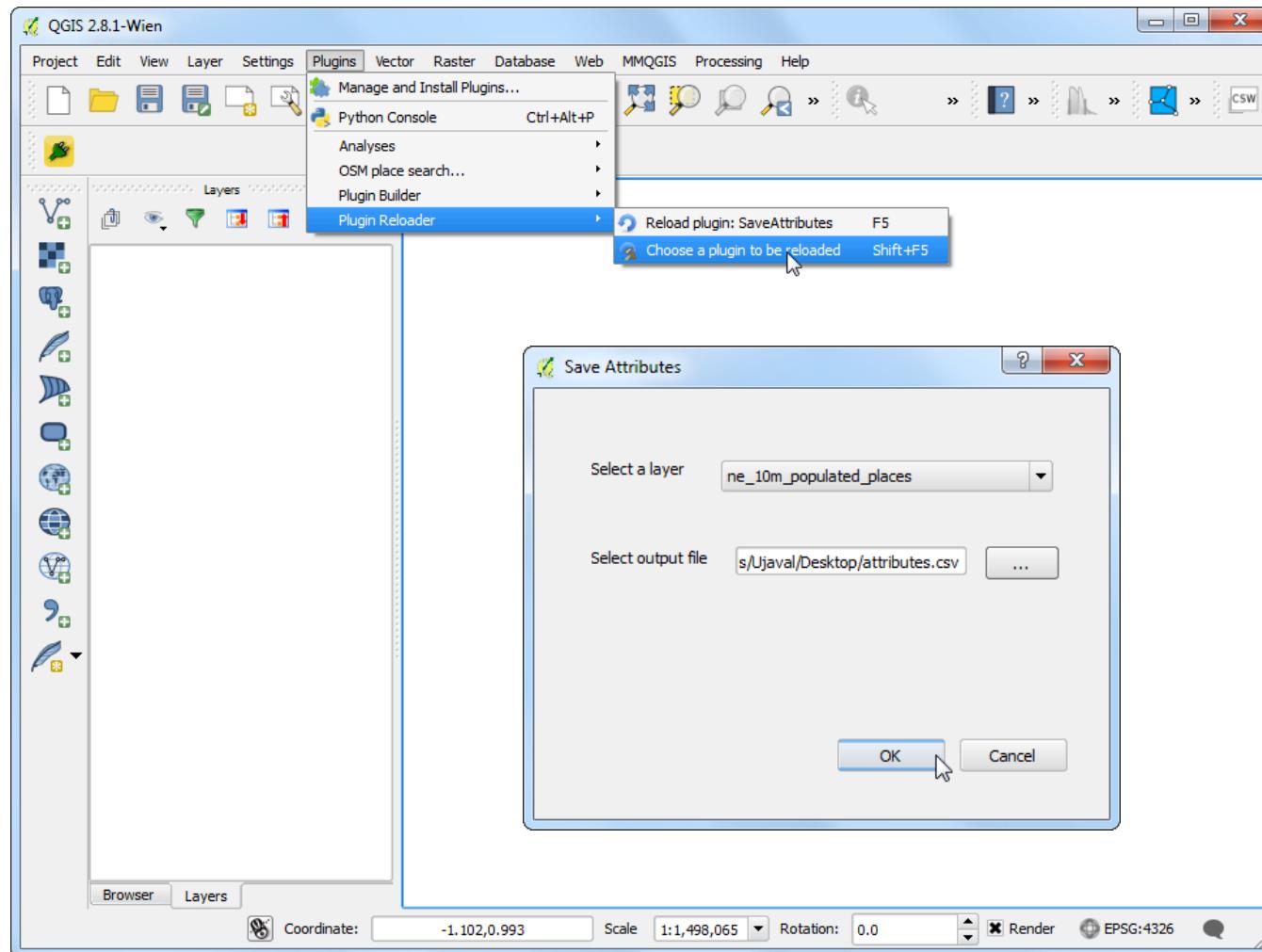
Upgrade all Uninstall plugin Reinstall plugin

Close Help

The screenshot shows the QGIS Plugin Manager interface. On the left, there's a sidebar with categories like All, Installed, Not installed, etc., and a search bar. The main area shows a list of installed plugins. The 'Plugin Reloader' plugin is highlighted. To the right, a detailed view of the plugin is shown with its title, a note that it's experimental, a brief description, a 5-star rating, download counts, tags, author information, and version details. At the bottom, there are buttons for managing the plugin: Upgrade all, Uninstall plugin, Reinstall plugin, Close, and Help.

Configure Plugin Reloader, then reload to see changes

- The first time choose your plugin folder to reload.
- Then just click the Reload plugin button or menu



When testing plugins running in QGIS there are always errors.

Debugging is about finding and fixing those errors.

- To debug plugins we can read the Python Stacktrace error messages. These indicate the line where the error occurred and the type of error.
- To trace variable's values or check progress in the algorithm we can use the print command. The results appear in the QGIS Python Console (Plugins > Python Console)
- However, this does not offer complete debugging tools.
- You should use the remote debugging tools in PyCharm (Pro) for fixing difficult bugs. (see Lab 0 slides)

<http://www.lutraconsulting.co.uk/blog/2014/10/17/getting-started-writing-qgis-python-plugins/>

<http://spatialgalaxy.com/2014/10/09/qgis-plugin-development-with-pb-tool/>

The following tutorials do not use the pb_tool: ignore the compile commands and use the pb_tool shortcuts in PyCharm instead:

<https://anitagraser.com/2014/04/26/getting-started-writing-qgis-2-x-plugins/>

<http://www.digital-geography.com/build-qgis-plugin/#.WCjciOHx6Rs>

PyQGIS Developer Cookbook – easy to follow:

http://docs.qgis.org/2.18/en/docs/pyqgis_developer_cookbook/

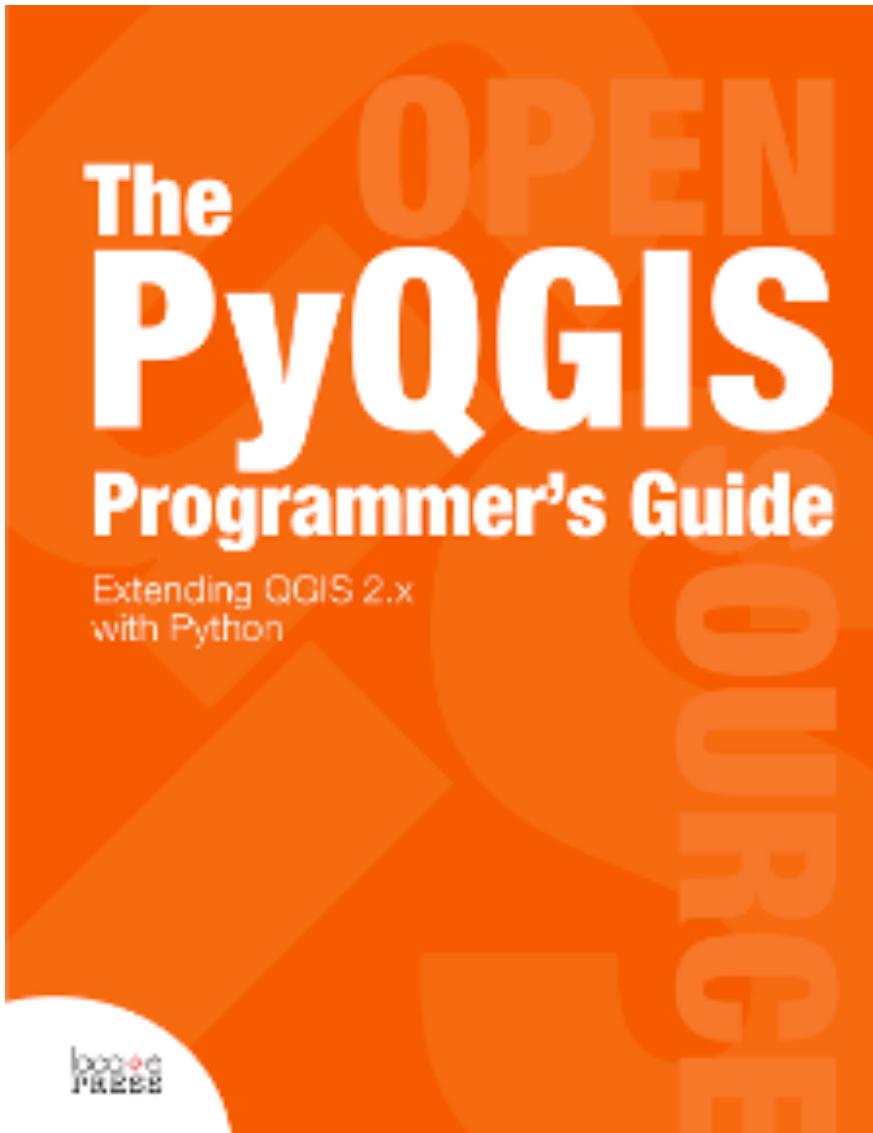
Official explanation for plug-in development:

http://docs.qgis.org/2.18/en/docs/pyqgis_developer_cookbook/plugins.html

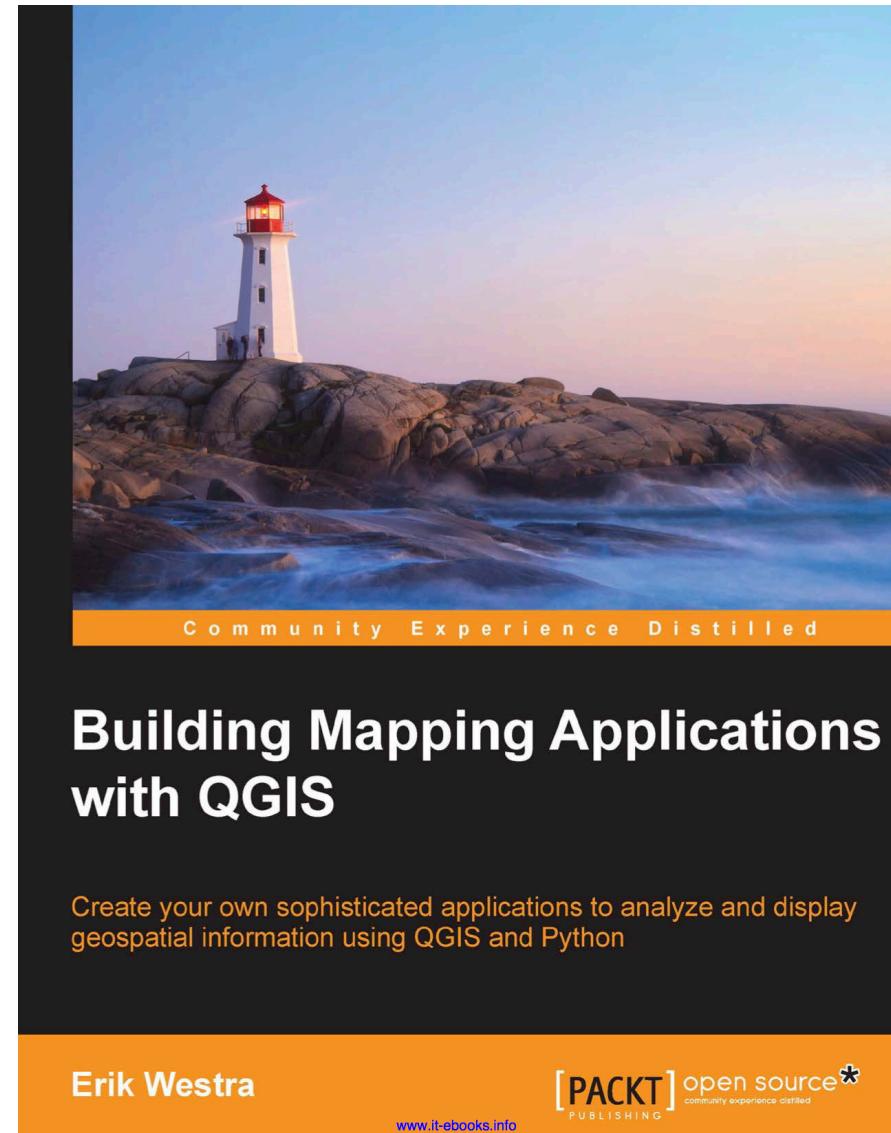
QGIS API – the complete reference (C++ style):

<http://qgis.org/api/2.18/>

<http://qgis.org/api/2.18/classQgisInterface.html>



Chapters 3, 5, 10, 14



Chapters 3 and 4