

24. Modul készítése QGIS alá Python-nal

Készítsünk saját modult QGIS alá. Használjuk ehhez a Plugin Builder modult. Az új modul segítségével az egyes rétegek attribútum tábláit CSV fájlként tudjuk menteni.

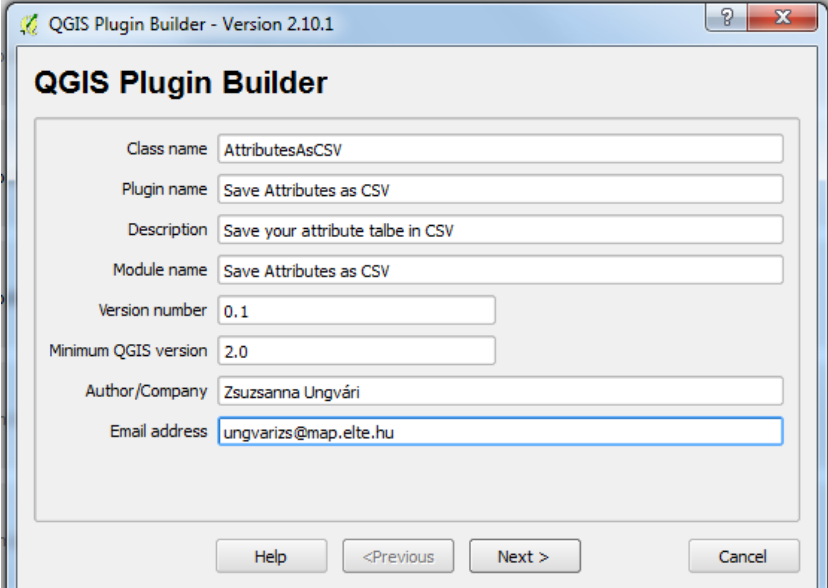
Ez a fejezet Ujaval Gandhi leckéje alapján készült.

[HTTP://WWW.QGISTUTORIALS.COM/EN/DOCS/BUILDING_A_PYTHON_PLUGIN.HTML](http://www.qgistutorials.com/en/docs/building_a_python_plugin.html)

Megoldás

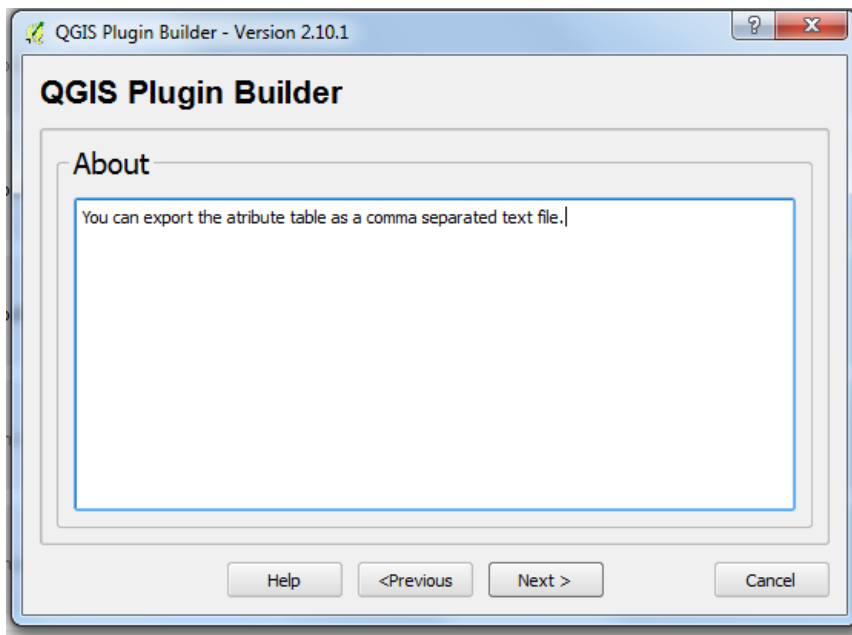
Telepítsük a Plugin Builder modult. Nyissuk meg a Plugins→Manage and Install Plugin menüt. Keressük meg az All-nál a Plugin Builder-t. Telepítsük, majd indítsuk el.

Töltsük ki a párbeszédpanelt.



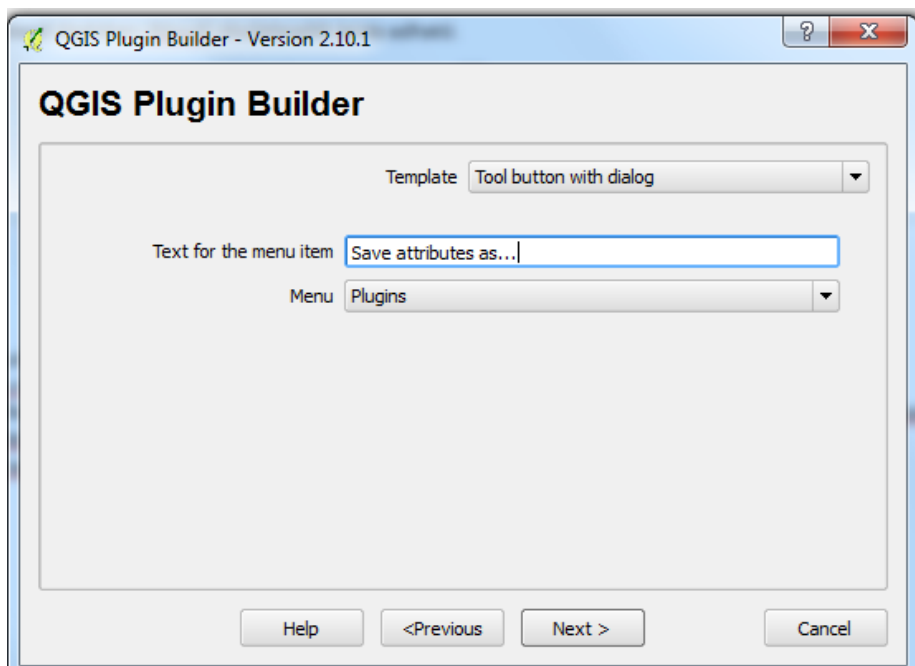
Megadhatjuk a plugin nevét (ez szerepel majd a fejlécben), az osztály nevét (Python osztály neve), a modul nevét, ebben ne legyen lehetőleg se ékezet, se szóköz. Ezen kívülé egy rövid leírást, a plugin verziószámát, a minimális QGIS verziót, a szerző nevét és e-mail címért.

A következő lépésben (About) részletesebb leírás adható.

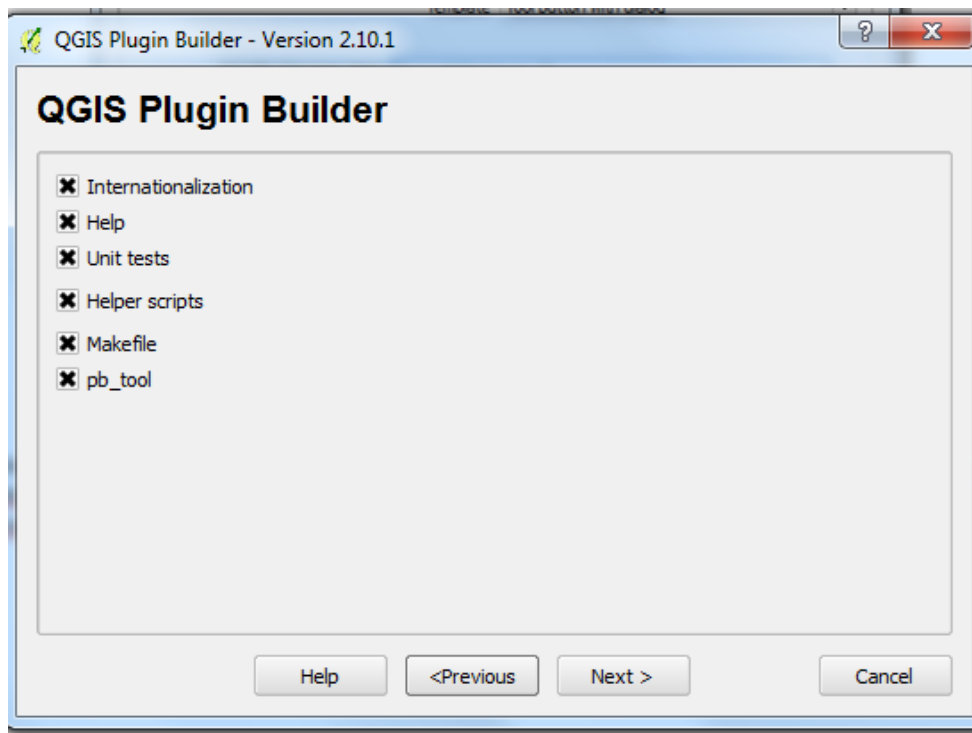


Válasszuk Template-nek a Tool button with dialog módot. Ezzel egy egyszerű párbeszédpanel jön létre.

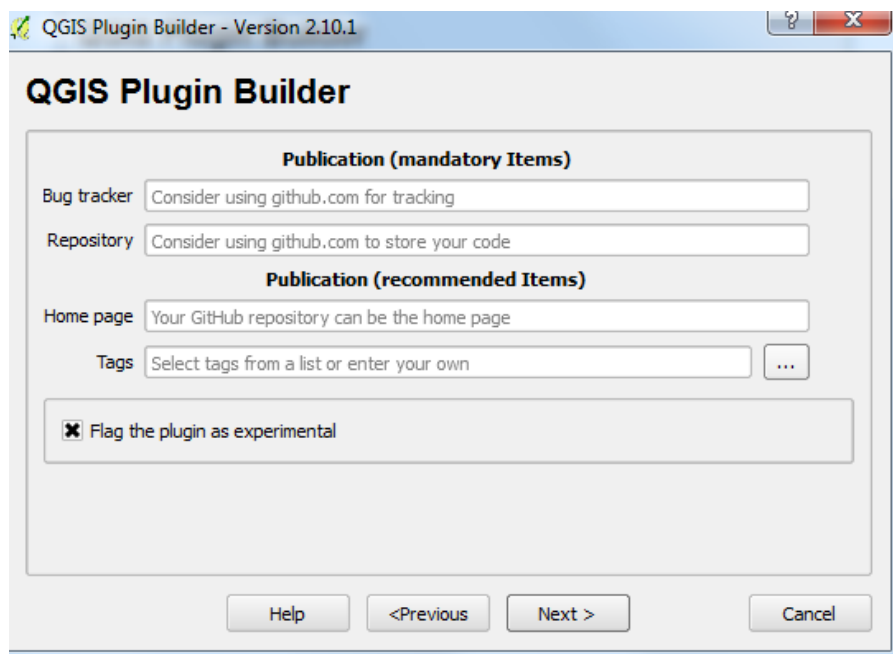
A Menu-nél kiválasztható, melyik Főmenübe kerüljön bele, legyen ez most a Plugin.



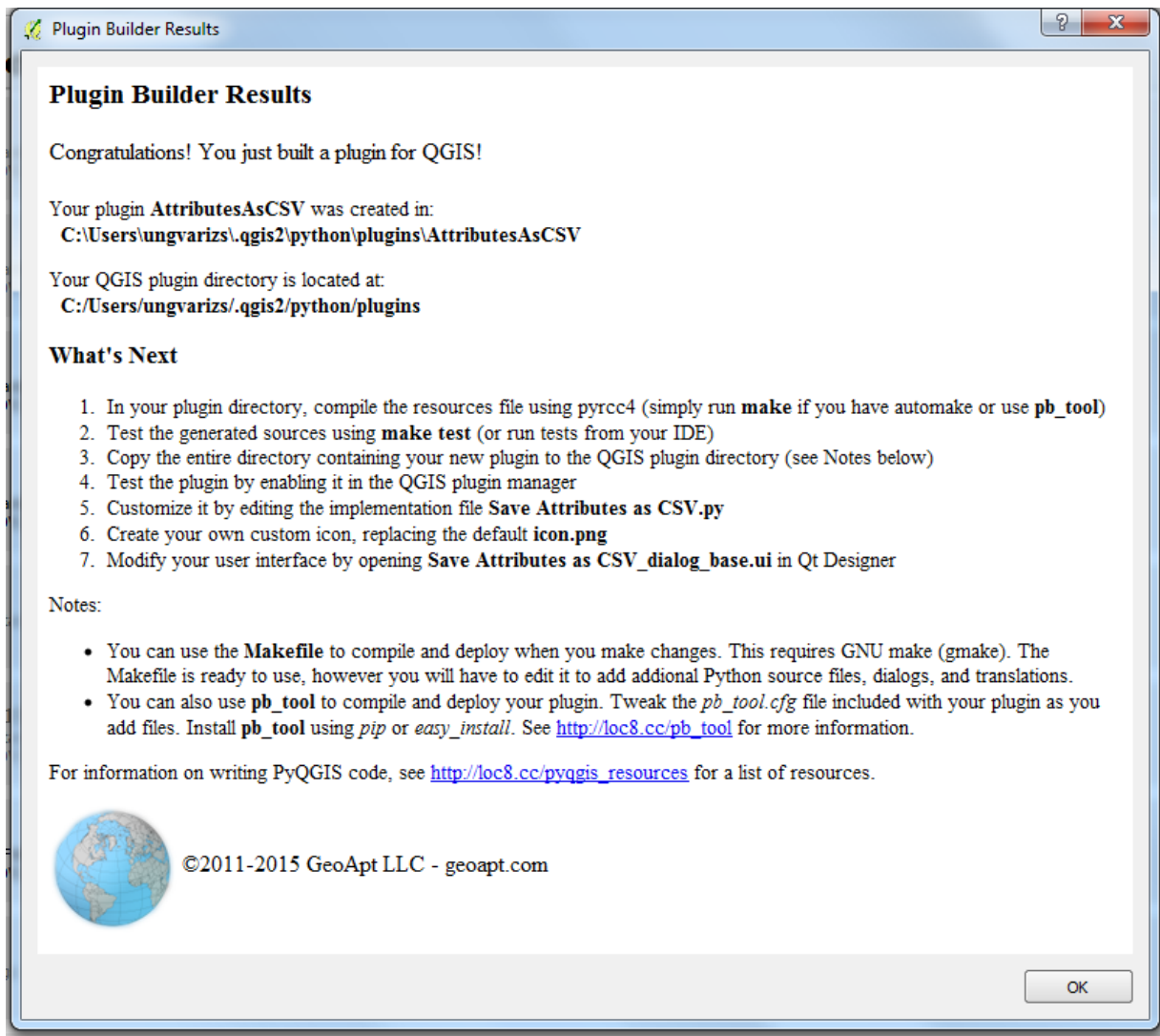
A következő lépésben megadható, milyen fájlokat készítsen el. (Internationalization: többnyelvűsítés, Help : Súlyó létrehozása), Unit tests (tesztelés), Helper scripts, Makefile: a modul futtatásra előkészítő szkript, pb_tool).



A Bug tracker-be, és a Repository-ba lehet majd feltölteni a kész kódot. Ha nem írunk be semmit, általában nem enged továbblépni. Később ez is módosítható a létrejött metadata.txt fájlban. A Flag the plugin as experimental azt jelenti, hogy a modul a kísérleti modulok közé kerül majd be. Ebben az esetben ne felejtjük el bekapcsolni a Manage and Install Plugin menüben a Settings-nél a Show also experimental plugin lehetőséget.



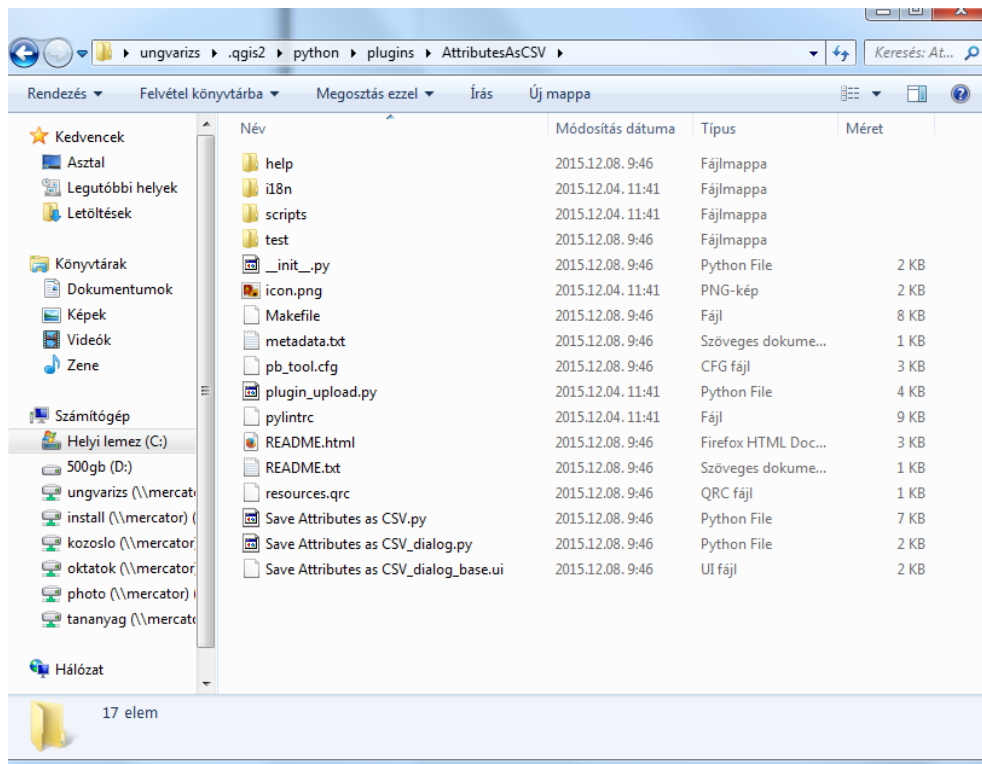
Következő lépésben ki kell választani a plugin helyét. A QGIS a plugineket a C:/Users/[sajátfelhasznalónev]/.qgis2/python/plugin mappába teszi, illetve innen olvassa be.



A fenti kép a sikeres létrehozást illusztrálja.

Nyissuk meg ezt a mappát, le kell fordítnunk a **RESOURCES . QRC** fájlt, hogy a QGIS a létrejött modult be tudja olvasni. Ezt a **PYRRC4 . EXE**-vel tudjuk megtenni a következő módon:
Nyissuk meg a konzolt.

(Linux alatt a Terminal ablak kell, Windows alatt a Start menüből a cmd.exe-t indítsuk el.)

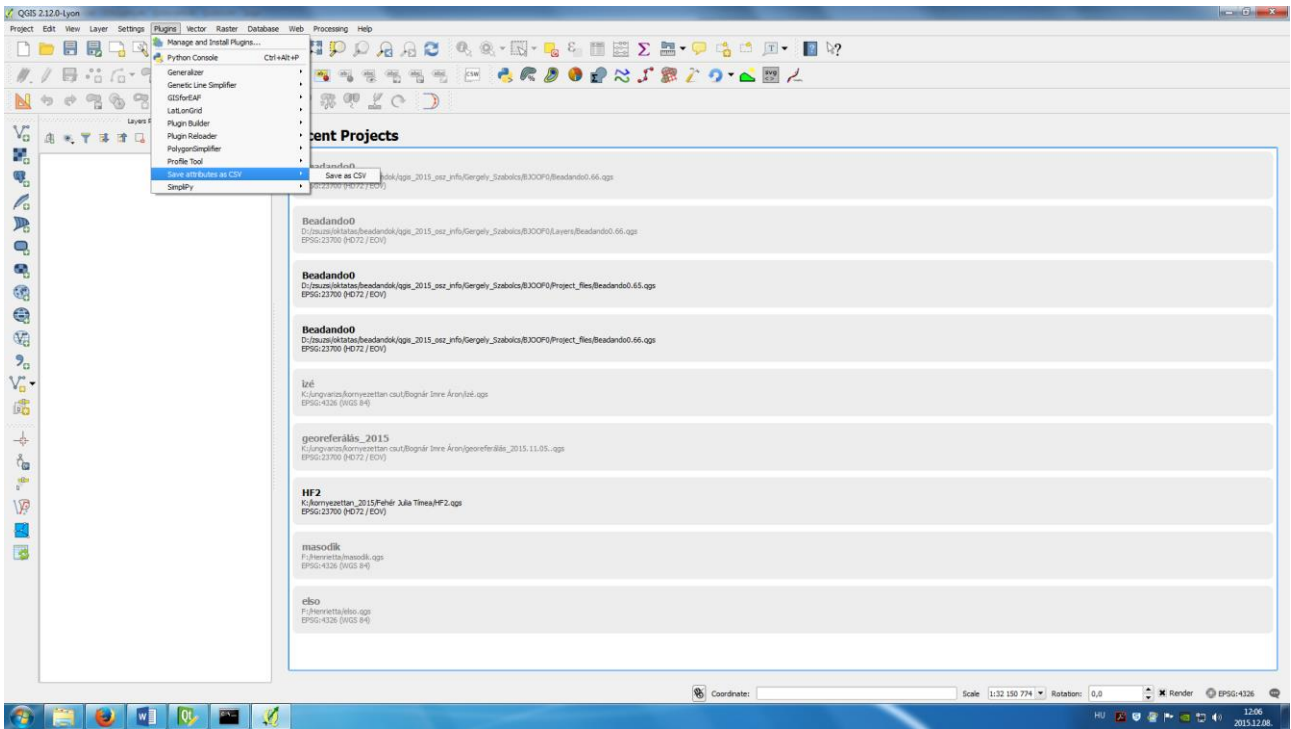


Lépjünk be a cd paranccsal a plugin mappájába: (cd=chdir parancs, mappaváltás)
`cd C:/Users/ungvarizs/.qgis2/plugins/save_csv`

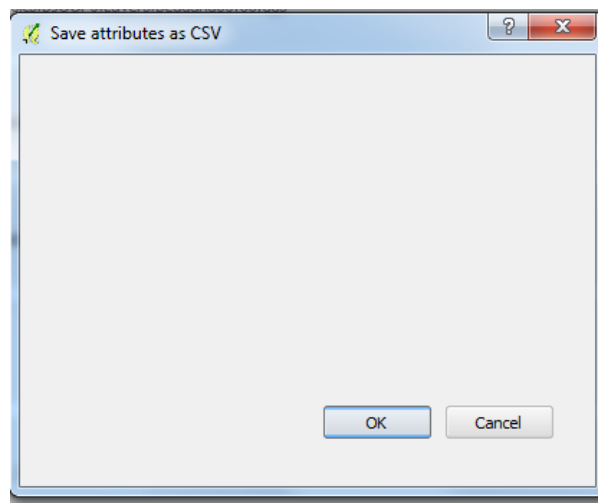
Fordítsuk le a kódot (QRC fájl) Python kódra (RESOURCES_RC.PY).
`C:/OSGeo4W64/BIN/PYRCC4.EXE -O RESOURCES_RC.PY RESOURCES.QRC`

Megjegyzés (a pyrcc4.exe a fent megadott könyvtárban van, ha áthelyeztük a .qgis2/python/plugins/ mappába a modulunkat, a pyrcc4-nek meg kell adni az elérési útját, különben addig nem fordítódik le a fájl.)

Ha hiba nélkül lefordult, nézzük meg QGIS-ben, a Plugin menüben:



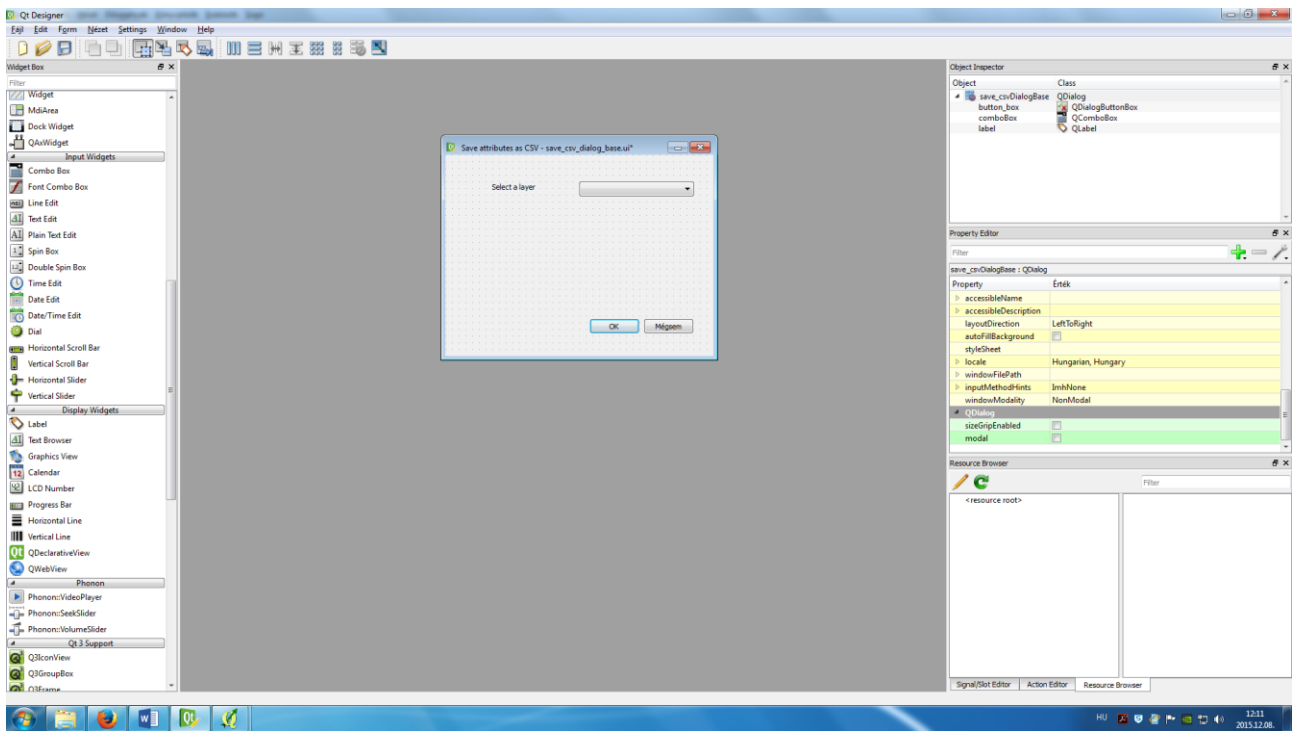
Így néz ki, egyelőre még nem csinál semmit. Most kell hozzá megadni a funkciókat.



Először is nyissuk meg Qt Designer programot a Start menüből. Ez az OSGEO4W installerrel települ.

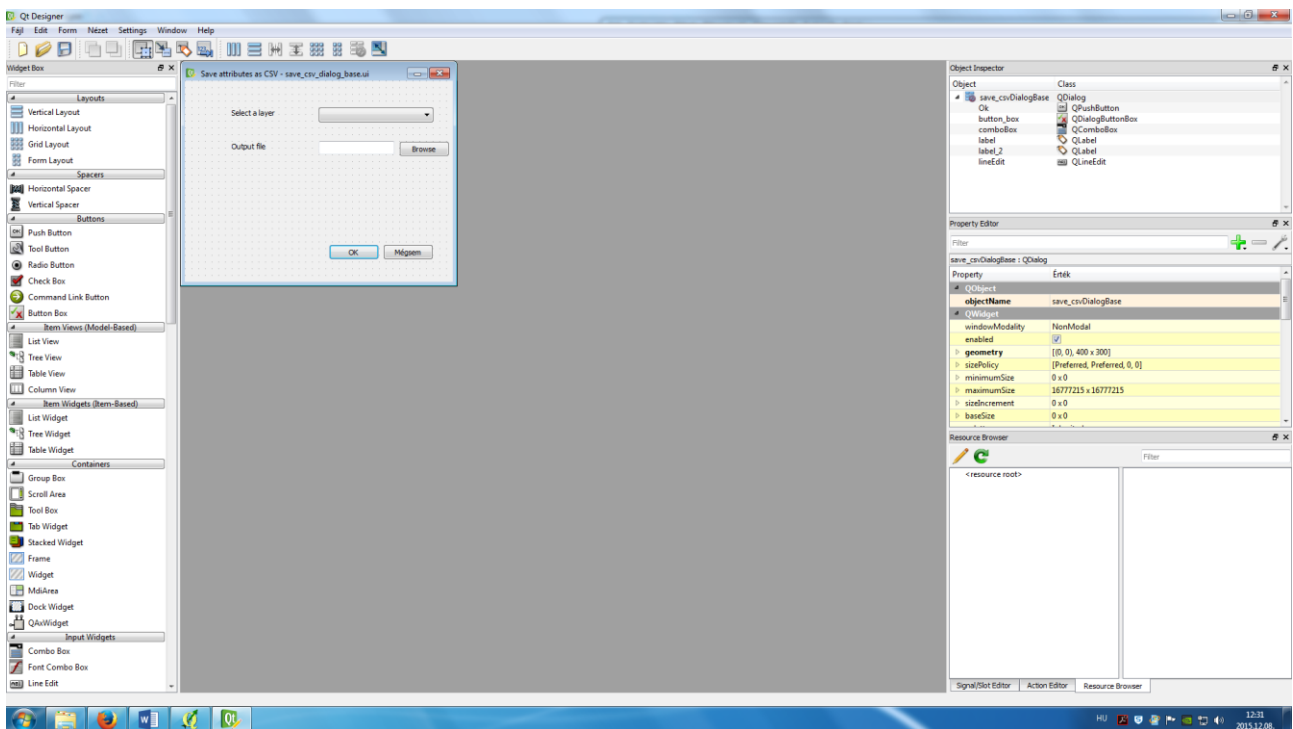
Ebben a programban tudjuk a plugin „kinézetét” egyszerűen megszerkeszteni, a program maga egy user interface designer, vagyis felhasználói felület tervező.

Nyissuk meg a modul mappájában lévő `SAVE_CSV_DIALOG_BASE.UI` UI kiterjesztésű fájlt.



Tegyük fel egy comboBox-ot, amellyel a felhasználó a behívott rétegek közül tud választani. Ehhez tegyük fel egy Text labelt is (szövege: Select layer), amelyre magyarázatokat fűzhetünk. Mindkettő az Input Widget-ben található, és csak rá kell húzni a modul felületére a bal egérgombbal.

Majd adjunk hozzá egy újabb Text labelt (Output file name felirattal), egy LineEditet, és hozzá „Browse” feliratú pushButton gombot. A Line editbe írható be majd az elérési út. A gomb segítségével pedig kiválaszthatjuk a mentés helyét, és a fájl nevét.



A modul felhasználói felülete ezzel készen van, most meg kell írunk azokat a kódrészleteket, amelyek a létrehozott objektumokat vezérlik.

Nyissuk meg a modul nevével azonos `.PY` kiterjesztésű fájlt a modul mappájából, vagyis az esetben a `SAVE_CSV.PY` fájlt.

A Python kódok bármilyen szövegszerkesztővel megnyithatók (pl. Notepad++ stb.), de erre a célra ajánlom az IDLE-t, ez a Python beépített szerkesztője.

A következő kódrészleteket kell beszúrni:

A fájl elején az importálandó moduloknál tegyük be a `QFileDialog`-o, ez az objektum a `PyQt4.Gui` modulban található.

```
from PyQt4.QtGui import QAction, QIcon, QFileDialog
```

A `run(self)` függvényhez szúrjuk be ezt a kódrészletet:

```
def run(self):
```

```
    """Töltsük be az épp behívott rétegeket a legördíthető listába, a comboBox-ba, ehhez  
    hozzunk létre egy listát. A layers nevű változóban listaként tárolódnak a rétegek nevei, ezeket  
    adjuk át a comboBox elemeinek, egyenként."""
```

```
        layers=self.iface.legendInterface().layers()
```

```
        layer_list=[]
```

```
        for layer in layers:
```

```
            layer_list.append(layer.name())
```

```
        self.dlg.comboBox.addItem(layer_list)
```

#Csináljunk egy új függvényt, amely ha a Browse gombra kattintunk, megadja a mentendő fájl nevét és teljes elérési útját.

```
def select_output_file(self):
```

```
    filename = QFileDialog.getSaveFileName(self.dlg, "Select output file ", "", '*.csv')
```

```
    self.dlg.lineEdit.setText(filename)
```

#Tegyük be az `__init__` metódus végére a következő két sort, ezzel töröljük a `lineEdit` szöveges box tartalmát (ha van), és feltöltjük a fájlnevel és elérési úttal.

```
self.dlg.lineEdit.clear()
```

```
self.dlg.pushButton.clicked.connect(self.select_output_file)
```

Végül írjuk meg a CSV fájlba írás kódját. Folytassuk a `run(self)` függvényt ezzel:

```
if result:
```

```
    filename=self.dlg.lineEdit.text() #Kivesszük a lineEdit tartalmát
```

```
    output_file=open(filename,'w') #Új szöveges fájl létrehozása írásra.
```

```
    selectedLayerIndex=self.dlg.comboBox.currentIndex() #szerezzük meg a kiválasztott réteg  
azonosítóját
```

```
    selectedLayer=layers[selectedLayerIndex]
```

```
    fields=selectedLayer.pendingFields() #Nézzük meg milyen mezők vannak a választott
```

```
rétegen
```

```
    fieldnames=[field.name() for field in fields]
```

```
    for f in selectedLayer.getFeatures(): #Menjünk végig a réteg elemein, és írjuk ki az adatokat  
fájlba
```

```
        line=','.join(unicode(f[x]) for x in fieldnames)+ '\n'
```

```
        unicode_line=line.encode('utf-8')
```

```
output_file.write(unicode_line)
output_file.close() #Zárjuk be a CSV-t, ezzel létrehozódik.
```

Készen vagyunk, teszteljük a modult!

Megjegyzés, kiegészítés

- A Python kis- és nagybetű érzékeny.
- Elgépeléskor, hibánál előfordulhat, hogy a modul nem töltődik be, nem látszik a menüsoron.
- A Python-ban szigorúan ügyelni kell a behúzásokra (tabulátorral vagy négy szóközzel), ugyanis ez jelzi, hogy az adott blokk a ciklus v. elágazás része-e. Egyes szövegszerkesztőkben ez nem követhető jól, ezért ajánlom az IDLE-t.
- A QtDesigner-ben a modulra felhelyezett objektumok (lineEdit, comboBox stb.) neve, értéke, megjelenése módosítható. Ehhez jelöljük ki az elemet, és a jobb oldali sávon, a Property Editorban változtassuk pl. a gomb szövegét a text tulajdonságnál, és az objectName-nél az objektum neve.