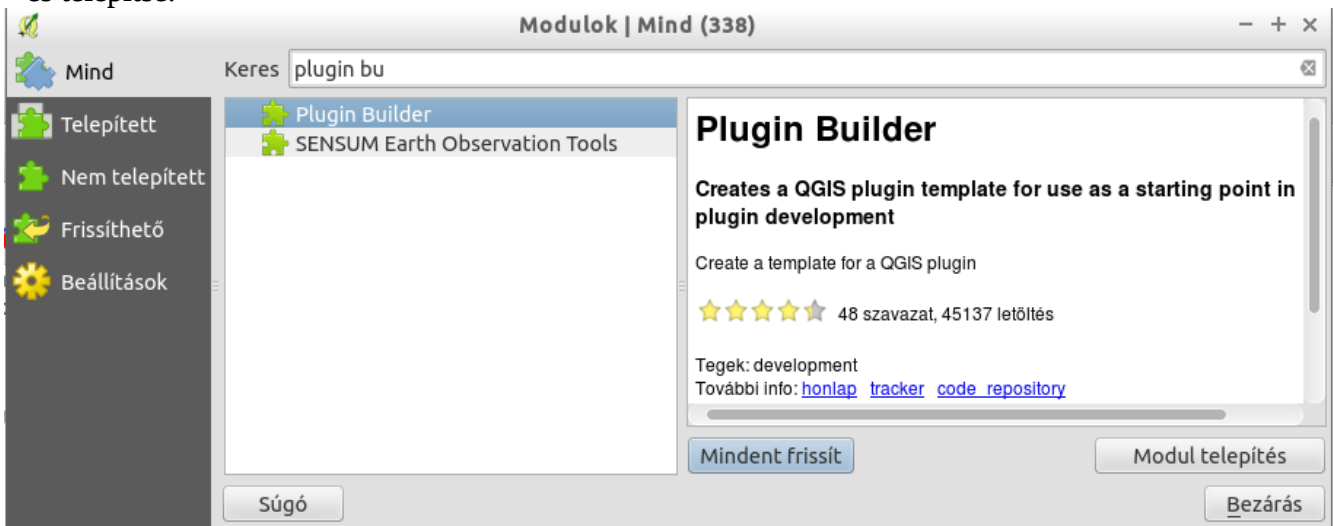


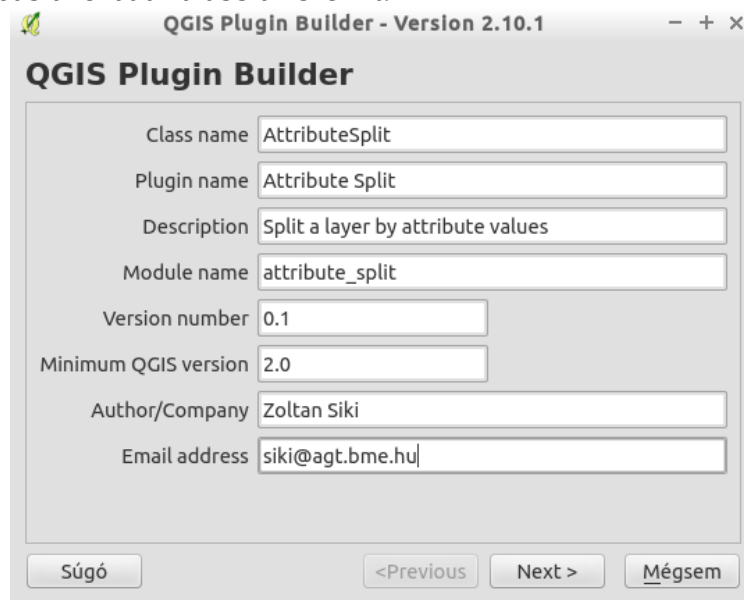
Python modul készítés QGIS 2.8

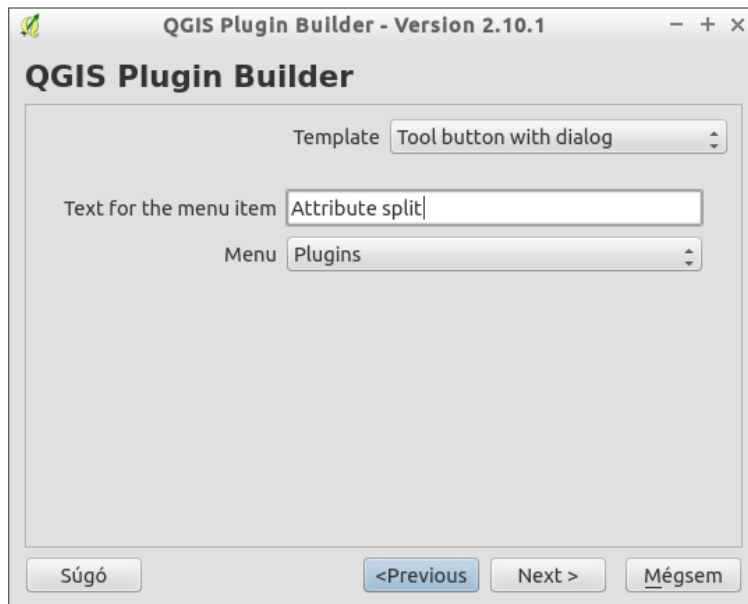
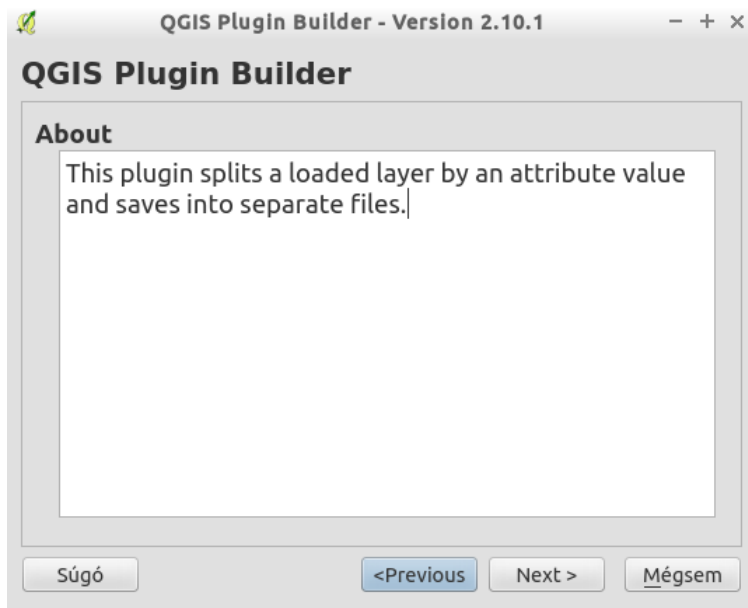
Egy Python modul készítését mutatom be, mely egy betöltött réteget tartalmát több shape fájlba menti egy attribútum érték alapján. Hasznos lehet például akkor, ha DXF fájlból betöltött adatokat az AutoCAD rétegeknek megfelelően szét szeretnénk választani.

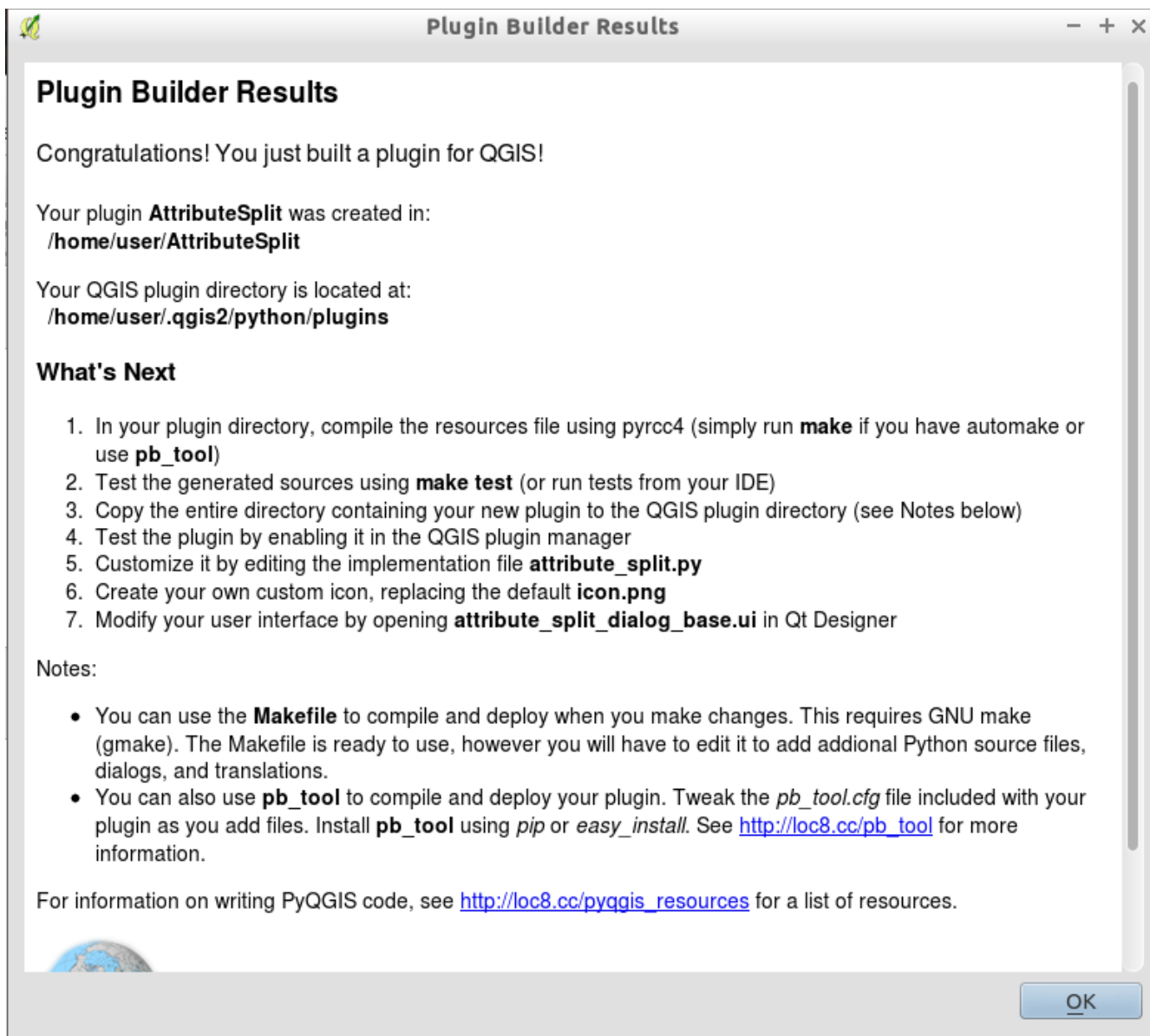
Először telepítsük a *Plugin Builder* modult. A **Modulok/Modul kezelés és telepítés** menüpontra kiválasztása után a megjelenő párbeszéd ablakban a **Mind** fülön keresse ki a *Plugin Builder* modult és telepítse.



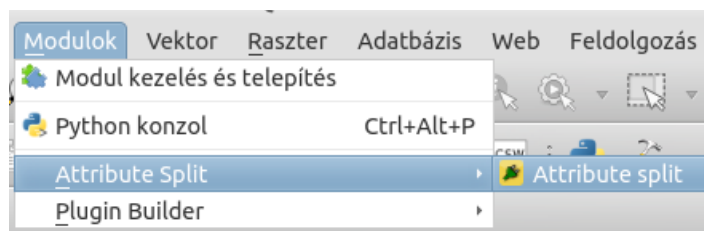
Az eszközsorban egy kárpitos kalapács jelenik meg, kattintson rá a telepített modul elindításához. Töltse ki a párbeszédablakokat az alábbiak szerint.



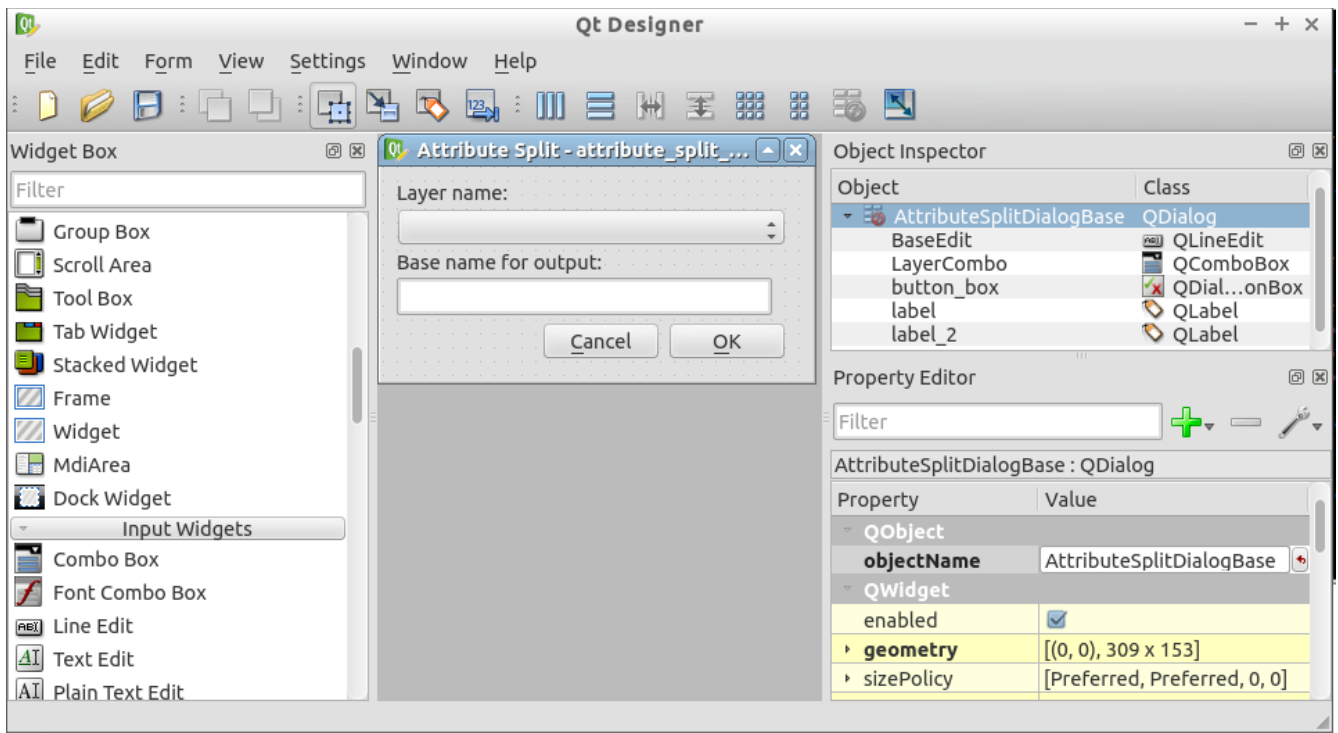




Lépjen ki a QGIS programból. Mozgassa át az AttributeSplit könyvtárat a `/home/user/.qgis2/python/plugin` könyvtárba, hogy a QGIS megtalálja. Lépjen be az új modul könyvtárába és adja ki a `make` parancsot, ez hozza létre a `resource.py` fájlt, mely a modul futtatásához szükséges. Majd indítsa újra a QGIS-t és kapcsolja be az új modult.



Már működőképes a modulunk, próbáljuk ki! Már van két nyomógombunk. A párbeszédablakot töltsük meg tartalommal a **Qt 4 Designer** segítségével, megtalálja a menüben vagy egy terminál ablakból indítsa el a `designer-qt4` programot és nyissuk meg az `attribute_split_dialog_base.ui` fájlt. A Display widgets csoportból húzzunk be két címkét (Label) az Input Widgets csoportból egy Combo Box-ot és egy LineEdit elemet, az alábbi ábra szerint.



Próbáljuk ki most is a módosított modulunkat, de előtte a modul könyvtárban adjuk ki a *make* parancsot. A modul elindítás után az új GUI elemeknek meg kell jelenniük.

Következő lépésben töltjük fel a rétegek legördülő listáját a vektoros rétegek neveivel. Ehhez használunk néhány segédfüggvényt a *realcentroid* modul *util.py* fájljából. Telepítsük a *realcentroid* modult és a *util.py* fájlt másoljuk át az *AttributeSplit* modul könyvtárba. Majd módosítsuk az *attribute_split_dialog.py* fájlt

At elejére tegyünk be néhány szükséges importot

```
from PyQt4 import QtGui, uic
from qgis.core import * # get some constants
from qgis.utils import * # get some utils
import util # get my utility function
```

Bővítsük a modult egy új tagfüggvénnyel

```
self.setupUi(self)
```

```
def showEvent(self, event):
    self.LayerCombo.clear() # remove previous entries
    self.BaseEdit.clear() # clear textbox
    names = util.getLayerNames([Qgis.Polygon, Qgis.Line, Qgis.Point])
    self.LayerCombo.addItem(names)
```

Importáljuk az `attribute_split.py` fájlba a `util.py`-t és a `VectorFileWriter`-t és a `QgsExpression`-t

```
import os.path
import util.py
from qgis.core import VectorFileWriter, QgsExpression, QgsFeatureRequest
```

És bővítsük ki a `run` metódust.

```
        result = self.dlg.exec_()
        # See if OK was pressed
        if result and len(self.dlg.LayerCombo.currentText()) and
len(self.dlg.BaseEdit.text()):
            vlayer = util.getMapLayerByName(self.dlg.LayerCombo.currentText())
            vprovider = vlayer.dataProvider()
            field = self.dlg.BaseEdit.text()
            fields = vprovider.fields()
            id = fields.indexFromName(field)
            uniquevalues = vprovider.uniqueValues(id)
            for uniquevalue in uniquevalues:
                writer = QgsVectorFileWriter(uniquevalue, vprovider.encoding(),
fields, vprovider.geometryType(), vprovider.crs())
                exptext = self.dlg.BaseEdit.text() + " = '" + uniquevalue + "'"
                exp = QgsExpression(exptext)
                request = QgsFeatureRequest(exp)
                for feature in vlayer.getFeatures(request):
                    writer.addFeature(feature)
                #writer.addFeatures(vlayer.getFeatures(request))

            del writer
```