

Автоматическое обновление репозитория расширений QGIS из SVN

[Обсудить в форуме](#) Комментариев — 2

Эта страница опубликована в основном списке статей сайта по адресу <http://gis-lab.info/qa/qgis-repo-update.html>

Советы разработчикам расширений как автоматически обновлять хранилище расширений QGIS прямо из SVN.

Одно из достоинств QGIS для разработчика — наличие репозитория расширений (пока только для Python). Так, в отличие от ArcScripts, где все хранится в едином хранилище, в QGIS такое хранилище может устроить каждый. Подключиться к нему может пользователь, знающий адрес этого репозитория. Помимо этого, так как к каждому плагину идет параметр «Version», менеджер расширений QGIS умеет проверять обновления плагинов ([подробнее](#)). Т.е. разработчику расширения остается только изменять версию и периодически выкладывать новый архив с расширением в репозиторий. Все прекрасно, но допустим у вас есть еще и SVN, в котором хранится исходный код расширения. В этом случае процесс обновления расширения становится несколько избыточным, так как после отправки кода в SVN, нужно его пакетировать, положить в репозиторий расширений, обновить xml репозитория согласно новой версии, чтобы QGIS увидел, что расширение обновилось.

Для того, чтобы автоматизировать этот процесс, можно использовать возможность SVN запускать нужный скрипт после нужного события, в нашем случае после коммита кода (post-commit).

Примечание: здесь и далее под «репозиторием» имеется в виду не хранилище кода расширения в SVN, а специализированное хранилище плагинов QGIS.

Подключение post-commit хука

Допустим расширение разработано, проверенно и работает локально. Репозиторий расширений создан, код расширения регулярно обновляется в SVN.

Создадим и сделаем исполняемым скрипт post-commit, срабатывающий после успешного коммита в svn.

```
cd PATH_TO_REPOS/hooks          # Путь к папке hook
cp post-commit.tmpl post-commit  # Включим хук, скопировав его из шаблона
chmod u+x post-commit           # Разрешим исполнять
```

Теперь, отредактируем наш post-commit, заставив его выполнить все нужные нам действия. Наш post-commit будет выполнять следующие действия:

1. Получать обновленный код через веб (wget)
2. Архивировать его в папку с расширением (zip)
3. Запускать программу на Python, обновляющую XML репозитория (getsetversion.py)
4. Отправлять сообщение администратору об обновлении (mail)

```
#!/bin/sh
```

```
REPOS="$1"
REV="$2"
```

```
NAME="testplugin"
NAMEP="TestPlugin"
```

```
cd /usr/local/www/programs/qgis
/usr/local/bin/wget -rNH http://svn.gis-lab.info/$NAME
cd $NAME
```

```
rm index.html
cd ..
/usr/local/bin/zip -mq9 $NAME.zip $NAME/*
rm -rf $NAME
/usr/local/bin/python getsetversion.py http://svn.gis-lab.info/$NAME/__init__.py $NAMEP
mail -s "$REPOS""$REV" sim@gis-lab.info
```

Приведенный выше код является bash-скриптом. Переменная \$REPOS содержит полный путь к svn-хранилищу, \$REV — последнюю ревизию кода, \$NAME — содержит имя папки, \$NAMEP — имя расширения как оно задано в XML репозитория.

Теперь рассмотрим как автоматически увеличивать версию расширения в XML репозитории.

Обновление XML-описания репозитория

Как уже разбиралась более подробно в статье «[Организация репозитория расширений QGIS](#)», информация о расширениях хранится в XML файле. Фрагмент, отвечающий за одно расширение, выглядит следующим образом:

```
<pyqgis_plugin name="TestPlugin" version="0.1.29">
  <description>This is the test plugin</description>
  <homepage>http://gis-lab.info/qa/qgis-dev-python.html</homepage>
  <qgis_minimum_version>1.0</qgis_minimum_version>
  <file_name>testplugin.zip</file_name>
  <author_name>GIS-Lab</author_name>
  <download_url>http://gis-lab.info/programs/qgis/testplugin.zip</download_url>
</pyqgis_plugin>
```

Как можно видеть, после обновления кода расширения нужно обновить только параметр version тэга pyqgis_plugin. Это и происходит при запуске этой строки из bash-скрипта, разобранный выше:

```
/usr/local/bin/python getsetversion.py http://svn.gis-lab.info/$NAME/__init__.py $NAMEP
```

Разберем код установки версии в файл repo.xml — getsetversion.py, где хранится и должна обновляться информация о расширениях.

В коде самого расширения версия хранится в файле __init__.py в фрагменте следующего типа:

```
def version():
    return "Version " + "0.1.32"
```

Будем извлекать ее оттуда, парсить XML описания репозитория и заменять в нем версию расширения на извлеченную.

```
import xml.dom.minidom,sys,urllib
from xml.dom.minidom import parse, parseString

xmlfile = "/usr/local/www/programs/qgis/qgis-repo.xml"
initfile = urllib.urlopen(sys.argv[1])
pluginname = sys.argv[2]

#in_file = open(initfile, 'rU')
text = initfile.readlines()
for line in text:
    if "Version" in line:
        if "return" in line:
            theline = line

theline = theline.replace('\n','')
alist = theline.split(' ')
cnt = len(alist)
version = alist[cnt-2]
version = version.replace(' ','')
```

```
parsedoc = parse(xmlfile)
elems = parsedoc.getElementsByTagName("pyqgis_plugin")
for elem in elems:
    valname = elem.attributes["name"].value
    if valname == pluginname:
        elem.attributes["version"].value = version

final = open(xmlfile, "w")
final.writelines(parsedoc.toxml())
```

Таким же способом можно делать резервные копии и много чего другого.

Теперь разработчик расширения делает только svn commit и никаких лишних действий вручную по сборке расширения для распространения. Пользователь же расширения, благодаря QGIS [автоматически извещается](#) о доступном обновлении.

Ссылки по теме

- [Разработка простого расширения для QGIS на Python](#)
- [Организация и работа с репозиториями расширений QGIS](#)

[Обсудить в форуме](#) Комментариев — 2

Последнее обновление: 2014-05-15 00:08

Дата создания: 19.02.2009

Автор(ы): [Максим Дубинин](#)