

PyQt for Autodesk Maya 2013 64bit

Reviewed by Cyrille Fauvel – Autodesk Developer Network (September 2012)

Additional Qt instructions available [here](#) -

<http://around-the-corner.typepad.com/adn/2012/10/building-qt-pyqt-pyside-for-maya-2013.html>

PyQt [<http://www.riverbankcomputing.co.uk>] is a python binding to the Qt library. Because Maya uses Qt internally, you can use the PyQt modules in Maya python scripts to create custom UI. PyQt does not have the same licensing as Maya, Qt, or Python. Please consult the PyQt website for information about licensing for PyQt.

Download PyQt: <http://www.riverbankcomputing.com/static/Downloads/PyQt4/>

Download SIP: <http://www.riverbankcomputing.com/software/sip/download>

The main change for building PyQt for Autodesk Maya 2013 (vs Maya 2012) is that when building PyQt for 2013 you will need to use the Microsoft Visual Studio 2010 Service Pack 1 compiler.

The following are instructions for building a copy of the PyQt modules that have been known to work with Maya.

Maya 2013 uses Qt4.7.1 which is binary compatible with the latest version of PyQt - 4.9.4 (at time of writing, September 2012).

Note that it's important to use the Maya modified version of the Qt source code. A copy of the customized Qt 4.7.1 source is available from Autodesk's Open Source web-site (<http://www.autodesk.com/igplsource>) and includes text files describing how to configure, build and install Qt for each platform supported by Maya.

Linux x86 – 64 bit

The following instructions only apply to Linux 64-bit builds of PyQt.

Download PyQt 4 and SIP from <http://www.riverbankcomputing.com/static/Downloads>

So for example for the latest version for Linux this would currently be:

Download PyQt: <http://www.riverbankcomputing.com/static/Downloads/PyQt4/PyQt4-x11-gpl-4.9.4.tar.gz>

Download SIP: <http://www.riverbankcomputing.com/static/Downloads/sip4/sip-4.13.3.tar.gz>

Building Qt

Follow the instructions for building Qt from the API docs (Developer Resources > API Guide > Setting up your build environment > Linux environments (64 bit), in the Maya Documentation).

*The following assumes Qt libraries have been installed into /usr/local/Trolltech/Qt-4.7.1/

Building SIP

1. cd sip-4.13.3
2. /usr/autodesk/maya2013-x64/bin/mayapy ./configure.py
3. make
4. sudo make install

This installs the SIP library to /usr/autodesk/maya2013-x64/lib/python2.6/site-packages/

Building PyQt

1. cd PyQt-x11-gpl-4.9.4
2. /usr/autodesk/maya2013-x64/bin/mayapy configure.py -q /usr/local/Trolltech/Qt-4.7.1/bin/qmake

The following message may be printed by the configure process:

Creating pyqtconfig.py...

Exception TypeError: TypeError("'NoneType' object is not callable",) in <bound method Popen.__del__ of <subprocess.Popen object at 0x1004ab7d0>> ignored

3. make -j4
4. sudo make install

This installs PyQt libraries into /usr/autodesk/maya2013-x64/lib/python2.6/site-packages/PyQt4

OSX x86 64 bit

The following instructions only apply to Mac 64-bit builds of PyQt.

Download PyQt 4 and SIP from <http://www.riverbankcomputing.com/static/Downloads>

So for example for the latest version for Mac OSX this would currently be:

Download PyQt: <http://www.riverbankcomputing.co.uk/static/Downloads/PyQt4/PyQt-mac-gpl-4.9.4.tar.gz>

Download SIP: <http://www.riverbankcomputing.co.uk/static/Downloads/sip4/sip-4.13.3.tar.gz>

Building Qt

Follow the instructions for building Qt from the API docs (Developer Resources > API Guide > Setting up your build environment > Mac OS X environment, in the Maya Documentation)

* The following assumes Qt has been installed into `/Users/myHomeDir/qt-4.7.1` per instructions given in the Maya modified source code build instructions.

Note that in case you build on Lion (10.7) that you need the MacOS Snow Leopard SDK (10.6) to build Qt and Maya plug-ins. Therefore add the `'-sdk /Developer/SDKs/MacOSX10.6.sdk'` to the Qt configure command line.

Building SIP

1. `tar -zxvf sip-4.13.3.tar.gz`
2. `cd sip-4.13.3`
3. `/Applications/Autodesk/maya2013/Maya.app/Contents/bin/mayapy ./configure.py -arch=x86_64`
4. `make`
5. `sudo make install`

This installs sip library into

`/Applications/Autodesk/maya2013/Maya.app/Contents/Frameworks/Python.framework/Versions/Current/lib/python2.6/site-packages`

Building PyQt

1. `tar -zxvf PyQt-mac-gpl-4.9.4.tar.gz`
2. `cd PyQt-mac-gpl-4.9.4`
3. `export QTDIR=/Users/myHomeDir/qt-4.7.1`
4. `export PATH=/Users/myHomeDir/qt-4.7.1/bin:$PATH`
5. `export QMAKESPEC=macx-g++`
6. `export DYLD_LIBRARY_PATH=/Users/myHomeDir/qt-4.7.1/lib`

7. `/Applications/Autodesk/maya2013/Maya.app/Contents/bin/mayapy configure.py`
`LIBDIR_QT=/Users/myHomeDir/qt-4.7.1/lib INCDIR_QT=/Users/myHomeDir/qt-4.7.1/include`
`MOC=/Users/myHomeDir/qt-4.7.1/bin/moc -w --no-designer-plugin`

The following message may be printed by the configure process:

Creating pyqtconfig.py...

Exception TypeError: TypeError("'NoneType' object is not callable",) in <bound method Popen.__del__ of <subprocess.Popen object at 0x1004ab7d0>> ignored

8. `grep -r warnold ./*`

for all files which are returned (if any), edit the file and replace the path

/Users/warnold/qt/qt-4.7.1

by your own path (i.e we used /Users/myHomeDir/qt-4.7.1 in the Qt build instructions file coming with the modified Autodesk Qt source - use the path you set in the '-prefix' flag). In theory, you should find these 3 following files – if not you're all good to continue.

./pyqtconfig.py

./qtdirs.out

./QtCore/Makefile

9. `make -j 8`
10. `sudo make install`

This installs PyQt libraries into

/Applications/Autodesk/maya2013/Maya.app/Contents/Frameworks/Python.framework/Versions/Current/lib/python2.6/site-packages/PyQt4

At this point we have PyQt installed, but the binaries are improperly linked for Maya. So you must correct that.

11. `for mod in phonon QtCore QtDeclarative QtDesigner QtDesignerComponents QtGui QtHelp QtMultimedia QtNetwork QtOpenGL QtScript QtScriptTools QtSql QtSvg QtWebKit QtXml QtXmlPatterns; do sudo find /Applications/Autodesk/maya2013/Maya.app/Contents/Frameworks/Python.framework/Versions/Current/lib/python2.6/site-packages/PyQt4 -name "*so" -exec install_name_tool -change ${mod}.framework/Versions/4/{mod} @executable_path/${mod} {} \;; done;`

Maya only includes the Qt binaries that it actually uses, so there are several installed PyQt modules which will not work because they won't find the missing libraries. This can in theory be solved by copying in the missing libraries from the Qt build earlier.

Windows x64

The following instructions only apply to 64-bit builds of Windows PyQt. You will need to have Visual Studio 2010 SP1 with x64 tools installed. These instructions may also work with Visual Studio Express 2010 Edition.

Download PyQt 4 and SIP from <http://www.riverbankcomputing.com/static/Downloads>

So for example for the latest version for Windows this would currently be:

Download PyQt: <http://www.riverbankcomputing.com/static/Downloads/PyQt4/PyQt-win-gpl-4.9.4.zip>

Download SIP: <http://www.riverbankcomputing.com/static/Downloads/sip4/sip-4.13.3.zip>

Building Qt

Follow the instructions for building Qt from the API docs (Developer Resources > API Guide > Setting up your build environment > Windows environment (32-bit and 64-bit), in the Maya Documentation)

* The following assumes you have built Qt x64 into c:\qt-adsk-4.7.1

Building SIP

1. Extract the archive to a folder (e.g. C:\sip-4.13.3)
2. Start up a Visual Studio 2010 x64 Win64 Command Prompt
3. cd C:\sip-4.13.3
4. set LIB=%LIB%;C:\Program Files\Autodesk\Maya2013\lib
(This is required so the build can find python26.lib)
5. "C:\Program Files\Autodesk\Maya2013\bin\mayapy" configure.py
6. nmake
7. nmake install

SIP is installed to Maya site-packages.

Building PyQt

1. Extract the archive to a folder (e.g. C:\PyQt-win-gpl-4.9.4)
2. Start up a Visual Studio 2010 x64 Win64 Command Prompt
3. cd C:\PyQt-win-gpl-4.9.4

The PyQt configure.py script searches the PATH for an installed QT. If you have other installed QTs, make sure they aren't in the PATH

4. set QTDIR=C:\qt-adsk-4.7.1
5. set PATH=C:\qt-adsk-4.7.1\bin;%PATH%
6. set QMAKESPEC=C:\qt-adsk-4.7.1\mkspecs\win32-msvc2010

7. "C:\Program Files\Autodesk\Maya2013\bin\mayapy" configure.py -w LIBDIR_QT=C:\qt-adsk-4.7.1\lib INCDIR_QT=C:\qt-adsk-4.7.1\include MOC=C:\qt-adsk-4.7.1\bin\moc.exe -w --no-designer-plugin
8. set INCLUDE=%INCLUDE%;C:\Program Files\Autodesk\Maya2013\include\python2.6;C:\Program Files\Autodesk\Maya2013\Python\include;C:\Python26\include
9. nmake

This may take a long time. If it results in errors referring to "D:\qt", search and replace all occurrences of "D:\qt\4.7.1-vc10" to "C:\qt-adsk-4.7.1" in all Makefiles under the PyQt folder, for example in Visual Studio 2010 you can use Edit > Find and Replace > Replace in Files.

10. nmake install

PyQt4 is installed to Maya site-packages.

Test PyQt in Maya

1. Start Maya
2. Start the Maya Script Editor and in a Python tab, copy and paste 1 of the samples below
3. Execute the code

Sample #1

```
import sip
sip.setapi('QString', 2)
sip.setapi('QVariant', 2)
from PyQt4 import QtGui, QtCore
btn = QtGui.QPushButton(QtGui.__file__)
btn.show()
```

Sample #2

```
import sip
import maya.OpenMayaUI as mui
from PyQt4.QtCore import *
from PyQt4.QtGui import *

def getMayaWindow():
    ptr = mui.MQtUtil.mainWindow()
    return sip.wrapinstance(long(ptr), QObject)

class Form(QDialog):
    def __init__(self, parent=None):
        super(Form, self).__init__(parent)
        self.setWindowTitle('Test Dialog')
        self.setObjectName('mainUI')
        self.mainLayout = QVBoxLayout(self)
        self.myButton = QPushButton('myButton')
        self.mainLayout.addWidget(self.myButton)

global app
global form
app = QApplication()
form = Form(getMayaWindow())
form.show()
```