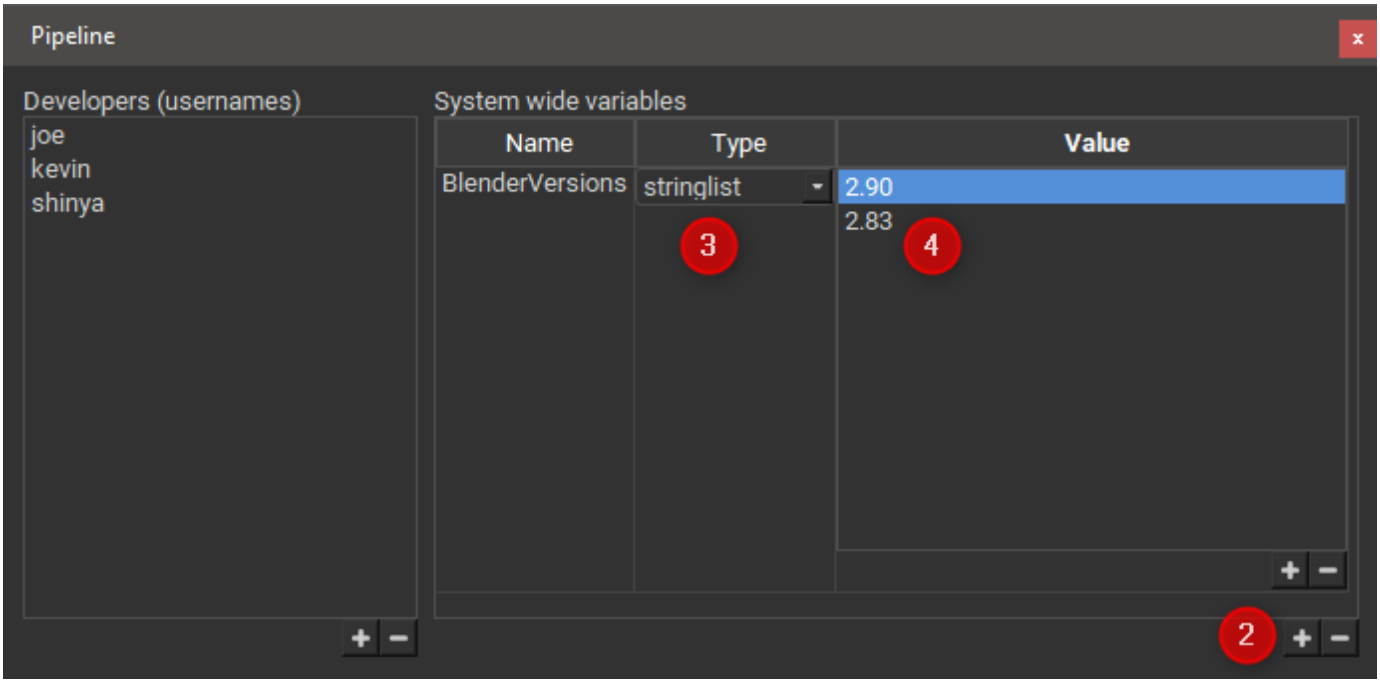
_qubeuiPanelPluginsHowTo

To create your own Panel Plugin you need to have access to the plugins/panels directory in the Qube! UI installation folder. We're going to create a panel that will open a selected job's Blender file in a specific version of Blender. Note that if you don't have Blender or Blender jobs then don't worry just replace anything to do with Blender with the application you want to open.

Using the Pipeline Panel we can add system-wide Pipeline Variables. We're going to add one to keep track of the versions of Blender we have installed on the users' workstations, note that these Pipeline Variables are set for everyone, not just you.

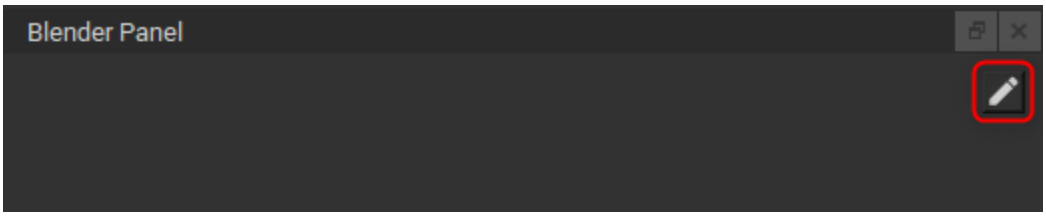
1. Open the Pipeline Panel.
2. Add a new Pipeline Variable and name it BlenderVersions in the dialog that appears.
3. Choose 'stringlist' for the type.
4. Add a couple of version numbers to the list, I have 2.83 and 2.90 installed so I'll add them.



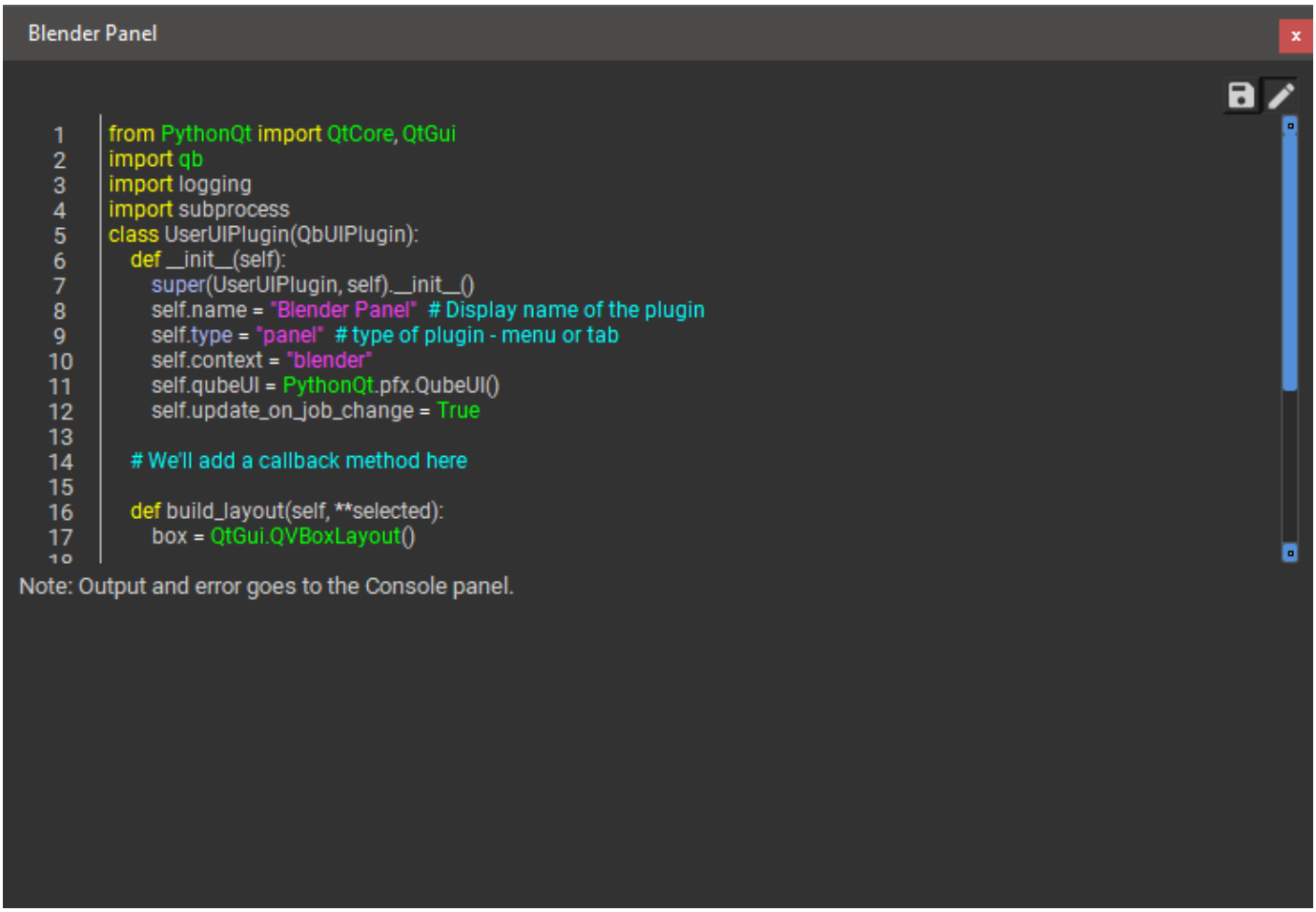
It's easiest to start with an existing panel plugin so you can make a copy of either of the two built-in examples, but for now just create a new .py python file and enter the following text:

```
from PythonQt import QtCore, QtGui
import qb
import logging
import subprocess
class UserUIPlugin(QbUIPlugin):
    def __init__(self):
        super(UserUIPlugin, self).__init__()
        self.name = "Blender Panel"  # Display name of the plugin
        self.type = "panel"  # type of plugin - menu or tab
        self.context = "blender"
        self.qubeUI = PythonQt.pfx.QubeUI()
        self.update_on_job_change = True

    # We'll add a callback method here

    def build_layout(self, **selected):
        box = QtGui.QVBoxLayout()

        # We'll add UI code here

        vspacer = QtGui.QSpacerItem(5, 5, QtGui.QSizePolicy.Fixed,
QtGui.QSizePolicy.Expanding);
        box.addItem(vspacer)
        hbox = QtGui.QHBoxLayout()
        hbox.addLayout(box)
        hspacer = QtGui.QSpacerItem(5, 5, QtGui.QSizePolicy.Expanding,
QtGui.QSizePolicy.Fixed);
        hbox.addItem(hspacer)
        return hbox
```

Save the file in the plugins/panels directory where the QubeUI executable sits on your platform. Make sure that you have write access to the file and open (or restart) the Qube! UI. You should see your new Panel Plugin, drag it somewhere sensible. If you're a Pipeline developer you should also see a pencil button in the top right, this allows you to edit the plugin inline.



Click it now and you should be looking at something like this.



Go to line 19, it should be a comment. Replace the comment with the following code.

```
# Make a new horizontal layout for our Blender version label and version combo.
        blenderVersionLayout = QtGui.QHBoxLayout()

        # Add a label for our version selector
        blenderVersionLabel = QtGui.QLabel("Blender version")
        blenderVersionLayout.addWidget(blenderVersionLabel)

        # Add a combo box and fill it with the contents of our pipeline variable
        self.blenderVersionCombo = QtGui.QComboBox()
        blenderVersions = self.qubeUI.pipelineVariable('BlenderVersions') # 1
        self.blenderVersionCombo.addItems(blenderVersions) # 2
        self.blenderVersionCombo.setFixedWidth(50)
        blenderVersionLayout.addWidget(self.blenderVersionCombo)
        box.addLayout(blenderVersionLayout)

        # Add a button that opens Blender
        openInBlenderButton =  QtGui.QPushButton()
        openInBlenderButton.text = 'Open in Blender'
        openInBlenderButton.connect('clicked()', self.openInBlender) # 3
        box.addWidget(openInBlenderButton)
```

1. Here we access our Pipeline Variable to give us a list of supported Blender versions.
2. We add the Blender versions to the version selector combo.

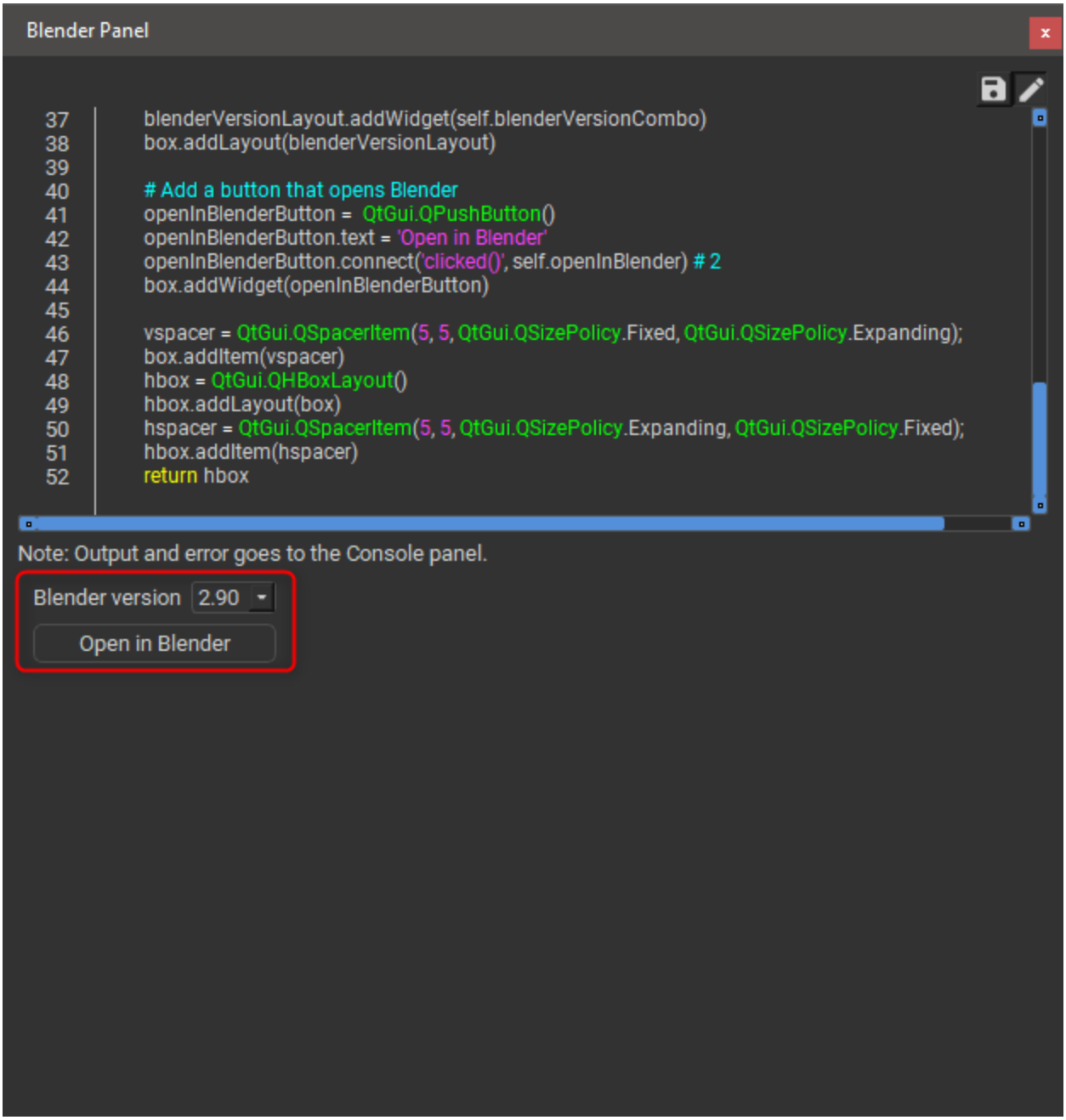
3. We connect a slot, we have yet to write, to the openInBlender button's clicked() signal.

Line 14 should be another comment, we'll replace that with the callback method code below.

```
# The openInBlenderButton's callback (slot)
    def openInBlender(self):
        jobs = self.qubeUI.selectedJobs() # 1
        for job in jobs:
            blender_path = "C:/Program Files/Blender Foundation/Blender
%s/blender.exe" % self.blenderVersionCombo.currentText # 2
            blender_scene_path = job.packages()['sceneFile'].value() # 3
            subprocess.call([blender_path, blender_scene_path]) # 4
```

1. Here we get the selected jobs from the Qube! UI as a list of Job objects.
2. We build the path to Blender using the currently selected blender version.
3. We use the job object to access the job's package and pull out the Blender scene file.
4. Finally we launch Blender with the scene file as the argument.

If you don't type anything for around 3 seconds your Panel Plugin should look something like this with our label, combo box and button all visible.



Save the plugin using the save button (the floppy disk button next to the pencil) and close the editor by clicking the pencil button. Select a Blender job in the a job list and click the "Open in Blender" button, if all went well the scene related to that job should open with the chosen version of Blender.