Creating a new SimpleCmd

Creating a SimpleCmd and adding it to the QubeGUI is a relatively straight-forward process. Since the QubeGUI will scan the simplecmds directory for any Python files, it will pick up any newly developed files. These files will contain a create() command that returns a list of SimpleCmd objects.

1. Open the simplecmds directory (SimpleCmds Location).
2. Create a text file with a Python (.py) extension and open it in your preferred text editor.
3. Add an import SimpleCmd to the top of the file. Then write a create() function that returns an array of SimpleCmd instances.
4. For testing purposes, add an if *__name__* == '*__main__': so the submission dialog can be launched directly by running Python *_script*.py. Here is a simple example:

Example

```
import sys

# You will need to have your PYTHONPATH, PATH, or modify sys.path
# in such a way that this import is possible. The path should be
# $QBDIR/api/python/qb/gui
from simplecmd import SimpleCmd

# user defined simpleCmds *must* define a create method.
def create():
 # This creates a simpleCmd called "Example Echo"
 # hasRange=False means that this will be a cmdline job - only one
 #  command for the job
 # canChunk=False means that there is no chunking option.  This is
 #  implied by hasRange=False.
 # help='...' provides the mouse-over tool tip and auto-generated
 #  documentation.
 cmdjob = SimpleCmd('Example Echo', hasRange=False,
      canChunk=False, help='basic echo test')

 # Adding an option group creates a collapsable, named section in
 # the submission UI.  It is purely cosmetic.
 cmdjob.add_optionGroup('Main options', collapsed=False)

 # add_option is the way to add parameters to the submission UI.
 # This one is of type 'string' which means a text field will be displayed
 # and associated with a variable called, in this case, 'message'.
 # The 'default' parameter pre-populates the field with the given text.  If
 #  the text is not changed, the value will be ignored.  Had we used 'value'
 #  instead of 'default', the given value will always be used, regardless of
 #  whether or not it was changed by the user.
 # The 'required' parameter tells the submission UI that this field must be
 #  populated with a non-default value in order for the submission to happen.
 #  it will also color the field red.
 cmdjob.add_option( 'message' , 'string' , 'Message to echo',
      default="Default message goes here", required=True)

 # This is the command template.  '%(message)s' will be replaced with
 # a parameter called 'message'.  We defined that parameter just above.
 cmdjob.command = 'echo %(message)s'

 # The create method *must* return a list of the SimpleCmd(s) it creates.
 return [cmdjob]

if __name__ == '__main__':
 import wx
 import submit
 import logging
 import simplecmd logging.basicConfig(level=logging.DEBUG)
 app = submit.TestApp(redirect=False)
 cmds = create()

 for cmd in cmds:
  simplecmd.createSubmitDialog(cmd)

 app.MainLoop()
```

5. Launch the QubeGUI. The new SimpleCmd should show up under the **Submit** menu.