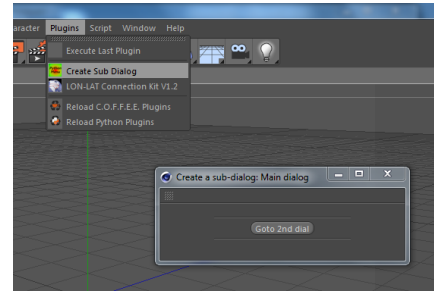


Python Tutorial – Creating a sub dialog

Here a short tutorial on creating and using sub dialogs.

Here is what we are going to do:

- 1) We create a simple plugins.CommandData plugin that will display the main dialog.
- 2) Once the button “Goto 2nd dialog” in the main dialog is selected, we are going to display a sub /second dialog.
- 3) This dialog (type MODAL) is closed again when the button ”Done” is selected.



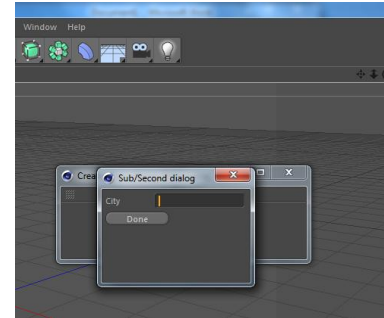
The main thing to remember here is that the first / the main dialog must be opened with the Pluginid and the sub /2nd dialog not! See also remark below.

Main dialog:

```
self.dialog.Open(dlgtype=c4d.DLG_TYPE_ASYNC, pluginid=PLUGIN_ID, defaultw=200, defaulth=150, xpos=-1, ypos=-1)
```

Sub / 2nd dialog

```
self.dialog2.Open(dlgtype=c4d.DLG_TYPE_MODAL, defaultw=200, defaulth=150, xpos=-1, ypos=-1)
```



Also, the main dialog is an ASYNC dialog and the sub dialog is a MODAL dialog.

This tutorial was possible thanks to Niklas with his help on http://www.plugincafe.com/forum/forum_topics.asp?FID=7 He made the following comment, which explained a lot of things:
“Pass the pluginid to a dialog only for the dialog that belongs to the CommandData itself. It is also only necessary when you want to restore the dialog (in CommandData.Restore()) which in turn only works for asynchronous dialogs.”

Let's have a look at the code.

Initialize:

- Load the plugin icon
- Register the CommandPlugin
- Class MyMenuPlugin(plugins.CommandData)
Here we open the main dialog using self.dialog.Open()

```
class MyMenuPlugin(plugins.CommandData):
    dialog = None
    def Execute(self, doc):
        # create the MAIN dialog
        if self.dialog is None:
            self.dialog = MyDialog()
        return self.dialog.Open(dlgtype=c4d.DLG_TYPE_ASYNC, pluginid=PLUGIN_ID, defaultw=200, defaulth=150, xpos=-1, ypos=-1)

    def RestoreLayout(self, sec_ref):
        # manage the MAIN dialog
        if self.dialog is None:
            self.dialog = MyDialog()
        return self.dialog.Restore(pluginid=PLUGIN_ID, secret=sec_ref)

if __name__ == "__main__":

    path, fn = os.path.split(__file__)
    bmp = bitmaps.BaseBitmap()
    bmp.InitWith(os.path.join(path, "res/icons/", "icon.tif"))

    okyn = plugins.RegisterCommandPlugin(PLUGIN_ID, "Create Sub Dialog", 0, bmp, "CSD", MyMenuPlugin())
    if (okyn): print "Initialized."
```

Main dialog class MyDialog(gui.GeDialog)

- Create the dialog (layout(self))
Normally you would also Init the dialog, but for this simple dialog, this is not done.
- Handle the dialog (Command(self, id, msg))
Handle the Button MY_BUTTON
When this button is selected the sub dialog is defined and displayed.
self.dialog2.Open(dlgtype=c4d.DLG_TYPE_MODAL, defaultw=200, defaulth=150, xpos=-1, ypos=-1)
Again, the Pluginid is not needed and we define a MODAL type dialog

class MyDialog(gui.GeDialog):

```
def CreateLayout(self): #main dialog
    self.SetTitle("Create a sub-dialog: Main dialog")
    self.GroupBegin(id=1013, flags=c4d.BFH_SCALEFIT, cols=1)
    self.element = self.AddStaticText(id=1003, flags=c4d.BFH_SCALEFIT, name=" ")
    self.AddSeparatorH(300, flags=c4d.BFH_SCALE)
    self.AddButton(MY_BUTTON, c4d.BFV_MASK, initw=100, name="Goto 2nd dialog")
    self.AddSeparatorH(300, flags=c4d.BFH_SCALE)
    self.GroupEnd()
    return True

#handle main dialog
def Command(self, id, msg):
    if (id == MY_BUTTON):
        self.dialog2 = MyDialog2()
        self.dialog2.Open(dlgtype=c4d.DLG_TYPE_MODAL, defaultw=200, defaulth=150, xpos=-1, ypos=-1)
    return True
```

Sub dialog class MyDialog2(gui.GeDialog)

- Create the dialog (layout(self))
Normally you would also Init the dialog, but for this simple dialog, this is not done.
- Handle the dialog (Command(self, id, msg))
Handle the Button Done
Get the input and close the dialog.

So, in fact about the same as the main dialog!

class MyDialog2(gui.GeDialog): #Sub / Second dialog

```
def CreateLayout(self):
    self.SetTitle("Sub/Second dialog")
    self.GroupBegin(id=1011, flags=c4d.BFH_FIT, cols=2)
    self.element = self.AddStaticText(id=1003, flags=c4d.BFH_LEFT, initw=100, name="City")
    self.AddEditText(MY_CITY, c4d.BFH_SCALEFIT, 100, 0)
    self.GroupEnd()

    self.GroupBegin(id=1013, flags=c4d.BFH_SCALEFIT, cols=1)
    self.AddButton(MY_BUTTON, c4d.BFV_MASK, initw=100, name="Done")
    self.GroupEnd()
    return True

#handle sub-dialog
def Command(self, id, msg):
    if (id == MY_BUTTON):
        print "city: ", self.GetString(MY_CITY)
        self.Close()
    return True
```

Here is the complete code:

```
import c4d
import os
from c4d import gui, plugins, bitmaps

PLUGIN_ID = 1011310 # Test ID

MY_CITY      = 11001
MY_BUTTON    = 11004

class MyDialog2(gui.GeDialog): #Sub / Second dialog
    def CreateLayout(self):
        self.SetTitle("Sub/Second dialog")
        self.GroupBegin(id=1011, flags=c4d.BFH_FIT, cols=2)
        self.element = self.AddStaticText(id=1003, flags=c4d.BFH_LEFT, initw=100, name="City")
        self.AddEditText(MY_CITY, c4d.BFH_SCALEFIT, 100, 0)
        self.GroupEnd()

        self.GroupBegin(id=1013, flags=c4d.BFH_SCALEFIT, cols=1)
        self.AddButton(MY_BUTTON, c4d.BFV_MASK, initw=100, name="Done")
        self.GroupEnd()
        return True

    #handle sub-dialog
    def Command(self, id, msg):
        if (id == MY_BUTTON):
            print "city: ", self.GetString (MY_CITY)
            self.Close()
            return True
        return True

class MyDialog(gui.GeDialog):
    def CreateLayout(self): #main dialog
        self.SetTitle("Create a sub-dialog: Main dialog")
        self.GroupBegin(id=1013, flags=c4d.BFH_SCALEFIT, cols=1)
        self.element = self.AddStaticText(id=1003, flags=c4d.BFH_SCALEFIT, name=" ")
        self.AddSeparatorH(300, flags=c4d.BFH_SCALE)
        self.AddButton(MY_BUTTON, c4d.BFV_MASK, initw=100, name="Goto 2nd dialog")
        self.AddSeparatorH(300, flags=c4d.BFH_SCALE)
        self.GroupEnd()
        return True

    #handle main dialog
    def Command(self, id, msg):
        if (id == MY_BUTTON):
            self.dialog2 = MyDialog2()
            self.dialog2.Open(dlgtype=c4d.DLG_TYPE_MODAL, defaultw=200, defaulth=150, xpos=-1, ypos=-1)
            return True
        return True

### Initialize #####

class MyMenuPlugin(plugins.CommandData):
    dialog = None
    def Execute(self, doc):
        # create the MAIN dialog
        if self.dialog is None:
            self.dialog = MyDialog()
            return self.dialog.Open(dlgtype=c4d.DLG_TYPE_ASYNC, pluginid=PLUGIN_ID, defaultw=200, defaulth=150, xpos=-1, ypos=-1)

    def RestoreLayout(self, sec_ref):
        # manage the MAIN dialog
        if self.dialog is None:
            self.dialog = MyDialog()
            return self.dialog.Restore(pluginid=PLUGIN_ID, secret=sec_ref)

if __name__ == "__main__":

    path, fn = os.path.split(__file__)
    bmp = bitmaps.BaseBitmap()
    bmp.InitWith(os.path.join(path, "res/icons/", "icon.tif"))

    okyn = plugins.RegisterCommandPlugin(PLUGIN_ID, "Create Sub Dialog",0, bmp, "OSM", MyMenuPlugin())
    if (okyn): print "Initialized."
```