

Lay-out formatting Cinema 4d Dialog (R13)

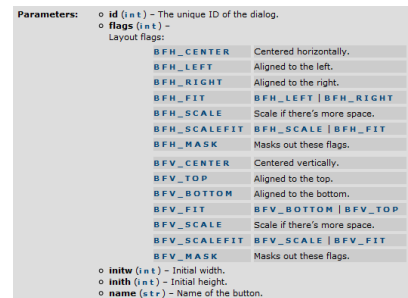
For this tutorial I used <http://villager-and-c4d.cocolog-nifty.com/blog/2011/12/c4d-python-r1-1.html>.

You define the widgets using some common arguments:

- widget **ID**
- **flags**, defining the layout
- **initw** (*int*) – Initial width.
- **inith** (*int*) – Initial height.
- specific for the widget

Note: Give each widget a unique id!

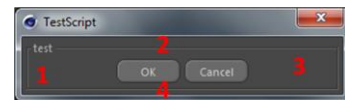
Note: Width / height are set automatically when you set it to 0.



Script 1: Grouping widgets

Use setTitle() to give your dialog a name.

You can 'group' widgets by placing them inside a Group. Additionally you can define groupborder and borderspace.



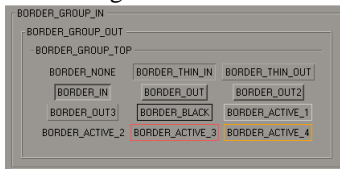
```
GroupBegin()  
GroupBorder()  
GroupBorderSpace()
```

Widgets ...

```
GroupEnd()
```

Note: Use [GroupBorderNoTitle\(\)](#) if you don't have a title. Otherwise there'll be a small gap in the border where the title would be.

Note: Border flags

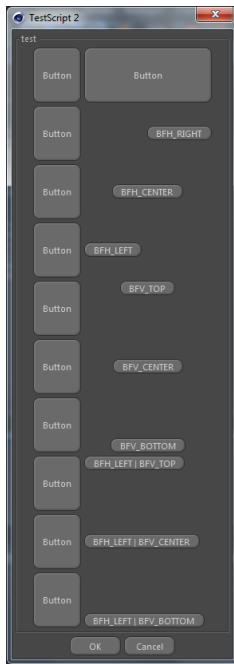


Note: Space Border GroupBorderSpace () are set in the clockwise direction from the left by the gap between the border and widgets (1.Left margin, 2.Top margin, 3.Right margin and 4.Bottom margin).

Python code:

```
self.SetTitle('TestScript')  
  
self.GroupBegin(10000, c4d.BFH_SCALEFIT, 1, title = 'test')  
self.GroupBorder(c4d.BORDER_GROUP_IN)  
self.GroupBorderSpace(20, 5, 20, 5)  
  
self.AddDlgGroup(c4d.DLG_OK|c4d.DLG_CANCEL)  
self.GroupEnd()
```

Script 2: Flags



Using flags you can define the lay-out of your widgets.

Note: Width / height are set automatically when you set initw and inith to 0.

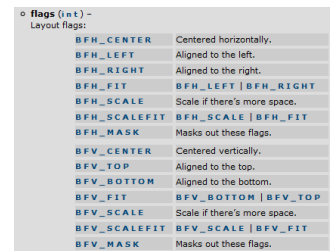
Python code:

```
self.GroupBegin(10000, c4d.BFH_SCALEFIT, 2, title = 'test')
self.GroupBorder(c4d.BORDER_GROUP_IN)
self.GroupBorderSpace(20, 5, 20, 5)
```

```
self.AddButton(1000, c4d.BFV_SCALEFIT, 0, 50, 'Button')
self.AddButton(1001, c4d.BFH_SCALEFIT, 0, 50, 'BFH_SCALEFIT')
self.AddButton(1002, c4d.BFV_SCALEFIT, 0, 50, 'Button')
self.AddButton(1003, c4d.BFH_RIGHT, 0, 0, 'BFH_RIGHT')
self.AddButton(1004, c4d.BFV_SCALEFIT, 0, 50, 'Button')
self.AddButton(1005, c4d.BFH_CENTER, 0, 0, 'BFH_CENTER')
self.AddButton(1006, c4d.BFV_SCALEFIT, 0, 50, 'Button')
self.AddButton(1007, c4d.BFH_LEFT, 0, 0, 'BFH_LEFT')
self.AddButton(1008, c4d.BFV_SCALEFIT, 0, 50, 'Button')
self.AddButton(1009, c4d.BFV_TOP, 0, 0, 'BFV_TOP')
self.AddButton(1010, c4d.BFV_SCALEFIT, 0, 50, 'Button')
self.AddButton(1011, c4d.BFV_CENTER, 0, 0, 'BFV_CENTER')
self.AddButton(1012, c4d.BFV_SCALEFIT, 0, 50, 'Button')
self.AddButton(1013, c4d.BFV_BOTTOM, 0, 0, 'BFV_BOTTOM')
self.AddButton(1014, c4d.BFV_SCALEFIT, 0, 50, 'Button')
```

```
self.AddButton(1015, c4d.BFH_LEFT|c4d.BFV_TOP, 0, 0, 'BFH_LEFT | BFV_TOP')
self.AddButton(1016, c4d.BFV_SCALEFIT, 0, 50, 'Button')
self.AddButton(1017, c4d.BFH_LEFT|c4d.BFV_CENTER, 0, 0, 'BFH_LEFT | BFV_CENTER')
self.AddButton(1018, c4d.BFV_SCALEFIT, 0, 50, 'Button')
self.AddButton(1019, c4d.BFH_LEFT|c4d.BFV_BOTTOM, 0, 0, 'BFH_LEFT | BFV_BOTTOM')

self.GroupEnd()
```



Script 3: Checkbox, Button

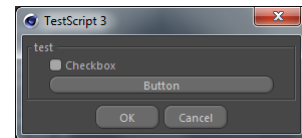
GeDialog.AddCheckbox (id, flags [, initw = 0] [, inith = 0] [, name ="])
 GeDialog.AddButton (id, flags [, initw = 0] [, inith = 0] [, name =")

Nothing special, except **name** defining the text displayed.

Python code:

```
self.GroupBegin(10000, c4d.BFH_SCALEFIT, 1, title = 'test')
self.GroupBorder(c4d.BORDER_GROUP_IN)
self.GroupBorderSpace(20, 5, 20, 5)

self.AddCheckbox(1001, c4d.BFH_SCALEFIT, 0, 0, 'Checkbox')
self.AddButton(1002, c4d.BFH_SCALEFIT, 0, 0, 'Button')
self.GroupEnd()
```



Script 4: Static text

GeDialog.AddStaticText (id, flags [, initw = 0] [, inith = 0] [, name ="] [, borderstyle = 0])

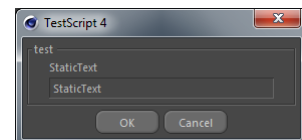
Besides the Group borderstyle, you can also define the borderstyle for StaticText. Border style is the same as the border of the group.

Python code:

```
self.GroupBegin(10000, c4d.BFH_SCALEFIT, 1, title = 'test')
self.GroupBorder(c4d.BORDER_GROUP_IN)
self.GroupBorderSpace(20, 5, 20, 5)

self.AddStaticText(1000, c4d.BFH_SCALEFIT, 0, 0, 'StaticText')
self.AddStaticText(1001, c4d.BFH_SCALEFIT, 0, 0, 'StaticText', c4d.BORDER_THIN_IN)

self.GroupEnd()
```

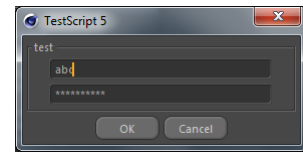


Script 5: Edittext

GeDialog.AddEditText (id, flags [, initw = 0] [, inith = 0] [, name = "] [, editflags = 0])

Edit flags: *EDITTEXT_PASSWORD* Password field.

When you enter text in a password field, text appears as ***.



Python code:

```
self.GroupBegin(10000, c4d.BFH_SCALEFIT, 1, title = 'test')
self.GroupBorder(c4d.BORDER_GROUP_IN)
self.GroupBorderSpace(20, 5, 20, 5)

self.AddEditText(1000, c4d.BFH_SCALEFIT, 100, 0)
self.AddEditText(1002, c4d.BFH_SCALEFIT, 100, 0, c4d.EDITTEXT_PASSWORD)

self.GroupEnd()
```

Script 6: MultiLine Edittext

GeDialog.AddMultiLineEditText (id, flags [, initw = 0] [, inith = 0] [, name = "] [, style = 0])

Using the style options you can define how the text is displayed.

style (int) –

A combination of the following flags:

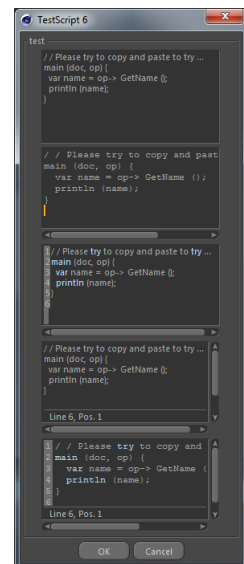
<i>DR_MULTILINE_MONOSPACED</i>	Monospaced font.
<i>DR_MULTILINE_SYNTAXCOLOR</i>	C.O.F.F.E.E. syntax highlighting.
<i>DR_MULTILINE_STATUSBAR</i>	Display a statusbar with the cursor position.
<i>DR_MULTILINE_HIGHLIGHTLINE</i>	Highlight lines.
<i>DR_MULTILINE_READONLY</i>	Read only multi line field.
<i>DR_MULTILINE_PYTHON</i>	Python syntax highlighting.
<i>DR_MULTILINE_WORDWRAP</i>	Word-warp multi line field.

Python code:

```
self.GroupBegin(10000, c4d.BFH_SCALEFIT, 1, title = 'test')
self.GroupBorder(c4d.BORDER_GROUP_IN)
self.GroupBorderSpace(20, 5, 20, 5)

self.AddMultiLineEditText(1000, c4d.BFH_SCALEFIT, 0, 90)
self.AddMultiLineEditText(1001, c4d.BFH_SCALEFIT, 0, 90, c4d.DR_MULTILINE_MONOSPACED)
self.AddMultiLineEditText(1002, c4d.BFH_SCALEFIT, 0, 90, c4d.DR_MULTILINE_SYNTAXCOLOR)
self.AddMultiLineEditText(1003, c4d.BFH_SCALEFIT, 0, 90, c4d.DR_MULTILINE_STATUSBAR)
self.AddMultiLineEditText(1004, c4d.BFH_SCALEFIT, 0, 90, c4d.DR_MULTILINE_MONOSPACED
|c4d.DR_MULTILINE_SYNTAXCOLOR | c4d.DR_MULTILINE_STATUSBAR)

self.GroupEnd()
```



Script 7: EditNumber / EditNumberArrows / Slider / EditSlider

GeDialog.AddEditNumber(id, flags[, initw=80][, inith=0])

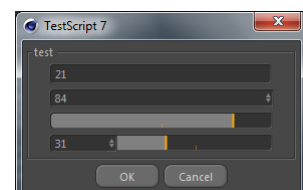
GeDialog.AddEditNumberArrows(id, flags[, initw=70][, inith=0])

GeDialog.AddSlider(id, flags[, initw=90][, inith=0])

GeDialog.AddEditSlider(id, flags[, initw=80][, inith=0])

These are the various ways to enter numbers:

Plain, with up- and down arrows and sliders.



Python code:

```
self.GroupBegin(10000, c4d.BFH_SCALEFIT, 1, title = 'test')
self.GroupBorder(c4d.BORDER_GROUP_IN)
self.GroupBorderSpace(20, 5, 20, 5)

self.AddEditNumber(1000, c4d.BFH_SCALEFIT, 80, 0)
self.AddEditNumberArrows(1001, c4d.BFH_SCALEFIT, 50, 0)
self.AddSlider(1002, c4d.BFH_SCALEFIT, 90, 0)
self.AddEditSlider(1003, c4d.BFH_SCALEFIT, 80, 0)

self.GroupEnd()
```

Script 8: Color, AddColorField / AddColorChooser

GeDialog.AddColorField (id, flags [, initw = 80] [, inith = 0])

GeDialog.AddColorChooser (id, flags [, initw = 80] [, inith = 0] [, layoutflags = False])

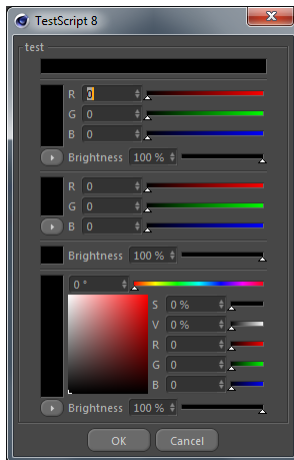
AddColorField is a bit unclear to me. I would use the ColorChooser directly. Clicking this field will open up the ColorChooser (Bodypaint Style).



layoutflags (int) –

AddColorChooser Flags:

<i>DR_COLORFIELD_NO_BRIGHTNESS</i>	Disable the brightness control.
<i>DR_COLORFIELD_NO_COLOR</i>	Disable the color control.
<i>DR_COLORFIELD_BODYPAINT</i>	Use the Bodypaint style.



Python code (separators are added to make it more clear):

```
self.GroupBegin(10000, c4d.BFH_SCALEFIT, 1, title = 'test')
self.GroupBorder(c4d.BORDER_GROUP_IN)
self.GroupBorderSpace(20, 5, 20, 5)

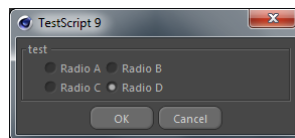
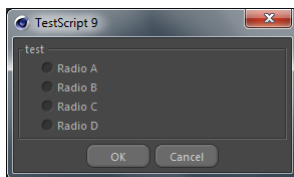
self.AddColorField(1000, c4d.BFH_SCALEFIT, 80, 12)

self.AddSeparatorH(0, flags=c4d.BFH_SCALEFIT)
self.AddColorChooser(1001, c4d.BFH_SCALEFIT, 80, 0)
self.AddSeparatorH(0, flags=c4d.BFH_SCALEFIT)
self.AddColorChooser(1002, c4d.BFH_SCALEFIT, 80, 0, c4d.DR_COLORFIELD_NO_BRIGHTNESS)
self.AddSeparatorH(0, flags=c4d.BFH_SCALEFIT)
self.AddColorChooser(1003, c4d.BFH_SCALEFIT, 80, 0, c4d.DR_COLORFIELD_NO_COLOR)
self.AddSeparatorH(0, flags=c4d.BFH_SCALEFIT)
self.AddColorChooser(1004, c4d.BFH_SCALEFIT, 80, 0, c4d.DR_COLORFIELD_BODYPAINT)

self.GroupEnd()
```

Script 9: Radiogroup

GeDialog.AddRadioGroup (id, flags [, columns = 0] [, rows = 0])



Using columns and rows, you can define how the radiogroup is grouped. The python code show the second example.

Python code (second screenshot with 2 rows and 2 columns):

```
self.GroupBegin(10000, c4d.BFH_SCALEFIT, 2, title = 'test')
self.GroupBorder(c4d.BORDER_GROUP_IN)
self.GroupBorderSpace(20, 5, 20, 5)

self.AddRadioGroup(1000, c4d.BFH_SCALEFIT, 2)
self.AddChild(1000, 0, 'Radio A')
self.AddChild(1000, 1, 'Radio B')
self.AddChild(1000, 2, 'Radio C')
self.AddChild(1000, 3, 'Radio D')

self.GroupEnd()
```

Script 10: Combobox

```
GeDialog.AddComboBox(id, flags[, initw=80][, inith=0][, specialalign=False])
```

Specialalign is not explained in the documentation (R13).

Python code:

```
self.GroupBegin(10000, c4d.BFH_SCALEFIT, 1, title = 'test')
self.GroupBorder(c4d.BORDER_GROUP_IN)
self.GroupBorderSpace(20, 5, 20, 5)

self.AddComboBox(1000, c4d.BFH_SCALEFIT, 80, 0, False)
self.AddChild(1000, 0, 'Combo A')
self.AddChild(1000, 1, 'Combo B')
self.AddChild(1000, 2, 'Combo C')
self.AddChild(1000, 3, 'Combo D')

self.GroupEnd()
```



Script 11: Separator Horizontal

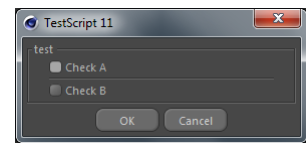
```
GeDialog.AddSeparatorH (initw[, flags=BFH_FIT])
```

Python code:

```
self.GroupBegin(10000, c4d.BFH_SCALEFIT, 1, title = 'test')
self.GroupBorder(c4d.BORDER_GROUP_IN)
self.GroupBorderSpace(20, 5, 20, 5)

self.AddCheckbox(1000, c4d.BFH_SCALEFIT, 0, 0, 'Check A')
self.AddSeparatorH(0, c4d.BFH_SCALEFIT)
self.AddCheckbox(1001, c4d.BFH_SCALEFIT, 0, 0, 'Check B')

self.GroupEnd()
```



Script 12: Separator Vertical

```
GeDialog.AddSeparatorV (inith[, flags=BFH_FIT])
```

Python code:

```
self.GroupBegin(10000, c4d.BFH_SCALEFIT, 3, title = 'test')
self.GroupBorder(c4d.BORDER_GROUP_IN)
self.GroupBorderSpace(20, 5, 20, 5)

self.AddCheckbox(1000, c4d.BFH_SCALEFIT, 0, 0, 'Check A')
self.AddSeparatorV(0, c4d.BFH_SCALEFIT)
self.AddCheckbox(1001, c4d.BFH_SCALEFIT, 0, 0, 'Check B')

self.GroupEnd()
```

