Tutorial: Manipulating layers in Cinema 4D using Python

I got a request on my blog.grooff.eu for a tutorial on manipulating layers using Python.
E.g. Creating new layers, adding objects to layers and adjusting the layer options (solo, etc.).

Note: A lot of the information / knowledge I got from reading the PluginCafe Python forum!

So, here we go, but first some basics.

Methods dealing with layers are exposed in several classes: LayerObject, BaseDocument and BaseList2D.
LayerObject (representation of a layer) inherits BaseList2D:
C4DAtom->GeListNode->BaseList2D->LayerObject.
Please have a look at the documentation for all the details.

Getting the information of all layers is done using GetLayerObjectRoot() and GetChildren().
This will return a list with all layers and checks whether the layer is active / selected:

```
root = doc.GetLayerObjectRoot()          #Gets the layer manager
LayersList = root.GetChildren()          #Get Layer list
LayersList = root.GetChildren()
for layers in LayersList:
        print layers.GetName()
        if (layers.GetBit(c4d.BIT_ACTIVE)): print "Active / selected"
```

Layers (layerobject) are, as many other things in Cinema 4D, derived from a GeListNode. Thus you can use the standard methods to navigate within a gelistnode tree (GetChildren, Next() and so on). Use basedocument.getlayerobjectroot() to get the root.

Layer data consists of:
dict{solo: bool, view: bool, render: bool, manager: bool, locked: bool, generators: bool, expressions: bool, animation: bool, color: Vector}

Note: Setting "solo" requires an additional "Button click simulation"

Creating a layer
There are several ways to create a layer:

- Use a CallCommand, c4d.CallCommand(100004738)
  See example CreateLayer()

- Use GeListNode methods. E.g. InserUnder.
  See example CreateLayer2() and CreateInsertUnder (parentname, childname)

Add an object to a layer
Just get the layer and link it to the object.

```
object[c4d.ID_LAYER_LINK] = layer
```

See example CreateObjectOnLayer(object, objectname, layer):

Layer options
You can set the following settings:
{solo: bool, view: bool, render: bool, manager: bool, locked: bool, generators: bool, expressions: bool, animation: bool, color: Vector}

Get the layer data and set the data and write the layer data back again.

```
layer_data = layer.GetLayerData(doc)
layer_data['solo'] = True / False
layers.SetLayerData(doc,layer_data)
CallButton(layers, 100004726)                # Button simulation. Only needed for solo
```

Note: Setting "solo" requires an additional "Button click simulation"

See example LayerSoloOnoff() and ToggleLayerView ().

Below the full source code of the example script with a lot of comments.
I used it on R13 and R14.
If you have any questions, please let me know.

```
#
# Tutorial (R14): Manipulate layers
# Pim Grooff, @2013
#

import c4d
from c4d import *

def CreateObjectOnLayer(object, objectname, layer):

    object[c4d.ID_BASELIST_NAME] = objectname
    object[c4d.ID_LAYER_LINK] = layer               #Set the layer
    doc.InsertObject(object)
    c4d.EventAdd()
    return True

def CreateLayer2 (name, layercolor):

    layer = c4d.documents.LayerObject()
    layer[c4d.ID_LAYER_COLOR] = layercolor

    # Set layer name (another option tp set the name)
    layer.SetName(name)

    # Get the invisible root layer
    layerRoot = doc.GetLayerObjectRoot()

    # Insert the layer under the parent
    layer.InsertUnder(layerRoot)

def CreateLayer (layername, layercolor):

    doc = c4d.documents.GetActiveDocument()
    root = doc.GetLayerObjectRoot()             #Gets the layer manager
    LayersList = root.GetChildren()             #Get Layer list

    # check if layer already exist
    layerexist = False
    for layers in LayersList:
        name = layers.GetName()
        if (name == layername): layerexist = True

    #print "layerexist: ", layerexist
    if (not layerexist):
        c4d.CallCommand(100004738)  # New Layer
        c4d.EventAdd()

        #rename new layer
        LayersList = root.GetChildren()   #redo getchildren, because a new one was added.
        for layers in LayersList:
            name = layers.GetName()
            if (name == "Layer"):
                layers.SetName(layername)
                layers.SetBit(c4d.BIT_ACTIVE)   # set layer active
                layers[c4d.ID_LAYER_COLOR] = layercolor
                c4d.EventAdd()


    return layers # end createlayer

def GetLayer (layername):

    doc = c4d.documents.GetActiveDocument()
    root = doc.GetLayerObjectRoot()     #Gets the layer manager
    LayersList = root.GetChildren()

    for layers in LayersList:
        name = layers.GetName()
        if (name == layername):
            return layers

    #gui.MessageDialog ("Layer does not exist: " + layername)
    return None #end GetLayer
```

```
def ToggleLayerView (layername):

    doc = c4d.documents.GetActiveDocument()
    root = doc.GetLayerObjectRoot()#Gets the layer manager
    LayersList = root.GetChildren()

    for layers in LayersList:
        name = layers.GetName()
        if (name == layername):         # if the layer is selected
            layer_data = layers.GetLayerData(doc)
            layer_data['view'] = not layer_data['view']
            layers.SetLayerData(doc,layer_data)
            c4d.EventAdd()

    return # end layerviewonoff

def LayerSoloOnoff (layername, onoff):

    # all properties:
    # dict{solo: bool, view: bool, render: bool, manager: bool,
    #       locked: bool, generators: bool, expressions: bool,
    #       animation: bool, color: Vector}

    doc = c4d.documents.GetActiveDocument()
    root = doc.GetLayerObjectRoot()#Gets the layer manager
    LayersList = root.GetChildren()

    for layers in LayersList:
        name = layers.GetName()
        if (name == layername):         # if the layer is selected
            layer_data = layers.GetLayerData(doc)

            layer_data['solo'] = onoff

            #layer_data['manager'] = True
            #layer_data['locked'] = True

            layers.SetLayerData(doc,layer_data)
            CallButton(layers, 100004726) # Button simulation. Only needed for solo
            c4d.EventAdd()

    return # end layerviewonoff

def CreateInsertUnder (parentname, childname):

    # Create a new layer
    layer = c4d.documents.LayerObject()

    # Set layer name (Calls C4DAtom.SetName())
    layer.SetName(childname)

    parentlayer = GetLayer (parentname)

    # Insert the layer under the parent
    layer.InsertUnder(parentlayer)

    c4d.EventAdd()

def main():

    layername = "layer 01"
    if (GetLayer (layername) is None):
        # create layer and set it as active
        layer = CreateLayer (layername, c4d.Vector(0,255,0))           #Create layer with color green

        # create sphere on active layer
        CreateObjectOnLayer(c4d.BaseObject(c4d.Osphere), "Sphere", layer)

    layer = CreateLayer ("layer 02", c4d.Vector(255,255,0))            #Create layer
    layer = CreateLayer ("layer 03", c4d.Vector(255,0,0))              #Create layer

    #set active layer back to "layer 02"
    layer = GetLayer ("layer 02")
    # create a Cube on active layer 02
    CreateObjectOnLayer(c4d.BaseObject(c4d.Ocube), "Cube", layer)
```

```
    # Toggle layer view setting
    # ToggleLayerView ("layer 01")

    # set Layer Solo off
    LayerSoloOnoff ("layer 01", True)

    CreateInsertUnder ("layer 02", "InsertUnder Layer02")

    CreateLayer2("layer 222 (white)", c4d.Vector(255,255,255))


if __name__=='__main__':
    main()
```