
Scribus Scripter API Documentation Documentation

Release version 1.5

Jain Basil Aliyas

Jun 21, 2017

Contents

1	Scripter2 API Specification	3
2	Tutorials	13
3	Python Scripts	15
4	Frequently Asked Questions	17
5	Indices and tables	19

Scribus is an Open Source program that brings professional page layout to Linux/UNIX, Mac OS X, OS/2 Warp 4/eComStation and Windows desktops with a combination of press-ready output and new approaches to page design. Underneath a modern and user-friendly interface, Scribus supports professional publishing features, such as color separations, CMYK and Spot Color support, ICC color management, and versatile PDF creation.

Scripter2 is newly developed Scripting Engine for Scribus which enables experienced users to develop python scripts to perform custom operations like generation of table of contents, barcode or more.

Scripter2 API Specification

Scripter

class **Scripter**

Scripter serves as the entry point to Scripting Engine of Scribus. It provides necessary methods to create new document, open an existing document etc., as documented in this section.

Scripter Attributes

`Scripter.activeDocument`

Instance of active document.

`Scripter.activeWindow`

Instance of active scribus window.

Scripter Methods

`Scripter.openDocument` (*docName*)

Open an existing scribus document.

DocumentAPI

class **DocumentAPI**

DocumentAPI provides properties and methods needed to interact with a Scribus document. An instance of DocumentAPI may be obtained from *Scripter*, for eg.

```
document = Scripter.activeDocument
```

DocumentAPI Attributes

DocumentAPI.name

Read the name of current document.:

```
print document.name
```

This will be empty for a new document.

DocumentAPI.available

Returns True if the document is available, else False.:

```
>>> document = Scripter.activeDocument
>>> document.available
True
>>>
```

DocumentAPI.margins

Return an object of type *Margins*

DocumentAPI.modified

This attribute is used to check whether the document is modified or not. It can also be used to set a document as modified.:

```
>>> doc = Scripter.activeDocument
>>> doc.modified
False
>>> doc.modified = True
>>> doc.modified
True
>>>
```

DocumentAPI.activePage

DocumentAPI.pageCount

DocumentAPI.activeItem

DocumentAPI.dimensions

DocumentAPI.items

DocumentAPI.selection

DocumentAPI.selectionCount

DocumentAPI.colors

DocumentAPI.layers

DocumentAPI.masterPages

DocumentAPI.styles

DocumentAPI Methods

DocumentAPI.close()

Closes a document. Returns True is successful, else return False.


```
>>> document = Scripter.activeDocument
>>> document.close()
True
>>>
```

`DocumentAPI.save()`

Saves the current document, and return True is successful. If the document is a new one (i.e., you are trying to save it for the first time), calling this function will result in invocation of Save As dialog, where you can enter the name for your document.

```
>>> document = Scripter.activeDocument
>>> document.save()
True
>>>
```

`DocumentAPI.saveAs(name)`

`DocumentAPI.setInformation(author, title, description)`

`DocumentAPI.newLayer(name)`

`DocumentAPI.removeLayer(name)`

`DocumentAPI.getActiveLayer()`

Make it property

`DocumentAPI.setActiveLayer(name)`

Make it property

`DocumentAPI.getActiveLayerName()`

Make it property

`DocumentAPI.newColorCMYK(c, m, y, k)`

`DocumentAPI.newColorRGB(r, g, b)`

`DocumentAPI.getColor(name)`

Make it property

`DocumentAPI.supportedImageTypes()`

`DocumentAPI.exportAsImages(directoryName, scale, quality, dpi, overwrite)`

Margins

class Margins

Margins represents the margins of a document.

Margins Attributes

Margins.top

This attribute is used to read/write the top margin of the document.:

```
>>> document = Scripter.activeDocument
>>> margins = document.margins
>>> margins.top
40.0
>>> margins.top = 60
```

```
>>> margins.top
60.0
>>>
```

Margins.**bottom**

This attribute is used to read/write the bottom margin of the document.:

```
>>> document = Scripter.activeDocument
>>> margins = document.margins
>>> margins.bottom
40.0
>>> margins.bottom = 60
>>> margins.bottom
60.0
>>>
```

Margins.**right**

This attribute is used to read/write the right margin of the document.

```
>>> document = Scripter.activeDocument
>>> margins = document.margins
>>> margins.right
40.0
>>> margins.right = 60
>>> margins.right
60.0
>>>
```

Margins.**left**

This attribute is used to read/write the left margin of the document.

```
>>> document = Scripter.activeDocument
>>> margins = document.margins
>>> margins.left
40.0
>>> margins.left = 60
>>> margins.left
60.0
>>>
```

Margins Methods

Margins.**set** (*left, top, bottom, right*)

set () is used to set the margins of a document. All the parameters are of type float.

LayerAPI

class LayerAPI

LayerAPI represents a layer in Scribus document. The operations on a layer can be achieved through this class.

LayerAPI Attributes

LayerAPI.**name**
LayerAPI.**id**
LayerAPI.**level**
LayerAPI.**printable**
LayerAPI.**viewable**
LayerAPI.**editable**
LayerAPI.**flowControl**
LayerAPI.**outlineMode**
LayerAPI.**transparency**
LayerAPI.**blendMode**
LayerAPI.**active**
LayerAPI.**items**

ItemAPI

class ItemAPI

ItemAPI represents an item in Scribus document. *TextAPI* and *ScribusImageItem* are inherited from this class to provide more specific attributes and methods to it. All the attributes and methods of *ItemAPI* are applicable to *TextAPI* and *ImageAPI*.

ItemAPI Attributes

ItemAPI.**itemName**
ItemAPI.**fillColor**
ItemAPI.**lineColor**
ItemAPI.**fillShade**
ItemAPI.**lineShade**
ItemAPI.**fillTransparency**
ItemAPI.**lineTransparency**
ItemAPI.**locked**
ItemAPI.**sizeLocked**
ItemAPI.**flipVertical**
ItemAPI.**flipHorizontal**
ItemAPI.**lineWidth**
ItemAPI.**customLineStyle**
ItemAPI.**startArrowIndex**
ItemAPI.**endArrowIndex**

`ItemAPI.printEnabled`

`ItemAPI.xPos`

`ItemAPI.yPos`

`ItemAPI.width`

`ItemAPI.height`

`ItemAPI.rotation`

`ItemAPI.reversed`

`ItemAPI.cornerRadius`

`ItemAPI.columnGap`

`ItemAPI.columns`

ItemAPI Methods

`ItemAPI.move(dx, dy)`

`ItemAPI.moveAbs(x, y)`

`ItemAPI.rotate(rotation)`

`ItemAPI.rotateAbs(rotation)`

`ItemAPI.resize(width, height)`

TextAPI

class TextAPI

TextAPI inherits *ItemAPI*. It adds attributes and methods that are specific to Text Items in a Scribus document.

TextAPI Attributes

`TextAPI.font`

Used to get/set the font of text frame.

```
>>> textItem.font
u'Arial Regular'
>>> textItem.font = "Times New Roman Regular"
>>>
```

`TextAPI.fontSize`

Used to get/set the font size of text frame.

```
>>> textItem.fontSize
12.0
>>> textItem.fontSize = 18.0
>>>
```

`TextAPI.text`

Used to get/set the text in text frame.

```
>>> textItem.text
u'This is a sample text'
>>> textItem.text = "Sample text has been changed"
>>>
```

Note that using this function to set text content will erase the existing content in text frame.

TextAPI.**textLines**

Used to get the number of lines in the content of text frame.

```
>>> textItem.textLines
1
>>>
```

TextAPI.**textLength**

Used to get the length of text content in the frame.

```
>>> textItem.textLength
28
>>>
```

TextAPI.**lineSpacing**

Used to get/set the linespacing for text frame.

```
>>> textItem.lineSpacing
15.0
>>> textItem.lineSpacing = 16.0
>>>
```

TextAPI.**distances**

Used to get the text distances of the frame. It will return a list.

```
>>> textItem.distances
[0.0, 0.0, 0.0, 0.0]
>>>
```

TextAPI.**deleteText**

Used to delete the text in a text frame.

```
>>> textItem.deleteText
>>>
```

TextAPI.**traceText**

Used to trace the text frame.

```
>>> textItem.traceText
True
>>>
```

TextAPI.**hyphenate**

TextAPI.**dehyphenate**

TextAPI.**PDFBookMark**

TextAPI Methods

```
TextAPI.insertText (text, position)
TextAPI.setLineSpacing (mode)
TextAPI.setDistances (left, right, top, bottom)
TextAPI.setTextColor (color)
TextAPI.setTextStroke (color)
TextAPI.setTextScalingV (value)
TextAPI.setTextScalingH (value)
TextAPI.setTextShade (shade)
TextAPI.selectText (start, selectCount)
TextAPI.linkToTextFrame (name)
TextAPI.unLinkTextFrames ()
```

ImageAPI

class ImageAPI

ImageAPI is used to access an Image Item in a Scribus document. It inherits the attributes and methods of *ItemAPI*, along with the following Attributes and Methods.

ImageAPI Attributes

ImageAPI.xScale

Used to read/write the Image Scale along X Axis.

```
>>> print imageItem.xScale #reading the xScale
1.0
>>> imageItem.xScale = 0.5 #setting the value of xScale
>>>
```

ImageAPI.yScale

Used to read/write the Image Scale along Y Axis.

```
>>> print imageItem.yScale #reading the yScale
1.0
>>> imageItem.yScale = 2.0 #setting the value of yScale
>>>
```

ImageAPI.xOffset

Used to read/write the Image Offset along X Axis.

```
>>> print imageItem.xOffset #reading the xOffset
0.0
>>> imageItem.xOffset = 2.0 #setting the value of xOffset
>>>
```

ImageAPI.yOffset

Used to read/write the Image Offset along Y Axis.

```
>>> print imageItem.yOffset #reading the yOffset
0.0
>>> imageItem.yOffset = 2.0 #setting the value of yOffset
>>>
```

ImageAPI Methods

ImageAPI.**load** (*filename*)

Load an image into the Image Frame.

Parameters **filename** – name of the image file to be loaded into the image frame.

```
>>> imageItem.load("/Users/scribus/logo.jpg")
>>>
```

ImageAPI.**scale** (*x*, *y*)

Set the scaling of loaded image.

Parameters

- **x** – Set the scaling along X Axis.
- **y** – Set the scaling along Y Axis.

ImageAPI.**offset** (*x*, *y*)

Set the offsets of loaded image.

Parameters

- **x** – Set the offset along X Axis.
- **y** – Set the offset along Y Axis.

ImageAPI.**setBrightness** (*amount*)

Set the brightness of the loaded image.

```
>>> imageItem.setBrightness(100)
>>>
```

ImageAPI.**setGrayscale** ()

Set the loaded image to Grayscale.

```
>>> imageItem.setGrayscale()
>>>
```

ImageAPI.**scaleToFrame** (*scale*, *proportional*)

Scale the loaded image to image frame.

Parameters

- **scale** – Set True for scaleToFrame.
- **proportional** – Set True for scaling proportionally.

```
>>> imageItem.scaleToFrame(True, True)
>>>
```


CHAPTER 2

Tutorials

Add a menu to Scribus MenuBar

The following script will add a new menu to Scribus MainWindow menubar.:

```
# -*- coding: utf-8 -*-
from __future__ import with_statement
from PyQt4.QtGui import QApplication, QMenu, QAction

def add_test():
    menuBar = Scriber.dialogs.mainWindow.qt.menuBar()
    item_action = None
    for action in menuBar.actions():
        if action.text() == "&View":
            item_action = action
    menuBar.insertMenu(item_action, QMenu("&TTest", menuBar))

def run():
    add_test()

run()
```

Frequently Asked Questions

How to submit scripts and tutorials to this documentation?

Scripter documentation is generated using Sphinx documentation system, and is maintained in Github Repository. You may fork this repository, add your submissions and commit it to your forked repository.

To submit to Scribus Scripter official documentation, you should submit a pull request to official git repository (repository from which you forked). Moderators will evaluate your script/tutorial and add it to official documentation, if valid.

Repository Details

- *Official Repository (Maintained by Jain Basil Aliyas)* <<http://github.com/jainbasil/scripter-doc>>

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`

A

active (LayerAPI attribute), 7
activeDocument (Scripter attribute), 3
activeItem (DocumentAPI attribute), 4
activePage (DocumentAPI attribute), 4
activeWindow (Scripter attribute), 3
available (DocumentAPI attribute), 4

B

blendMode (LayerAPI attribute), 7
bottom (Margins attribute), 6

C

close() (DocumentAPI method), 4
colors (DocumentAPI attribute), 4
columnGap (ItemAPI attribute), 8
columns (ItemAPI attribute), 8
cornerRadius (ItemAPI attribute), 8
customLineStyle (ItemAPI attribute), 7

D

dehyphenate (TextAPI attribute), 9
deleteText (TextAPI attribute), 9
dimensions (DocumentAPI attribute), 4
distances (TextAPI attribute), 9
DocumentAPI (built-in class), 3

E

editable (LayerAPI attribute), 7
endArrowIndex (ItemAPI attribute), 7
exportAsImages() (DocumentAPI method), 5

F

fillColor (ItemAPI attribute), 7
fillShade (ItemAPI attribute), 7
fillTransparency (ItemAPI attribute), 7
flipHorizontal (ItemAPI attribute), 7
flipVertical (ItemAPI attribute), 7
flowControl (LayerAPI attribute), 7

font (TextAPI attribute), 8
fontSize (TextAPI attribute), 8

G

getActiveLayer() (DocumentAPI method), 5
getActiveLayerName() (DocumentAPI method), 5
getColor() (DocumentAPI method), 5

H

height (ItemAPI attribute), 8
hyphenate (TextAPI attribute), 9

I

id (LayerAPI attribute), 7
ImageAPI (built-in class), 10
insertText() (TextAPI method), 10
ItemAPI (built-in class), 7
itemName (ItemAPI attribute), 7
items (DocumentAPI attribute), 4
items (LayerAPI attribute), 7

L

LayerAPI (built-in class), 6
layers (DocumentAPI attribute), 4
left (Margins attribute), 6
level (LayerAPI attribute), 7
lineColor (ItemAPI attribute), 7
lineShade (ItemAPI attribute), 7
lineSpacing (TextAPI attribute), 9
lineTransparency (ItemAPI attribute), 7
lineWidth (ItemAPI attribute), 7
linkToTextFrame() (TextAPI method), 10
load() (ImageAPI method), 11
locked (ItemAPI attribute), 7

M

Margins (built-in class), 5
margins (DocumentAPI attribute), 4
masterPages (DocumentAPI attribute), 4

modified (DocumentAPI attribute), 4
move() (ItemAPI method), 8
moveAbs() (ItemAPI method), 8

N

name (DocumentAPI attribute), 4
name (LayerAPI attribute), 7
newColorCMYK() (DocumentAPI method), 5
newColorRGB() (DocumentAPI method), 5
newLayer() (DocumentAPI method), 5

O

offset() (ImageAPI method), 11
openDocument() (Scripter method), 3
outlineMode (LayerAPI attribute), 7

P

pageCount (DocumentAPI attribute), 4
PDFBookMark (TextAPI attribute), 9
printable (LayerAPI attribute), 7
printEnabled (ItemAPI attribute), 7

R

removeLayer() (DocumentAPI method), 5
resize() (ItemAPI method), 8
reversed (ItemAPI attribute), 8
right (Margins attribute), 6
rotate() (ItemAPI method), 8
rotateAbs() (ItemAPI method), 8
rotation (ItemAPI attribute), 8

S

save() (DocumentAPI method), 5
saveAs() (DocumentAPI method), 5
scale() (ImageAPI method), 11
scaleToFrame() (ImageAPI method), 11
Scripter (built-in class), 3
selection (DocumentAPI attribute), 4
selectionCount (DocumentAPI attribute), 4
selectText() (TextAPI method), 10
set() (Margins method), 6
setActiveLayer() (DocumentAPI method), 5
setBrightness() (ImageAPI method), 11
setDistances() (TextAPI method), 10
setGrayscale() (ImageAPI method), 11
setInformation() (DocumentAPI method), 5
setLineSpacing() (TextAPI method), 10
setTextColor() (TextAPI method), 10
setTextScalingH() (TextAPI method), 10
setTextScalingV() (TextAPI method), 10
setTextShade() (TextAPI method), 10
setTextStroke() (TextAPI method), 10
sizeLocked (ItemAPI attribute), 7

startArrowIndex (ItemAPI attribute), 7
styles (DocumentAPI attribute), 4
supportedImageTypes() (DocumentAPI method), 5

T

text (TextAPI attribute), 8
TextAPI (built-in class), 8
textLength (TextAPI attribute), 9
textLines (TextAPI attribute), 9
top (Margins attribute), 5
traceText (TextAPI attribute), 9
transparency (LayerAPI attribute), 7

U

unLinkTextFrames() (TextAPI method), 10

V

viewable (LayerAPI attribute), 7

W

width (ItemAPI attribute), 8

X

xOffset (ImageAPI attribute), 10
xPos (ItemAPI attribute), 8
xScale (ImageAPI attribute), 10

Y

yOffset (ImageAPI attribute), 10
yPos (ItemAPI attribute), 8
yScale (ImageAPI attribute), 10