

## Document related Commands

### **closeDoc(...)**

```
closeDoc()
```

Closes the current document without prompting to save.

May throw [NoDocOpenError](#) if there is no document to close

### **docChanged(...)**

```
docChanged(bool)
```

Enable/disable save icon in the Scribus icon bar and the Save menu item. It's useful to call this procedure when you're changing the document, because Scribus won't automatically notice when you change the document using a script.

### **getUnit(...)**

```
getUnit() -> integer (Scribus unit constant)
```

Returns the measurement units of the document. The returned value will be one of the UNIT\_\* constants: UNIT\_INCHES, UNIT\_MILLIMETERS, UNIT\_PICAS, UNIT\_POINTS.

### **getDocName(...)**

```
getDocName() -> string
```

Returns the name of current file

### **haveDoc(...)**

```
haveDoc() -> bool
```

Returns true if there is a document open.

### **loadStylesFromFile(...)**

```
loadStylesFromFile("filename")
```

Loads paragraph styles from the Scribus document at "filename" into the current document.

### **masterPageNames(...)**

```
masterPageNames()
```

Returns a list of the names of all master pages in the document.

### **newDoc(...)**

```
newDoc(size, margins, orientation, firstPageNumber, unit, facingPages, firstSideLeft) -> bool
```

WARNING: Obsolete procedure! Use [newDocument](#) instead.

Creates a new document and returns true if successful. The parameters have the following meaning:

- size = A tuple (width, height) describing the size of the document. You can use predefined constants named PAPER\_<paper\_type> e.g. PAPER\_A4 etc.
- margins = A tuple (left, right, top, bottom) describing the document margins
- orientation = the page orientation - constants PORTRAIT, LANDSCAPE
- firstPageNumber = is the number of the first page in the document used for page numbering. While you'll usually want 1, it's useful to have higher numbers if you're creating a document in several parts.
- unit: this value sets the measurement units used by the document. Use a predefined constant for this, one of: UNIT\_INCHES, UNIT\_MILLIMETERS, UNIT\_PICAS, UNIT\_POINTS.
- facingPages = FACINGPAGES, NOFACINGPAGES
- firstSideLeft = FIRSTPAGELEFT, FIRSTPAGERIGHT

The values for width, height and the margins are expressed in the given unit for the document. PAPER\_\* constants are expressed in points. If your document is not in points, make sure to account for this.

example: [newDoc](#)(PAPER\_A4, (10, 10, 20, 20), LANDSCAPE, 1, UNIT\_POINTS, FACINGPAGES, FIRSTPAGERIGHT)

### **newDocument(...)**

```
newDocument(size, margins, orientation, firstPageNumber, unit, pagesType, firstPageOrder, numPages)  
-> bool
```

Creates a new document and returns true if successful. The parameters have the following meaning:

- size = A tuple (width, height) describing the size of the document. You can use predefined constants named PAPER\_<paper\_type> e.g. PAPER\_A4 etc.
- margins = A tuple (left, right, top, bottom) describing the document margins
- orientation = the page orientation - constants PORTRAIT, LANDSCAPE
- firstPageNumber = is the number of the first page in the document used for pagenumbering. While you'll usually want 1, it's useful to

have higher numbers if you're creating a document in several parts.

- unit: this value sets the measurement units used by the document. Use a predefined constant for this, one of: UNIT\_INCHES, UNIT\_MILLIMETERS, UNIT\_PICAS, UNIT\_POINTS.
- pagesType = One of the predefined constants PAGE\_n. PAGE\_1 is single page, PAGE\_2 is for double sided documents, PAGE\_3 is for 3 pages fold and PAGE\_4 is 4-fold.
- firstPageOrder = What is position of first page in the document. Indexed from 0 (0 = first).
- numPage = Number of pages to be created.

The values for width, height and the margins are expressed in the given unit for the document. PAPER\_\* constants are expressed in points. If your document is not in points, make sure to account for this.

example: newDocument(PAPER\_A4, (10, 10, 20, 20), LANDSCAPE, 7, UNIT\_POINTS, PAGE\_4, 3, 1)

May raise ScribusError if is firstPageOrder bigger than allowed by pagesType.

#### **openDoc(...)**

```
openDoc("name")
```

Opens the document "name".

May raise ScribusError if the document could not be opened.

#### **placeEPS(...)**

placeEPS("filename", x, y) Places the EPS "filename" onto the current page, x and y specify the coordinate of the topleft corner of the EPS placed on the page. If loading was successful, the selection contains the imported EPS.

#### **placeODG(...)**

placeODG("filename", x, y) Places the ODG "filename" onto the current page, x and y specify the coordinate of the topleft corner of the ODG placed on the page. If loading was successful, the selection contains the imported ODG.

#### **placeSVG(...)**

placeSVG("filename", x, y) Places the SVG "filename" onto the current page, x and y specify the coordinate of the topleft corner of the SVG placed on the page. If loading was successful, the selection contains the imported SVG.

#### **placeSXD(...)**

placeSXD("filename", x, y) Places the SXD "filename" onto the current page, x and y specify the coordinate of the topleft corner of the SXD placed on the page. If loading was successful, the selection contains the imported SXD.

#### **saveDoc(...)**

```
saveDoc()
```

Saves the current document with its current name, returns true if successful. If the document has not already been saved, this may bring up an interactive save file dialog. If the save fails, there is currently no way to tell.

#### **saveDocAs(...)**

```
saveDocAs("name")
```

Saves the current document under the new name "name" (which may be a full or relative path).

May raise ScribusError if the save fails.

#### **setInfo(...)**

```
setInfo("author", "info", "description") -> bool
```

Sets the document information. "Author", "Info", "Description" are strings.

#### **setMargins(...)**

```
setMargins(lr, rr, tr, br)
```

Sets the margins of the document, Left(lr), Right(rr), Top(tr) and Bottom(br) margins are given in the measurement units of the document - see UNIT\_<type> constants.

#### **setDocType(...)**

```
setDocType(facingPages, firstPageLeft)
```

Sets the document type. To get facing pages set the first parameter to FACINGPAGES, to switch facingPages off use NOFACINGPAGES instead. If you want to be the first page a left side set the second parameter to FIRSTPAGELEFT, for a right page use FIRSTPAGERIGHT.

#### **setUnit(...)**

```
setUnit(type)
```

Changes the measurement unit of the document. Possible values for "unit" are defined as constants UNIT\_<type>.

May raise `ValueError` if an invalid unit is passed.

**scrollDocument(...)**

`scrollDocument(x, y)`

Scroll the document in main GUI window by x and y.

**zoomDocument(...)**

`zoomDocument(double)`

Zoom the document in main GUI window. Actions have whole number values like 20.0, 100.0, etc. Zoom to Fit uses -100 as a marker.