Page related Commands

applyMasterPage(...)
```
applyMasterPage(Master Page name, page nr)
```

Applies the named master page to the indicated page. Some examples of usage are on the wiki..

closeMasterPage(...)
```
closeMasterPage()
```

Closes the currently active master page, if any, and returns editing to normal. Begin editing with editMasterPage().

createMasterPage(...)
```
createMasterPage(pageName)
```

Creates a new master page named pageName. Begin editing with editMasterPage().

currentPage(...)
```
currentPage() -> integer
```

Returns the number of the current working page. Page numbers are counted from 1 upwards, no matter what the displayed first page number of your document is.

deleteMasterPage(...)
```
deleteMasterPage(pageName)
```

Delete the named master page.

deletePage(...)
```
deletePage(nr)
```

Deletes the given page. Does nothing if the document contains only one page. Page numbers are counted from 1 upwards, no matter what the displayed first page number is.

May raise IndexError if the page number is out of range

editMasterPage(...)
```
editMasterPage(pageName)
```

Enables master page editing and opens the named master page for editing. Finish editing with closeMasterPage().

getAllObjects(...)
```
getAllObjects() -> list
```

Returns a list containing the names of all objects on the current page.

getHGuides(...)
```
getHGuides() -> list
```

Returns a list containing positions of the horizontal guides. Values are in the document's current units - see UNIT_<type> constants.

getPageType(...)
```
getPageType() -> integer
```

Returns the type of the Page, 0 means left Page, 1 is a middle Page and 2 is a right Page

getVGuides(...)
```
getVGuides()
```

See getHGuides.

getPageItems(...)
```
getPageItems() -> list
```

Returns a list of tuples with items on the current page. The tuple is: (name, objectType, order) E.g. [('Text1', 4, 0), ('Image1', 2, 1)] means that object named 'Text1' is a text frame (type 4) and is the first at the page...

getPageMargins(...)
```
getPageMargins()
```

Returns the document page margins as a (top, left, right, bottom) tuple in the document's current units. See UNIT_<type> constants and getPageSize().

getPageNMargins(...)
```
getPageNMargins(nr)
```

Returns a tuple with a particular page's margins measured in the document's current units. See UNIT_<type> constants and getPageSize().

getPageSize(...)
```
getPageSize() -> tuple
```

Returns a tuple with document page dimensions measured in the document's current units. See UNIT_<type> constants and getPageMargins()

getPageNSize(...)
```
getPageNSize(nr) -> tuple
```

Returns a tuple with a particular page's size measured in the document's current units. See UNIT_<type> constants and getPageMargins()

gotoPage(...)
```
gotoPage(nr)
```

Moves to the page "nr" (that is, makes the current page "nr"). Note that gotoPage doesn't (curently) change the page the user's view is displaying, it just sets the page that script commands will operates on.

May raise IndexError if the page number is out of range.

importPage(...)

importPage("fromDoc", (pageList), [create, imortwhere, importwherePage]) Imports a set of pages (given as a tuple) from an existing document (the file name must be given). This functions maps the "Page->Import" dropdown menu function. fromDoc: string; the filename of the document to import pages from pageList: tuple with page numbers of pages to import create: number; 0 to replace existing pages, 1 (default) to insert new pages importWhere: number; the page number (of the current document) at which import the pages importWherePage: number; used if create==1; 0 to create pages before selected page; 1 to create pages after selected page; 2 (default) to create pages at the end of the document

newPage(...)
```
newPage(where [,"masterpage"])
```

Creates a new page. If "where" is -1 the new Page is appended to the document, otherwise the new page is inserted before "where". Page numbers are counted from 1 upwards, no matter what the displayed first page number of your document is. The optional parameter "masterpage" specifies the name of the master page for the new page.

May raise IndexError if the page number is out of range

pageCount(...)
```
pageCount() -> integer
```

Returns the number of pages in the document.

redrawAll(...)
```
redrawAll()
```

Redraws all pages.

savePageAsEPS(...)
```
savePageAsEPS("name")
```

Saves the current page as an EPS to the file "name".

May raise ScribusError if the save failed.

setHGuides(...)
```
setHGuides(list)
```

Sets horizontal guides. Input parameter must be a list of guide positions measured in the current document units - see UNIT_<type> constants.

Example: setHGuides(getHGuides() + [200.0, 210.0] # add new guides without any lost
setHGuides([90,250]) # replace current guides entirely

setRedraw(...)
```
setRedraw(bool)
```

Disables page redraw when bool = False, otherwise redrawing is enabled. This change will persist even after the script exits, so make sure to call setRedraw(True) in a finally: clause at the top level of your script.

setVGuides(...)
```
setVGuides()
```

See setHGuides.