Using Dialogs from Scribus

fileDialog(...)

```
fileDialog("caption", ["filter", "defaultname", haspreview, issave, isdir])
-> string with filename
```

Shows a File Open dialog box with the caption "caption". Files are filtered with the filter string "filter". A default filename or file path can also supplied, leave this string empty when you don't want to use it. A value of True for haspreview enables a small preview widget in the FileSelect box. When the issave parameter is set to True the dialog acts like a "Save As" dialog otherwise it acts like a "File Open Dialog". When the isdir parameter is True the dialog shows and returns only directories. The default for all of the opional parameters is False.

The filter, if specified, takes the form 'comment (*.type *.type2 ...)'. For example 'Images (*.png *.xpm *.jpg)'.

Refer to the Qt-Documentation for QFileDialog for details on filters.

Example: fileDialog('Open input', 'CSV files (*.csv)')
Example: fileDialog('Save report', defaultname='report.txt', issave=True)

fileQuit(...)

```
fileQuit()
```

Quit Scribus.

getGuiLanguage(...)

```
getGuiLanguage() -> string
```

Returns a string with the -lang value.

messagebarText(...)

```
messagebarText("string")
```

Writes the "string" into the Scribus message bar (status line). The text must be UTF8 encoded or 'unicode' string(recommended).

messageBox(...)

```
messageBox("caption", "message", icon=ICON_NONE, button1=BUTTON_OK|BUTTONOPT_DEFAULT,
button2=BUTTON_NONE, button3=BUTTON_NONE) -> integer
```

Displays a message box with the title "caption", the message "message", and an icon "icon" and up to 3 buttons. By default no icon is used and a single button, OK, is displayed. Only the caption and message arguments are required, though setting an icon and appropriate button(s) is strongly recommended. The message text may contain simple HTML-like markup.

Returns the number of the button the user pressed. Button numbers start at 1.

For the icon and the button parameters there are predefined constants available with the same names as in the Qt Documentation. These are the BUTTON_* and ICON_* constants defined in the module. There are also two extra constants that can be binary-ORed with button constants:

- BUTTONOPT_DEFAULT Pressing enter presses this button.
- BUTTONOPT_ESCAPE Pressing escape presses this button.

Usage examples:

```
result = messageBox('Script failed',
                    'This script only works when you have a text frame selected.',
                    ICON_ERROR)
result = messageBox('Monkeys!', 'Something went ook! <i>Was it a monkey?</i>',
                    ICON_WARNING, BUTTON_YES|BUTTONOPT_DEFAULT,
                    BUTTON_NO, BUTTON_IGNORE|BUTTONOPT_ESCAPE)
```

Defined button and icon constants: BUTTON_NONE, BUTTON_ABORT, BUTTON_CANCEL, BUTTON_IGNORE, BUTTON_NO, BUTTON_NOALL, BUTTON_OK, BUTTON_RETRY, BUTTON_YES, BUTTON_YESALL, ICON_NONE, ICON_INFORMATION, ICON_WARNING, ICON_CRITICAL.

newDocDialog(...)

```
newDocDialog() -> bool
```

Displays the "New Document" dialog box. Creates a new document if the user accepts the settings. Does not create a document if the user presses cancel. Returns true if a new document was created.

newStyleDialog(...)

```
newStyleDialog() -> string
```

Shows 'Create new paragraph style' dialog. Function returns real style name or None when user cancels the dialog.

statusMessage(...)

```
messagebarText("string")
```

Writes the "string" into the Scribus message bar (status line). The text must be UTF8 encoded or 'unicode' string(recommended).

progressReset(...)

```
progressReset()
```

Cleans up the Scribus progress bar previous settings. It is called before the new progress bar use. See progressSet.

progressSet(...)

```
progressSet(nr)
```

Set the progress bar position to "nr", a value relative to the previously set progressTotal. The progress bar uses the concept of steps; you give it the total number of steps and the number of steps completed so far and it will display the percentage of steps that have been completed. You can specify the total number of steps with progressTotal(). The current number of steps is set with progressSet(). The progress bar can be rewound to the beginning with progressReset(). [based on info taken from Trolltech's Qt docs]

progressTotal(...)

```
progressTotal(max)
```

Sets the progress bar's maximum steps value to the specified number. See progressSet.

valueDialog(...)

```
valueDialog(caption, message [,defaultvalue]) -> string
```

Shows the common 'Ask for string' dialog and returns its value as a string Parameters: window title, text in the window and optional 'default' value.

Example: valueDialog('title', 'text in the window', 'optional')