



**Technologies of Ohio, Inc.**  
 Home of filePro \* Development Software

432 W. Gypsy Lane Road  
 Bowling Green, OH 43402  
 Tel. (800) 847-4740  
 Fax (877) 606-6853  
 sales@fpotech.com

**New functions for creating XLSX documents from filePro**

**Table of Contents**

<a href="#"><u>XL_OPEN</u></a>	<a href="#"><u>XL_SAVE</u></a>	<a href="#"><u>XL_ADDSHEET</u></a>	<a href="#"><u>XL_ADDCELL</u></a>	<a href="#"><u>XL_ADDCELL2</u></a>
<a href="#"><u>XL_FORMAT</u></a>	<a href="#"><u>XL_COLWIDTH</u></a>	<a href="#"><u>XL_FONT</u></a>	<a href="#"><u>XL_BORDER</u></a>	<a href="#"><u>XL_FILL</u></a>
<a href="#"><u>XL_ADD_DT</u></a>	<a href="#"><u>XL_ADD_DT2</u></a>	<a href="#"><u>XL_CHART</u></a>	<a href="#"><u>XL_CHART2</u></a>	<a href="#"><u>XL_CHARTSHEET</u></a>
<a href="#"><u>XL_SERIES</u></a>	<a href="#"><u>XL_SERIES2</u></a>	<a href="#"><u>XL_PROTECTSHEET</u></a>	<a href="#"><u>XL_PROTECTCHARTSHEET</u></a>	<a href="#"><u>XL_ERROR</u></a>
<a href="#"><u>XL_SETPOS</u></a>	<a href="#"><u>XL_SETPOS2</u></a>	<a href="#"><u>XL_NEXTROW</u></a>	<a href="#"><u>XL_NEXTCOL</u></a>	<a href="#"><u>XL_STYLE</u></a>
<a href="#"><u>XL_IMAGE</u></a>	<a href="#"><u>XL_IMAGE2</u></a>	<a href="#"><u>XL_LASTCMD</u></a>	<a href="#"><u>XL_MARGINS</u></a>	<a href="#"><u>XL_LANDSCAPE</u></a>
<a href="#"><u>XL_PORTRAIT</u></a>	<a href="#"><u>XL_GRIDLINES</u></a>	<a href="#"><u>XL_FITPAGES</u></a>	<a href="#"><u>XL_PAPERTYPE</u></a>	<a href="#"><u>XL_CENTERH</u></a>
<a href="#"><u>XL_CENTERV</u></a>	<a href="#"><u>XL_PRINTACROSS</u></a>	<a href="#"><u>XL_SETHEADER</u></a>	<a href="#"><u>XL_SETFOOTER</u></a>	<a href="#"><u>XL_SETBACKGROUND</u></a>
<a href="#"><u>XL_HIDEZEROS</u></a>	<a href="#"><u>XL_SHOWROWCOL</u></a>			

**Important note:** A value described as a handle references a numeric typed field of size 8.

**Example:** ‘declare handle(8,.0)’. This can also be an untyped variable, provided it’s large enough to store the returned handle value.

**Start building an XLSX output file**

```
e = XL_OPEN(file [, name])
```

**Parameters:**

**file:** Path to the file to create. If full path is not given the generated file will be placed in the PFTMP or equivalent directory.

**name:** The name for the default sheet that will be created. Default name is “Sheet1”.

**Notes:**

If the filename provided does not end in “.xlsx” it will be added on creation.

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

Only one XLSX file can be created at a time.

[Back to top](#)

## Saving the current XLSX file

```
e = XL_SAVE ([password])
```

### Parameters:

**password:** If specified, encrypt the XLSX output file using Agile encryption (AES128).

### Notes:

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

Encrypted XLSX files cannot be opened with most third-party programs such as LibreOffice and OpenOffice. They are fully supported by Excel however. The documents are saved in an encrypted CFB file.

## Adding a new sheet to the XLSX document

```
handle = XL_ADDSHEET ([name])
```

### Parameters:

**name:** The name for the sheet to be created. Defaults to auto naming the sheet based on the Sheet1, Sheet2, ..., SheetN template.

### Notes:

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

## Adding a new cell to the XLSX document

```
e = XL_ADDCELL([data [, style [, sheet [, row [, col]]]])
```

### Parameters:

**data:** Data to be inserted into the document. A cell starting with ‘=’ will be treated as a formula.

**style:** Handle to style to be used for this cell. Leave blank to use the default style.

**sheet:** Handle to sheet to insert the cell on. Leave blank, “0”, or “-1” to use the default sheet.

**row:** Row to place the cell (0 indexed)

**col:** Column to place the cell (0 indexed)

**Notes:**

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

Using an empty or negative row/column value will cause the cell to be added using the auto counter in the sheet, incrementing the column value after the cell is added. Specifying a location will reposition the auto counter. Formulas can be used as part of the data as well by prefixing the string with ‘=’.

**Adding a new cell to the XLSX document**

```
e = XL_ADDCELL2([data [, style [, sheet [, cell]]]])
```

**Parameters:**

**data:** Data to be inserted into the document. A cell starting with ‘=’ will be treated as a formula.

**style:** Handle to style to be used for this cell. Leave blank to use the default style.

**sheet:** Handle to sheet to insert the cell on. Leave blank, “0”, or “-1” to use the default sheet.

**cell:** The Excel style cell to insert the cell. e.g. “A1” “D6” “F6”.

**Notes:**

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

Using an empty cell number will cause the cell to be added using the auto counter in the sheet, incrementing the column value after the cell is added. Specifying a location will reposition the auto counter. Formulas can be used as part of the data as well by prefixing the string with ‘=’.

**Creating a new format to use with the XLSX document**

```
handle = XL_FORMAT(format)
```

**Parameters:**

**format:** Excel format string to format a style. e.g.

“\$ #,###,nnn.nn”

“% ##n.n”

“m/d/yyyy”

**Notes:**

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

[Back to top](#)

### Changing the default column width for a sheet between a range

```
e = XL_COLWIDTH(width, firstcol, lastcol [, sheet])
```

#### Parameters:

**width:** Width of the column(s). e.g. "24", "12.5", "11"

**firstcol:** Zero based column index or column letter to set from.

**lastcol:** Zero based column index or column letter to set to.

**sheet:** Handle to sheet to change the cell widths.

#### Notes:

Returns "1" on success and "-1" on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

### Creating a new font to use with the XLSX document

```
handle = XL_FONT(font, [size [, attr [, color]])
```

#### Parameters:

**font:** Name of the font to use.

**size:** Point size of the font. e.g. "11", "8.42", "12", default "11.0"

**attr:** List of attributes to apply to this font, separated by commas. e.g. "bold,italic"

#### Values:

"bold", "italic", "underline", "strike", "hidden", "wrap", "shrink"  
"fill", "left", "center", "right", "justify", "top", "bottom", "vjustify",  
"vcenter"

**color:** The RGB hex value to set the font color. e.g. "000000", "ADD8E6"

#### Notes:

Returns "1" on success and "-1" on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

### Creating a new border to use with the XLSX document

```
handle = XL_BORDER(borderstyle [, color])
```

#### Parameters:

**borderstyle:** The style to use with this border. Must be one of the following values:

"thin", "medium", "dashed", "dotted", "thick", "hair",

[Back to top](#)

“medium\_dashed”, “dash\_dot”, “medium\_dash\_dot”,  
“dash\_dot\_dot”, “medium\_dash\_dot\_dot”, “slant\_dash\_dot”

**color:** The RGB hex value to set the font color. e.g. “000000”, “ADD8E6”

**Notes:**

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

### Creating a new fill to use with the XLSX document

```
handle = XL_FILL(bg [, fg [, fill]])
```

**Parameters:**

**bg:** The RGB hex value to set the background fill color. e.g. “000000”  
“ADD8E6”

**fg:** The RGB hex value to set the foreground fill color. e.g. “000000” “ADD8E6”

**fill:** The fill pattern to use, defaults to “solid” fill. Value must be one of the following:

“solid”, “medium\_gray”, “dark\_gray”, “light\_gray”,  
“dark\_horizontal”, “dark\_vertical”, “dark\_down”, “dark\_up”,  
“dark\_grid”, “dark\_trellis”, “light\_horizontal”, “light\_vertical”,  
“light\_down”, “light\_up”, “light\_grid”, “light\_trellis”, “gray\_125”,  
“gray\_0625”

**Notes:**

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

### Combining two fields into a single spreadsheet datetime fields and inserting it as a new cell in the XLSX document

```
e = XL_ADD_DT(date, time [, style [, sheet [, row [, col]]]])
```

**Parameters:**

**date:** filePro date field.

**time:** filePro time field.

**style:** Handle to style to be used for this cell. Leave blank to use the default style.

**sheet:** Handle to sheet to insert the cell on. Leave blank, “0”, or “-1” to use the default sheet.

**row:** Row to place the cell (0 indexed)

**col:** Column to place the cell (0 indexed)

[Back to top](#)

**Notes:**

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

**Combining two fields into a single spreadsheet datetime fields and inserting it as a new cell in the XLSX document**

```
e = XL_ADD_DT2(date, time [, style [, sheet [, cell]])
```

**Parameters:**

**date:** filePro date field.

**time:** filePro time field.

**style:** Handle to style to be used for this cell. Leave blank to use the default style.

**sheet:** Handle to sheet to insert the cell on. Leave blank, “0”, or “-1” to use the default sheet.

**cell:** The Excel style cell to insert the cell. e.g. “A1”, “D6”, “F6”

**Notes:**

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

**Adding a new chart to the XLSX document**

```
handle = XL_CHART(type [, title [, xname [, yname [, row [, col [, stylenum [, sheet [, xoff [, yoff [, xscale [, yscale ]]]]]]]]]]]))
```

**Parameters:**

**type:** Type of chart to create. Must be one of the following values:

“area”, “area\_stacked”, “area\_stacked\_percent”, “bar”, “bar\_stacked”, “bar\_stacked\_percent”, “column”, “column\_stacked”, “column\_stacked\_percent”, “doughnut”, “line”, “line\_stacked”, “line\_stacked\_percent”, “pie”, “scatter”, “scatter\_straight”, “scatter\_straight\_markers”, “scatter\_smooth”, “scatter\_smooth\_markers”, “radar”, “rader\_with\_markers”, “radar\_filled”

**title:** The title for this chart.

**xname:** The title for the x-axis.

**yname:** The title for the y-axis.

**row:** Row to place the cell (0 indexed)

**col:** Column to place the cell (0 indexed)

[Back to top](#)

**stylenum:** Number of the built in Excel style to use. Must be between “1” and “48”. The default style is 2. The value is one of the 48 built in styles available on the “Design” tab in Excel 2007.

**sheet:** Handle to sheet to insert the chart on. Leave blank, “0”, or “-1” to use the default sheet.

**xoff:** X axis offset to place the chart, in pixels.

**yoff:** Y axis offset to place the chart, in pixels.

**xscale:** Scale the chart along the x axis. e.g. "1", "0.5" "2". Value cannot be negative.

**yscale:** Scale the chart along the y axis. e.g. "1", "0.5" "2". Value cannot be negative.

**Notes:**

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

The chart functions do not use the auto counter found in the sheets and instead will default to “0”, “0” or “A1” when used for insertion.

**Adding a new chart to the XLSX document**

```
handle = XL_CHART2(type [, title [, xname [, yname [, cell [, stylenum  
[, sheet [, xoff [, yoff [, xscale [, yscale]]]]]]]]]]))
```

**Parameters:**

**type:** Type of chart to create. Must be one of the following values:

“area”, “area\_stacked”, “area\_stacked\_percent”, “bar”, “bar\_stacked”,  
“bar\_stacked\_percent”, “column”, “column\_stacked”,  
“column\_stacked\_percent”, “doughnut”, “line”, “line\_stacked”,  
“line\_stacked\_percent”, “pie”, “scatter”, “scatter\_straight”,  
“scatter\_straight\_markers”, “scatter\_smooth”, “scatter\_smooth\_markers”,  
“radar”, “rader\_with\_markers”, “radar\_filled”

**title:** The title for this chart.

**xname:** The title for the x-axis.

**yname:** The title for the y-axis.

**cell:** The Excel style cell to insert the cell. e.g. “A1”, “D6”, “F6”

**stylenum:** Number of the built in Excel style to use. Must be between “1” and “48”. The default style is 2. The value is one of the 48 built in styles available on the “Design” tab in Excel 2007.

**sheet:** Handle to sheet to insert the chart on. Leave blank, “0”, or “-1” to use the default sheet.

[Back to top](#)

**xoff:** X axis offset to place the chart, in pixels.

**yoff:** Y axis offset to place the chart, in pixels.

**xscale:** Scale the chart along the x axis. e.g. "1", "0.5" "2". Value cannot be negative.

**yscale:** Scale the chart along the y axis. e.g. "1", "0.5" "2". Value cannot be negative.

**Notes:**

Returns "1" on success and "-1" on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

The chart functions do not use the auto counter found in the sheets and instead will default to "0", "0" or "A1" when used for insertion.

**Adding a new chartsheet to the XLSX document. A chartsheet is a full chart that occupies its own sheet and cannot contain any cells.**

```
handle = XL_CHARTSHEET(type [, title [, xname [, yname [, stylenum]]]])
```

**Parameters:**

**type:** Type of chart to create. Must be one of the following values:

"area", "area\_stacked", "area\_stacked\_percent", "bar", "bar\_stacked",  
"bar\_stacked\_percent", "column", "column\_stacked",  
"column\_stacked\_percent", "doughnut", "line", "line\_stacked",  
"line\_stacked\_percent", "pie", "scatter", "scatter\_straight",  
"scatter\_straight\_markers", "scatter\_smooth", "scatter\_smooth\_markers",  
"radar", "rader\_with\_markers", "radar\_filled"

**title:** The title for this chart.

**xname:** The title for the x-axis.

**yname:** The title for the y-axis.

**stylenum:** Number of the built in Excel style to use. Must be between "1" and "48". The default style is 2. The value is one of the 48 built in styles available on the "Design" tab in Excel 2007.

**Notes:**

Returns "1" on success and "-1" on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

**Adding a series to a chart or chartsheet**

```
e = XL_SERIES(chartnum, sheet, namerow, namecol, cfirstrow, cfirstcol,  
clastrow, clastcol, vfirstrow, vfirstcol, vlastrow, vlastcol)
```



**Parameters:**

**chartnum:** Handle to a chart or chartsheet to add series.

**sheet:** Handle to sheet to get values from. Leave blank, "0", or "-1" to use the default sheet.

**namerow:** Series name row (0 indexed).

**namecol:** Series name column (0 indexed).

**cfirstrow:** Categories first row (0 indexed).

**cfirstcol:** Categories first column (0 indexed).

**clastrow:** Categories last row (0 indexed).

**clastcol:** Categories last col (0 indexed).

**vfirstrow:** Values first row (0 indexed).

**vfirstcol:** Values first column (0 indexed).

**vlastrow:** Values last row (0 indexed).

**vlastcol:** Values last column (0 indexed).

**Notes:**

Returns "1" on success and "-1" on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

**Adding a series to a chart or chartsheet**

`e = XL_SERIES2(chartnum, sheet, namecell, cfirst, clast, vfirst, vlast)`

**Parameters:**

**chartnum:** Handle to a chart or chartsheet to add series.

**sheet:** Handle to sheet to get values from. Leave blank, "0", or "-1" to use the default sheet.

**namecell:** Series name Excel style cell. e.g. "A1" "D6" "F6".

**cfirst:** Categories first Excel style cell. e.g. "A1" "D6" "F6".

**clast:** Categories last Excel style cell. e.g. "A1" "D6" "F6".

**vfirst:** Values first Excel style cell. e.g. "A1" "D6" "F6".

**vlast:** Values last Excel style cell. e.g. "A1" "D6" "F6".

**Notes:**

Returns "1" on success and "-1" on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

[Back to top](#)

### Adding a password to restrict editing of a sheet

```
e = XL_PROTECTSHEET(sheet, password)
```

#### Parameters:

**sheet:** Handle to sheet to protect. Leave blank, “0”, or “-1” to use the default sheet.

**password:** Password used to protect this sheet.

#### Notes:

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

### Adding a password to restrict editing of a chartsheet

```
e = XL_PROTECTCHARTSHEET(cs, password)
```

#### Parameters:

**cs:** Handle to chartsheet to protect.

**password:** Password used to protect this sheet.

#### Notes:

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

### Returning the last error in the XLSX

```
e = XL_ERROR()
```

#### Notes:

Returns the last error generated by the XLSX set of functions.

Returns the last error string generated by the XLSX engine.

### Setting the auto counter position for a sheet

```
e = XL_SETPOS(row [, col [, sheet]])
```

#### Parameters:

**row:** Row to move auto counter to (0 indexed).

**col:** Column to move auto counter to (0 indexed).

**sheet:** Handle of sheet to set. Leave blank, “0”, or “-1” to use the default sheet.

[Back to top](#)

**Notes:**

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

**Setting the auto counter position for a sheet**

**e = XL\_SETPOS2 (cell [, sheet])**

**Parameters:**

**cell:** The Excel style cell to insert the cell. e.g. “A1”, “D6”, “F6”

**sheet:** Handle of sheet to set. Leave blank, “0”, or “-1” to use the default sheet.

**Notes:**

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

**Moving the auto counter down a row for a sheet**

**e = XL\_NEXTROW ([sheet])**

**Parameters:**

**sheet:** Handle of sheet to set. Leave blank, “0”, or “-1” to use the default sheet.

**Notes:**

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

**Moving the auto counter down a col for a sheet**

**e = XL\_NEXTCOL ([sheet])**

**Parameters:**

**sheet:** Handle of sheet to set. Leave blank, “0”, or “-1” to use the default sheet.

**Notes:**

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

[Back to top](#)

## Adding a new style to the XLSX document

```
e = XL_STYLE([font [, fill [, fmt [, btop [, bbot [, bleft [, bright]]]]]])
```

### Parameters:

**font:** Handle to font object to use.

**fill:** Handle to fill object to use.

**fmt:** Handle to format object to use.

**btop:** Handle to border object to use for top border.

**bbot:** Handle to border object to use for bottom border.

**bleft:** Handle to border object to use for left border.

**bright:** Handle to border object to use for right border.

### Notes:

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

## Adding a new image to the XLSX document

```
e = XL_IMAGE(img [, row [, col [, sheet [, xoff [, yoff [, scalex [, scaley [, flag]]]]]])
```

### Parameters:

**img:** Path to image file to use.

**row:** Row to insert the image on (0 indexed).

**col:** Column to insert the image on (0 indexed).

**sheet:** Handle of sheet to insert image. Leave blank, “0”, or “-1” to use the default sheet.

**xoff:** X-axis offset for the image, in pixels.

**yoff:** Y-axis offset for the image, in pixels.

**scalex:** Scale the image along the x-axis. e.g. "1", "0.5" "2". Value cannot be negative

**scaley:** Scale the image along the y-axis. e.g. "1", "0.5" "2". Value cannot be negative

**flag:** Option of how to position image.

“0” - Default Positioning.

“1” - Move and size image with the cells.

“2” - Move but don't size image with the cells.

“3” - Don't move or size the image with the cells.

[Back to top](#)

“4” - Same as "1" but wait to apply hidden cells.

**Notes:**

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

The image functions only support PNG, JPEG, and BMP files.

### Adding a new image to the XLSX document

```
e = XL_IMAGE2(img [, cell [, sheet [, xoff [, yoff [, scalex [, scaley [, flag]]]]]]])
```

**Parameters:**

**img:** Path to image file to use.

**cell:** The Excel style cell to insert the cell. e.g. “A1”, “D6”, “F6”

**sheet:** Handle of sheet to insert image. Leave blank, “0”, or “-1” to use the default sheet.

**xoff:** X-axis offset for the image, in pixels.

**yoff:** Y-axis offset for the image, in pixels.

**scalex:** Scale the image along the x-axis. e.g. "1", "0.5" "2". Value cannot be negative

**scaley:** Scale the image along the y-axis. e.g. "1", "0.5" "2". Value cannot be negative

**flag:** Option of how to position image.

“0” - Default Positioning.

“1” - Move and size image with the cells.

“2” - Move but don't size image with the cells.

“3” - Don't move or size the image with the cells.

“4” - Same as "1" but wait to apply hidden cells.

**Notes:**

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

The image functions only support PNG, JPEG, and BMP files.

### Getting debug information about the last XLSX call

```
e = XL_LASTCMD ()
```

**Notes:**

Returns the last evaluated command parsed string.

[Back to top](#)

### Setting the worksheet print margins

```
e = XL_MARGINS([left, [right, [top, [bottom, [sheet]]]])
```

#### Parameters:

**left:** Left margin in inches, e.g. "0.5", "1", "0.75". A blank or negative value will use the default of "0.7".

**right:** Right margin in inches, e.g. "0.5", "1", "0.75". A blank or negative value will use the default of "0.7".

**top:** Top margin in inches, e.g. "0.5", "1", "0.75". A blank or negative value will use the default of "0.7".

**bottom:** Bottom margin in inches, e.g. "0.5", "1", "0.75". A blank or negative value will use the default of "0.7".

**sheet:** Handle of sheet to set the margins. Leave blank, "0", or "-1" to use the default sheet.

#### Notes:

Returns "1" on success and "-1" on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

### Setting the worksheet to print in landscape mode

```
e = XL_LANDSCAPE([sheet])
```

#### Parameters:

**sheet:** Handle of sheet to change mode. Leave blank, "0", or "-1" to use the default sheet.

#### Notes:

Returns "1" on success and "-1" on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

### Setting the worksheet to print in portrait mode

```
e = XL_PORTRAIT([sheet])
```

#### Parameters:

**sheet:** Handle of sheet to change mode. Leave blank, "0", or "-1" to use the default sheet.

#### Notes:

Returns "1" on success and "-1" on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

[Back to top](#)

### Setting the worksheet to print in portrait mode

```
e = XL_GRIDLINES(option [, sheet])
```

#### Parameters:

**option:** Which gridlines to print. Cannot be blank. Must be one of the following values:

“hide\_all”, “show\_all”, “show\_screen”, “show\_print”

**sheet:** Handle of sheet to change mode. Leave blank, “0”, or “-1” to use the default sheet.

#### Notes:

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

### Fitting the printed area to a specific number of pages both vertically and horizontally

```
e = XL_FITPAGES([height, [width, [sheet]]])
```

#### Parameters:

**height:** Number of pages vertically. A value of "0" or blank will set the height as necessary.

**width:** Number of pages horizontally. A value of "0" or blank will set the height as necessary.

**sheet:** Handle of sheet to change mode. Leave blank, “0”, or “-1” to use the default sheet.

#### Notes:

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

### Setting the paper format for the printed output of a worksheet

```
e = XL_PAPERTYPE(type [, sheet])
```

#### Parameters:

**type:** The paper format to use with a printed worksheet. Must be one of the following values:

“default”, “letter”, “tabloid”, “ledger”, “legal”, “statement”,  
“executive”, “a3”, “a4”, “a5”, “b4”, “b5”, “folio”, “quarto”,  
“10x14”, “11x17”, “note”, “envelope”, “envelope\_9”,  
“envelope\_10”, “envelope\_11”, “envelope\_12”,  
“envelope\_14”, “c”, “d”, “e”, “envelope\_dl”, “envelope\_c3”,  
“envelope\_c4”, “envelope\_c5”, “envelope\_c6”,  
“envelope\_c65”, “envelope\_b4”, “envelope\_b5”,  
“envelope\_b6”, “monarch”, “fanfold”, “german\_std\_fanfold”,  
“german\_legal\_fanfold”

[Back to top](#)

**sheet:** Handle of sheet to change type. Leave blank, “0”, or “-1” to use the default sheet.

**Notes:**

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

### Centering the worksheet data horizontally between the margins on the printed page

**e = XL\_CENTERH([sheet])**

**Parameters:**

**sheet:** Handle of sheet to change mode. Leave blank, “0”, or “-1” to use the default sheet.

**Notes:**

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

### Centering the worksheet data vertically between the margins on the printed page

**e = XL\_CENTERV([sheet])**

**Parameters:**

**sheet:** Handle of sheet to change mode. Leave blank, “0”, or “-1” to use the default sheet.

**Notes:**

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

### Changing the default print direction to across then down

**e = XL\_PRINTACROSS([sheet])**

**Parameters:**

**sheet:** Handle of sheet to change mode. Leave blank, “0”, or “-1” to use the default sheet.

**Notes:**

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.



## Setting the printed page header and footer

```
e = XL_SETHEADER(string [, margin, [limage, [cimage, [rimage, [sheet]]]])
```

```
e = XL_SETFOOTER(string [, margin, [limage, [cimage, [rimage, [sheet]]]])
```

### Parameters:

**string:** The header/footer definition string. See below for format options. Cannot be blank.

**margin:** The margin in inches to use for the header/footer. A blank, "0", or negative value will use the default margin of "0.3".

**limage:** Full path to an image to use in place of the left image placeholder.

**cimage:** Full path to an image to use in place of the center image placeholder.

**rimage:** Full path to an image to use in place of the right image placeholder.

**sheet:** Handle of sheet to set header/footer. Leave blank, "0", or "-1" to use the default sheet.

Control	Category	Description
&L &C &R	Justification	Left Center Right
&P &N &D &T &F &A &Z	Information	Page number Total number of pages Date Time File Name Worksheet name Worksheet path
&fontsize &"font,style" &U &E &S &X &Y	Font	Font size Font name and style Single underline Double underline Strikethrough Superscript Subscript
&[Picture] &G	Images	Image placeholder Same as &[Picture]
&&	Miscellaneous	Literal ampersand &

[Back to top](#)

#### Notes:

Text in headers and footers can be justified to the left, center and right by prefixing the text with the control characters &L, &C and &R. For example, "&LHello, World!", "&CHello, World!", "&RHello, World!"

For simple text, if the justification is not specified the text will be center aligned. However, you must prefix the text with &C if you use any other formatting.

You are limited to 3 images in a header/footer.

The image types supported are PNG, JPEG, and BMP files. There is a hard limit of 255 characters in a header/footer string, including control characters. Strings longer than this will not be written to the document.

Returns "1" on success and "-1" on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

### Setting the background image for a worksheet

```
e = XL_SETBACKGROUND (image [, sheet])
```

#### Parameters:

**image:** Full path to an image to use as the sheet background.

**sheet:** Handle of sheet to set background image. Leave blank, "0", or "-1" to use the default sheet.

#### Notes:

Returns "1" on success and "-1" on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

The image types supported are PNG, JPEG, and BMP files.

### Hiding zero values in worksheet cells

```
e = XL_HIDEZEROS ([sheet])
```

#### Parameters:

**sheet:** Handle of sheet to change mode. Leave blank, "0", or "-1" to use the default sheet.

#### Notes:

Returns "1" on success and "-1" on error. [XL\\_ERROR\(\)](#) can be called to return the last error.

[Back to top](#)

## Showing row and column headers on the printed page

```
e = XL_SHOWROWCOL([sheet])
```

### Parameters:

**sheet:** Handle of sheet to change mode. Leave blank, “0”, or “-1” to use the default sheet.

### Notes:

Returns “1” on success and “-1” on error. [XL\\_ERROR\(\)](#) can be called to return the last error.