This document contains reference information on the following Excel macro functions:

# T

[TABLE](#xlfcttable), [TAB.ORDER](#xlfcttaborder), [TERMINATE](#xlfctterminate), [TEXT.BOX](#xlfcttextbox), [TEXTREF](#xlfcttextref), [TEXT.TO.COLUMNS](#xlfcttexttocolumns), [TRACER.CLEAR](#xlfcttracerclear), [TRACER.DISPLAY](#xlfcttracerdisplay), [TRACER.ERROR](#xlfcttracererror), [TRACER.NAVIGATE](#xlfcttracernavigate), [TTESTM](#xlfctttestm)

# U

[UNDO](#xlfctundo), [UNGROUP](#xlfctungroup), [UNHIDE](#xlfctunhide), [UNLOCKED.NEXT, UNLOCKED.PREV](#xlfctunlockednext), [UNREGISTER](#xlfctunregister), [UPDATE.LINK](#xlfctupdatelink)

# V

[VBA.INSERT.FILE](#xlfctvbainsertfile), [VBA.MAKE.ADDIN](#xlfctvbamakeaddin), [VIEW.3D](#xlfctview3d), [VIEW.DEFINE](#xlfctviewdefine), [VIEW.DELETE](#xlfctviewdelete), [VIEW.GET](#xlfctviewget), [VIEW.SHOW](#xlfctviewshow), [VLINE](#xlfctvline), [VOLATILE](#xlfctvolatile), [VPAGE](#xlfctvpage), [VSCROLL](#xlfctvscroll)

# W

[WAIT](#xlfctwait), [WHILE](#xlfctwhile), [WINDOW.MAXIMIZE](#xlfctwindowmaximize), [WINDOW.MINIMIZE](#xlfctwindowminimize), [WINDOW.MOVE](#xlfctwindowmove), [WINDOW.RESTORE](#xlfctwindowrestore), [WINDOWS](#xlfctwindows), [WINDOW.SIZE](#xlfctwindowsize), [WINDOW.TITLE](#xlfctwindowtitle), [WORKBOOK.ACTIVATE](#xlfctworkbookactivate), [WORKBOOK.ADD](#xlfctworkbookadd), [WORKBOOK.COPY](#xlfctworkbookcopy), [WORKBOOK.DELETE](#xlfctworkbookdelete), [WORKBOOK.HIDE](#xlfctworkbookhide), [WORKBOOK.INSERT](#xlfctworkbookinsert), [WORKBOOK.MOVE](#xlfctworkbookmove), [WORKBOOK.NAME](#xlfctworkbookname), [WORKBOOK.NEW](#xlfctworkbooknew), [WORKBOOK.NEXT](#xlfctworkbooknext), [WORKBOOK.OPTIONS](#xlfctworkbookoptions), [WORKBOOK.PREV](#xlfctworkbookprev), [WORKBOOK.PROTECT](#xlfctworkbookprotect), [WORKBOOK.SCROLL](#xlfctworkbookscroll), [WORKBOOK.SELECT](#xlfctworkbookselect), [WORKBOOK.TAB.SPLIT](#xlfctworkbooktabsplit), [WORKBOOK.UNHIDE](#xlfctworkbookunhide), [WORKGROUP](#xlfctworkgroup), [WORKSPACE](#xlfctworkspace)

# Z

[ZOOM](#xlfctzoom), [ZTESTM](#xlfctztestm)

# TABLE

Equivalent to clicking the Table command on the Data menu. Creates a table based on the input values and formulas you define on a worksheet. Use data tables to perform a "what-if" analysis by changing certain constant values in your workbook to see how values in other cells are affected.

**Syntax**

**TABLE**(row\_ref, column\_ref)

**TABLE**?(row\_ref, column\_ref)

Row\_ref    specifies the one cell to use as the row input for your table.

* Row\_ref should be either an external reference to a single cell on the active worksheet, such as !$A$1 or !Price, or an R1C1-style reference to a single cell in the form of text, such as "R1C1", "R[-1]C[-1]", or "Price".
* If row\_ref is an R1C1-style reference, it is assumed to be relative to the active cell in the selection.

Column\_ref    specifies the one cell to use as the column input for your table. Column\_ref is subject to the same restrictions as row\_ref.

Return to [top](#T)

# TAB.ORDER

This function determines the order in which dialog controls will be selected when the user presses the TAB key.

**Syntax**

**TAB.ORDER**?( )

**Remarks**

* This function brings up the Tab Order dialog box and allows the user to select the order in which buttons will be selected when the TAB key is pressed.
* The BRING.TO.FRONT and SEND.TO.BACK macro functions can also be used to programmatically set up the tab order.

**Related Functions**

BRING.TO.FRONT   Puts the selected object or objects on top of all other objects

SEND.TO.BACK   Puts the selected object or objects behind all other objects

Return to [top](#T)

# TERMINATE

Closes a dynamic data exchange (DDE) channel previously opened with the INITIATE function. Use TERMINATE to close a channel after you have finished communicating with another application.

**Syntax**

**TERMINATE**(**channel\_num**)

**Important**Microsoft Excel for the Macintosh requires system software version 7.0 or later for this function.

Channel\_num    is the number returned by a previously run INITIATE function. Channel\_num identifies a DDE channel to close.

If TERMINATE is not successful, it returns the #VALUE! error value.

**Related Functions**

EXEC   Starts another application

INITIATE   Opens a channel to another application

Return to [top](#T)

# TEXT.BOX

Replaces characters in a text box or button with the text you specify.

**Syntax**

**TEXT.BOX**(**add\_text**, object\_id\_text, start\_num, num\_chars)

Add\_text    is the text you want to add to the text box or button.

Object\_id\_text    is the name of the text box or button to which you want to add text (for example, "Text 1" or "Button 2"). If object\_id\_text is omitted, it is assumed to be the selected item.

Start\_num    is a number specifying the position of the first character you want to replace (or the position at which you want to insert characters if you do not want to replace any). If start\_num is omitted, it is assumed to be 1.

Num\_chars    is the number of characters you want to replace. If num\_chars is 0, then no characters are replaced, and add\_text is inserted starting at the position start\_num. If num\_chars is omitted, all the characters are replaced.

**Examples**

The following macro formula replaces the first five characters in a text box named "Text 5" with the text "Net Income":

TEXT.BOX("Net Income", "Text 5", 1, 5)

The following macro formula inserts the words "Account Summary for 1991" at the beginning of a text box named "Text 6":

TEXT.BOX("Account Summary for 1991", "Text 6", 1, 0)

**Related Functions**

CREATE.OBJECT   Creates an object

FONT.PROPERTIES   Applies a font to the selection

GET.OBJECT   Returns information about an object

Return to [top](#T)

# TEXTREF

Converts text to an absolute reference in either A1- or R1C1-style. Use TEXTREF to convert references stored as text to references so that you can use them with other functions, such as OFFSET.

**Syntax**

**TEXTREF**(**text**, a1)

Text    is a reference in the form of text.

A1    is a logical value specifying the reference type of text. If a1 is TRUE, text is assumed to be an A1-style reference; if FALSE or omitted, text is assumed to be an R1C1-style reference.

**Remarks**

* If you use TEXTREF by itself in a cell, you will get the value contained in the cell specified by text, not the reference itself, because references are automatically converted into the contents of the referenced cell.
* If you use TEXTREF as a reference argument to a function, Microsoft Excel does not convert the reference to a value.

**Tip**You can convert a reference to text with REFTEXT, manipulate it with the REPLACE and MID functions, and convert it back to a reference with TEXTREF.

**Examples**

TEXTREF("B7", TRUE) equals the reference value $B$7

TEXTREF("R5C5", FALSE) equals the reference value R5C5

TEXTREF("B7", FALSE) equals the #REF! error value, because "B7" can't be interpreted as an R1C1-style reference.

**Related Functions**

DEREF   Returns the values of the cells in a reference

REFTEXT   Converts a reference to text

Return to [top](#T)

# TEXT.TO.COLUMNS

Equivalent to clicking the Text To Columns command on the Data menu when a column of data is to be separated into multiple columns. Parses text into columns of data.

**Syntax**

**TEXT.TO.COLUMNS**(destination\_ref, data\_type, text\_delim, consecutive\_delim, tab, semicolon, comma, space, other, other\_char, field\_info)

The following arguments correspond to check boxes, option buttons and text buttons in the Text To Columns Wizard, which is started with the Text To Columns command on the Data menu.

Destination\_Ref    is a single cell reference to specifies where to place the converted text.

Data\_Type    is a number specifying whether that data is delimited (1) or fixed width (2)

Text\_Delim    denotes how text strings are represented, and can be one of the following values:

|  |  |
| --- | --- |
| **Number** | **Text\_Delim** |
| 1 | " |
| 2 | ' |
| 3 | none |

Consecutive\_delim    is a logical value corresponding to the Treat Consecutive Delimiters as One check box which, if TRUE, allows consecutive delimiters (such as ",,,") to be treated as a single delimiter. If FALSE, all consecutive delimiters are considered separate column breaks.

Tab    is a logical value which, if TRUE, specifies that the column has tab delimiters. If FALSE, the column does not have tab delimiters. Tab is ignored if data\_type is 2.

Semicolon    is a logical value which, if TRUE, specifies that the column has semicolon delimiters. If FALSE, the column does not have semicolon delimiters. Semicolon is ignored if data\_type is 2.

Comma    is a logical value which, if TRUE, specifies that the column has comma delimiters. If FALSE, the column does not have comma delimiters. Comma is ignored if data\_type is 2.

Space    is a logical value which, if TRUE, specifies that the column has space delimiters. If FALSE, the column does not have space delimiters. Space is ignored if data\_type is 2.

Other    is a logical value which, if TRUE, specifies that the column has custom delimiters. If FALSE, the column does not have custom delimiters. Other is ignored if data\_type is 2.

Other\_Char    is a single character that specifies a delimiter not included in the list of delimiters. Other\_Char is ignored if data\_type is 2 and if the argument other is FALSE.

Field\_Info    is an array which consists of the following elements: "column number, data\_format", if data\_type is 1; or "start\_pos, data\_format" if data\_type is 2. The second number defines the column's data format, and can be one of the following.

|  |  |
| --- | --- |
| **2nd Number** | **Data Format** |
| 1 | General |
| 2 | Text |
| 3 | Date, in the form MDY |
| 4 | Date, in the form DMY |
| 5 | Date, in the form YMD |
| 6 | Date, in the form MYD |
| 7 | Date, in the form DYM |
| 8 | Date, in the form YDM |
| 9 | Do not import column (skip) |

Return to [top](#T)

# TRACER.CLEAR

Equivalent to clicking the Remove All Arrows button on the Auditing toolbar on a worksheet. Clears all tracer arrows on the worksheet.

**Syntax**

**TRACER.CLEAR**()

**Remark**

Returns the #VALUE! error value if not available; for example, the selection is something other than worksheet.

**Related Function**

TRACER.DISPLAY   Allows tracer arrow to be displayed showing relationship among cells

Return to [top](#T)

# TRACER.DISPLAY

Equivalent to clicking the Trace Precedents or Trace Dependents buttons on the Auditing toolbar on a worksheet. Allows tracer arrow to be graphically displayed showing relationship among cells.

**Syntax**

**TRACER.DISPLAY**(direction, create)

Direction    is logical value which, if TRUE, displays tracer arrows for precedents. If FALSE tracer arrows for dependents are displayed.

Create    is a logical value which, if TRUE displays the next level of tracer arrows in the direction specified by direction. If FALSE, removes the current level of tracer arrows in the direction specified by direction. A level is the number of "arrows" away from the source cell.

**Remark**

Returns the #VALUE! error value if not available; for example, the selection is something other than a worksheet, or the cell(s) cannot be traced.

**Related Function**

TRACER.CLEAR   Clears all tracer arrows on the worksheet

Return to [top](#T)

# TRACER.ERROR

Equivalent to clicking the Trace Error button on the Auditing toolbar on a worksheet. Allow tracer arrows to be graphically displayed showing which cells are generating an error value.

**Syntax**

**TRACER.ERROR**()

Returns TRUE if Microsoft Excel successfully found the cell at which the error occurred. Returns FALSE if an error is not found.

**Remark**

* Returns the #VALUE! error value if not available; for example, the selection is something other than worksheet, or cell(s) that cannot be traced.
* If you need to know if there is an error in a cell, use ISERROR().

**Related Functions**

TRACER.DISPLAY   Allows tracer arrow to be displayed

TRACER.CLEAR   Clears all tracer arrows on the worksheet

Return to [top](#T)

# TRACER.NAVIGATE

Equivalent to double-clicking on a displayed tracer arrow. Moves the selection from one end of a tracer arrow to the other. If it is an error tracer arrow, then the selection goes to the end of the branch.

**Syntax**

**TRACER.NAVIGATE**(direction, arrow\_num, ref\_num)

Direction    is a logical value which, if TRUE, moves the selection to the arrow endpoint in the precedents direction. If FALSE, moves the selection to the arrow endpoint in the dependents direction.

Arrow\_num    is a number specifying which reference a tracer arrow will follow. For example, a 1 indicates that the arrow will follow the first reference in the formula. 1 is the default.

Ref\_num    If the arrow is an external reference arrow with multiple links, this argument tells which of the links to follow. Refer to the Links dialog, which is displayed with the Links command from the Edit menu. If ref\_num is 1, the link in the first reference in the Links dialog box will be followed. The default is 1.

**Remarks**

* Returns TRUE if successful. Returns FALSE if arrow\_num exceeds the number of tracer arrows or if there are no tracer arrows.
* Returns FALSE if ref\_num exceeds the number of links.
* Returns the #VALUE! error value if not available; for example, if the selection is something other than a worksheet, or the active cell does not contain an arrow.

**Related Function**

TRACER.DISPLAY   Allows tracer arrow to be displayed showing which cells formulas in other cells depend on

Return to [top](#T)

# TTESTM

Performs a two-sample Student's t-Test for means, assuming equal variances.

If this function is not available, you must install the Analysis ToolPak add-in.

**Syntax**

**TTESTM**(**inprng1, inprng2**, outrng, labels, alpha, difference)

**TTESTM**?(inprng1, inprng2, outrng, labels, alpha, difference)

Inprng1    is the input range for the first data set.

Inprng2    is the input range for the second data set.

Outrng    is the first cell (the upper-left cell) in the output table or the name, as text, of a new sheet to contain the output table. If FALSE, blank, or omitted, places the output table in a new workbook.

Labels    is a logical value.

* If labels is TRUE, then labels are in the first row or column of the input ranges.
* If labels is FALSE or omitted, all cells in inprng1 and inprng2 are considered data. The output table will include default row or column headings.

Alpha    is the confidence level for the test. If omitted, alpha is 0.05.

Difference    is the hypothesized difference in means. If omitted, difference is 0.

**Related Functions**

PTTESTM   Performs a paired two-sample Student's t-Test for means

PTTESTV   Performs a two-sample Student's t-Test, assuming unequal variances

Return to [top](#T)

# UNDO

Equivalent to clicking the Undo command on the Edit menu. Reverses certain actions and commands. UNDO is available in the same situations as the Undo command.

**Syntax**

**UNDO**( )

Return to [top](#T)

# UNGROUP

Separates a grouped object into individual objects. Use UNGROUP to separate a group of objects so that you can format, move, or size one of the objects.

If the selection is not a grouped object, UNGROUP returns FALSE.

**Syntax**

**UNGROUP**( )

**Related Function**

GROUP   Groups selected objects

Return to [top](#T)

# UNHIDE

Equivalent to clicking the Unhide command on the Window menu. Use UNHIDE to display hidden windows.

**Syntax**

**UNHIDE**(**window\_text**)

Window\_text    is the name of the window to unhide. If window\_text is not the name of an open workbook, an error value is returned and the macro is interrupted. You cannot unhide a window of an add-in workbook.

**Tip**You can use UNHIDE to activate an embedded chart in order to edit and format it. Use the HIDE function to de-activate the chart window.

**Related Functions**

GET.WINDOW   Returns information about a window

HIDE   Hides the active window

Return to [top](#T)

# UNLOCKED.NEXT, UNLOCKED.PREV

Equivalent to pressing TAB or SHIFT+TAB to move to the next or previous unlocked cell in a protected worksheet. Use these functions when you want to control which cell is active on a protected sheet.

**Syntax**

**UNLOCKED.NEXT**( )

**UNLOCKED.PREV**( )

**Related Functions**

CELL.PROTECTION   Controls protection for the selected cells

PROTECT.DOCUMENT   Controls protection for the active sheet

Return to [top](#T)

# UNREGISTER

Unregisters a previously registered dynamic link library (DLL) or code resource. You can use UNREGISTER to free memory that was allocated to a DLL or code resource when it was registered. There are two syntax forms of this function. Use syntax 1 when you want Microsoft Excel to unregister a function or code resource according to its use count. Use syntax 2 when you want Microsoft Excel to unregister a function or code resource regardless of the use count.

**Syntax 1**

**UNREGISTER**(**register\_id**)

Register\_id    is the register ID returned by the REGISTER or REGISTER.ID function, which corresponds to the function or code resource to be removed from memory.

Microsoft Excel counts the number of times you register a function or code resource. This number is called the use count. Each time you unregister a function or code resource, its use count is decremented by 1. When the use count equals 0, Microsoft Excel frees the allocated memory. Therefore, if you register a function or code resource more than once, you must use a corresponding number of UNREGISTER functions to ensure that it is completely unregistered.

**Note**Because Microsoft Excel for Windows and Microsoft Excel for the Macintosh use different types of code resources, UNREGISTER has a slightly different syntax form when used in each operating environment.

**Syntax 2a**

For Microsoft Excel for Windows

**UNREGISTER**(**module\_text**)

**Syntax 2b**

For Microsoft Excel for the Macintosh

**UNREGISTER**(**file\_text**)

Module\_text or file\_text    is text specifying the name of the dynamic link library (DLL) that contains the function (in Microsoft Excel for Windows) or the name of the file that contains the code resource (in Microsoft Excel for the Macintosh).

If you use this syntax of UNREGISTER, all functions in the DLL (or all code resources in the file) are immediately unregistered, regardless of the use count.

**Examples**

Assuming that a REGISTER function in cell A5 of a macro sheet has already run (and has run only once), the following macro formula unregisters the corresponding function or code resource:

UNREGISTER(A5)

You could also use REGISTER.ID to return the register ID, instead of specifying a cell reference:

UNREGISTER(REGISTER.ID("User", "GetTickCount")

Assuming that you have registered several different functions from the USER.EXE DLL of Microsoft Windows, the following macro formula unregisters all functions in that DLL:

UNREGISTER("User")

**Tip**If you register a function or code resource, and use the optional function\_text argument to specify a custom name that will appear in the Paste Function dialog box, this custom name will not be removed by the UNREGISTER function. To remove the custom name, use the SET.NAME function without its second argument.

**Related Function**

REGISTER   Registers a code resource

Return to [top](#T)

# UPDATE.LINK

Equivalent to clicking the Links command on the Edit menu and clicking the Update Now button with a link selected in the Links dialog box. Updates a link to another document. Use UPDATE.LINK to get the newest information from a supporting document.

**Syntax**

**UPDATE.LINK**(link\_text, type\_of\_link)

Link\_text    is a text string describing the full path of the link as displayed in the Links dialog box. If link\_text is omitted, only links from the active workbook to other Microsoft Excel workbooks are updated.

Type\_of\_link    is a number from 1 to 4 that specifies the type of link to update.

|  |  |
| --- | --- |
| **Type\_of\_link** | **Link document type** |
| 1 or omitted | Microsoft Excel link |
| 2 | DDE link |
| 3 | Not available |
| 4 | Not available |

**Related Functions**

CHANGE.LINK   Changes supporting links

GET.LINK.INFO   Returns information about a link

OPEN.LINKS   Opens specified supporting documents

Return to [top](#T)

# VBA.INSERT.FILE

Inserts a text file containing code directly into your Visual Basic module.

**Syntax**

**VBA.INSERT.FILE**(**filename\_text**)

Filename\_text    is the name of a text file that contains Microsoft Visual Basic code that is inserted into the currently active module.

Return to [top](#T)

# VBA.MAKE.ADDIN

Converts a workbook containing Visual Basic procedures into an add-in.

**Syntax**

**VBA.MAKE.ADDIN**(**filename\_text**)

Filename\_text    is the name of the workbook that you want to convert to an add-in.

**Remarks**

For information about creating add-ins with Visual Basic, see Chapter 13, "Creating Automatic Procedures and Add-in Applications" in the Visual Basic User's Guide.

Return to [top](#T)

# VIEW.3D

Equivalent to clicking the 3-D View command on the Format menu in Microsoft Excel version 4.0, available when a chart sheet is the active sheet. Adjusts the view of the active 3-D chart. Use VIEW.3D to emphasize different parts of your chart by viewing it from different angles and perspectives.

**Syntax**

**VIEW.3D**(elevation, perspective, rotation, axes, height%, autoscale)

**VIEW.3D**?(elevation, perspective, rotation, axes, height%, autoscale)

Elevation    is a number from -90 to 90 specifying the viewing elevation of the chart and is measured in degrees. Elevation corresponds to the Elevation box in the 3-D View dialog box in Microsoft Excel version 4.0.

* If elevation is 0, you view the chart straight on. If elevation is 90, you view the chart from above (a "bird's eye view"). If elevation is -90, you view the chart from below.
* If elevation is omitted, the current value is used..
* Elevation is limited to 0 to 44 for 3-D bar charts and 0 to 80 for 3-D pie charts.

Perspective    is a number from 0 to 100% specifying the perspective of the chart. Perspective corresponds to the Perspective box in the 3-D View dialog box in Microsoft Excel version 4.0.

* A higher perspective value simulates a closer view.
* If perspective is omitted, the current value is used..
* Perspective is ignored on 3-D bar and pie charts.

Rotation    is a number from 0 to 360 specifying the rotation of the chart around the value (z) axis and is measured in degrees. Rotation corresponds to the Rotation box in the 3-D View dialog box in Microsoft Excel version 4.0. As you rotate the chart, the back and side walls are moved so that they do not block the chart.

* If rotation is omitted, the current value is used..
* Rotation is limited to 0 to 44 for 3-D bar charts.

Axes    is a logical value specifying whether axes are fixed in the plane of the screen or can rotate with the chart. Axes corresponds to the Right Angle Axes check box in the 3-D View dialog box in Microsoft Excel version 4.0.

* If axes is TRUE, Microsoft Excel locks the axes.
* If axes is FALSE, Microsoft Excel allows the axes to rotate.
* If axes is omitted and the chart view is 3-D layout, axes is assumed to be FALSE.
* Axes is TRUE for 3-D bar charts and ignored for 3-D pie charts.

Height%    is a number from 5 to 500 specifying the height of the chart as a percentage of the length of the base. Height% corresponds to the Height box in the 3-D View dialog box in Microsoft Excel version 4.0. Height% is useful for controlling the appearance of charts with many series or data points. If height% is omitted, the current value is used..

Autoscale    is a logical value corresponding to the Auto Scaling check box in the 3-D View dialog box in Microsoft Excel version 4.0. If TRUE, automatic scaling is used; if FALSE, it is not; if omitted, the current setting is not changed.

**Related Function**

FORMAT.MAIN   Formats a main chart

Return to [top](#T)

# VIEW.DEFINE

Equivalent to clicking the Add button in the Custom Views dialog box in Microsoft Excel 97 or later, which appears when you click the Custom Views command on the View menu. In Microsoft Excel 97, the Custom Views command replaced the View Manager command that was available in Microsoft Excel 95 and earlier versions. Creates or replaces a view.

If this function is not available in Microsoft Excel 95 or in earlier versions, you must install the View Manager add-in.

**Syntax**

**VIEW.DEFINE**(**view\_name**, print\_settings\_log, row\_col\_log)

View\_name    is text enclosed in quotation marks and specifies the name of the view you want to define for the sheet. If the workbook already contains a view with view\_name, the new view replaces the existing one.

Print\_settings\_log    is a logical value that, if TRUE or omitted, includes current print settings in the view or, if FALSE, does not include current print settings in the view.

Row\_col\_log    is a logical value that, if TRUE or omitted, includes current row and column settings in the view or, if FALSE, does not include current row and column settings in the view.

**Related Functions**

VIEW.DELETE   Removes a view from the active workbook

VIEW.SHOW   Shows a view

Return to [top](#T)

# VIEW.DELETE

Equivalent to selecting a view and clicking the Delete button in the Custom Views dialog box, which appears when you click the Custom Views command on the View menu. In Microsoft Excel 97 or later, the Custom Views command replaced the View Manager command that was available in Microsoft Excel 95 and earlier versions. Removes a view from the active workbook.

If this function is not available in Microsoft Excel 95 or in earlier versions, you must install the View Manager add-in.

**Syntax**

**VIEW.DELETE**(**view\_name**)

View\_name    is text enclosed in quotation marks and specifies the name of the view in the current workbook that you want to delete.

**Remarks**

VIEW.DELETE returns the #VALUE error value if view\_name is invalid or if the workbook is protected.

**Related Functions**

VIEW.DEFINE   Creates or replaces a view

VIEW.SHOW   Shows a view

Return to [top](#T)

# VIEW.GET

Equivalent to displaying a list of views in the Custom Views dialog box, which appears when you click the Custom Views command on the View menu. In Microsoft Excel 97 or later, the Custom Views command replaces the View Manager command that was available in Microsoft Excel 95 and earlier versions. Returns an array of views from the active workbook.

If this function is not available in Microsoft Excel 95 or in earlier versions, you must install the View Manager add-in.

**Syntax**

**VIEW.GET**(**type\_num**, view\_name)

Type\_num    is a number from 1 to 3 that specifies the type of information to return, as shown in the following table.

|  |  |
| --- | --- |
| **Type\_num** | **Result** |
| 1 | Returns an array of views from all the sheets in the active workbook or the #N/A error value if none are defined. |
| 2 | Returns TRUE if print settings are included in the specified view. Returns FALSE if print settings are not included. Returns the #VALUE! error value if the name is invalid or the workbook is protected. |
| 3 | Returns TRUE if row and column settings are included in the specified view. Returns FALSE if row and column settings are not included. Returns the #VALUE! error value if the name is invalid or the workbook is protected. |

View\_name    is text enclosed in quotation marks and specifies the name of a view in the active workbook. View\_name is required if type\_num is 2 or 3.

**Example**

The following macro formula returns an array of views from the active workbook:

VIEW.GET(1)

**Related Functions**

VIEW.DEFINE   Creates or replaces a view

VIEW.DELETE   Removes a view from the active workbook

VIEW.SHOW   Shows a view

Return to [top](#T)

# VIEW.SHOW

Equivalent to selecting a view and clicking the Show button in the Custom Views dialog box, which appears when you click the Custom Views command on the View menu. In Microsoft Excel 97 or later, the Custom Views command replaces the View Manager command that was available in Microsoft Excel 95 and earlier versions. Shows a view.

If this function is not available in Microsoft Excel 95 or in earlier versions, you must install the View Manager add-in.

**Syntax**

**VIEW.SHOW**(**view\_name**)

**VIEW.SHOW**?(view\_name)

View\_name    is text enclosed in quotation marks and specifies the name of a view in the active workbook.

**Remarks**

VIEW.SHOW returns the #VALUE error value if view\_name is invalid or the workbook is protected.

**Related Functions**

VIEW.DEFINE   Creates or replaces a view

VIEW.DELETE   Removes a view from the active workbook

Return to [top](#T)

# VLINE

Scrolls through the active window vertically by the number of rows you specify.

**Syntax**

**VLINE**(**num\_rows**)

Num\_rows    is a number that specifies how many rows to scroll.

* If num\_rows is positive, Microsoft Excel scrolls down by the number of rows indicated by num\_rows.
* If num\_rows is negative, Microsoft Excel scrolls up by the number of rows indicated by num\_rows.

**Related Functions**

HLINE   Horizontally scrolls through the active window by columns

HPAGE   Horizontally scrolls through the active window one window at a time

HSCROLL   Horizontally scrolls through a worksheet by percentage or by column number

VPAGE   Vertically scrolls through the active window one window at a time

VSCROLL   Vertically scrolls through a worksheet by percentage or by row number

Return to [top](#T)

# VOLATILE

Specifies whether a custom worksheet function is volatile or nonvolatile. A volatile custom function is recalculated every time a calculation occurs on the worksheet.

**Syntax**

**VOLATILE**(logical)

Logical    is a logical value specifying whether the custom function is volatile or nonvolatile. If logical is TRUE or omitted, the function is volatile; if FALSE, nonvolatile.

**Remarks**

* VOLATILE must precede every other formula in the custom function except RESULT and ARGUMENT.
* Normally, a worksheet recalculates a cell containing a nonvolatile custom function only when any part of the complete formula in the cell is recalculated. Use VOLATILE(TRUE) to recalculate the function every time the worksheet is recalculated.
* Most custom functions are nonvolatile by default, but custom functions with reference arguments are volatile by default. Use VOLATILE(FALSE) to prevent these functions from being recalculated unnecessarily often.

**Related Function**

RESULT   Specifies the data type a custom function returns

Return to [top](#T)

# VPAGE

Vertically scrolls through the active window one window at a time. Use VPAGE to change the displayed area of a worksheet or macro sheet.

**Syntax**

**VPAGE**(**num\_windows**)

Num\_windows    specifies the number of windows to scroll through the active window vertically. A window is defined as the number of visible rows. If 20 rows are visible in the window, VPAGE scrolls in increments of 20 rows.

* If num\_windows is positive, VPAGE scrolls down.
* If num\_windows is negative, VPAGE scrolls up.

**Related Functions**

HPAGE   Horizontally scrolls through the active window one window at a time

HSCROLL   Horizontally scrolls through a worksheet by percentage or by column number

VLINE   Vertically scrolls through the active window by rows

VSCROLL   Vertically scrolls through a worksheet by percentage or by row number

Return to [top](#T)

# VSCROLL

Vertically scrolls through the active sheet by percentage or by row number.

**Syntax**

**VSCROLL**(**position**, row\_logical)

Position    specifies the row you want to scroll to. Position can be an integer representing the row number or a fraction or percentage representing the vertical position of the row in the sheet. If position is 0, VSCROLL scrolls through your sheet to its top edge, which is row 1. If position is 1, VSCROLL scrolls through your sheet to its bottom edge, which is row 16, 384 in Microsoft Excel 95 or earlier, or row 65,536 in Microsoft Excel 97 or later. For charts that do not size with the window, use a fraction or percentage.

Row\_logical    is a logical value specifying how the function scrolls.

* If row\_logical is TRUE, VSCROLL scrolls through the sheet to row position.
* If row\_logical is FALSE or omitted, VSCROLL scrolls through the sheet to the vertical position represented by the fraction position.

**Remarks**

* To scroll to a specific row n, either use VSCROLL(n, TRUE) or VSCROLL(n/16384) in Microsoft Excel 95 or earlier; in Microsoft Excel 97 or later, you should use VSCROLL(n/65536). To scroll to row 138, for example, enter VSCROLL(138, TRUE) (in any version) or VSCROLL(138/16384) in earlier versions of Microsoft Excel or VSCROLL(138/65536) in Microsoft Excel 97 or later
* If you are recording a macro and move the scroll box several times in a row, the recorder only records the final location of the scroll box, omitting any intermediate steps. Remember that scrolling does not change the active cell or the selection.

**Related Functions**

FORMULA.GOTO   Selects a named area or reference on any open workbook

HLINE   Horizontally scrolls through the active window by columns

HPAGE   Horizontally scrolls through the active window one window at a time

HSCROLL   Horizontally scrolls through a sheet by percentage or by column number

SELECT   Selects a cell, object, or chart item

VLINE   Vertically scrolls through the active window by rows

VPAGE   Vertically scrolls through the active window one window at a time

Return to [top](#T)

# WAIT

Pauses the macro until the time specified by the serial number.

**Syntax**

**WAIT**(**serial\_number**)

Serial\_number    is the date-time code used by Microsoft Excel for date and time calculations. You can give serial\_number as text, such as "4:30 PM", or as a formula, such as NOW()+"00:00:04", instead of as a number. The text or formula is automatically converted to a serial number. For more information about serial\_number, see NOW.

**Important**WAIT suspends all Microsoft Excel activity and may prevent you from performing other operations on your computer. Background processes, such as printing and recalculation, are continued.

**Example**

Use WAIT with NOW to pause a macro for a length of time or until the time specified by the serial number. For example, the following macro formula waits 3 seconds from the time the functions are evaluated:

WAIT(NOW()+"00:00:03")

**Related Function**

ON.TIME   Runs a macro at a specific time

Return to [top](#T)

# WHILE

Carries out the statements between the WHILE function and the next NEXT function until logical\_test is FALSE. Use WHILE-NEXT loops to carry out a series of macro formulas while a certain condition remains TRUE.

**Syntax**

**WHILE**(**logical\_test**)

Logical\_test    is a value or formula that evaluates to TRUE or FALSE. If logical\_test is FALSE the first time the WHILE function is reached, the macro skips the loop and resumes running at the statement after the next NEXT function. If there is no NEXT function in the same column, WHILE displays an error message and interrupts the macro.

**Remarks**

If you know exactly how many times you'll need to carry out the statements within a loop, in most cases you should use a FOR-NEXT loop. Also, avoid creating an infinite loop by making sure logical\_test does not always evaluate to TRUE.

**Example**

The following statement starts a loop that executes while the value in the current cell is less than 5:

=WHILE(TYPE(ACTIVE.CELL()<5))

The following statement starts a loop that executes until the position in the open file identified as FileNumber reaches the end of the file:

=WHILE(FPOS(FileNumber)<=FSIZE(FileNumber))

**Related Functions**

FOR   Starts a FOR-NEXT loop

FOR.CELL   Starts a FOR.CELL-NEXT loop

IF   Specifies an action to take if a logical test is TRUE

NEXT   Ends a FOR-NEXT, FOR.CELL-NEXT, or WHILE-NEXT loop

Return to [top](#T)

# WINDOW.MAXIMIZE

Changes the active window from its normal size to full size. In Microsoft Excel for Windows, using WINDOW.MAXIMIZE is equivalent to pressing CTRL+F10 or double-clicking the title bar. In Microsoft Excel for the Macintosh, using WINDOW.MAXIMIZE is equivalent to double-clicking the title bar or clicking the zoom box.

**Syntax**

**WINDOW.MAXIMIZE**(window\_text)

Window\_text    specifies which window to switch to and maximize. Window\_text is text enclosed in quotation marks or a reference to a cell containing text. If window\_text is omitted, the active window is maximized.

**Remarks**

WINDOW.MAXIMIZE replaces FULL(TRUE) in earlier versions of Microsoft Excel.

**Related Functions**

WINDOW.MINIMIZE   Minimizes a window

WINDOW.MOVE   Moves a window

WINDOW.RESTORE   Restores a window to its previous size

WINDOW.SIZE   Changes the size of a window

Return to [top](#T)

# WINDOW.MINIMIZE

Shrinks a window to an icon. In Microsoft Excel for Windows, using WINDOW.MINIMIZE is equivalent to clicking the minimize button on a workbook window. In Microsoft Excel for the Macintosh, the minimize feature is not supported.

**Syntax**

**WINDOW.MINIMIZE**(window\_text)

Window\_text    specifies which window to minimize.

* Window\_text is text enclosed in quotation marks or a reference to a cell containing text.
* If window\_text is omitted, Microsoft Excel minimizes the active window.

**Remarks**

If a window is already minimized, WINDOW.MINIMIZE has no effect.

**Related Functions**

WINDOW.MAXIMIZE   Maximizes a window

WINDOW.MOVE   Moves a window

WINDOW.RESTORE   Restores a window to its previous size

WINDOW.SIZE   Changes the size of a window

Return to [top](#T)

# WINDOW.MOVE

Equivalent to clicking the Move command on the Control menu in Microsoft Excel for Windows or moving a window by dragging its title bar or its icon. Moves the active window so that its upper-left corner is at the specified horizontal and vertical positions. The dialog-box form, WINDOW.MOVE?, is supported only in Microsoft Excel for Windows.

**Syntax**

**WINDOW.MOVE**(x\_pos, y\_pos, window\_text)

**WINDOW.MOVE**?(x\_pos, y\_pos, window\_text)

X\_pos    is the horizontal position to which you want to move the window. X\_pos is measured in points. A point is 1/72nd of an inch.

* In Microsoft Excel for Windows, x\_pos is measured from the left edge of your workspace to the left edge of the window.
* In Microsoft Excel for the Macintosh, x\_pos is measured from the left edge of your screen to the left edge of the window.
* If x\_pos is omitted, the window does not move horizontally.

Y\_pos    is the vertical position to which you want to move the window. Y\_pos in measured in points from the bottom edge of the formula bar to the top edge of the window. If y\_pos is omitted, the window does not move vertically.

Window\_text    specifies which window to restore.

* Window\_text is text enclosed in quotation marks or a reference to a cell containing text.
* If window\_text is omitted, it is assumed to be the name of the active window.

**Remarks**

* If the window is minimized, WINDOW.MOVE moves the icon on the workspace. Measurements are relative to the upper-left corner of the workspace and the icon.
* WINDOW.MOVE does not change the size of the window or affect whether the specified window is active or inactive.
* In Microsoft Excel for the Macintosh, if window\_text is "Clipboard", WINDOW.MOVE moves the Clipboard. The Clipboard must already be available; if it is not available, use the SHOW.CLIPBOARD function before using the WINDOW.MOVE function.
* WINDOW.MOVE replaces MOVE in earlier versions of Microsoft Excel.

**Related Functions**

FORMAT.MOVE   Moves the selected object

WINDOW.MAXIMIZE   Maximizes a window

WINDOW.MINIMIZE   Minimizes a window

WINDOW.RESTORE   Restores a window to its previous size

WINDOW.SIZE   Changes the size of a window

Return to [top](#T)

# WINDOW.RESTORE

Changes the active window from maximized or minimized size to its previous size. In Microsoft Excel for Windows, using WINDOW.RESTORE is equivalent to pressing CTRL+F5 or double-clicking the title bar (or double-clicking the icon if it is minimized). In Microsoft Excel for the Macintosh, using WINDOW.RESTORE is equivalent to double-clicking the title bar or clicking the zoom box.

**Syntax**

**WINDOW.RESTORE**(window\_text)

Window\_text    specifies which window to switch to and restore.

* Window\_text is text enclosed in quotation marks or a reference to a cell containing text.
* If window\_text is omitted, Microsoft Excel restores the active window.

**Remarks**

* If the window is minimized, WINDOW.RESTORE restores the icon to its previous size. This operation is equivalent to double-clicking the icon.
* WINDOW.RESTORE replaces FULL(FALSE) in earlier versions of Microsoft Excel.

**Related Functions**

WINDOW.MAXIMIZE   Maximizes a window

WINDOW.MINIMIZE   Minimizes a window

WINDOW.MOVE   Moves a window

WINDOW.SIZE   Changes the size of a window

Return to [top](#T)

# WINDOWS

Returns the names of the specified open Microsoft Excel windows, including hidden windows. Use WINDOWS to get a list of active windows for use by other macro functions that return information about or manipulate windows, such as GET.WINDOW and ACTIVATE. The names are returned as a horizontal array of text values, in order of their appearance on your screen. The first name is the active window, the next name is the window directly under the active window, and so on.

**Syntax**

**WINDOWS**(type\_num, match\_text)

Type\_num    is a number that specifies which types of workbooks are returned by WINDOWS, according to the following table.

|  |  |
| --- | --- |
| **Type\_num** | **Returns window names from these types of documents** |
| 1 or omitted | All windows except those belonging to add-in workbooks |
| 2 | Add-in workbooks only |
| 3 | All types of workbooks |

Match\_text    specifies the windows whose names you want returned and can include wildcard characters. If match\_text is omitted, WINDOWS returns the names of all open windows.

**Tips**

* You can change the output of a horizontal array to vertical with the TRANSPOSE function.
* You can use WINDOWS with the INDEX function to select individual window names from the array for use in other functions that take window names as arguments.
* You can use the COLUMNS functions to count the number of entries in the array, which is the number of windows.

**Examples**

If the active window, named BOOK1, is on top of a window named MACROS:2, which is on top of a window named MACROS:1, then:

WINDOWS() equals {"BOOK1", "MACROS:2", "MACROS:1"}

**Related Functions**

ACTIVATE   Switches to a window

DOCUMENTS   Returns the names of the specified open workbooks

GET.WINDOW   Returns information about a window

NEW.WINDOW   Creates a new window for an existing sheet or macro sheet

ON.WINDOW   Runs a macro when you switch to a window

Return to [top](#T)

# WINDOW.SIZE

Equivalent to clicking the Size command on the Control menu or to adjusting the sizing borders (in Microsoft Excel for Windows) or the sizing box (in Microsoft Excel for the Macintosh) of the window with the mouse. Changes the size of the active window by moving its lower-right corner so that the window has the width and height you specify. WINDOW.SIZE does not change the position of the upper-left corner of the window, nor does it affect whether the specified window is active or inactive.

**Syntax**

**WINDOW.SIZE**(**width, height**, window\_text)

**WINDOW.SIZE**?(width, height, window\_text)

Width    specifies the width of the window and is measured in points. A point is 1/72nd of an inch.

Height    specifies the height of the window and is measured in points.

Window\_text    specifies which window to size.

* Window\_text is text enclosed in quotation marks or a reference to a cell containing text.
* If window\_text is omitted, it is assumed to be the name of the active window.

**Remarks**

* In Microsoft Excel for Windows, an error occurs if you try to resize a window that has already been minimized to an icon or enlarged to its maximum size. You must first restore the window to its original size using the WINDOW.RESTORE function. For more information, see WINDOW.RESTORE.
* WINDOW.SIZE replaces SIZE in earlier versions of Microsoft Excel.

**Related Functions**

FORMAT.SIZE   Sizes an object

WINDOW.MAXIMIZE   Maximizes a window

WINDOW.MINIMIZE   Minimizes a window

WINDOW.MOVE   Moves a window

WINDOW.RESTORE   Restores a window to its previous size

Return to [top](#T)

# WINDOW.TITLE

Changes the title of the active window to the title you specify. The title appears at the top of the workbook window. Use WINDOW.TITLE to control window titles when you're using Microsoft Excel to create a custom application.

**Syntax**

**WINDOW.TITLE**(text)

Text    is the title you want to assign to the window. If text is omitted, it is assumed to be the name of the workbook as it is stored on your disk. Empty text ("") specifies no title.

**Important**   WINDOW.TITLE changes the name of the window, not the actual name of the workbook as it is stored on your disk. To change the name of the workbook, use the SAVE.AS function.

**Remarks**

* The window name you create using WINDOW.TITLE will appear on the Window menu, and will be returned by the WINDOWS function, but not by the DOCUMENTS function. You must use the new window name in theACTIVATE function and the ON.WINDOW function.
* If you want to communicate with a Microsoft Excel workbook using DDE functions like INITIATE or REQUEST, you must specify the filename of the workbook and not the window title specified with the WINDOW.TITLE function.
* If you use NEW.WINDOW to create new windows on the workbook, the window title will be restored to its original name.

**Example**

The following macro formula changes the title of the active window to First Quarter.

WINDOW.TITLE("First Quarter")

**Related Functions**

APP.TITLE   Changes the title of the application workspace

SAVE.AS   Specifies a new filename, file type, protection password, or write-reservation password, or to create a backup file

Return to [top](#T)

# WORKBOOK.ACTIVATE

Equivalent to activating a worksheet by clicking on its tab.

**Syntax**

**WORKBOOK.ACTIVATE**(**sheet\_name**)

Sheet\_name    is the name of the sheet you want to activate. You can use WORKBOOK.ACTIVATE to activate a sheet within the active workbook, or you can also activate a sheet in another workbook by using the [workbook]sheet\_name reference for the sheet\_name argument.

**Related Functions**

WORKBOOK.SELECT   Selects one or more sheets for group editing

WORKBOOK.OPTIONS   Changes the settings of a workbook sheet

Return to [top](#T)

# WORKBOOK.ADD

Macro SheetsOnly

Equivalent to clicking the Add button in the workbook contents window in Microsoft Excel version 4.0. Moves a sheet between workbooks. For Microsoft Excel version 5.0 or later use WORKBOOK.MOVE.

**Syntax**

**WORKBOOK.ADD**(**name\_array**, dest\_book, position\_num)

**WORKBOOK.ADD**?(name\_array, dest\_book, position\_num)

Name\_array    is the name of a sheet, or an array of names of sheets, that you want to move.

Dest\_book     is the name of the workbook to which you want to add name\_array. If dest\_book is omitted, it is assumed to be the active workbook.

Position\_num    is a number that specifies the position of the sheet within the workbook.

**Related Functions**

WORKBOOK.MOVE   Moves one or more sheets between workbooks or changes a sheet's position within a workbook

WORKBOOK.COPY   Copies one or more documents from their current workbook to another workbook

Return to [top](#T)

# WORKBOOK.COPY

Equivalent to clicking the Move or Copy Sheet command on the Edit menu. Copies one or more sheets from their current positions to the specified position in the same workbook or to another workbook.

**Syntax**

**WORKBOOK.COPY**(**name\_array**, dest\_book, position\_num)

**WORKBOOK.COPY**?(name\_array, dest\_book, position\_num)

Name\_array    is the name of a sheet, or an array of names of sheets, that you want to copy in the currently active workbook.

Dest\_book    is the name of the workbook to which you want to copy name\_array. If dest\_book is the current workbook, name\_array is copied within the workbook. If dest\_book is omitted, the copy of name\_array becomes a separate workbook.

Position\_num    is a number that specifies the target position for the sheet within the new workbook. The first position is 1.

* If position\_num is specified, Microsoft Excel inserts the copy of the sheet at the specified position in the workbook.
* If position\_num is omitted, Microsoft Excel places the sheet at the last position in the workbook.
* If dest\_book is omitted, position\_num is ignored.

**Remarks**

* If the structure of the workbook is protected, you cannot copy sheets within the workbook or to another workbook.
* You cannot copy a hidden sheet.

**Related Function**

WORKBOOK.MOVE   Moves one or more documents from one workbook to another workbook or to another position in the same workbook

Return to [top](#T)

# WORKBOOK.DELETE

Equivalent to clicking the Delete Sheet command on the Edit menu. Deletes a sheet or group of sheets from the current workbook.

**Syntax**

**WORKBOOK.DELETE**(sheet\_text)

Sheet\_text    is the name of the sheet to delete. If omitted, the currently active sheet or sheets is deleted.

**Remarks**

* This function prompts for confirmation. To suppress the prompt, use the ERROR function. For example, =ERROR(FALSE).
* If the structure of the workbook is protected, you cannot delete any of its sheets.
* If you want to delete Sheet1:Sheet10, you must select them first with WORKBOOK.SELECT(). You can also place the sheets in an array first, as in {"Sheet1", "Sheet2", "Sheet3",...}.
* You cannot delete the last visible sheet in a workbook.

Return to [top](#T)

# WORKBOOK.HIDE

Equivalent to clicking the Sheet command on the Format menu, and then clicking Hide on the Sheet submenu. Hides sheets in the active workbook.

**Syntax**

**WORKBOOK.HIDE**(sheet\_text, very\_hidden)

Sheet\_text    is the name of the sheet to hide. If omitted, the currently selected sheet(s) are hidden.

Very\_hidden    specifies how the sheet is hidden. If TRUE, then the sheet name does not appear in the Unhide dialog box. After using this argument, use WORKBOOK.UNHIDE to unhide the sheet. If FALSE or omitted, hides the sheet but does not prevent the sheet's name from appearing in the Unhide dialog box.

**Remarks**

* If the structure of the workbook is protected, you cannot hide any sheets in the workbook.
* You cannot hide the last visible sheet in a workbook.
* To hide Sheet1:Sheet10, select them first with the WORKBOOK.SELECT function. You can also place the sheets in an array first, as in {"Sheet1", "Sheet2", "Sheet3",...}.

Return to [top](#T)

# WORKBOOK.INSERT

Equivalent to clicking the Worksheet, Chart, or Macro commands on the Insert menu. Inserts one or more new sheets into the current workbook.

**Syntax**

**WORKBOOK.INSERT**(**type\_num**)

**WORKBOOK.INSERT**?(**type\_num**)

**Type\_num**    specifies the type of sheet to insert.

|  |  |
| --- | --- |
| **Type\_num** | **Type of sheet** |
| 1 | Worksheet |
| 2 | Chart |
| 3 | Microsoft Excel 4.0 Macro Sheet |
| 4 | Microsoft Excel 4.0 International Macro Sheet |
| 5 | (Reserved) |
| 6 | Microsoft Excel Visual Basic Module |
| 7 | Dialog |
| Quoted text | Template |

If omitted, the type of the active sheet is used.

If the current selection contains one sheet, then only one sheet is inserted. If the selection contains more than one sheet and the active sheet is a worksheet, then an equal number of sheets is inserted to the left of the selected group of sheets.

**Remarks**

* The new sheets are always inserted to the left of the current selection.
* If the workbook structure is protected, you cannot insert new sheets.

Return to [top](#T)

# WORKBOOK.MOVE

Equivalent to clicking the Move or Copy Sheet command on the Edit menu. Moves one or more sheets between workbooks or changes a sheet's position within a workbook.

**Syntax**

**WORKBOOK.MOVE**(**name\_array**, dest\_book, position\_num)

**WORKBOOK.MOVE**?(name\_array, dest\_book, position\_num)

Name\_array    is the name of a sheet or an array of names of sheets in the active workbook that you want to move.

Dest\_book    is the name of the workbook to which you want to move name\_array. If dest\_book is omitted, WORKBOOK.MOVE moves the sheet out of the workbook and puts it in a new separate workbook. If dest\_book is the same as the current book, then the sheet is moved within the workbook.

Position\_num    is a number that specifies the target position for the sheet within dest\_book. The first position is 1.

* If position\_num is specified, Microsoft Excel inserts the sheet at the specified position in the workbook.
* If position\_num is omitted, Microsoft Excel moves the sheet to the last position in the workbook. If you move the last sheet out of a workbook, the workbook closes.

**Related Function**

WORKBOOK.COPY   Copies one or more documents from their current workbook into another workbook

Return to [top](#T)

# WORKBOOK.NAME

Equivalent to clicking the Rename command on the Sheet submenu of the Format menu. Renames a sheet in a workbook.

**Syntax**

**WORKBOOK.NAME**(**oldname\_text**, **newname\_text**)

**WORKBOOK.NAME**?(oldname\_text, newname\_text)

Oldname\_text    is the name of the sheet that you want to rename.

Newname\_text    is the new name of the sheet.

**Remarks**

* If you try to rename a sheet using a sheet name that already exists in the workbook, Microsoft Excel displays an error message and interrupts the macro.
* If the structure of the workbook is protected, you cannot rename any of the sheets in the workbook.

Return to [top](#T)

# WORKBOOK.NEW

Adds a sheet to a workbook. This function is for compatibility with Microsoft Excel version 4.0. To add a new sheet to a workbook in Microsoft Excel version 5, use the WORKBOOK.INSERT function.

**Syntax**

**WORKBOOK.NEW**( )

**WORKBOOK.NEW**?( )

In both forms of this function, you will be prompted for the name of the workbooks.

**Related Function**

WORKBOOK.INSERT   Adds sheets to workbooks

Return to [top](#T)

# WORKBOOK.NEXT

Activates the next sheet in the active workbook.

**Syntax**

**WORKBOOK.NEXT**( )

**Remarks**

* If the last sheet in the workbook is active, this function has no effect.
* This function skips over hidden sheets in the workbook.

Return to [top](#T)

# WORKBOOK.OPTIONS

Equivalent to selecting the Options button in a workbook contents window in Microsoft Excel version 4.0. This function is for compatibility with Microsoft Excel version 4.0. To change the name of a sheet in Microsoft Excel version 5.0, use the WORKBOOK.NAME function.

**Syntax**

**WORKBOOK.OPTIONS**(**sheet\_name**, bound\_logical, new\_name)

Sheet\_name    is the name of a sheet.

Bound\_logical    is for compatibility with Microsoft Excel version 4.0. In Microsoft Excel version 5.0 and later versions, this should be TRUE or omitted because FALSE returns an error.

New\_name    is the sheet name to assign to sheet\_name. If new\_name is omitted, then the sheet name is not changed.

**Related Functions**

GET.WORKBOOK   Returns information about a workbook sheet

WORKBOOK.NAME   Changes the name of a sheet in a workbook

WORKBOOK.SELECT   Selects the specified sheets in a workbook

Return to [top](#T)

# WORKBOOK.PREV

Activates the previous sheet in the workbook.

**Syntax**

**WORKBOOK.PREV**( )

**Remarks**

* If the first sheet in the workbook is active, this function has no effect.
* This function skips over hidden sheets in the workbook.

Return to [top](#T)

# WORKBOOK.PROTECT

Equivalent to clicking the Protect Workbook command on the Protection submenu of the Tools menu. Controls protection of workbooks.

**Syntax**

**WORKBOOK.PROTECT**(structure, windows, password)

**WORKBOOK.PROTECT**?(structure, windows, password)

Structure    specifies whether the structure of the workbook is protected. If TRUE, the structure is protected. If FALSE or omitted, the structure is not protected.

Windows    specifies whether the windows of the workbook are protected. If TRUE, the windows are protected. If FALSE or omitted, the windows are not protected.

Password    specifies whether to protect the workbook with a password. If omitted no password is used. When recording a macro, this argument is not recorded. In the dialog box form of this function, you can specify a password.

**Remarks**

To protect a sheet in a workbook, use the PROTECT.DOCUMENT function.

**Related Function**

PROTECT.DOCUMENT   Protects a sheet in a workbook

Return to [top](#T)

# WORKBOOK.SCROLL

Scrolls through the sheets in a workbook.

**Syntax**

**WORKBOOK.SCROLL**(**num\_sheets**, firstlast\_logical)

Num\_sheets. is a number specifying how many sheets to scroll and the direction of scrolling. Positive numbers scroll forward by that many sheets. Negative numbers scroll backward by that many sheets. Zero (0) does not scroll.

Firstlast\_logical    specifies whether to scroll to the first or last sheet in the workbook. If TRUE, scrolls to the last sheet in the workbook. If FALSE, scrolls to the first sheet in the workbook. If this argument is specified, then num\_sheets is ignored.

Return to [top](#T)

# WORKBOOK.SELECT

Equivalent to selecting a sheet or group of sheets in the active workbook. If you select a group of sheets, subsequent commands effect all the sheets in the group.

**Syntax**

**WORKBOOK.SELECT**(name\_array, active\_name, replace)

Name\_array    is a horizontal array of text names of sheets you want to select. If name\_array is omitted, no sheets are selected.

Active\_name    is the name of a single sheet in the workbook that you want to be the active sheet. If active\_name is omitted, the first sheet in name\_array is made the active sheet.

Replace    specifies whether the currently selected sheets or macro sheets are to be replaced by name\_array. If TRUE or omitted, then the current sheet selection is replaced by name\_array. If FALSE, then name\_array will be appended to the current sheet.

**Related Functions**

GET.WORKBOOK   Returns information about a workbook

SELECT   Selects a cell, worksheet object, or chart item

Return to [top](#T)

# WORKBOOK.TAB.SPLIT

Sets the ratio of the tabs to the horizontal scrollbar.

**Syntax**

**WORKBOOK.TAB.SPLIT**(ratio\_num)

Ratio\_num    is the ratio of tabs to horizontal scroll, as a decimal value between 0 and 1. If omitted defaults to 6.

**Remarks**

* If the structure of the workbook is protected, you cannot use this function.
* Use GET.WINDOW(28) to find out what the current ratio is.

**Related Function**

GET.WINDOW   Returns information about a workbook window

Return to [top](#T)

# WORKBOOK.UNHIDE

Equivalent to clicking the Unhide command on the sheet submenu of the Format menu. Unhides one or more sheets in the current workbook.

**Syntax**

**WORKBOOK.UNHIDE**(sheet\_text)

**WORKBOOK.UNHIDE**?(sheet\_text)

Sheet\_text. specifies the sheet that you want to unhide. If sheet\_text is omitted, then this function unhides the sheets in the order that they would appear in the workbook.

**Remarks**

If the workbook is protected, you cannot unhide any sheets in the book.

**Related Function**

WORKBOOK.HIDE   Hides sheets in the active workbook

Return to [top](#T)

# WORKGROUP

Equivalent to clicking the Group Edit command on the Options menu in Microsoft Excel version 4.0. Creates a group. This function is provided for compatibility only. In Microsoft Excel version 5.0 and later versions, you can create a group by using the WORKBOOK.SELECT function.

**Syntax**

**WORKGROUP**(name\_array)

**WORKGROUP**?(name\_array)

Name\_array    is the list of workbooks or sheets in workbooks that you want grouped.

* If name\_array is omitted, the most recently created group is recreated.
* If no group has been created during the current Microsoft Excel session, all open, unhidden worksheets are created as a group.
* If you specify just the name of a workbook, WORKGROUP adds the first sheet of the workbook to the group.

**Remarks**

WORKGROUP returns the #VALUE! error value and interrupts the macro if it can't find any of the sheets in name\_array or if any of the sheets is a chart or module.

**Related Functions**

FILL.GROUP   Fills the contents of the active worksheet's selection to the same area on all other worksheets in the group

WORKBOOK.SELECT   Selects one or more sheets in a workbook

Return to [top](#T)

# WORKSPACE

Changes the workspace settings for a workbook. This function is provide for compatibility with Microsoft Excel version 4.0 only. In Microsoft Excel version 5.0 and later versions, you can change workbook settings with OPTIONS.GENERAL function.

**Syntax**

**WORKSPACE**(fixed, decimals, r1c1, scroll, status, formula, menu\_key, remote, entermove, underlines, tools, notes, nav\_keys, menu\_key\_action, drag\_drop, show\_info)

**WORKSPACE**?(fixed, decimals, r1c1, scroll, status, formula, menu\_key, remote, entermove, underlines, tools, notes, nav\_keys, menu\_key\_action, drag\_drop, show\_info)

Arguments correspond to check boxes and text boxes in the Workspace dialog box. Arguments corresponding to check boxes are logical values. If an argument is TRUE, the check box is selected; if FALSE, the check box is cleared; if omitted, the current setting is not changed.

Fixed    corresponds to the Fixed Decimal check box.

Decimals    specifies the number of decimal places. Decimals is ignored if fixed is FALSE or omitted.

R1c1    corresponds to the R1C1 check box.

Scroll    corresponds to the Scroll Bars check box.

Status    corresponds to the Status Bar check box.

Formula    corresponds to the Formula Bar check box.

Menu\_key    is a text value indicating an alternate menu key, and corresponds to the Alternate Menu Or Help Key box.

Remote    corresponds to the Ignore Remote Requests check box.

**Important**Microsoft Excel for the Macintosh requires system software version 7.0 or later for this argument.

Entermove    corresponds to the Move Selection After Enter/Return check box.

Underlines    is a number corresponding to the Command Underline options as shown in the following table.

**Note**This argument is only available in Microsoft Excel for the Macintosh.

|  |  |
| --- | --- |
| **If underlines is** | **Command underlines are** |
| 1 | On |
| 2 | Off |
| 3 | Automatic |

Tools    is a logical value. If TRUE, the Standard toolbar is displayed; if FALSE, all visible toolbars are hidden. If omitted, the current toolbar display is not changed.

Notes    corresponds to the Note Indicator check box.

Nav\_keys    corresponds to the Alternate Navigation Keys check box. In Microsoft Excel for the Macintosh, nav\_keys is ignored.

Menu\_key\_action    is the number 1 or 2 specifying options for the alternate menu or Help key. In Microsoft Excel for the Macintosh, menu\_key\_action is ignored.

|  |  |
| --- | --- |
| **Menu\_key\_action** | **Alternate menu or Help key activates** |
| 1 or omitted | Microsoft Excel menus |
| 2 | Lotus 1-2-3 Help |

Drag\_drop    corresponds to the Cell Drag And Drop check box.

Show\_info    corresponds to the Info Window check box.

**Related Function**

GET.WORKSPACE   Returns information about the workspace

Return to [top](#T)

# ZOOM

Equivalent to choosing the Zoom command from the View menu. Enlarges or reduces a sheet in the active window. Use ZOOM when you need to view more cells than would normally fit in the active windows, or fewer cells at a larger size.

**Syntax**

**ZOOM**(magnification)

Magnification    is a logical value or a number specifying the amount of enlargement or reduction.

* Magnification can be a number from 10 to 400 specifying the percentage of enlargement or reduction.
* If magnification is TRUE or omitted, the current selection is enlarged or reduced to completely fill the active window.
* If magnification is FALSE, the sheet is restored to normal 100% magnification.

**Related Function**

PRINT.PREVIEW   Previews pages and page breaks before printing.

Return to [top](#T)

# ZTESTM

Performs a two-sample z-test for means, assuming the two samples have known variances.

If this function is not available, you must install the Analysis ToolPak add-in.

**Syntax**

**ZTESTM**(**inprng1, inprng2**, outrng, labels, alpha, difference, **var1, var2**)

**ZTESTM**?(inprng1, inprng2, outrng, labels, alpha, difference, var1, var2)

Inprng1    is the input range for the first data set.

Inprng2    is the input range for the second data set.

Outrng    is the first cell (the upper-left cell) in the output table or the name, as text, of a new sheet to contain the output table. If FALSE, blank, or omitted, places the output table in a new workbook.

Labels    is a logical value.

* If labels is TRUE, then the first row or column of the input ranges contains labels.
* If labels is FALSE or omitted, all cells in inprng1 and inprng2 are considered data. Microsoft Excel will then generate the appropriate data labels for the output table.

Alpha    is the confidence level for the test. If omitted, alpha is 0.05.

Difference    is the hypothesized difference in means. If omitted, difference is 0.

Var1    is the variance of the first data set.

Var2    is the variance of the second data set.

Return to [top](#T)