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

# Q

[QUERY.GET.DATA](#xlfctquerygetdata), [QUERY.REFRESH](#xlfctqueryrefresh), [QUIT](#xlfctquit)

# R

[RANDOM](#xlfctrandom), [RANKPERC](#xlfctrankperc), [REFTEXT](#xlfctreftext), [REGISTER](#xlfctregister), [REGRESS](#xlfctregress), [RELREF](#xlfctrelref), [REMOVE.LIST.ITEM](#xlfctremovelistitem), [REMOVE.PAGE.BREAK](#xlfctremovepagebreak), [RENAME.COMMAND](#xlfctrenamecommand), [RENAME.OBJECT](#xlfctrenameobject), [REPLACE.FONT](#xlfctreplacefont), [REPORT.DEFINE](#xlfctreportdefine), [REPORT.DELETE](#xlfctreportdelete), [REPORT.GET](#xlfctreportget), [REPORT.PRINT](#xlfctreportprint), [REQUEST](#xlfctrequest), [RESET.TOOL](#xlfctresettool), [RESET.TOOLBAR](#xlfctresettoolbar), [RESTART](#xlfctrestart), [RESULT](#xlfctresult), [RESUME](#xlfctresume), [RETURN](#xlfctreturn), [ROUTE.DOCUMENT](#xlfctroutedocument), [ROUTING.SLIP](#xlfctroutingslip), [ROW.HEIGHT](#xlfctrowheight), [RUN](#xlfctrun)

# S

[SAMPLE](#xlfctsample), [SAVE](#xlfctsave), [SAVE.AS](#xlfctsaveas), [SAVE.COPY.AS](#xlfctsavecopyas), [SAVE.DIALOG](#xlfctsavedialog), [SAVE.TOOLBAR](#xlfctsavetoolbar), [SAVE.WORKBOOK](#xlfctsaveworkbook), [SAVE.WORKSPACE](#xlfctsaveworkspace), [SCALE](#xlfctscale), [SCALE Syntax 1](#xlfctscale1), [SCALE Syntax 2](#xlfctscale2), [SCALE Syntax 3](#xlfctscale3), [SCALE Syntax 4](#xlfctscale4), [SCALE Syntax 5](#xlfctscale5), [SCENARIO.ADD](#xlfctscenarioadd), [SCENARIO.CELLS](#xlfctscenariocells), [SCENARIO.DELETE](#xlfctscenariodelete), [SCENARIO.EDIT](#xlfctscenarioedit), [SCENARIO.GET](#xlfctscenarioget), [SCENARIO.MERGE](#xlfctscenariomerge), [SCENARIO.SHOW](#xlfctscenarioshow), [SCENARIO.SHOW.NEXT](#xlfctscenarioshownext), [SCENARIO.SUMMARY](#xlfctscenariosummary), [SCROLLBAR.PROPERTIES](#xlfctscrollbarproperties), [SELECT](#xlfctselect), [SELECT Syntax 1](#xlfctselect1), [SELECT Syntax 2](#xlfctselect2), [SELECT Syntax 3](#xlfctselect3), [SELECT.ALL](#xlfctselectall), [SELECT.CHART](#xlfctselectchart), [SELECT.END](#xlfctselectend), [SELECTION](#xlfctselection), [SELECT.LAST.CELL](#xlfctselectlastcell), [SELECT.LIST.ITEM](#xlfctselectlistitem), [SELECT.PLOT.AREA](#xlfctselectplotarea), [SELECT.SPECIAL](#xlfctselectspecial), [SEND.KEYS](#xlfctsendkeys), [SEND.MAIL](#xlfctsendmail), [SEND.TO.BACK](#xlfctsendtoback), [SERIES](#xlfctseries), [SERIES.AXES](#xlfctseriesaxes), [SERIES.ORDER](#xlfctseriesorder), [SERIES.X](#xlfctseriesx), [SERIES.Y](#xlfctseriesy), [SET.CONTROL.VALUE](#xlfctsetcontrolvalue), [SET.CRITERIA](#xlfctsetcriteria), [SET.DATABASE](#xlfctsetdatabase), [SET.DIALOG.DEFAULT](#xlfctsetdialogdefault), [SET.DIALOG.FOCUS](#xlfctsetdialogfocus), [SET.EXTRACT](#xlfctsetextract), [SET.LIST.ITEM](#xlfctsetlistitem), [SET.NAME](#xlfctsetname), [SET.PAGE.BREAK](#xlfctsetpagebreak), [SET.PREFERRED](#xlfctsetpreferred), [SET.PRINT.AREA](#xlfctsetprintarea), [SET.PRINT.TITLES](#xlfctsetprinttitles), [SET.UPDATE.STATUS](#xlfctsetupdatestatus), [SET.VALUE](#xlfctsetvalue), [SHORT.MENUS](#xlfctshortmenus), [SHOW.ACTIVE.CELL](#xlfctshowactivecell), [SHOW.BAR](#xlfctshowbar), [SHOW.CLIPBOARD](#xlfctshowclipboard), [SHOW.DETAIL](#xlfctshowdetail), [SHOW.DIALOG](#xlfctshowdialog), [SHOW.INFO](#xlfctshowinfo), [SHOW.LEVELS](#xlfctshowlevels), [SHOW.TOOLBAR](#xlfctshowtoolbar), [SIZE](#xlfctsize), [SLIDE.COPY.ROW](#xlfctslidecopyrow), [SLIDE.CUT.ROW](#xlfctslidecutrow), [SLIDE.DEFAULTS](#xlfctslidedefaults), [SLIDE.DELETE.ROW](#xlfctslidedeleterow), [SLIDE.EDIT](#xlfctslideedit), [SLIDE.GET](#xlfctslideget), [SLIDE.PASTE](#xlfctslidepaste), [SLIDE.PASTE.ROW](#xlfctslidepasterow), [SLIDE.SHOW](#xlfctslideshow), [SOLVER.ADD](#xlfctsolveradd), [SOLVER.CHANGE](#xlfctsolverchange), [SOLVER.DELETE](#xlfctsolverdelete), [SOLVER.FINISH](#xlfctsolverfinish), [SOLVER.GET](#xlfctsolverget), [SOLVER.LOAD](#xlfctsolverload), [SOLVER.OK](#xlfctsolverok), [SOLVER.OPTIONS](#xlfctsolveroptions), [SOLVER.RESET](#xlfctsolverreset), [SOLVER.SAVE](#xlfctsolversave), [SOLVER.SOLVE](#xlfctsolversolve), [SORT](#xlfctsort), [SOUND.NOTE](#xlfctsoundnote), [SOUND.PLAY](#xlfctsoundplay), [SPELLING](#xlfctspelling), [SPELLING.CHECK](#xlfctspellingcheck), [SPLIT](#xlfctsplit), [SQL.BIND](#xlfctsqlbind), [SQL.CLOSE](#xlfctsqlclose), [SQL.ERROR](#xlfctsqlerror), [SQL.EXEC.QUERY](#xlfctsqlexecquery), [SQL.GET.SCHEMA](#xlfctsqlgetschema), [SQL.OPEN](#xlfctsqlopen), [SQL.RETRIEVE](#xlfctsqlretrieve), [SQL.RETRIEVE.TO.FILE](#xlfctsqlretrievetofile), [STANDARD.FONT](#xlfctstandardfont), [STANDARD.WIDTH](#xlfctstandardwidth), [STEP](#xlfctstep), [STYLE](#xlfctstyle), [SUBSCRIBE.TO](#xlfctsubscribeto), [SUBTOTAL.CREATE](#xlfctsubtotalcreate), [SUBTOTAL.REMOVE](#xlfctsubtotalremove), [SUMMARY.INFO](#xlfctsummaryinfo)

# QUERY.GET.DATA

Builds a new query using the supplied information. The application Microsoft Query nor any dialog boxes are displayed.

**Syntax**

**QUERY.GET.DATA**(**connection\_string**, **query\_text**, keep\_query\_def, field\_names, row\_numbers, destination)

**QUERY.GET.DATA**?(connection\_string, query\_text, keep\_query\_def, field\_names, row\_numbers, destination)

Connection\_string    supplies information, such as the data source name, user ID, and passwords, necessary to making a SQL connection to an external data source. For example: "DSN=Myserver; Server=server1; UID=dbayer; PWD=buyer1; Database=nwind".

* You must define the data source name (DSN) used in connection\_string before you try to connect to it.
* You can enter connection\_string as an array or a string. If connection\_string exceeds 250 characters, you must enter it as an array.
* If QUERY.GET.DATA is unable to access the data source using connection\_string, it returns the #N/A error value.

Query\_text    is the SQL language query to be executed on the data source.

Keep\_query\_def    is a logical value that, if TRUE or omitted, preserves the query definition. If FALSE, the query definition is lost and the data from the query no longer constitutes a data range.

Field\_names    is a logical value that, if TRUE or omitted, places field names from Microsoft Query into the first row of the data range. If FALSE, the field names are discarded.

Row\_numbers    is a logical value that, if TRUE, places row numbers from Microsoft Query into the first column in the data range. If FALSE or omitted, the row numbers are discarded.

Destination    is the location as a cell reference where you want the data placed. If destination is in a current data range then that data range is changed to reflect the new SQL statement. The default destination is the currently selected cell or range.

**Remarks**

* If the information provided is not sufficient to create the query then the error value #REF! is returned.
* If Microsoft Query is unavailable or can not be found, #N/A is returned.
* If connection string is longer than 255 characters, the string will be truncated at the last semi-colon.

**Related Function**

QUERY.REFRESH   Refreshes the data in a data range returned by Microsoft Query

Return to [top](#Q)

# QUERY.REFRESH

Refreshes the data in a data range returned to a worksheet from Microsoft Query. This function is equivalent to the Refresh button on the External Data toolbar.

**Syntax**

**QUERY.REFRESH**(reference)

Reference    is the reference to a single cell inside a data range. If reference is not in a data range then the error value #REF! is returned.

**Related Function**

QUERY.GET.DATA   Builds a new query using the supplied information

Return to [top](#Q)

# QUIT

Equivalent to clicking the Exit command on the File menu in Microsoft Excel for Windows. Equivalent to clicking the Quit command on the File menu in Microsoft Excel for the Macintosh. Quits Microsoft Excel and closes any open workbooks. If open workbooks have unsaved changes, Microsoft Excel displays a message asking if you want to save them. You can use QUIT in an Auto\_Close macro to force Microsoft Excel to quit when a particular workbook is closed.

**Syntax**

**QUIT**( )

**Caution**If you have cleared error-checking with an ERROR(FALSE) function, QUIT will not ask whether you want to save changes.

**Remarks**

When you use the QUIT function, Microsoft Excel does not run any Auto\_Close macros before closing the workbook.

**Examples**

The following function displays a confirmation alert and quits Microsoft Excel if the user clicks OK:

IF(ALERT("Are you sure you want to quit Microsoft Excel?",1), QUIT(),)

**Related Function**

FILE.CLOSE   Closes the active workbook

Return to [top](#Q)

# RANDOM

Fills a range with independent random or patterned numbers drawn from one of several distributions.

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

RANDOM provides six different random distributions and one patterned data option. Because the distributions require different argument lists, there are seven syntax forms for RANDOM.

**Syntax 1**

Uniform distribution

**RANDOM**(outrng, variables, points, **distribution**, seed, **from, to**)

**RANDOM**?(outrng, variables, points, distribution, seed, from, to)

**Syntax 2**

Normal distribution

**RANDOM**(outrng, variables, points, **distribution**, seed, **mean, standard\_dev**)

**RANDOM**?(outrng, variables, points, distribution, seed, mean, standard\_dev)

**Syntax 3**

Bernoulli distribution

**RANDOM**(outrng, variables, points, **distribution**, seed, **probability**)

**RANDOM**?(outrng, variables, points, distribution, seed, probability)

**Syntax 4**

Binomial distribution

**RANDOM**(outrng, variables, points, **distribution**, seed, **probability, trials**)

**RANDOM**?(outrng, variables, points, distribution, seed, probability, trials)

**Syntax 5**

Poisson distribution

**RANDOM**(outrng, variables, points, **distribution**, seed, **lambda**)

**RANDOM**?(outrng, variables, points, distribution, seed, lambda)

**Syntax 6**

Patterned distribution

**RANDOM**(outrng, variables, points, **distribution**, seed, **from, to, step, repeat\_num, repeat\_seq**)

**RANDOM**?(outrng, variables, points, distribution, seed, from, to, step, repeat\_num, repeat\_seq)

**Syntax 7**

Discrete distribution

**RANDOM**(outrng, variables, points, **distribution**, seed, **inprng**)

**RANDOM**?(outrng, variables, points, distribution, seed, inprng)

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.

Variables    is the number of random number sets to generate. RANDOM will generate variables columns of random numbers. If omitted, variables is equal to the number of columns in the output range.

Points    is the number of data points per random number set. RANDOM will generate points rows of random numbers for each random number set. If omitted, points is equal to the number of rows in the output range. Points is ignored when distribution is 6 (Patterned).

Distribution    indicates the type of number distribution.

|  |  |
| --- | --- |
| **Distribution** | **Distribution type** |
| 1 | Uniform |
| 2 | Normal |
| 3 | Bernoulli |
| 4 | Binomial |
| 5 | Poisson |
| 6 | Patterned |
| 7 | Discrete |

Seed    is an optional value with which to begin random number generation. Seed is ignored when distribution is 6 (Patterned) or 7 (Discrete).

From    is the lower bound.

To    is the upper bound.

Mean    is the mean.

Standard\_dev    is the standard deviation.

Probability    is the probability of success on each trial.

Trials    is the number of trials.

Lambda    is the Poisson distribution parameter.

Step    is the increment between from and to.

Repeat\_num    is the number of times to repeat each value.

Repeat\_seq    is the number of times to repeat each sequence of values.

Inprng    is a two-column range of values and their probabilities.

Return to [top](#Q)

# RANKPERC

Returns a table that contains the ordinal and percent rank of each value in a data set.

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

**Syntax**

**RANKPERC**(**inprng**, outrng, grouped, labels)

**RANKPERC**?(inprng, outrng, grouped, labels)

Inprng    is the input range.

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.

Grouped    is a text character that indicates whether the data in the input range is organized by row or column.

* If grouped is "C" or omitted, then the data is organized by column.
* If grouped is "R", then the data is organized by row.

Labels    is a logical value that describes where the labels are located in the input range, as shown in the following table:

|  |  |  |
| --- | --- | --- |
| **Labels** | **Grouped** | **Labels are in** |
| TRUE | "C" | First row of the input range. |
| TRUE | "R" | First column of the input range. |
| FALSE or omitted | (ignored) | No labels. All cells in the input range are data. |

Return to [top](#Q)

# REFTEXT

Converts a reference to an absolute reference in the form of text. Use REFTEXT when you need to manipulate references with text functions. After manipulating the reference text, you can convert it back into a normal reference by using TEXTREF.

**Syntax**

**REFTEXT**(**reference**, a1)

Reference    is the reference you want to convert.

A1    is a logical value specifying A1-style or R1C1-style references.

* If a1 is TRUE, REFTEXT returns an A1-style reference.
* If a1 is FALSE or omitted, REFTEXT returns an R1C1-style reference.

**Examples**

REFTEXT(C3, TRUE) equals "$C$3"

REFTEXT(B2:F2) equals "R2C2:R2C6"

If the active cell is B9 on the active sheet named SHEET1, then:

REFTEXT(ACTIVE.CELL()) equals "[Book1]SHEET1!R9C2"

REFTEXT(ACTIVE.CELL(), TRUE) equals "[Book1]SHEET1!$B$9"

**Related Functions**

ABSREF   Returns the absolute reference of a range of cells to another range

DEREF   Returns the values of cells in the reference

RELREF   Returns a relative reference

TEXTREF   Converts text to a reference

Return to [top](#Q)

# REGISTER

Registers the specified dynamic link library (DLL) or code resource and returns the register ID. You can also specify a custom function name and argument names that will appear in the Paste Function dialog box. If you register a command (macro\_type = 2), you can also specify a shortcut key. Because Microsoft Excel for Windows and Microsoft Excel for the Macintosh use different types of code resources, REGISTER has a slightly different syntax form when used in each operating environment.

**Important**   This function is provided for advanced users only. If you use the CALL function incorrectly, you could cause errors that will require you to restart your computer.

**Syntax 1**

For Microsoft Excel for Windows

**REGISTER**(**module\_text**, procedure, type\_text, function\_text, argument\_text, macro\_type, category, shortcut\_text, help\_topic, function\_help, argument\_help1, argument\_help2,...)

**Syntax 2**

For Microsoft Excel for the Macintosh

**REGISTER**(**file\_text**, resource, type\_text, function\_text, argument\_text, macro\_type, category, shortcut\_text, help\_topic, function\_help, argument\_help1, argument\_help2,...)

Module\_text or file\_text    is text specifying the name of the 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).

Procedure or resource    is text specifying the name of the function in the DLL (in Microsoft Excel for Windows) or the name of the code resource (in Microsoft Excel for the Macintosh). In Microsoft Excel for Windows, you can also use the ordinal value of the function from the EXPORTS statement in the module-definition file (.DEF). In Microsoft Excel for the Macintosh, you can also use the resource ID number. The ordinal value or resource ID number should not be in text form.

This argument may be omitted for stand-alone DLLs or code resources. In this case, REGISTER will register all functions or code resources and then return module\_text or file\_text.

Type\_text    is text specifying the data type of the return value and the data types of all arguments to the DLL or code resource. The first letter of type\_text specifies the return value.

Function\_text    is text specifying the name of the function as you want it to appear in the Paste Function dialog box. If you omit this argument, the function will not appear in the Paste Function dialog box.

Argument\_text    is text specifying the names of the arguments you want to appear in the Paste Function dialog box. Argument names should be separated by commas.

Macro\_type    specifies the macro type: 1 for a function or 2 for a command. If macro\_type is omitted, it is assumed to be 1 (function).

Category    specifies the function category in the Paste Function dialog box in which you want the registered function to appear. You can use the category number or the category name for category. If you use the category name, be sure to enclose it in double quotation marks. If category is omitted, it is assumed to be 14 (User Defined).

|  |  |
| --- | --- |
| **Category number** | **Category name** |
| 1 | Financial |
| 2 | Date & Time |
| 3 | Math & Trig |
| 4 | Text |
| 5 | Logical |
| 6 | Lookup & Matrix |
| 7 | Database |
| 8 | Statistical |
| 9 | Information |
| 10 | Commands (macro sheets only) |
| 11 | Actions (macro sheets only) |
| 12 | Customizing (macro sheets only) |
| 13 | Macro Control (macro sheets only) |
| 14 | User Defined |

Shortcut\_text    is a character specifying the shortcut key for the registered command. The shortcut key is case-sensitive. This argument is used only if macro\_type = 2 (command). If shortcut\_text is omitted, the command will not have a shortcut key.

Help\_topic    is the reference (including path) to the help file that you want displayed when the user clicks the Help button when your custom function is displayed.

Function\_help    is a text string describing your custom function when it is selected in the Paste Function dialog box. The maximum number of characters is 255.

Argument\_help1, argument\_help2    are 1 to 21 text strings that describes you custom function's arguments when the function is selected in the Paste Function dialog box.

Example      
Syntax 1

In Microsoft Excel for Windows, the following macro formula registers the GetTickCount function from Microsoft Windows. This function returns the number of milliseconds that have elapsed since Microsoft Windows was started.

REGISTER("User", "GetTickCount", "J")

Assuming that the REGISTER function is in cell A5, after your macro registers GetTickCount, you can use the CALL function to return the number of milliseconds that have elapsed:

CALL(A5)

Example      
Syntax 1 with optional function\_text

You can use the following macro formula to register the GetTickCount function from Microsoft Windows and assign the custom name GetTicks to it. To do this, include "GetTicks" as the optional function\_text argument to the REGISTER function.

REGISTER("User", "GetTickCount", "J", "GetTicks", , 1, 9)

After the function is registered, the custom name GetTicks will appear in the Information function category (category = 9) in the Paste Function dialog box.

You can call the function from the same macro sheet on which it was registered using the following formula:

GetTicks()

You can call the function from another sheet or macro sheet by including the name of the original macro sheet in the formula. For example, assuming the macro sheet on which GetTicks was registered is named MACRO1.XLS, the following formula calls the function from another sheet:

MACRO1.XLS!GetTicks()

**Tip**You can use functions in a DLL or code resource directly on a sheet without first registering them from a macro sheet. Use syntax 2a or 2b of the CALL function. For more information, see CALL.

**Related Function**

UNREGISTER   Removes a registered code resource from memory

Return to [top](#Q)

# REGRESS

Performs multiple linear regression analysis.

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

**Syntax**

**REGRESS**(**inpyrng, inpxrng**, constant, labels, confid, soutrng, residuals, sresiduals, rplots, lplots, routrng, nplots, poutrng)

**REGRESS**?(inpyrng, inpxrng, constant, labels, confid, soutrng, residuals, sresiduals, rplots, lplots, routrng, nplots, poutrng)

Inpyrng    is the input range for the y-values (dependent variable).

Inpxrng    is the input range for the x-values (independent variable).

Constant    is a logical value. If constant is TRUE, the y-intercept is assumed to be zero (the regression line passes through the origin). If constant is FALSE or omitted, the y-intercept is assumed to be a non-zero number.

Labels    is a logical value.

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

Confid    is an additional confidence level to apply to the regression. If omitted, confid is 95%.

Soutrng    is the first cell (the upper-left cell) in the output table or the name, as text, of the new sheet to contain the summary output table. If FALSE, blank, or omitted, places the summary output table in a new workbook. Microsoft Excel version 5.0 uses a single output table for REGRESS; Microsoft Excel version 4.0 used three different output tables for summary, residual, and probability data.

Residuals    is a logical value. If residuals is TRUE, REGRESS includes residuals in the output table. If residuals is FALSE or omitted, residuals are not included.

Sresiduals    is a logical value. If sresiduals is TRUE, REGRESS includes standardized residuals in the output table. If sresiduals is FALSE or omitted, standardized residuals are not included.

Rplots    is a logical value. If rplots is TRUE, REGRESS generates separate charts for each x versus the residual. If rplots is FALSE or omitted, separate charts are not generated.

Lplots    is a logical value. If lplots is TRUE, REGRESS generates a chart showing the regression line fitted to the observed values. If lplots is FALSE or omitted, the chart is not generated.

Routrng    is the first cell (the upper-left cell) in the residuals output table or the name, as text, of the new sheet to contain the residuals output table. If FALSE, blank, or omitted, places the residuals output table in a new worksheet. This argument is for compatibility with Microsoft Excel version 4.0 only and is ignored in later versions.

Nplots    is a logical value. If nplots is TRUE, REGRESS generates a chart of normal probabilities. If nplots is FALSE or omitted, the chart is not generated.

Poutrng    is the first cell (the upper-left cell) in the probability data output table or the name, as text, of the new sheet to contain the probability output table. If FALSE, blank, or omitted, places the probability output table in a new worksheet. This argument is for compatibility with Microsoft Excel version 4.0 only and is ignored in later versions.

Return to [top](#Q)

# RELREF

Returns the reference of a cell or cells relative to the upper-left cell of rel\_to\_ref. The reference is given as an R1C1-style relative reference in the form of text, such as "R[1]C[1]".

**Syntax**

**RELREF**(**reference, rel\_to\_ref**)

Reference    is the cell or cells to which you want to create a relative reference.

Rel\_to\_ref    is the cell from which you want to create the relative reference.

**Tip**   If you know the absolute reference of a cell that you want to include in a formula, but your formula requires a relative reference, use RELREF to generate the relative reference. This is especially useful with the FORMULA function, since its formula\_text argument requires R1C1-style references, and RELREF returns relative R1C1-style references. You can also use the FORMULA.CONVERT function to convert absolute references to relative references.

**Examples**

RELREF($A$1, $C$3) equals "R[-2]C[-2]"

RELREF($A$1:$E$5, $C$3:$G$7) equals "R[-2]C[-2]:R[2]C[2]"

RELREF($A$1:$E$5, $C$3) equals "R[-2]C[-2]:R[2]C[2]"

**Related Functions**

ABSREF   Returns the absolute reference of a range of cells to another range

DEREF   Returns the value of the cells in the reference

FORMULA   Enters values into a cell or range or onto a chart

FORMULA.CONVERT   Changes the reference style and type

Return to [top](#Q)

# REMOVE.LIST.ITEM

Removes an item in a list box or drop-down box.

**Syntax**

**REMOVE.LIST.ITEM**(**index\_num**, count\_num)

Index\_num    specifies the index of the item to remove, from 1 to the number of items in the list. Specify zero to remove all items in the list.

Count\_num   Specifies the number of items to delete starting from index\_num. If omitted, only one item is removed.

**Remarks**

If count\_num + index\_num is greater than the number of items in the list, all items starting with index\_num to the end of the list are removed.

**Examples**

REMOVE.LIST.ITEM(3,2) removes two items starting with the third item

REMOVE.LIST.ITEM(3) removes only the third item

**Related Function**

LISTBOX.PROPERTIES   Sets the properties of a list box and drop-down controls on worksheet and dialog sheets

Return to [top](#Q)

# REMOVE.PAGE.BREAK

Equivalent to clicking the Remove Page Break command on the Insert menu. Removes manual page breaks that you set with the SET.PAGE.BREAK function or the Page Break command on the Insert menu. If the active cell is not below or to the right of a manual page break, REMOVE.PAGE.BREAK takes no action. If the entire sheet is selected, REMOVE.PAGE.BREAK removes all manual page breaks. REMOVE.PAGE.BREAK does not remove automatic page breaks.

**Syntax**

**REMOVE.PAGE.BREAK**( )

**Related Function**

SET.PAGE.BREAK   Sets manual page breaks

Return to [top](#Q)

# RENAME.COMMAND

Changes the name of a built-in or custom menu command or the name of a menu. Use RENAME.COMMAND to change the name of a command on a menu, for example, when you create two custom commands that toggle on the menu. Examples of two built-in commands that toggle are the Page Break and Remove Page Break commands on the Insert menu.

**Syntax**

**RENAME.COMMAND**(**bar\_num, menu, command, name\_text**, position)

Bar\_num    can be either the number of one of the Microsoft Excel built-in menu bars or the number returned by a previously run ADD.BAR function. See ADD.COMMAND for a list of ID numbers for built-in menu bars.

Menu    can be either the name of a menu as text or the number of a menu. Menus are numbered starting with 1 from the left of the screen.

Command    can be either the name of the command as text or the number of the command to be renamed (the first command on a menu is command 1). If command is 0, RENAME.COMMAND renames the menu instead of the command. Because other macros can change the position of custom menu commands, you should use the name of the command rather than a number whenever possible.

If the specified menu bar, menu, or command does not exist, RENAME.COMMAND returns the #VALUE! error value and interrupts the macro.

Name\_text    is the new name for the command.

Position    is the name of the command on a submenu that you want to rename. If you use position, you must use command as the name of the submenu.

**Tip**   To specify an access key for the new name, precede the character you want to use with an ampersand (&). The access key is indicated by an underline under one letter of a menu or command name. In Microsoft Excel for the Macintosh, you can use the General tab in the Options dialog box to turn command underlining on or off. To see the Options dialog box, click Options on the Tools menu.

**Example**

To rename the Save All command as Global Save, and to make the letter "G" in Global Save an access key, use the following macro formula:

RENAME.COMMAND(10, "File", "Save All", "&Global Save")

**Related Functions**

ADD.BAR   Adds a menu bar

ADD.COMMAND   Adds a command to a menu

CHECK.COMMAND   Adds or deletes a check mark to or from a command

DELETE.COMMAND   Deletes a command from a menu

ENABLE.COMMAND   Enables or disables a menu or custom command

Return to [top](#Q)

# RENAME.OBJECT

Renames the selected object or group. This is useful for giving objects names more relevant to their usage. This is also useful if it is uncertain how the object may have been named.

**Syntax**

**RENAME.OBJECT**(new\_name)

New\_name    is the new name to be given to the selected object.

**Related Functions**

GET.OBJECT   Returns information about a specified object

INSERT.OBJECT   Equivalent to clicking the Object command on the Insert menu

SELECT Syntax 2   Selects objects on worksheets

Return to [top](#Q)

# REPLACE.FONT

Replaces one of the four built-in fonts in Microsoft Excel for Windows version 2.1 or earlier with a new font and style. This function is included only for macro compatibility. To change the font of the selected cell or range as part of a macro, use the FONT.PROPERTIES function.

**Syntax**

**REPLACE.FONT**(font\_num, name\_text, size\_num, bold, italic, underline, strike, color, outline, shadow)

**Related Function**

FONT.PROPERTIES   Sets various font attributes

Return to [top](#Q)

# REPORT.DEFINE

Equivalent to clicking the Report Manager command on the View menu and then clicking the Add option in the Report Manager dialog box. Creates or replaces a report definition. If this function is not available, you must install the Report Manager add-in.

**Syntax**

**REPORT.DEFINE**(**report\_name, sections\_array**, pages\_logical)

Report\_name    specifies the name of the report. If the workbook already contains a report with report\_name, the new report replaces the existing one.

Sections\_array    is an array that contains one or more rows of view, scenario, and sheet name that define the report. The sheet name is the sheet on which the view and scenario are defined. If the sheet name is not specified, the current sheet is used when REPORT.DEFINE is run.

Pages\_logical    is a logical value that, if TRUE or omitted, specifies continuous page numbers for multiple sections or, if FALSE, resets page numbers to 1 for each new section.

**Remarks**

* REPORT.DEFINE returns the #VALUE error value if report\_name is invalid or if the workbook is protected.
* If there are no reports defined, this function will bring up the Add Report dialog box.

**Related Functions**

REPORT.DELETE   Removes a report from the active workbook

REPORT.PRINT   Prints a report

REPORT.GET   Returns information about reports defined for the active workbook

Return to [top](#Q)

# REPORT.DELETE

Equivalent to clicking the Report Manager command on the View menu and then selecting a report in the Report Manager dialog box and clicking the Delete button. Removes a report definition from the active workbook.

If this function is not available, you must install the Report Manager add-in.

**Syntax**

**REPORT.DELETE**(**report\_name**)

Report\_name    specifies the name of the report to be removed. Report\_name can be any text that does not contain quotation marks.

**Remarks**

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

**Related Functions**

REPORT.DEFINE   Creates a report

REPORT.PRINT   Prints a report

REPORT.GET   Returns information about reports defined for the active workbook

Return to [top](#Q)

# REPORT.GET

Returns information about reports defined for the active workbook. Use REPORT.GET to return information you can use in other macro commands that manipulate reports.

If this function is not available, you must install the Report Manager add-in.

**Syntax**

**REPORT.GET**(**type\_num**, report\_name)

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

|  |  |
| --- | --- |
| **Type\_num** | **Returns** |
| 1 | An array of reports from all sheets in the active workbook or the #N/A error value if none are specified |
| 2 | An array of views, scenarios, and sheet names for the specified report in the active workbook. REPORT.GET returns the #N/A error value if the scenario check box is not selected. Returns the #VALUE! error value if name is invalid or the workbook is protected. |
| 3 | If continuous page numbers are used, returns TRUE. If page numbers start at 1 for each section, returns FALSE. Returns the #VALUE! error value if report\_name is invalid or the workbook is protected. |

Report\_name    specifies the name of a report in the active workbook.

**Remarks**

Report\_name is required if type\_num is 2 or 3.

**Example**

The following macro formula returns an array of reports from the active workbook.

REPORT.GET(1)

**Related Functions**

REPORT.DEFINE   Creates a report

REPORT.DELETE   Removes a report from the active workbook

REPORT.PRINT   Prints a report

Return to [top](#Q)

# REPORT.PRINT

Equivalent to clicking the Print button in the Report Manager dialog box. Prints a report.

If this function is not available, you must install the Report Manager add-in.

**Syntax**

**REPORT.PRINT**(**report\_name**, copies\_num, show\_print\_dlg\_logical)

**REPORT.PRINT**?(report\_name, copies\_num)

Report\_name    specifies the name of a report in the active workbook.

Copies\_num    is the number of copies you want to print. If omitted, the default is 1.

Show\_print\_dlg\_logical    is a logical value that, if TRUE, displays a dialog box asking how many copies to print, or, if FALSE or omitted, prints the report immediately using existing print settings.

**Remarks**

REPORT.PRINT returns the #VALUE! error value if report\_name is invalid or if the workbook is protected.

**Related Functions**

REPORT.DEFINE   Creates a report

REPORT.DELETE   Removes a report from the active workbook

Return to [top](#Q)

# REQUEST

Requests an array of a specific type of information from an application with which you have a dynamic data exchange (DDE) link. Use REQUEST with other Microsoft Excel DDE functions to move information from another application into Microsoft Excel.

**Syntax**

**REQUEST**(**channel\_num, item\_text**)

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

Channel\_num    is a number returned by a previously run INITIATE function. Channel\_num refers to a channel through which Microsoft Excel communicates with another program.

Item\_text    is a code indicating the type of information you want to request from another application. The form of item\_text depends on the application connected to channel\_num.

REQUEST returns the data as an array. For example, suppose the remote data to be returned came from a sheet that looked like the following illustration.



REQUEST would return that data as the following array:

{1, 2, 3;4, 5, 6}

If REQUEST is not successful, it returns the following error values.

|  |  |
| --- | --- |
| **Value returned** | **Situation** |
| #VALUE! | Channel\_num is not a valid channel number. |
| #N/A | The application you are accessing is busy doing something else. |
| #DIV/0! | The application you are accessing does not respond after a certain length of time, or you have pressed ESC or COMMAND+PERIOD to cancel. |
| #REF! | The request is refused. |

**Tip**Use the ERROR.TYPE function to distinguish between the different error values.

**Example**

Suppose you had opened a DDE channel to Microsoft Word for Windows. WChan contains the number of the open channel. In Microsoft Excel for Windows, the following function returns the text specified by the bookmark named BMK1.

=REQUEST(WChan, "BMK1")

**Related Functions**

EXECUTE   Carries out a command in another application

INITIATE   Opens a channel to another application

POKE   Sends data to another application

SEND.KEYS   Sends a key sequence to another application

TERMINATE   Closes a dynamic data exchange (DDE) channel previously opened with the INITIATE function

Return to [top](#Q)

# RESET.TOOL

Resets a button to its original button face.

**Syntax**

**RESET.TOOL**(**bar\_id, position**)

Bar\_id    is the number or name of the toolbar containing the button you want to reset. For detailed information about bar\_id, see ADD.TOOL.

Position    specifies the position of the button within the toolbar. Position starts with 1 at the left side (if horizontal) or at the top (if vertical).

**Related Functions**

ADD.TOOL   Adds one or more buttons to a toolbar

DELETE.TOOL   Deletes a button from a toolbar

RESET.TOOLBAR   Resets a button to its original button face

Return to [top](#Q)

# RESET.TOOLBAR

Resets built-in toolbars to the default Microsoft Excel set.

**Syntax**

**RESET.TOOLBAR**(bar\_id)

Bar\_id    specifies the number or name of the toolbar that you want to reset. For detailed information about bar\_id, see ADD.TOOL.

**Remarks**

If RESET.TOOLBAR successfully resets the toolbar, it returns TRUE. If you try to reset a custom toolbar, RESET.TOOLBAR returns #VALUE! and takes no other action.

**Related Functions**

ADD.TOOL   Adds one or more tools to a toolbar

DELETE.TOOLBAR   Deletes custom toolbars

Return to [top](#Q)

# RESTART

Removes a number of RETURN statements from the stack. When one macro calls another, the RETURN statement at the end of the second macro returns control to the calling macro. You can use the RESTART function to determine which macro regains control.

**Syntax**

**RESTART**(level\_num)

Level\_num    is a number specifying the number of previous RETURN statements you want to be ignored. If level\_num is omitted, the next RETURN statement will halt macro execution.

For example, if the currently running macro has two "ancestors" (the current macro was called by one macro that, in turn, was called by another macro), using RESTART(1) in the third macro returns control to the first calling macro when the RETURN statement is encountered. The RESTART(1) formula removes one level of RETURN statements from Microsoft Excel's memory so that the second macro is skipped.

**Remarks**

RESTART is particularly useful if you frequently use macros to call other macros that in turn call other macros. Use RESTART in combination with IF statements to prevent macro execution from returning to macros that called, either directly or indirectly, the currently running macro.

**Related Functions**

HALT   Stops all macros from running

RETURN   Ends the currently running macro

Return to [top](#Q)

# RESULT

Specifies the type of data a macro or custom function returns. Use RESULT to make sure your macros, custom functions, or subroutines return values of the correct data type.

**Syntax**

**RESULT**(type\_num)

Type\_num    is a number specifying the data type.

|  |  |
| --- | --- |
| **Type\_num** | **Type of returned data** |
| 1 | Number |
| 2 | Text |
| 4 | Logical |
| 8 | Reference |
| 16 | Error |
| 64 | Array |

* Type\_num can be the sum of the numbers in the preceding table to allow for more than one possible result type. For example, if type\_num is 12, which equals 4 + 8, the result can be a logical or a reference value.
* If you omit type\_num, it is assumed to be 7. Since 7 equals 1 + 2 + 4, the value returned can be a number (1), text (2), or logical value (4).

**Examples**

The following function specifies that a custom function's return value can be a number or a logical value (4+1=5):

RESULT(5)

**Related Functions**

ARGUMENT   Passes an argument to a macro

RETURN   Ends the currently running macro

Return to [top](#Q)

# RESUME

Equivalent to choosing the Resume button on the toolbar. Resumes a paused macro. Returns TRUE if successful or the #VALUE! error value if no macro is paused. A macro can be paused by using the PAUSE function or choosing Pause from the Single Step dialog box, which appears when you choose the Step Into button from the Macro dialog box.

**Syntax**

**RESUME**(type\_num)

Type\_num    is a number from 1 to 4 specifying how to resume.

|  |  |
| --- | --- |
| **Type\_num** | **How Microsoft Excel resumes** |
| 1 or omitted | If paused by a PAUSE function, continues running the macro. If paused from the Single Step dialog box, returns to that dialog box. |
| 2 | Halts the paused macro |
| 3 | Continues running the macro |
| 4 | Opens the Single Step dialog box |

**Tip**   You can use Microsoft Excel's ON functions to resume based on an event. For an example, see ENTER.DATA.

**Remarks**

* If one macro runs a second macro that pauses, and you need to halt only the paused macro, use RESUME(2) instead of HALT. HALT halts all macros and prevents resuming or returning to any macro.
* If the macro was paused from the Single Step dialog box, RESUME returns to the Single Step dialog box.

**Related Functions**

HALT   Stops all macros from running

PAUSE   Pauses a macro

RETURN   Ends the currently running macro

Return to [top](#Q)

# RETURN

Ends the currently running macro. If the currently running macro is a subroutine macro that was called by another macro, control is returned to the calling macro. If the currently running macro is a custom function, control is returned to the formula that called the custom function. If the currently running macro is a command macro started by the user with the Run button in the Macro dialog box or a shortcut key or by clicking an object, control is returned to the user.

**Syntax**

**RETURN**(value)

Value    specifies what to return.

* If the macro is a custom function or a subroutine, value specifies what value to return. However, not all subroutines return values; the last line in macros that do not return values is =RETURN().
* If the macro is a command macro run by the user, value should be omitted.

**Remarks**

RETURN signals the end of a macro. Every macro must end with a RETURN or HALT function, but not every macro returns values.

**Example**

The following function returns the sum of the range B1:B10:

RETURN(SUM(B1:B10))

**Related Functions**

BREAK   Interrupts a FOR-NEXT, FOR.CELL-NEXT, or WHILE-NEXT loop

HALT   Stops all macros from running

RESULT   Specifies the data type a custom function returns

Return to [top](#Q)

# ROUTE.DOCUMENT

Routes the workbook using the defined routing slip information.

**Syntax**

**ROUTE.DOCUMENT**()

**Remarks**

If there is no routing slip, returns #N/A. If an error occurs or routing is not enabled for the system, returns #VALUE!.

**Related Functions**

SEND.MAIL   Sends the active workbook using email

ROUTING.SLIP   Adds or Edits the routing slip attached to the current workbook

Return to [top](#Q)

# ROUTING.SLIP

Equivalent to clicking the Add Routing Slip command on the File menu. Adds or Edits the routing slip attached to the current workbook.

**Syntax**

**ROUTING.SLIP**(recipients,subject, message, route\_num, return\_logical, status\_logical)

**ROUTING.SLIP**?(recipients,subject, message, route\_num, return\_logical, status\_logical)

Recipients    is the name of the person to whom you want to send the mail. The name should be given as text.

* To specify more than one name, give the list of names as an array. For example, ROUTING.SLIP({"John", "Paul", "George", "Ringo"}) would send the active workbook to the four names in the array. You can also refer to a range on a sheet or macro sheet that contains a list of names to whom you want the mail to be sent.
* Specifying recipients while a routing is in progress only modifies the non-grayed recipients (that is, those recipients who have not received the message yet). Recipients who have already received, reviewed and forwarded the routed workbook cannot be modified.

Subject    is a text string containing the subject text used for the mail messages used to route the workbook. If omitted, the default subject line is "Routing: name", where name is the file name or title as displayed in the Summary Info dialog box, if available.

Message    is a text string containing the body text used for the mail messages used to route the workbook.

Route\_num    is a number indicating the type of routing method.

|  |  |
| --- | --- |
| **Route\_num** | **Method** |
| 1 or omitted | One after another routing |
| 2 | All at once routing |

Return\_logical    is a logical value which, if TRUE or omitted, indicates that the routing should be returned to the originator when the routing is complete. If FALSE, the routing will end with the last recipient in the To list box in the Routing Slip Dialog box.

Status\_logical    is a logical value corresponding to the Track Status check box in the Routing Slip dialog box. If TRUE or omitted, status tracking messages for the routing are sent. FALSE means that no status tracking is performed.

**Remarks**

* If this function is used on a workbook that is already being routed, the route\_num, status\_logical and return\_logical arguments are ignored (they cannot be changed).
* When arguments are omitted and a routing slip already exists, the omitted arguments are replaced by the current values of the routing slip.

**Related Functions**

ROUTE.DOCUMENT   Routes the workbook using the defined routing slip information

SEND.MAIL   Sends the active workbook using email

Return to [top](#Q)

# ROW.HEIGHT

Equivalent to choosing the Height command on the Row submenu of the Format menu. Changes the height of the rows in a reference.

**Syntax**

**ROW.HEIGHT**(height\_num, reference, standard\_height, type\_num)

**ROW.HEIGHT**?(height\_num, reference, standard\_height, type\_num)

Height\_num    specifies how high you want the rows to be in points. If standard\_height is TRUE, height\_num is ignored.

Reference    specifies the rows for which you want to change the height.

* If reference is omitted, the reference is assumed to be the current selection.
* If reference is specified, it must be either an external reference to the active worksheet, such as !$2:$4 or !Database, or an R1C1-style reference in the form of text or a name, such as "R1:R3", "R[-4]:R[-2]", or Database.
* If reference is a relative R1C1-style reference in the form of text, it is assumed to be relative to the active cell.

Standard\_height    is a logical value that sets the row height as determined by the font in each row.

* If standard\_height is TRUE, Microsoft Excel sets the row height to a standard height that may vary from row to row depending on the fonts used in each row, ignoring height\_num.
* If standard\_height is FALSE or omitted, Microsoft Excel sets the row height according to height\_num.

Type\_num    is a number from 1 to 3 corresponding to selecting the Hide, Unhide, or AutoFit commands from the Row submenu.

|  |  |
| --- | --- |
| **Type\_num** | **Action taken** |
| 1 | Hides the row selection by setting the row height to 0 |
| 2 | Unhides the row selection by setting the row height to the value set before the selection was hidden |
| 3 | Sets the row selection to an AutoFit height, which varies from row to row depending on how large the font is in any cell in each row or on how many lines of text are wrapped |

**Remarks**

* If any of the argument settings conflict, such as when standard\_height is TRUE and type\_num is 3, Microsoft Excel uses the type\_num argument and ignores any arguments that conflict with type\_num.
* If you are recording a macro while using a mouse, and you change row heights by dragging the row border, Microsoft Excel records the reference of the rows using R1C1-style references in the form of text. If Uses Relative References is selected, Microsoft Excel uses R1C1-style relative references. If Uses Relative References is not selected, Microsoft Excel uses R1C1-style absolute references.

**Related Function**

COLUMN.WIDTH   Sets the widths of the specified columns

Return to [top](#Q)

# RUN

Equivalent to choosing the Run button in the Macro dialog box, which appears when you choose the Macros command on the Macro submenu of the Tools menu. Runs a macro.

**Syntax**

**RUN**(reference, step)

**RUN**?(reference, step)

Reference    is a reference to the macro you want to run or a number from 1 to 4 specifying an Auto macro to run.

|  |  |
| --- | --- |
| **If reference is** | **Specifies** |
| 1 | All Auto\_Open macros on the active workbook |
| 2 | All Auto\_Close macros |
| 3 | All Auto\_Activate macros |
| 4 | All Auto\_Deactivate macros |

* If reference is a range of cells, RUN begins with the macro function in the upper-left cell of reference.
* If the macro sheet containing the macro is not in the active workbook, reference can be an external reference to the name of the macro, such as RUN([BOOK1]Macro!Months) or an external R1C1-style reference to the location of the macro, such as RUN("[Book1]Macro!R2C3"). The reference must be in text form.
* If reference is omitted, the macro function in the active cell is carried out, and macro execution continues down that column.

Step    is a logical value specifying that the macro is to be run in single-step mode. If step is TRUE, Microsoft Excel runs the macro in single-step mode; if FALSE or omitted, Microsoft Excel runs the macro normally.

**Remarks**

* RUN is recorded when you choose the Run button the Macro dialog box while recording a macro. The reference you enter in the Run dialog box is recorded as text, with A1-style references converted to R1C1-style references.
* To run a macro from a macro sheet, you could alternatively enter the name of the macro as a formula, followed by a set of parentheses. For example, enter =[Book1]Macro!Months() instead of =RUN([Book1]Macro!Months).

**Related Function**

GOTO   Directs macro execution to another cell

Return to [top](#Q)

# SAMPLE

Samples data.

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

**Syntax**

**SAMPLE**(**inprng**, outrng, **method, rate**, labels)

**SAMPLE**?(inprng, outrng, method, rate, labels)

Inprng    is the input range.

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

Method    is a text character that indicates the type of sampling.

* If method is "P", then periodic sampling is used. The input range is sampled every nth cell, where n = rate.
* If method is "R", then random sampling is used. The output column will contain rate samples.

Rate    is the sampling rate, if method is "P" (periodic sampling). Rate is the number of samples to take if method is "R" (random sampling).

Labels    is a logical value.

* If labels is TRUE, then the first row or column of inprng contains labels.
* If labels is FALSE or omitted, all cells in inprng are considered data. Microsoft Excel generates appropriate data labels for the output table.

Return to [top](#Q)

# SAVE

Equivalent to choosing the Save command from the File menu. Saves the active workbook.

**Syntax**

**SAVE**( )

**Remarks**

Use the SAVE.AS function instead of SAVE when you want to change the filename or file type, specify a password, create a backup file, or save a file to a different directory or folder.

**Related Functions**

SAVE.AS   Saves a workbook and allows you to specify the name, file type, password, backup file, and location of the workbook

SAVE.WORKBOOK   Saves a workbook

Return to [top](#Q)

# SAVE.AS

Equivalent to clicking the Save As command on the File menu. Use SAVE.AS to specify a new filename, file type, protection password, or write-reservation password, or to create a backup file.

**Syntax**

**SAVE.AS**(document\_text, type\_num, prot\_pwd, backup, write\_res\_pwd, read\_only\_rec)

**SAVE.AS**?(document\_text, type\_num, prot\_pwd, backup, write\_res\_pwd, read\_only\_rec)

Document\_text    specifies the name of a workbook to save, such as SALES.XLS (in Microsoft Excel for Windows) or SALES (in Microsoft Excel for the Macintosh). You can include a full path in document\_text, such as C:\EXCEL\ANALYZE.XLS (in Microsoft Excel for Windows) or HARDDISK:FINANCIALS:ANALYZE (in Microsoft Excel for the Macintosh).

Type\_num    is a number specifying the file format in which to save the workbook.

|  |  |
| --- | --- |
| **Type\_num** | **File format** |
| 1 or omitted | Normal |
| 2 | SYLK |
| 3 | Text |
| 4 | WKS |
| 5 | WK1 |
| 6 | CSV |
| 7 | DBF2 |
| 8 | DBF3 |
| 9 | DIF |
| 10 | Reserved |
| 11 | DBF4 |
| 12 | Reserved |
| 13 | Reserved |
| 14 | Reserved |
| 15 | WK3 |
| 16 | Microsoft Excel 2.x |
| 17 | Template |
| 18 | Add-in macro (For compatibility only. In Microsoft Excel version 5.0, this saves as normal.) |
| 19 | Text (Macintosh) |
| 20 | Text (Windows) |
| 21 | Text (MS-DOS) |
| 22 | CSV (Macintosh) |
| 23 | CSV (Windows) |
| 24 | CSV (MS-DOS) |
| 25 | International macro |
| 26 | International add-in macro |
| 27 | Reserved |
| 28 | Reserved |
| 29 | Microsoft Excel 3.0 |
| 30 | WK1 / FMT |
| 31 | WK1 / Allways |
| 32 | WK3 / FM3 |
| 33 | Microsoft Excel 4.0 |
| 34 | WQ1 |
| 35 | Microsoft Excel 4.0 workbook |
| 36 | Formatted text (space delimited) |

The following table shows which values of type\_num apply to the six Microsoft Excel document types.

|  |  |
| --- | --- |
| **Document Type** | **Type\_num** |
| Worksheet | All except 10, 12-14, 18, 25-28, 36 |
| Chart sheet | All except 10, 12-14, 18, 25-28 |
| Visual Basic module | 1, 3, 17 |
| Dialog | 1, 17 |
| Macro sheet | 1-3, 6, 9, 16-29, 33 |
| Workbook | 1, 15, 35 |

Prot\_pwd    corresponds to the Protection Password box in the Save Options dialog box in Microsoft Excel 95 or earlier versions, or the Password To Open box in Microsoft Excel 97 or later.

* Prot\_pwd is a password given as text or as a reference to a cell containing text. Prot\_pwd should be no more than 15 characters.
* If a file is saved with a password, the password must be supplied for the file to be opened.

Backup    is a logical value corresponding to the Always Create Backup check box in the Save Options dialog box and specifies whether to make a backup workbook. If backup is TRUE, Microsoft Excel creates a backup file; if FALSE, no backup file is created; if omitted, the status is unchanged.

Write\_res\_pwd    corresponds to the Write Reservation Password box in the Save Options dialog box in Microsoft Excel 95 or earlier versions, or the Password To Modify box in Microsoft Excel 97 or later. Allows the user to write to a file. If a file is saved with a password and the password is not supplied when the file is opened, the file is opened read-only.

Read\_only\_rec    is a logical value corresponding to the Read-Only Recommended check box in the Save Options dialog box.

* If read\_only\_rec is TRUE, Microsoft Excel saves the workbook as a read-only recommended workbook; if FALSE, Microsoft Excel saves the workbook normally; if omitted, Microsoft Excel uses the current settings.
* When you open a workbook that was saved as read-only recommended, Microsoft Excel displays a message recommending that you open the workbook as read-only.

**Related Functions**

CLOSE   Closes the active window

GET.DOCUMENT   Returns information about a workbook

SAVE   Saves the active workbook

SAVE.WORKBOOK   Saves a workbook

Return to [top](#Q)

# SAVE.COPY.AS

Saves a copy of the current workbook using a different name but all the current workbook settings, such as passwords and file protection. Does not affect the current workbook. Use this command if you need a temporary copy of the current workbook; for example, to include in an electronic mail message.

**Syntax**

**SAVE.COPY.AS**(document\_text)

Document\_text    is the name you want to give the copy of the workbook.

**Example**

Suppose that you are creating a macro that makes changes to a file called BUDGET95.XLS. Use the following function to save a copy of this file called TEMP.XLS without affecting BUDGET95.XLS:

SAVE.COPY.AS("temp.xls")

Return to [top](#Q)

# SAVE.DIALOG

Displays the standard Microsoft Excel File Save As dialog box and gets a file name from the user. This function returns the path and file name of the file that has been saved. Use SAVE.AS to automatically save a file with a particular format and other properties.

**Syntax**

**SAVE.DIALOG**(init\_filename, title, button\_text, file\_filter, filter\_index)

Init\_filename   Specifies the suggested filename for saving. If omitted, the active workbook's name is used, as returned by the GET.DOCUMENT(1) function.

Title    Specifies the default window title on Microsoft Excel for Windows. For Microsoft Excel for the Macintosh, title specifies the prompt string. If omitted, "File Save As" will be used for Microsoft Excel for Windows, and "Save As:" For Microsoft Excel for the Macintosh.

Button\_text    is the text used for the save button in the dialog. If omitted, "Save" will be used as the default. This argument is ignored on the Microsoft Excel for Windows.

File\_filter    is the file filtering criteria to use, as text. For Microsoft Excel for Windows, file\_filter consists of two parts, a descriptive phrase denoting the file type followed by a comma and then the MS-DOS wildcard file filter specification, as in "Text Files (\*.TXT), \*.TXT, Add-in Files (\*.XLA), \*.XLA". Groups of filter specifications are also separated by commas. Each separate pair is listed in the file type drop-down list box. File\_filter can include an asterisk (\*) to represent any sequence of characters and a question mark (?) to represent any single character. For Microsoft Excel for the Macintosh, file\_filter consists of file type codes separated by commas, as in "TEXT,XLA,XLS4". Spaces are significant and should not be inserted before or after the comma separators unless they are part of the file type code.

Filter\_index    specifies the index number of the default file filtering criteria from 1 to the number of filters specified in file\_filter. If omitted or greater than the number of filters present, 1 will be used as the starting index number. The argument is ignored on Microsoft Excel for the Macintosh.

**Remarks**

* To use multiple MS-DOS wildcard expressions within file\_filter for a single file filter type, separate the wildcard expressions with semicolons, as in "VB Files (\*.bas; \*.txt), \*.bas;\*.txt".
* If file\_filter is omitted, "ALL Files (\*.\*), \*.\*" will be used as the default in Microsoft Excel for Windows. The default for Microsoft Excel for the Macintosh is all file types.
* If the user cancels the dialog box, FALSE is returned.

**Examples**

SAVE.DIALOG("TRAVEL.XLS","How do you want to save this file?",,  
"Text Files (\*.TXT), \*.TXT, Add-in Files (\*.XLA), \*.XLA, ALL FILES (\*.\*), \*.\*") opens a File Save As dialog box titled "How do you want to save this file?", with "TRAVEL.XLS" as the suggested file name, and with three file filter criteria in the drop-down list box.

**Related Function**

OPEN.DIALOG   Displays the standard Microsoft Excel File Open dialog box with the specified file filters

Return to [top](#Q)

# SAVE.TOOLBAR

Saves one or more toolbar definitions to a specified file.

**Syntax**

**SAVE.TOOLBAR**(bar\_id, filename)

Bar\_id    is either the name or number of a toolbar whose definition you want to save or an array of toolbar names or numbers whose definitions you want to save. Use an array to save several toolbar definitions at the same time. For detailed information about bar\_id, see ADD.TOOL. If bar\_id is omitted, all toolbar definitions are saved.

Filename    is text specifying the name of the destination file. If filename does not exist, Microsoft Excel creates a new file. If filename exists, Microsoft Excel overwrites the file. If filename is omitted, Microsoft Excel saves the toolbar or toolbars in Username8.xlb, where "username" is your Windows or network logon name. With Microsoft Windows, Username8.xlb is stored in the directory where Windows is installed; with Apple Macintosh, EXCEL TOOLBARS is stored in the System:Preferences folder

**Examples**

In Microsoft Excel for Windows, the following macro formula saves Toolbar6 as \EXCDT\TOOLFILE.XLB.

SAVE.TOOLBAR("Toolbar6", "\EXCDT\TOOLFILE.XLB")

In Microsoft Excel for the Macintosh, the following macro formula saves Toolbar6 as TOOLFILE.

SAVE.TOOLBAR("Toolbar6", "TOOLFILE")

**Related Functions**

ADD.TOOL   Adds one or more tools to a toolbar

ADD.TOOLBAR   Creates a new toolbar with the specified tools

OPEN   Opens a workbook

Return to [top](#Q)

# SAVE.WORKBOOK

Equivalent to clicking the Save Workbook command on the File menu in Microsoft Excel version 4.0. Provided for compatibility with Microsoft Excel version 4.0. Saves the workbook to which the active sheet belongs. To save Microsoft Excel version 5.0 or later workbooks, use SAVE.AS.

**Syntax**

**SAVE.WORKBOOK**(document\_text, type\_num, prot\_pwd, backup, write\_res\_pwd, read\_only\_rec)

**SAVE.WORKBOOK**?(document\_text, type\_num, prot\_pwd, backup, write\_res\_pwd, read\_only\_rec)

For a description of the arguments, see SAVE.AS.

**Related Functions**

CLOSE   Closes the active window

GET.DOCUMENT   Returns information about a workbook

SAVE   Saves the active workbook

SAVE.AS   Saves a workbook and allows you to specify the name, file type, password, backup file, and location of the workbook

Return to [top](#Q)

# SAVE.WORKSPACE

Equivalent to clicking the Save Workspace command on the File menu. Saves the currently opened workbook or workbooks as a workspace.

**Syntax**

**SAVE.WORKSPACE**(name\_text)

**SAVE.WORKSPACE**?(name\_text)

Name\_text    is the name of the workspace to save.

**Related Function**

SAVE.AS   Specifies a new filename.

Return to [top](#Q)

# SCALE

Changes the position, formatting, and scaling of axes in a chart. There are five syntax forms of this function.

Syntax 1   Changes the position, formatting, and scaling of the category axis in 2-D charts

Syntax 2   Changes the position, formatting, and scaling of the value axis in 2-D charts

Syntax 3   Changes the position, formatting, and scaling of the category axis in 3-D charts

Syntax 4   Changes the position, formatting, and scaling of the series axis in 3-D charts

Syntax 5   Changes the position, formatting, and scaling of the value axis in 3-D charts

Return to [top](#Q)

# SCALE Syntax 1

Equivalent to clicking the Selected Axis command on the Format menu when a chart's category (x) axis is selected, and then clicking the Scale tab. There are five syntax forms of this function. Syntax 1 of SCALE applies if the selected axis is a category (x) axis on a 2-D chart and the chart is not an xy (scatter) chart. Use this syntax of SCALE to change the position, formatting, and scaling of the category axis.

**Syntax 1**

**SCALE**(cross, cat\_labels, cat\_marks, between, max, reverse)

**SCALE**?(cross, cat\_labels, cat\_marks, between, max, reverse)

Arguments correspond to text boxes and check boxes in the Scale tab on the Format Axis dialog box. Arguments corresponding to check boxes are logical values. If an argument is TRUE, Microsoft Excel selects the check box; if FALSE, Microsoft Excel clears the check box.

Cross    is a number corresponding to the Value (Y) Axis Crosses At Category number text box. The default is 1. Cross is ignored if max is set to TRUE.

Cat\_labels    is a number corresponding to the Number Of Categories Between Tick Mark Labels text box. The default is 1.

Cat\_marks    is a number corresponding to the Number Of Categories Between Tick Marks text box. The default is 1.

Between    corresponds to the Value (Y) Axis Crosses Between Categories check box. This argument only applies if cat\_labels is set to a number other than 1.

Max    corresponds to the Value (Y) Axis Crosses At Maximum Category check box. If max is TRUE, it overrides any setting for cross.

Reverse    corresponds to the Categories In Reverse Order check box.

**Related Functions**

AXES   Controls whether axes on a chart are visible

GRIDLINES   Controls whether chart gridlines are visible

Syntax 2   Changes the position, formatting, and scaling of the value axis in 2-D charts

Syntax 3   Changes the position, formatting, and scaling of the category axis in 3-D charts

Syntax 4   Changes the position, formatting, and scaling of the series axis in 3-D charts

Syntax 5   Changes the position, formatting, and scaling of the value axis in 3-D charts

Return to [top](#Q)

# SCALE Syntax 2

Equivalent to clicking the Selected Axes command on the Format menu when a chart's value (y) axis is selected, and then clicking the Scale tab. There are five syntax forms of this function. Syntax 2 of SCALE applies if the selected axis is a value (y) axis on a 2-D chart, or either axis on an xy (scatter) chart. Use this syntax of SCALE to change the position, formatting, and scaling of the value axis.

**Syntax 2**

**SCALE**(min\_num, max\_num, major, minor, cross, logarithmic, reverse, max)

**SCALE**?(min\_num, max\_num, major, minor, cross, logarithmic, reverse, max)

The first five arguments correspond to the five range variables on the Scale tab. Each argument can be either the logical value TRUE or a number:

* If an argument is TRUE, Microsoft Excel selects the Auto check box.
* If an argument is a number, that number is used for the variable.

Min\_num    corresponds to the Minimum check box and is the minimum value for the value axis.

Max\_num    corresponds to the Maximum check box and is the maximum value for the value axis.

Major    corresponds to the Major Unit check box and is the major unit of measure.

Minor    corresponds to the Minor Unit check box and is the minor unit of measure.

Cross    corresponds to the Category (X) Axis Crosses At text box for the value (y) axis of a 2-D chart or the Value (Y) Axis Crosses At text box for the category (x) axis of an xy (scatter) chart.

The last three arguments are logical values corresponding to check boxes on the Scale tab . If an argument is TRUE, Microsoft Excel selects the check box; if FALSE, Microsoft Excel clears the check box.

Logarithmic    corresponds to the Logarithmic Scale check box.

Reverse    corresponds to the Values In Reverse Order check box.

Max    corresponds to the Category (X) Axis Crosses At Maximum Value check box.

**Related Functions**

Syntax 1   Changes the position, formatting, and scaling of the category axis in 2-D charts

Syntax 3   Changes the position, formatting, and scaling of the category axis in 3-D charts

Syntax 4   Changes the position, formatting, and scaling of the series axis in 3-D charts

Syntax 5   Changes the position, formatting, and scaling of the value axis in 3-D charts

Return to [top](#Q)

# SCALE Syntax 3

Equivalent to clicking the Selected Axes command on the Format menu when a chart's category (x) axis is selected, and then click the Scale tab. There are five syntax forms of this function. Syntax 3 of SCALE applies if the selected axis is a category (x) axis on a 3-D chart. Use this syntax of SCALE to change the position, formatting, and scaling of the category axis.

**Syntax 3**

**SCALE**(cat\_labels, cat\_marks, reverse, between)

**SCALE**?(cat\_labels, cat\_marks, reverse, between)

Cat\_labels    is a number corresponding to the Number Of Categories Between Tick-Mark Labels box. The default is 1. Cat\_labels can also be a logical value. If TRUE, an automatic setting will be used. If FALSE, or omitted, the number will be used.

Cat\_marks    is a number corresponding to the Number Of Categories Between Tick Marks text box. The default is 1. Cat\_marks can also be a logical value. If TRUE, an automatic setting will be used. If FALSE, or omitted, the number will be used.

Reverse    corresponds to the Categories In Reverse Order check box. If reverse is TRUE, Microsoft Excel selects the check box; if FALSE, Microsoft Excel clears the check box.

Between    corresponds to the Value (Z) Axis Crosses Between Categories check box. If between is TRUE, Microsoft Excel selects the check box and the data points appear between categories. If between is FALSE or omitted, Microsoft Excel clears the check box.

**Related Functions**

Syntax 1   Changes the position, formatting, and scaling of the category axis in 2-D charts

Syntax 2   Changes the position, formatting, and scaling of the value axis in 2-D charts

Syntax 4   Changes the position, formatting, and scaling of the series axis in 3-D charts

Syntax 5   Changes the position, formatting, and scaling of the value axis in 3-D charts

Return to [top](#Q)

# SCALE Syntax 4

Equivalent to clicking the Selected Axes command on the Format menu when a chart's value (y) axis is selected, and then clicking the Scale tab. There are five syntax forms of this function. Syntax 4 of SCALE applies if the selected axis is a series (y) axis on a 3-D chart. Use this syntax of SCALE to change the position, formatting, and scaling of the series axis.

**Syntax 4**

Series (y) axis, 3-D chart

**SCALE**(series\_labels, series\_marks, reverse)

**SCALE**?(series\_labels, series\_marks, reverse)

Series\_labels    is a number corresponding to the Number Of Series Between Tick Labels text box. The default is 1. Series\_labels can also be a logical value. If TRUE, and automatic setting will be used. If FALSE, or omitted, the number will be used.

Series\_marks    is a number corresponding to the Number Of Series Between Tick Marks text box. The default is 1. Series\_marks can also be a logical value. If TRUE, and automatic setting will be used. If FALSE, or omitted, the number will be used.

Reverse    is a logical value that corresponds to the Series In Reverse Order check box on the Scale tab. If reverse is TRUE, Microsoft Excel displays the series in reverse order; if FALSE or omitted, Microsoft Excel displays the series normally.

**Related Functions**

Syntax 1   Changes the position, formatting, and scaling of the category axis in 2-D charts

Syntax 2   Changes the position, formatting, and scaling of the value axis in 2-D charts

Syntax 3   Changes the position, formatting, and scaling of the category axis in 3-D charts

Syntax 5   Changes the position, formatting, and scaling of the value axis in 3-D charts

Return to [top](#Q)

# SCALE Syntax 5

Equivalent to clicking the Selected Axes command on the Format menu when a chart's value (z) axis is selected, and then clicking the Scale tab. There are five syntax forms of this function. Syntax 5 of SCALE applies if the selected axis is a value (z) axis on a 3-D chart. Use this syntax of SCALE to change the position, formatting, and scaling of the value axis.

**Syntax 5**

**SCALE**(min\_num, max\_num, major, minor, cross, logarithmic, reverse, min)

**SCALE**?(min\_num, max\_num, major, minor, cross, logarithmic, reverse, min)

The first five arguments correspond to the five range variables in the Format Axis dialog box, as shown in the following list. Each argument can be either the logical value TRUE or a number.

* If TRUE or omitted, the Auto check box is selected.
* If a number, that number is used.

Min\_num    corresponds to the Minimum check box and is the minimum value for the value axis.

Max\_num    corresponds to the Maximum check box and is the maximum value for the value axis.

Major    corresponds to the Major Unit check box and is the major unit of measure.

Minor    corresponds to the Minor Unit check box and is the minor unit of measure.

Cross    corresponds to the Floor (XY Plane) Crosses At check box.

The last three arguments are logical values corresponding to check boxes on the Scale tab. If an argument is TRUE, Microsoft Excel selects the check box; if FALSE, Microsoft Excel clears the check box.

Logarithmic    corresponds to the Logarithmic Scale check box.

Reverse    corresponds to the Values In Reverse Order check box.

Min    corresponds to the Floor (XY Plane) Crosses At Minimum Value check box.

**Related Functions**

Syntax 1   Changes the position, formatting, and scaling of the category axis in 2-D charts

Syntax 2   Changes the position, formatting, and scaling of the value axis in 2-D charts

Syntax 3   Changes the position, formatting, and scaling of the category axis in 3-D charts

Syntax 4   Changes the position, formatting, and scaling of the series axis in 3-D charts

Return to [top](#Q)

# SCENARIO.ADD

Equivalent to clicking the Scenarios command on the Tools menu and then clicking the Add button. Defines the specified values as a scenario. A scenario is a set of values to be used as input for a model on your worksheet.

**Syntax**

**SCENARIO.ADD**(**scen\_name**, value\_array, changing\_ref, scen\_comment, locked, hidden)

Scen\_name    is the name of the scenario you want to define.

Value\_array    is a horizontal array of values you want to use as input for the model on your worksheet.

* Any entry that would be valid for a cell in your model can be a value in value\_array.
* The values must be arranged in the same order as the model's changing cells. The changing cells are listed in the Changing Cells box in the Scenario Manager dialog box.
* If value\_array is omitted, it is assumed to contain the current values of the changing cells.

Changing\_ref    is a reference to cells you want to define as changing cells for a scenario.

* If omitted, uses the changing cells for the last scenario defined for the sheet.
* If changing\_ref contains nonadjacent references, you must separate the reference areas by commas (or other list separator). If you are using A1-style references, then you must enclose reference in an extra set of parentheses.

Scen\_comment    is text specifying a descriptive comment for the scenario defined by scen\_name.

Locked    is a logical value that corresponds to the Prevent Changes check box in the Add or Edit Scenario dialogs boxes. If TRUE or omitted , prevents users from changing values in a scenario. If FALSE, users are allowed to make changes to the scenario. The locking will not become enabled until the sheet is protected with the Protect Sheet command from the Protection submenu on the Tools menu.

Hidden    is a logical value that corresponds to the Hide check box in the Add or Edit Scenario dialog boxes. If TRUE, the scenario will be hidden from view from the users and will not appear in the Scenario Manager dialog box. If FALSE or omitted, the scenario will remain unhidden. The scenario will not become hidden until the sheet is protected with the Protect Sheet command from the Protection submenu on the Tools menu.

**Related Functions**

REPORT.DEFINE   Creates a report

SCENARIO.GET   Returns the specified information about the scenarios defined on your worksheet

Return to [top](#Q)

# SCENARIO.CELLS

Equivalent to clicking the Scenarios command on the Tools menu and then editing the Changing Cells box. Defines the changing cells for a model on your worksheet. Changing cells are the cells into which values will be entered when you display a scenario. If you have only one set of changing cells on your sheet, SCENARIO.CELLS will change the changing cells for all scenarios. If your sheet has scenarios defined with multiple sets of changing cell, this function returns an error and the macro is halted. This function is provided for compatibility with Microsoft Excel version 4.0. Use SCENARIO.EDIT with the changing\_ref argument instead of SCENARIO.CELLS if you want to change the changing cells of a scenario.

**Syntax**

**SCENARIO.CELLS**(**changing\_ref**)

**SCENARIO.CELLS**? (changing\_ref)

Changing\_ref    is a reference to the cells you want to define as changing cells for the model. If changing\_ref contains nonadjacent references, you must separate the reference areas by commas and enclose changing\_ref in an extra set of parentheses.

**Related Function**

SCENARIO.EDIT   Equivalent to clicking the Scenarios command on the Tools menu and then clicking the Edit button

Return to [top](#Q)

# SCENARIO.DELETE

Equivalent to clicking the Scenarios command on the Tools menu, clicking a scenario, and then clicking the Delete button. Deletes the specified scenario.

**Syntax**

**SCENARIO.DELETE**(**scen\_name**)

Scen\_name    is the name of the scenario you want to delete.

**Related Functions**

SCENARIO.GET   Returns the specified information about the scenarios defined on your worksheet

SCENARIO.ADD   Equivalent to clicking the Scenario Manager command on the Tools menu and then clicking the Add button

SCENARIO.EDIT   Equivalent to clicking the Scenario Manager command on the Tools menu and then clicking the Edit button

Return to [top](#Q)

# SCENARIO.EDIT

Equivalent to clicking the Scenarios command from the Tools menu and then clicking the Edit button.

**Syntax**

**SCENARIO.EDIT**(**scen\_name**, new\_scenname, value\_array, changing\_ref, scen\_comment, locked, hidden)

**SCENARIO.EDIT**?(scen\_name, new\_scenname, value\_array, changing\_ref, scen\_comment, locked, hidden)

Scen\_name    is the name of the scenario that you want to edit.

New\_scenname    is the new name you want to give to the scenario.

Value\_array    is a horizontal array of values that you want to use for the scenario.

* If value\_array is omitted but changing\_ref is specified, Scenario Manager uses the values in changing\_ref as value\_array.
* Value\_array must match the dimensions of changing\_ref for the scenario being edit.

Changing\_ref    is a reference to cells you want to define as changing cells for a scenario.

Scen\_comment    is text specifying a descriptive comment for the scenario you want to edit.

Locked    is a logical value that corresponds to the Prevent Changes check box in the Add or Edit Scenario dialogs boxes. If TRUE or omitted , prevents users from changing values in a scenario. If FALSE, users are allowed to make changes to the scenario. The locking will not become enabled until the sheet is protected with the Protect Sheet command from the Protection submenu on the Tools menu.

Hidden    is a logical value that corresponds to the Hide check box in the Add or Edit Scenario dialog boxes. If TRUE, the scenario will be hidden from view from the users. If FALSE or omitted, the scenario will remain unhidden. The scenario will not become hidden until the sheet is hidden with the Hide command from the Window menu.

**Related Functions**

SCENARIO.GET   Returns the specified information about the scenarios defined on your worksheet

SCENARIO.ADD   Equivalent to clicking the Scenario Manager command on the Tools menu and then clicking the Add button

SCENARIO.DELETE   Equivalent to clicking the Scenario Manager command on the Tools menu and then selecting a scenario and clicking the Delete button

Return to [top](#Q)

# SCENARIO.GET

Returns the specified information about the scenarios defined on your worksheet.

**Syntax**

**SCENARIO.GET**(**type\_num**, scen\_name)

Type\_num    is a number from 1 to 8 specifying the type of information you want.

|  |  |
| --- | --- |
| **Type\_num** | **Information returned** |
| 1 | A horizontal array of all scenario names in the form of text |
| 2 | A reference to the set of changing cells of scen\_name (specified in the Changing Cells box of the Scenario Manager dialog box). If scen\_name is omitted, the first scenario is used. |
| 3 | A reference to the result cells (specified in the Result Cells box in the Scenario Summary dialog box) |
| 4 | An array of scenario values for the scenario scen\_name . Each scenario is in a separate row. If scen\_name is omitted, the first scenario is used. |
| 5 | Comment, as text, for the scenario |
| 6 | Returns TRUE if the specified scenario is locked to prevent changes; FALSE, if unlocked. Scen\_name is required. |
| 7 | Returns TRUE if the specified scenario is hidden; FALSE, if visible to the user. Scen\_name is required. |
| 8 | Returns the user name of the person who last modified the scenario by either adding or editing a scenario. Scen\_name is required. |

Scen\_name    is the name of the scenario that you want information about. Ignored if type\_num equals 1 or 3.

**Remarks**

In the returned array of scenario values, the number of rows is the number of scenarios, and the number of columns is the number of changing cells.

Return to [top](#Q)

# SCENARIO.MERGE

Equivalent to choosing the Scenarios command from the Tools menu and then selecting Merge. This function merges scenarios from other sheets onto the active sheet. A scenario is a set of values to be used as input for a model on your worksheet.

**Syntax**

**SCENARIO.MERGE**(source\_file)

**SCENARIO.MERGE**?(source\_file)

Source\_file    is the name of the book and sheet from which you want to merge scenarios onto the active sheet.

**Related Function**

SCENARIO.GET   Returns the specified information about the scenarios defined on your worksheet

Return to [top](#Q)

# SCENARIO.SHOW

Equivalent to clicking the Scenarios command on the Tools menu and then selecting a scenario and clicking the Show button. Recalculates a model using the specified scenario and displays the result.

**Syntax**

**SCENARIO.SHOW**(**scen\_name**)

Scen\_name    is the name of the previously defined scenario whose values you want to switch to.

Return to [top](#Q)

# SCENARIO.SHOW.NEXT

Equivalent to clicking the Scenarios command on the Tools menu, selecting the next scenario from the Scenarios list, and clicking the Show button. Recalculates a model using the next scenario and displays the result.

**Syntax**

**SCENARIO.SHOW.NEXT**( )

**Remarks**

After displaying the last scenario, running SCENARIO.SHOW.NEXT again displays the first scenario.

Return to [top](#Q)

# SCENARIO.SUMMARY

Equivalent to clicking the Scenarios command on the Tools menu and then clicking the Summary button. Generates a table summarizing the results of all the scenarios for the model on your worksheet.

**Syntax**

**SCENARIO.SUMMARY**(result\_ref, report\_type)

**SCENARIO.SUMMARY**?(result\_ref, report\_type)

Result\_ref    is a reference to the result cells you want to include in the summary report. Normally, result\_ref refers to one or more cells containing the formulas that depend on the changing cell values for your model—that is, the cells that show the results of a particular scenario.

* If result\_ref is omitted, no result cells are included in the report.
* If result\_ref contains nonadjacent references, you must separate the reference areas by commas and enclose result\_ref in an extra set of parentheses.

Report\_type    is a number specifying the type of report desired.

|  |  |
| --- | --- |
| **Report\_type** | **Type of Report** |
| 1 or omitted | A scenario summary report (Microsoft Excel version 4.0) |
| 2 | A scenario PivotTable report. Requires result\_ref. |

**Remarks**

* SCENARIO.SUMMARY generates a summary table of the changing cell and result cell values for each scenario.
* The table is generated on a new sheet in the current workbook. The sheet becomes active after SCENARIO.SUMMARY runs.

Return to [top](#Q)

# SCROLLBAR.PROPERTIES

Sets the properties of the scroll bar and spinner button on a worksheet or dialog sheet.

**Syntax**

**SCROLLBAR.PROPERTIES**(value, min, max, inc, page, link, 3d\_shading)

**SCROLLBAR.PROPERTIES**?(value, min, max, inc, page, link, 3d\_shading)

Value    is the value of the control, and can range from min to max, inclusive. It designates where the scroll bar button is positioned along the scroll bar.

Min    is a number specifying the minimum value that the scroll bar can have. This number ranges from 0 to 30,000, but cannot be greater than the maximum value given in max.

Max    is a number specifying the maximum value that the scroll bar can have. This number ranges from 0 to 30,000.

Inc    is a number specifying the increment that the value is adjusted by when the scrollbar arrow is clicked.

Page    is a number specifying the increment that the value is adjusted by when the page scroll region of a scroll bar is clicked.

Link    is the cell on the macro sheet to which the scroll bar value is linked. Whenever the scroll bar control is changed, the value of the control is entered into the cell. Similarly, whenever the value in the cell is changed, the setting for the scroll bar is also changed. To clear the link, set this value to an empty string.

3d\_shading    is a logical value that specifies whether the scroll bar or spinner button appears as 3-D. If TRUE, the scroll bar or spinner button will appear as 3-D. If FALSE or omitted, the scroll bar or spinner button will not be 3-D. The argument is available for only worksheets.

**Related Functions**

PUSHBUTTON.PROPERTIES   Sets the properties of the push button control

EDITBOX.PROPERTIES   Sets the properties of an edit box on a worksheet or dialog sheet

Return to [top](#Q)

# SELECT

Equivalent to selecting cells or changing the active cell. There are three syntax forms of SELECT. Use syntax 1 to select a cell on a worksheet or macro sheet; use one of the other syntax forms to select worksheet or macro sheet objects or chart items.

Syntax 1   Selects cells

Syntax 2   Selects objects on worksheets

Syntax 3   Selects chart objects

Return to [top](#Q)

# SELECT Syntax 1

Equivalent to selecting cells or changing the active cell. There are three syntax forms of SELECT. Use syntax 1 to select a cell on a worksheet or macro sheet; use one of the other syntax forms to select worksheet or macro sheet objects or chart items.

**Syntax**

**SELECT**(selection, active\_cell)

Selection    is the cell or range of cells you want to select. Selection can be a reference to the active worksheet, such as !$A$1:$A$3 or !Sales, or an R1C1-style reference to a cell or range relative to the active cell in the current selection, such as "R[-1]C[-1]:R[1]C[1]". The reference must be in text form. If selection is omitted, the current selection is used.

Active\_cell    is the cell in selection you want to make the active cell. Active\_cell can be a reference to a single cell on the active worksheet, such as !$A$1, or an R1C1-style reference relative to the active cell, such as "R[-1]C[-1]". The reference must be in text form. If active\_cell is omitted, SELECT makes the cell in the upper-left corner of selection the active cell.

**Remarks**

* Active\_cell must be within selection. If it is not, an error message is displayed and SELECT returns the #VALUE! error value.
* If you are recording a macro using relative references, Microsoft Excel records the action using R1C1-style relative references in the form of text.
* If you are recording using absolute references, Microsoft Excel records the action using R1C1-style absolute references in the form of text.
* You cannot give an external reference to a specific sheet as the selection argument. The sheet on which you want to make a selection must be active when you use SELECT. Use FORMULA.GOTO to make a selection on another sheet in the same workbook or in another workbook.

**Tip**   You can enter data in a cell without selecting the cell by using the reference arguments to the CUT, COPY, or FORMULA functions.

**Examples**

The following macro formula selects cells C3:E5 on the active worksheet and makes C5 the active cell:

SELECT(!$C$3:$E$5, !$C$5)

If the active cell is C3, the following macro formula selects cells E5:G7 and makes cell F6 the active cell in the selection:

SELECT("R[2]C[2]:R[4]C[4]", "R[1]C[1]")

You can also make multiple nonadjacent selections with SELECT. The following macro formula selects a number of nonadjacent ranges:

SELECT("R1C1, R3C2:R4C3, R8C4:R10C5")

The following sequence of macro formulas moves the active cell right, left, down, and up within the selection, just as TAB, SHIFT+TAB, ENTER, and SHIFT+ENTER do:

SELECT(, "RC[1]")

SELECT(, "RC[-1]")

SELECT(, "R[1]C")

SELECT(, "R[-1]C")

Use SELECT with the OFFSET function to select a new range a specified distance away from the current range. For example, the following macro formula selects a range that is the same size as the current range, one column over:

SELECT(OFFSET(SELECTION(), 0, 1))

**Related Functions**

ACTIVE.CELL   Returns the reference of the active cell

SELECT.SPECIAL   Selects a group of cells belonging to a category

SELECTION   Returns the reference of the selection

SELECT Syntax 2   Selects objects on worksheets

SELECT Syntax 3   Selects chart objects

Return to [top](#Q)

# SELECT Syntax 2

Equivalent to selecting objects on a chart, worksheet, or macro sheet. There are three syntax forms of SELECT. Use syntax 2 to select an object on which to perform an action; use one of the other syntax forms to select cells on a worksheet or macro sheet or items on a chart.

**Syntax**

**SELECT**(object\_id\_text, replace)

Object\_id\_text    is text that identifies the object to select. Object\_id\_text can be the name of more than one object. To give the name of more than one object, use the following format:

SELECT("Oval 3, Arc 2, Line 4")

The last item in the object\_id\_text list will be the active object. The active object is important when moving and sizing a group of objects. A multiple selection of objects is moved and sized relative to the upper-left corner of the active object.

Replace    is a logical value that specifies whether previously selected objects are included in the selection. If replace is TRUE or omitted, Microsoft Excel only selects the objects specified by object\_id\_text; if FALSE, it includes any objects that were previously selected. For example, if a button is selected and a SELECT formula selects an arc and an oval, TRUE leaves only the arc and oval selected, and FALSE includes the button with the arc and oval.

**Remarks**

Objects can be identified by their object type and number as described in CREATE.OBJECT, or by the unique number that specifies the order of their creation. For example, if the third object you create is an oval, you could use either "oval 3" or "3" as object\_id\_text.

**Examples**

The following macro formulas each select a number of objects and specify Arc 2 as the active object:

SELECT("Oval 3, Arc 1, Line 4, Arc 2")

SELECT("3, 1, 4, 2")

**Related Functions**

FORMAT.MOVE   Moves the selected object

FORMAT.SIZE   Changes the size of the selected objects

GET.OBJECT   Returns information about an object

SELECTION   Returns the reference of the selection

SELECT Syntax 1   Selects cells

SELECT Syntax 3   Selects chart objects

Return to [top](#Q)

# SELECT Syntax 3

Selects a chart object as specified by the selection code item\_text. There are three syntax forms of SELECT. Use syntax 3 to select a chart item to which you want to apply formatting; use one of the other syntax forms to select cells or objects on a worksheet or macro sheet.

**Syntax**

**SELECT**(item\_text, single\_point)

Item\_text    is a selection code from the following table which specifies which chart object to select.

|  |  |
| --- | --- |
| **To select** | **Item\_text** |
| Entire chart | "Chart" |
| Plot area | "Plot" |
| Legend | "Legend" |
| Primary chart value axis | "Axis 1" |
| Primary chart category axis | "Axis 2" |
| Secondary chart value axis or 3-D series axis | "Axis 3" |
| Secondary chart category axis | "Axis 4" |
| Chart title | "Title" |
| Label for the primary chart value axis | "Text Axis 1" |
| Label for the primary chart category axis | "Text Axis 2" |
| Label for the primary chart series axis | "Text Axis 3" |
| nth floating text item | "Text n" |
| nth arrow | "Arrow n" |
| Major gridlines of value axis | "Gridline 1" |
| Minor gridlines of value axis | "Gridline 2" |
| Major gridlines of category axis | "Gridline 3" |
| Minor gridlines of category axis | "Gridline 4" |
| Major gridlines of series axis | "Gridline 5" |
| Minor gridlines of series axis | "Gridline 6" |
| Primary chart droplines | "Dropline 1" |
| Secondary chart droplines | "Dropline 2" |
| Primary chart hi-lo lines | "Hiloline 1" |
| Secondary chart hi-lo lines | "Hiloline 2" |
| Primary chart up bar | "UpBar1" |
| Secondary chart up bar | "UpBar2" |
| Primary chart down bar | "DownBar1" |
| Secondary chart down bar | "DownBar2" |
| Primary chart series line | "Seriesline1" |
| Secondary chart series line | "Seriesline2" |
| Entire series | "Sn" |
| Data associated with point m in series n if single\_point is TRUE | "SnPm" |
| Text attached to point m of series n | "Text SnPm" |
| Series title text of series n of an area chart | "Text Sn" |
| Base of a 3-D chart | "Floor" |
| Back of a 3-D chart | "Walls" |
| Corners of a 3-D chart | "Corners" |
| Trend line | "SnTm" |
| Error bars | "SnEm" |
| Legend Marker | "Legend Marker n" |
| Legend Entry | "Legend Entry n" |

For trend lines and error bars, the value m can be X or Y, depending on which point you want to select. If m is blank, selects both.

Single\_point    is a logical value that determines whether to select a single point. Single\_point is available only when item\_text is "SnPm".

* If single\_point is TRUE, Microsoft Excel selects a single point.
* If single\_point is FALSE or omitted, Microsoft Excel selects a single point if there is only one series in the chart or selects the entire series if there is more than one series in the chart.
* If you specify single\_point when item\_text is any value other than "SnPm", SELECT returns an error value.

**Examples**

SELECT("Chart") selects the entire chart.

SELECT("Dropline 2") selects the droplines of an overlay chart.

SELECT("S1P3", TRUE) selects the third point in the first series.

SELECT("Text S1") selects the series title text of the first series in an area chart.

**Related Functions**

SELECTION   Returns the reference of the selection

SELECT Syntax 1   Selects cells

SELECT Syntax 2   Selects objects on worksheets

Return to [top](#Q)

# SELECT.ALL

Equivalent to selecting all the sheets in a workbook.

**Syntax**

**SELECT.ALL**( )

Return to [top](#Q)

# SELECT.CHART

Equivalent to the Select Chart command on the Chart menu in Microsoft Excel version 4.0. This function is equivalent to using the third form of SELECT with "Chart" as the item\_text argument.

**Syntax**

**SELECT.CHART**( )

**Remarks**

This function is included for compatibility with macros written with Microsoft Excel for the Macintosh version 1.5 or earlier.

**Related Function**

SELECT   Selects a chart object

Return to [top](#Q)

# SELECT.END

Selects the cell at the edge of the range or the first cell of the next range in the direction specified. Equivalent to pressing CTRL+ARROW in Microsoft Excel for Windows or COMMAND+ARROW in Microsoft Excel for the Macintosh.

**Syntax**

**SELECT.END**(**direction\_num**)

Direction\_num    is a number from 1 to 4 indicating the direction in which to move.

|  |  |
| --- | --- |
| **Direction\_num** | **Direction** |
| 1 | Left (equivalent to CTRL+LEFT ARROW or COMMAND+LEFT ARROW) |
| 2 | Right (equivalent to CTRL+RIGHT ARROW or COMMAND+RIGHT ARROW) |
| 3 | Up (equivalent to CTRL+UP ARROW or COMMAND+UP ARROW) |
| 4 | Down (equivalent to CTRL+DOWN ARROW or COMMAND+DOWN ARROW) |

**Related Function**

SELECT.LAST.CELL   Selects the last cell on a worksheet or macro sheet that contains a formula, value, or format or that is referred to in a formula or name

Return to [top](#Q)

# SELECTION

Returns the reference or object identifier of the selection as an external reference. Use SELECTION to return information about the current selection for use in other macro formulas.

**Syntax**

**SELECTION**( )

If a cell or range of cells is selected, Microsoft Excel returns the corresponding external reference. If an object is selected, Microsoft Excel returns the object identifier listed in the following table.

|  |  |
| --- | --- |
| **Item selected** | **Identifier returned** |
| Imported graphic | Picture n |
| Linked graphic | Picture n |
| Chart picture | Picture n |
| Linked chart | Chart n |
| Range | Picture n |
| Linked range | Picture n |
| Text box | Text n |
| Button | Button n |
| Rectangle | Rectangle n |
| Oval | Oval n |
| Line | Line n |
| Arc | Arc n |
| Group | Group n |
| Freehand drawing or polygon | Drawing n |

SELECTION also returns the identifiers of chart items. The identifiers returned are the same as the identifiers you specify when you use the SELECT function. For a list of these identifiers, see the description of item\_text in SELECT.

If you select cells and use the value returned by SELECTION in a function or operation, you usually get the value contained in the selection instead of its reference. References are automatically converted to the contents of the reference. If you want to work with the actual reference, use SET.NAME to assign a name to it, even if the reference refers to objects. See the last example following. You can also use the REFTEXT function to convert the reference to text, which you can then store or manipulate.

**Remarks**

* If an object is selected, SELECTION returns the identifier of the object. If multiple objects are selected, it returns the identifiers of all the selected objects, as a string separated by commas.
* If more than 1024 characters would be returned, SELECTION returns the #VALUE! error value.

**Examples**

If the sheet in the active window is named SHEET1 in the workbook BOOK1, and if A1:A3 is the selection, then:

SELECTION() equals [BOOK1]SHEET1!A1:A3

The following macro formula moves the current selection one row down:

SELECT(OFFSET(SELECTION(), 1, 0))

The above formula is particularly useful for moving incrementally through a database to add or modify records.

The following macro formula defines the name "EntryRange" on the active sheet to refer to one row below the current selection on the active sheet:

DEFINE.NAME("EntryRange", OFFSET(SELECTION(), 1, 0))

The following macro formula defines the name "Objects" on your macro sheet to refer to the object names in the current multiple selection:

SET.NAME("Objects", SELECTION())

**Related Functions**

ACTIVE.CELL   Returns the reference of the active cell

SELECT   Selects a cell, graphic object, or chart

Return to [top](#Q)

# SELECT.LAST.CELL

Equivalent to choosing the Special button from the Go To dialog box and selecting the Last Cell option. The Go To dialog box appears when you choose the Go To command from the Edit menu. Selects the cell at the intersection of the last row and column that contains a formula, value, or format, or that is referred to in a formula or name.

**Syntax**

**SELECT.LAST.CELL**( )

**Related Function**

SELECT.END   Selects the last cell in a range

Return to [top](#Q)

# SELECT.LIST.ITEM

Selects an item in a list box or in a group box.

**Syntax**

**SELECT.LIST.ITEM**(**index\_num**, selected\_logical)

Index\_num    is the index number of the item to select. Using zero will deselect all items. Adding 1 to the number of items in the list will select all the items specified.

Selected\_logical    is a number that specifies the selection mode of the list box. Zero is single selection. 1 is simple multi-select. 2 is extended multi-select.

**Related Functions**

ADD.LIST.ITEM   Adds an item in a list box or drop-down control on a worksheet or dialog sheet control

REMOVE.LIST.ITEM   Removes an item in a list box or drop-down box

Return to [top](#Q)

# SELECT.PLOT.AREA

Equivalent to clicking the Select Plot Area command on the Chart menu in Microsoft Excel version 4.0. Selects the plot area of the active chart.

**Syntax**

**SELECT.PLOT.AREA**( )

**Remarks**

SELECT.PLOT.AREA is included only for compatibility with previous versions of Microsoft Excel for the Macintosh. SELECT.PLOT.AREA is the same as the SELECT("Plot") function.

**Related Function**

SELECT   Selects a cell, graphic object, or chart

Return to [top](#Q)

# SELECT.SPECIAL

Equivalent to clicking the Go To command on the Edit menu and then selecting the Special button. Use SELECT.SPECIAL to select groups of similar cells in one of a variety of categories.

**Syntax**

**SELECT.SPECIAL**(**type\_num**, value\_type, levels)

**SELECT.SPECIAL**?(type\_num, value\_type, levels)

Type\_num    is a number from 1 to 13 corresponding to options in the Go To Special dialog box and describes what to select.

|  |  |
| --- | --- |
| **Type\_num** | **Description** |
| 1 | Notes/comments |
| 2 | Constants |
| 3 | Formulas |
| 4 | Blanks |
| 5 | Current region |
| 6 | Current array |
| 7 | Row differences |
| 8 | Column differences |
| 9 | Precedents |
| 10 | Dependents |
| 11 | Last cell |
| 12 | Visible cells only (outlining) |
| 13 | All objects |

Value\_type    is a number specifying which types of constants or formulas you want to select. Value\_type is available only when type\_num is 2 or 3.

|  |  |
| --- | --- |
| **Value\_type** | **Selects** |
| 1 | Numbers |
| 2 | Text |
| 4 | Logical values |
| 16 | Error values |

These values can be added to select more than one type. The default for value\_type is 23, which select all value types.

Levels    is a number specifying how precedents and dependents are selected. Levels is available only when type\_num is 9 or 10. The default is 1.

|  |  |
| --- | --- |
| **Levels** | **Selects** |
| 1 | Direct only |
| 2 | All levels |

Return to [top](#Q)

# SEND.KEYS

Sends keystrokes to the active application just as if they were typed at the keyboard. Use SEND.KEYS to send keystrokes that perform actions and execute commands to applications you are running with Microsoft Excel's other dynamic data exchange (DDE) functions.

**Syntax**

**SEND.KEYS**(**key\_text**, wait\_logical)

**Note**This function is available only in Microsoft Excel for Windows.

Key\_text    is the key or key combination you want to send to another application. The format for key\_text is described in the ON.KEY function.

Wait\_logical    is a logical value that determines whether the macro continues before the actions caused by key\_text are carried out.

* If wait\_logical is TRUE, Microsoft Excel waits for the keys to be processed before returning control to the macro.
* If wait\_logical is FALSE or omitted, the macro continues running without waiting for the keys to be processed.

**Remarks**

If Microsoft Excel is the active application, wait\_logical is assumed to be FALSE, even if you enter wait\_logical as TRUE. This is because if wait\_logical is TRUE, Microsoft Excel waits for the keys to be processed in the other application before returning control to the macro. Microsoft Excel doesn't process keys while a macro is running.

**Example**

The following macro uses the Calculator application in Microsoft Excel for Windows to multiply some numbers, and then cuts the result and pastes it into Microsoft Excel.

=EXEC("CALC.EXE", 1)

=SEND.KEYS("10\*30", TRUE)

=SEND.KEYS("~", TRUE)

=SEND.KEYS("%ec", TRUE)

=APP.ACTIVATE(, FALSE)

=SELECT(!B1)

=PASTE()

=RETURN()

**Related Functions**

APP.ACTIVATE   Switches to an application

EXECUTE   Carries out a command in another application

ON.KEY   Runs a macro when a specified key is pressed

Return to [top](#Q)

# SEND.MAIL

Equivalent to clicking the Send Mail command on the File menu. Sends the active workbook using email.

**Syntax**

**SEND.MAIL**(**recipients**, subject, return\_receipt)

**SEND.MAIL**?(recipients, subject, return\_receipt)

**Important**   To use SEND.MAIL in Microsoft Excel for Windows, you must be using a mail client that supports the Messaging Applications Programming Interface (MAPI) or Vendor-Independent Messaging (VIM). To use SEND.MAIL in Microsoft Excel for the Macintosh, you must be using Microsoft Mail version 2.0 or later.

Recipients    is the name of the person to whom you want to send the mail. The name should be given as text.

* To specify more than one name, give the list of names as an array. For example, SEND.MAIL({"John", "Paul", "George", "Ringo"}) would send the active workbook to the four names in the array. You can also refer to a range on a sheet or macro sheet that contains a list of names to whom you want the mail to be sent.
* To send mail to users on different Microsoft Mail for the Macintosh servers, specify the server name along with the user name. The following text, as the recipients argument, sends mail to wandagr on server2, gregpr on the current server, and victorge on server7:

{"wandagr@server2", "gregpr", "victorge@server7"}

Subject    is a text string that specifies the subject of the message. If subject is omitted, the name of the active workbook is used as the subject.

Return\_receipt    is a logical value that corresponds to the Return Receipt check box. If return\_receipt is TRUE, Microsoft Excel selects the check box and sends a return receipt; if FALSE or omitted, Microsoft Excel clears the check box.

**Related Function**

OPEN.MAIL   Opens files sent via Microsoft Mail that Microsoft Excel can open

Return to [top](#Q)

# SEND.TO.BACK

Sends the selected object or objects to the back. Use SEND.TO.BACK to position selected objects behind other objects.

If the selection is not an object or a group of objects, SEND.TO.BACK returns the #VALUE! error value and interrupts the macro.

**Syntax**

**SEND.TO.BACK**( )

**Related Function**

BRING.TO.FRONT   Brings selected objects to the front

Return to [top](#Q)

# SERIES

Charts Only

Represents a data series in the active chart. SERIES is used only in charts; you cannot enter it on a sheet or macro sheet. You normally create or change data series by using the Chart Wizard or EDIT.SERIES macro function, which is equivalent to the Edit Series command on the Chart menu in Microsoft Excel version 4.0. However, you can edit a data series manually by selecting it, switching to the formula bar, and typing the changes.

**Syntax**

**SERIES**(name\_ref, categories, **values, plot\_order**)

Name\_ref    is the name of the data series. It can be an external reference to a single cell or a name defined as a single cell. Name\_ref can also be text enclosed in quotation marks (for example, "Projected Sales").

Categories    is an external reference to the name of the workbook and to the cells that contain one of the following sets of data:

* Category labels for all charts except xy (scatter) charts
* X-coordinate data for xy (scatter) charts

Values    is an external reference to the name of the workbook and to the cells that contain values (or y-coordinate data in scatter charts).

Plot\_order    is an integer specifying whether the series is plotted first, second, or third, and so on, in the chart. No two series can have the same plot\_order.

**Remarks**

* Categories and values can be arrays or references to a multiple selection, although they cannot be names that refer to a multiple selection. If you specify a multiple selection for any of these arguments, make sure you include the necessary sets of parentheses so that Microsoft Excel does not treat the components of the references as separate arguments.
* If either categories or values is a multiple selection, then all areas in that selection must be either vertical (more rows than columns) or horizontal (more columns than rows).

**Related Functions**

CHART.WIZARD   Creates and formats a chart

EDIT.SERIES   Creates or changes a chart series

Return to [top](#Q)

# SERIES.AXES

Equivalent to the Axis Tab in the Format Data Series dialog box. Changes the axis on which a series is plotted. This function is for compatibility with Microsoft Excel versions earlier than Microsoft Excel 97.

**Syntax**

**SERIES.AXES**(axis)

Axis    is a number specifying on which axis to plot the data series: use 1 for primary axis, 2 for secondary axis.

Return to [top](#Q)

# SERIES.ORDER

Changes the order of series in a chart.

**Syntax**

**SERIES.ORDER**(chart\_num, old\_series\_num, new\_series\_num)

Chart\_num    is the number of the group containing the series you want to change

Old\_series\_num    is the current number of the series in the group.

New\_series\_num    is the new number you want for the series in the group.

Return to [top](#Q)

# SERIES.X

Equivalent to the X Values tab in the Format Data Series dialog box. Specifies the category labels (x values) for a data series. This function is for compatibility with Microsoft Excel versions earlier than Microsoft Excel 97.

**Syntax**

**SERIES.X**(x\_ref)  
X-ref    is an external reference in the form of text specifying the range containing the category labels (or x values for a scatter (xy) chart) you want to use.

**Related Function**

SERIES.Y   Specifies the name and values for a data series

Return to [top](#Q)

# SERIES.Y

Equivalent to the Name and Values tab in the Format Data Series dialog box. Specifies the name and values for a data series. This function is for compatibility with Microsoft Excel versions earlier than Microsoft Excel 97.

**Syntax**

**SERIES.Y**(name\_ref, y\_ref)

Name\_ref    is text or an external reference in the form of text specifying the name for the data series that appears in the legend for the chart.

Y\_ref    is an external reference in the form of text specifying the range containing the values for the data series.

**Related Function**

SERIES.X   Specifies the category labels (x values) for a data series

Return to [top](#Q)

# SET.CONTROL.VALUE

Changes the value for the active control, such as a list box, drop-down box, check box, option button, scroll bar, and spinner button.

**Syntax**

**SET.CONTROL.VALUE**(value)

Value    is the value you want to change. The control interprets this value as follows:

|  |  |
| --- | --- |
| **Control** | **Value is** |
| List box | The index of the selected item. If zero, then no item is selected. |
| Drop-down box | The index of the selected item. If zero, then no item is selected. |
| Check box | 0 = Off 1 = On 2 = Mixed |
| Option button | 0= Off 1 = On |
| Scroll bar | The numeric value of the control, between the maximum and minimum values |
| Spinner button | The numeric value of the control, between the maximum and minimum values |

**Related Functions**

ADD.LIST.ITEM   Adds an item in a list box or drop-down control on a worksheet or dialog sheet control

REMOVE.LIST.ITEM   Removes an item in a list box or drop-down box

SELECT.LIST.ITEM   Selects an item in a list box or in a group box

CHECKBOX.PROPERTIES   Sets various properties of check box and option box controls

SCROLLBAR.PROPERTIES   Sets the properties of the scroll bar and spinner controls

Return to [top](#Q)

# SET.CRITERIA

Equivalent to clicking the Set Criteria command on the Data menu in Microsoft Excel version 4.0. Defines the name Criteria for the selected range on a sheet or macro sheet.

**Syntax**

**SET.CRITERIA**( )

**Related Functions**

SET.DATABASE   Equivalent to clicking the Set Database command on the Data menu in Microsoft Excel version 4.0

SET.EXTRACT   Equivalent to clicking the Set Extract command on the Data menu in Microsoft Excel version 4.0

Return to [top](#Q)

# SET.DATABASE

Equivalent to clicking the Set Database command on the Data menu in Microsoft Excel version 4.0. Defines the name Database for the selected range on a sheet or macro sheet.

**Syntax**

**SET.DATABASE**( )

**Related Functions**

SET.CRITERIA   Equivalent to clicking the Set Criteria command on the Data menu in Microsoft Excel version 4.0

SET.EXTRACT   Equivalent to clicking the Set Extract command on the Data menu in Microsoft Excel version 4.0

Return to [top](#Q)

# SET.DIALOG.DEFAULT

Sets which button is automatically pressed (the default button) when the user presses ENTER. While running, this default button is visually recognized by its thick border. This function is used only with a dialog sheet active.

**Syntax**

**SET.DIALOG.DEFAULT**(**object\_id\_text**)

Object\_id\_text    is the name of the button control to set as the default button, as in "Button 5".

**Related Function**

SET.DIALOG.FOCUS   Sets the focus of a dialog box

Return to [top](#Q)

# SET.DIALOG.FOCUS

Sets the focus of a dialog box. This function is used only with a dialog sheet active.

**Syntax**

**SET.DIALOG.FOCUS**(**object\_id\_text**)

Object\_id\_text    the name of the control or object as text to give the focus to, as in "Check box 4".

**Related Function**

SET.DIALOG.DEFAULT   Sets which button is automatically pressed (the default button) when the user presses ENTER

Return to [top](#Q)

# SET.EXTRACT

Equivalent to clicking the Set Extract command on the Data menu in Microsoft Excel version 4.0. Defines the name Extract for the selected range on the active sheet.

**Syntax**

**SET.EXTRACT**( )

**Related Functions**

SET.DATABASE   Equivalent to clicking the Set Database command on the Data menu in Microsoft Excel version 4.0

SET.CRITERIA   Equivalent to clicking the Set Criteria command on the Data menu in Microsoft Excel version 4.0

Return to [top](#Q)

# SET.LIST.ITEM

Sets the text of an item in a list box or drop-down box control.

**Syntax**

**SET.LIST.ITEM**(**text**, **index\_num**)

Text    specifies the text of the item to be added. Instead of text, an empty string may be inserted.

Index\_num    is the list index of the item to be changed, from 1 to the number of items in the list.

**Remarks**

If the list box or drop-down box was already filled using the LISTBOX.PROPERTIES function, then changing an item with SET.LIST.ITEM causes the fillrange contents to be discarded, leaving a list with one non-blank element and index\_num entries.

**Related Functions**

REMOVE.LIST.ITEM   Removes an item in a list box or drop-down box

SELECT.LIST.ITEM   Selects an item in a list box or in a group box

Return to [top](#Q)

# SET.NAME

Defines a name on a macro sheet to refer to a value. The defined name exists only on the macro sheet's list of names and does not appear in the global list of names for the workbook. The SET.NAME function is useful for storing values while the macro is calculating.

**Syntax**

**SET.NAME**(**name\_text**, value)

Name\_text    is the name in the form of text that refers to value.

Value    is the value you want to store in name\_text.

* If value is omitted, the name name\_text is deleted.
* If value is a reference, name\_text is defined to refer to that reference.

**Remarks**

* If you want to define a name as a constant value, you can use the following syntax instead of SET.NAME:

name\_text=value

See the first two examples following.

* SET.NAME defines names as absolute references, even if a relative reference is specified. See the third and fourth examples following.
* If you want name\_text to refer permanently to the value of a referenced cell rather than to the reference itself, you must use the DEREF function. Use of DEREF prevents name\_text from referring to a new value every time the contents of the referenced cell changes. See the last example following.

**Tips**

* If you need to return an array to a macro sheet (for example, if the macro needs a list of all open windows), assign a name to the array instead of placing the array information in a range of cells. For example:

SET.NAME("OpenDocuments", WINDOWS()) or  
SET.NAME("OpenDocuments", {"WORKSHEET1", "WORKSHEET2"})

* You can then use the INDEX function with the name you have defined to access items in the array stored in the name.
* When you're debugging a macro and want to know the current value assigned to a name created by SET.NAME, you can halt the macro, click Define on the Name submenu of the Insert menu, and select the name from the Define Name dialog box.

**Examples**

Each of these formulas defines the name Counter to refer to the constant number 1 on the macro sheet:

SET.NAME("Counter", 1)

Counter=1

Each of these formulas redefines Counter to refer to the current value of Counter plus 1:

SET.NAME("Counter", Counter+1)

Counter=Counter+1

The following macro formula defines the name Reference to refer to cell $A$1:

SET.NAME("Reference", A1)

The following macro formula defines the name Results to refer to the cells $A$1:$C$3:

SET.NAME("Results", A1:C3)

The following macro formula defines the name Range as the current selection:

SET.NAME("Range", SELECTION())

If $A$1 contains the value 2, the following macro formula defines the name Index to refer to the constant value 2:

SET.NAME("Index", DEREF(A1))

**Related Functions**

DEFINE.NAME   Defines a name on the active worksheet or macro sheet

SET.VALUE   Sets the value of a cell on a macro sheet

Return to [top](#Q)

# SET.PAGE.BREAK

Equivalent to clicking the Page Break command on the Insert menu. Sets manual page breaks. Use SET.PAGE.BREAK to override the automatic page breaks. Setting a manual page break changes the automatic page breaks that follow it.

The page break occurs above and to the left of the active cell and appears as dotted lines if you have set up a printer. If the active cell is in column A, a manual page break is added only above the cell. If the active cell is in row 1, a manual page break is added only at the left edge of the cell. If the row or column next to the active cell already has a page break, SET.PAGE.BREAK takes no action.

**Syntax**

**SET.PAGE.BREAK**( )

**Related Functions**

PRINT.PREVIEW   Previews pages and page breaks before printing

REMOVE.PAGE.BREAK   Removes manual page breaks

Return to [top](#Q)

# SET.PREFERRED

Changes the default format that Microsoft Excel uses when you create a new chart or when you format a chart PREFERRED macro function. When you use the SET.PREFERRED function, the format of the active chart becomes the preferred format.

**Syntax**

**SET.PREFERRED**(format)

Format    is the name of the format that you want as the default format for charts. If omitted, the format of the currently active chart is used. If format is "Built\_in", then Microsoft Excel will use the standard, built-in chart as the default. If the chart was created in Microsoft Excel version 4.0 and if format is "PREFERRED", then the preferred chart format used in Microsoft Excel version 4.0 will be used. Format is case sensitive.

**Related Function**

PREFERRED   Changes the format of the active chart to the preferred format

Return to [top](#Q)

# SET.PRINT.AREA

Defines the print area for the workbook—the area that prints when you click the Print command on the File menu. Equivalent to entering a range in the Print Area edit box on the Sheet tab in the Page Setup dialog box, which appears when you click the Page Setup command on the File menu.

**Syntax**

**SET.PRINT.AREA**(range)

Range    is the reference to the range that you want to be printed. If you specify no range by using a set of empty quotation marks (""), deletes the print area.

**Remarks**

* If you use SET.PRINT.AREA with a multiple selection and then use the PRINT function, the individual selections are printed one after the other in the order they were selected.
* To resume printing the entire worksheet, click the Page Setup command on the File menu and click the Sheet tab. Then delete the range in the Print Area edit box.

**Related Functions**

PRINT   Prints the active sheet

SET.PRINT.TITLES   Identifies text to print as titles

Return to [top](#Q)

# SET.PRINT.TITLES

Defines the print titles for the sheet. Use SET.PRINT.TITLES if you want Microsoft Excel to print the titles whenever it prints any cells in a row or column that intersect the print titles area; a cell need only share the row or column with a print title for the title to be printed above or to the left of that cell.

**Syntax**

**SET.PRINT.TITLES**(titles\_for\_cols\_ref, titles\_for\_rows\_ref)

**SET.PRINT.TITLES**?(titles\_for\_cols\_ref, titles\_for\_rows\_ref)

Titles\_for\_cols\_ref    is a reference to the row to be used as a title for columns.

* If you specify part of a row, Microsoft Excel expands the title to a full row.
* If you omit titles\_for\_cols\_ref, Microsoft Excel uses the existing row of column titles, if any.
* If you specify empty text (""), Microsoft Excel removes the row from the print titles definition.

Titles\_for\_rows\_ref    is a reference to the column to be used as a title for rows.

* If you specify part of a column, Microsoft Excel expands the title to a full column.
* If you omit titles\_for\_rows\_ref, Microsoft Excel uses the existing column of row titles, if any.
* If you specify empty text (""), Microsoft Excel removes the column from the print titles definition.

**Remarks**

* SET.PRINT.TITLES operates on the current sheet. If you specify a range that is invalid for the current sheet, Microsoft Excel returns the #VALUE error value.
* The print titles selection can be a multiple selection. Microsoft Excel names this selection Print\_Titles when SET.PRINT.TITLES is run.

**Related Functions**

DEFINE.NAME   Defines a name on the active worksheet or macro sheet

PRINT   Prints the active sheet

SET.PRINT.AREA   Defines the print area

Return to [top](#Q)

# SET.UPDATE.STATUS

Sets the update status of a link to automatic or manual. Use SET.UPDATE.STATUS to change the way a link is updated.

**Syntax**

**SET.UPDATE.STATUS**(**link\_text, status**, type\_of\_link)

Link\_text    is the path of the linked file for which you want to change the update status.

Status    is the number 1 or 2 and describes how you want the link to be updated.

|  |  |
| --- | --- |
| **Status** | **Update method** |
| 1 | Automatic |
| 2 | Manual |

Type\_of\_link    is a number from 1 to 4 that specifies what type of link you want to get information about.

|  |  |
| --- | --- |
| **Type\_of\_link** | **Link document type** |
| 1 | Not available |
| 2 | DDE/OLE link |
| 3 | Not available |
| 4 | Not available |

**Example**

In Microsoft Excel for Windows, the following macro formula sets the update status of the DDE link to Microsoft Word for Windows to manual:

SET.UPDATE.STATUS("WordDocument|'C:\MEMO.DOC'!DDE.LINK1", 2, 2)

**Related Functions**

GET.LINK.INFO   Returns information about a link

UPDATE.LINK   Updates a link to another document

Return to [top](#Q)

# SET.VALUE

Changes the value of a cell or cells on the macro sheet (not the worksheet) without changing any formulas entered in those cells. Use SET.VALUE to assign initial values and to store values during the calculation of a macro. SET.VALUE is especially useful for initializing a dialog box and the conditional test in a WHILE loop. SET.VALUE assigns values to a specific reference or to the name of a reference that has already been defined. For information about creating a new name or entering data on a worksheet, see "Remarks" later in this topic.

**Syntax**

**SET.VALUE**(**reference**, values)

Reference    specifies the cell or cells on the macro sheet to which you want to assign a new value or values. If the cell is empty, enters the value in the cell.

* If a cell in reference previously contained a formula, the formula is not changed, but the value of the cell might change. See the second example following.
* If reference is a reference to a range of cells, rather than to a single cell, then values should be an array of the same size. If not, Microsoft Excel expands it into multiple values using the normal rules for expanding arrays. See the third example following.

Values    is the value or set of values to which you want to assign the cell or cells in reference.

**Remarks**

Consider the following guidelines as you choose a function to set values on a worksheet or macro sheet:

* Use SET.VALUE to assign initial values to a reference (including names that have already been defined) on a macro sheet, and to store values during the calculation of a macro.
* Use FORMULA to enter values in a worksheet cell.
* Use SET.NAME to change the value of a name on a macro sheet (the name is created if it does not already exist). For more information, see SET.NAME.
* Use DEFINE.NAME to create or change the value of a name on a worksheet.

**Examples**

The following macro formula changes the value of cell A1 on the macro sheet to 1:

SET.VALUE($A$1, 1)

Suppose the name TempAverage refers to a cell containing the formula AVERAGE(Temp1, Temp2, Temp3). The following formula assigns the value 99 to this cell, even if the average of the arguments is not 99, without changing the formula in TempAverage:

SET.VALUE(TempAverage, 99)

The preceding formula is useful if a WHILE loop or some other conditional test depends on TempAverage and you want to force the conditional test to have a particular result. Of course, TempAverage is restored to its correct value as soon as it is recalculated. (Recall that unlike formulas in a worksheet, formulas in a macro sheet are not recalculated until the macro actually uses them.)

The following macro formula stores the values 1, 2, 3, and 4 in cells A1:B2:

SET.VALUE($A$1:$B$2, {1, 2;3, 4})

**Related Functions**

DEFINE.NAME   Defines a name on the active worksheet or macro sheet

FORMULA   Enters values into a cell or range or onto a chart

SET.NAME   Defines a name as a value

Return to [top](#Q)

# SHORT.MENUS

Equivalent to clicking the Short Menus command on the Options menu or the Chart menu in Microsoft Excel version 3.0 or earlier.

**Syntax**

**SHORT.MENUS**(logical)

Return to [top](#Q)

# SHOW.ACTIVE.CELL

Scrolls the active window so the active cell becomes visible. If an object is selected, SHOW.ACTIVE.CELL returns the #VALUE! error value and halts the macro.

**Syntax**

**SHOW.ACTIVE.CELL**( )

**Related Functions**

ACTIVE.CELL   Returns the reference of the active cell

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

Return to [top](#Q)

# SHOW.BAR

Displays the specified menu bar. Use SHOW.BAR to display a menu bar you have created with the ADD.BAR function or to display a built-in Microsoft Excel 95 or earlier version menu bar.

**Syntax**

**SHOW.BAR**(bar\_num)

Bar\_num    is the number of the menu bar you want to display. It can be the number of one of the Microsoft Excel built-in menu bars, the number returned by a previously executed ADD.BAR function, or a reference to a cell containing a previously executed ADD.BAR function.

If bar\_num is omitted, Microsoft Excel displays the appropriate menu bar for the active workbook as shown in the following table.

|  |  |
| --- | --- |
| **Bar\_num** | **Bar displayed** |
| 1 | A sheet or macro sheet (Microsoft Excel version 4.0) |
| 2 | A chart (Microsoft Excel version 4.0) |
| 3 | No active window |
| 4 | The Info window (Microsoft Excel 95 or earlier versions) |
| 5 | A sheet or macro sheet (short menus) |
| 6 | A chart (short menus) |
| 7 | Shortcut menus 1 (for Cells, Workbook tabs, Toolbars, VB Windows) |
| 8 | Shortcut menus 2 (for objects) |
| 9 | Shortcut menus 3 (for chart elements) |
| 10 | A sheet or macro sheet |
| 11 | A chart |
| 12 | A Visual Basic module |
| 13-35 | Reserved for use by shortcut menus. These numbers will return an error if a macro tries to do anything with them. |
| 37-51 | Custom menu bar for macro use |

**Remarks**

* When displaying a built-in menu bar, you can display only bars 1 or 5 if a sheet or macro sheet is active, bars 2 or 6 if a chart is active, and so on. If you try to display a chart menu bar while a sheet or macro sheet is active, SHOW.BAR returns an error and interrupts the current macro.
* Displaying a custom menu bar disables automatic menu-bar switching when different types of sheets are selected. For example, if a custom menu bar is displayed and you switch to a chart, neither of the two chart menus is automatically displayed as it would be when you are using the built-in menu bars. Automatic menu-bar switching is reenabled when a built-in bar is displayed using SHOW.BAR.

**Example**

The following macro formula displays short menus on a worksheet or macro sheet:

SHOW.BAR(5)

**Related Functions**

ADD.BAR   Adds a menu bar

DELETE.BAR   Deletes a menu bar

SHOW.TOOLBAR   Hides or displays a toolbar

Return to [top](#Q)

# SHOW.CLIPBOARD

Displays the contents of the Clipboard in a new window.

**Syntax**

**SHOW.CLIPBOARD**( )

**Remarks**

* In Microsoft Excel for Windows, the Clipboard must already be running if you want to display its contents in a new window. If it is not already running, you must run the SHOW.CLIPBOARD function twice, once to start the Clipboard application and again to display it in a new window.
* If the Clipboard contains cells, the window shows the size of the Clipboard contents in rows and columns. If the Clipboard contains text cut from the formula bar, the window displays the text.

Return to [top](#Q)

# SHOW.DETAIL

Expands or collapses the detail under the specified expand or collapse button.

**Syntax**

**SHOW.DETAIL**(**rowcol, rowcol\_num**, expand, show\_field)

Rowcol    is a number that specifies whether to operate on rows or columns of data.

|  |  |
| --- | --- |
| **Rowcol** | **Operates on** |
| 1 | Rows |
| 2 | Columns |
| 3 | The current cell's row or column. The second argument, rowcol\_num, is then ignored. |

Rowcol\_num    is a number that specifies the row or column to expand or collapse. If you are in A1 mode, you must still give the column as a number. If rowcol\_num is not a summary row or column, SHOW.DETAIL returns the #VALUE! error value and interrupts the macro.

Expand    is a logical value that specifies whether to expand or collapse the detail under the row or column. If expand is TRUE, Microsoft Excel expands the detail under the row or column; if FALSE, it collapses the detail under the row or column. If expand is omitted, the detail is expanded if it is currently collapsed and collapsed if it is currently expanded.

Show\_Field    is a string specifying the name of the field to add to a PivotTable report, if the selection is inside a PivotTable report. The new field is added as the new innermost field. Available for only innermost row or column fields.

**Related Function**

SHOW.LEVELS   Displays a specific number of levels of an outline

Return to [top](#Q)

# SHOW.DIALOG

Runs a dialog on a dialog sheet.

**Syntax**

**SHOW.DIALOG**(dialog\_sheet)

Dialog\_sheet    is the name of the dialog sheet to run. If omitted, the active sheet will be the sheet that is run. If this function is run on a sheet other than a dialog sheet, this function returns the #VALUE error value.

**Remarks**

Returns TRUE if the dialog box is closed by the user choosing an OK button. Returns FALSE if the dialog box is cancelled by choosing the Cancel button or the ESC key, or in Microsoft Excel for the Macintosh by pressing COMMAND+. (period).

**Related Function**

HIDE.DIALOG   Closes the dialog box that has the current focus

Return to [top](#Q)

# SHOW.INFO

This function should not be used. The Info Window has been removed from Microsoft Excel 97 or later.

**Related Functions**

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

GET.CELL   Returns information about the specified cell

SELECT   Selects a cell, worksheet object, or chart item

Return to [top](#Q)

# SHOW.LEVELS

Displays the specified number of row and column levels of an outline.

**Syntax**

**SHOW.LEVELS**(row\_level, col\_level)

Row\_level    specifies the number of row levels of an outline to display. If the outline has fewer levels than specified by row\_level, Microsoft Excel shows all levels. If row\_level is zero or omitted, no action is taken on rows.

Col\_level    specifies the number of column levels of an outline to display. If the outline has fewer levels than specified by col\_level, Microsoft Excel shows all levels. If col\_level is zero or omitted, no action is taken on columns.

**Remarks**

If you omit both arguments, SHOW.LEVELS returns the #VALUE! error value.

**Related Function**

SHOW.DETAIL   Expands or collapses a portion of an outline

Return to [top](#Q)

# SHOW.TOOLBAR

Equivalent to selecting the check box corresponding to a toolbar on the Toolbars tab in the Customize dialog box, which appears when you select the Customize command (View menu, Toolbars submenu). Hides or displays a toolbar. Use SHOW.TOOLBAR to display or hide a menu bar you have created with the ADD.BAR function or to display a built-in Microsoft Excel 95 or earlier version toolbar.

**Syntax**

**SHOW.TOOLBAR**(**bar\_id, visible**, dock, x\_pos, y\_pos, width, protect, tool\_tips, large\_buttons, color\_buttons)

Bar\_id    is a number or name of a toolbar corresponding to the toolbars you want to display. For detailed information about bar\_id, see ADD.TOOL.

Visible    is a logical value that, if TRUE, specifies that the toolbar is visible or, if FALSE, specifies that the toolbar is hidden.

Dock    specifies the docking location of the toolbar.

|  |  |
| --- | --- |
| **Dock** | **Position of toolbar** |
| 1 | Top of workspace |
| 2 | Left edge of workspace |
| 3 | Right edge of workspace |
| 4 | Bottom of workspace |
| 5 | Floating (not docked) |

X\_pos    specifies the horizontal position of the toolbar.

* If the toolbar is docked (not floating), x\_pos is measured horizontally from the left edge of the toolbar to the left edge of the toolbar's docking area.
* If the toolbar is floating, x\_pos is measured horizontally from the left edge of the toolbar to the right edge of the rightmost toolbar in the left docking area.
* X\_pos is measured in points. A point is 1/72nd of an inch.

Y\_pos    specifies the vertical position of the toolbar.

* If the toolbar is docked, y\_pos is measured vertically from the top edge of the toolbar to the top edge of the toolbar's docking area.
* If the toolbar is floating, y\_pos is measured vertically from the top edge of the toolbar to the top edge of the Microsoft Excel workspace.
* Y\_pos is measured in points.

Width    specifies the width of the toolbar and is measured in points. If you omit width, Microsoft Excel uses the existing width setting.

Protect    is a number specifying the degree to which you can modify a toolbar and its buttons. Each succeeding protect number retains the protection status of its previous numbers. For example, a protect status of 3 (a toolbar cannot become docked if it is floating) assumes the protection status of 0, 1, and 2 as well.

|  |  |
| --- | --- |
| **Protect** | **Description** |
| 0 | Default. Toolbars can be re-shaped, docked, and floating. Toolbar buttons can be removed from and moved to the toolbar. |
| 1 | Toolbars can be re-shaped, docked, and floating. Toolbar buttons can not be removed from nor moved to the toolbar. |
| 2 | A floating toolbar cannot be re-shaped. It can be docked. |
| 3 | A floating toolbar cannot be docked. If it is already docked, it cannot become floating. |
| 4 | The toolbar cannot be moved at all. If it is already floating, it cannot be re-shaped or moved. If it is docked, it cannot become un-docked. |

Tool\_tips    is a logical value that corresponds to the Show Screentips On Toolbars check box on the Options tab. If TRUE, ScreenTips will be displayed. If FALSE, ScreenTips will not be displayed.

Large Buttons    is a logical value that corresponds to the Large Icons check box on the Options tab. If TRUE, large icons will be displayed. If FALSE, large icons will not be displayed.

Color\_buttons    is a logical value that corresponds to the Color Toolbars check box. If TRUE, the toolbar buttons will be displayed in color. If FALSE, the toolbar buttons will not be displayed in color. This argument is for compatibility with Microsoft Excel version 5.0.

**Related Functions**

ADD.BAR   Adds a menu bar

ADD.TOOLBAR   Creates a new toolbar with the specified tools

Return to [top](#Q)

# SIZE

Equivalent to clicking the Size command on the Control menu in Microsoft Excel for Windows version 3.0 or earlier or to changing the size of a window by dragging its border. In Microsoft Excel for the Macintosh version 3.0 or earlier, equivalent to changing the size of a window by dragging its size box. This function is included only for macro compatibility and will be converted to WINDOW.SIZE when you open older macro sheets. For more information, see WINDOW.SIZE.

**Syntax**

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

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

**Related Function**

WINDOW.SIZE   Changes the size of the active window

Return to [top](#Q)

# SLIDE.COPY.ROW

This function should not be used in Microsoft Excel 95 or later because the Slide Show add-in is available only in Microsoft Excel version 5.0 or earlier versions.

Equivalent to clicking the Copy Row button on a slide show sheet. Copies the selected slides, each of which is defined on a single row, to the Clipboard.

If this function is not available, you must install the Slide Show add-in.

**Syntax**

**SLIDE.COPY.ROW**( )

**Remarks**

* SLIDE.COPY.ROW, SLIDE.CUT.ROW, SLIDE.DELETE.ROW, and SLIDE.PASTE.ROW return TRUE if successful, or FALSE if not successful. If the active sheet is not a slide show or is protected, these functions return the #N/A error value. If the current selection is not valid, these functions return the #VALUE! error value.

**Related Functions**

SLIDE.CUT.ROW   Cuts the selected slides and pastes them onto the Clipboard

SLIDE.DEFAULTS   Specifies default values for the active slide show sheet

SLIDE.DELETE.ROW   Deletes the selected slides

SLIDE.EDIT   Changes the attributes of the selected slide

SLIDE.GET   Returns information about a slide or slide show

SLIDE.PASTE   Pastes the contents of the Clipboard onto a slide

SLIDE.PASTE.ROW   Pastes previously cut or copied slides onto the current selection

SLIDE.SHOW   Starts a slide show in the active sheet

Return to [top](#Q)

# SLIDE.CUT.ROW

This function should not be used in Microsoft Excel 95 or later because the Slide Show add-in is available only in Microsoft Excel version 5.0 or earlier versions.

Equivalent to clicking the Cut Row button on a slide show sheet. Cuts the selected slides, each of which is defined on a single row, and pastes them onto the Clipboard. For more information, see SLIDE.COPY.ROW.

If this function is not available, you must install the Slide Show add-in.

**Syntax**

**SLIDE.CUT.ROW**( )

**Related Function**

SLIDE.COPY.ROW   Copies the selected slides and pastes them onto the Clipboard

Return to [top](#Q)

# SLIDE.DEFAULTS

This function should not be used in Microsoft Excel 95 or later because the Slide Show add-in is available only in Microsoft Excel version 5.0 or earlier versions.

Equivalent to clicking the Set Defaults button on a slide show sheet. Specifies the default values for the transition effect, speed, advance rate, and sound on the active slide show sheet.

If this function is not available, you must install the Slide Show add-in.

**Syntax**

**SLIDE.DEFAULTS**(effect\_num, speed\_num, advance\_rate\_num, soundfile\_text)

**SLIDE.DEFAULTS**?(effect\_num, speed\_num, advance\_rate\_num, soundfile\_text)

For a description of the arguments, see SLIDE.PASTE. If an argument is omitted, its default value is not changed.

**Remarks**

* SLIDE.DEFAULTS returns TRUE if it successfully changes the default values, or FALSE if you click the Cancel button when using the dialog-box form. If the active sheet is not a slide show or is protected, SLIDE.DEFAULTS returns the #N/A error value.

Return to [top](#Q)

# SLIDE.DELETE.ROW

This function should not be used in Microsoft Excel 95 or later because the Slide Show add-in is available only in Microsoft Excel version 5.0 or earlier versions.

Equivalent to clicking the Delete Row button on a slide show sheet. Deletes the selected slides, each of which is defined on a single row. For more information, see SLIDE.COPY.ROW.

If this function is not available, you must install the Slide Show add-in.

**Syntax**

**SLIDE.DELETE.ROW**( )

**Related Function**

SLIDE.COPY.ROW   Copies the selected slides and pastes them onto the Clipboard

Return to [top](#Q)

# SLIDE.EDIT

This function should not be used in Microsoft Excel 95 or later because the Slide Show add-in is available only in Microsoft Excel version 5.0 or earlier versions.

Equivalent to clicking the Edit button in a slide show sheet. Gives the currently selected slide the attributes you specify.

If this function is not available, you must install the Slide Show add-in.

**Syntax**

**SLIDE.EDIT**(effect\_num, speed\_num, advance\_rate\_num, soundfile\_text)

**SLIDE.EDIT**?(effect\_num, speed\_num, advance\_rate\_num, soundfile\_text)

For a description of the arguments, see SLIDE.PASTE.

**Remarks**

* SLIDE.EDIT returns TRUE if it successfully edits the slide, or FALSE if you click the Cancel button when using the dialog-box form. If the active sheet is not a slide show or is protected, SLIDE.EDIT returns the #N/A error value. If the current selection is not a valid slide, SLIDE.EDIT returns the #VALUE error value.

**Related Function**

SLIDE.PASTE   Pastes the contents of the Clipboard onto a slide

Return to [top](#Q)

# SLIDE.GET

This function should not be used in Microsoft Excel 95 or later because the Slide Show add-in is available only in Microsoft Excel version 5.0 or earlier versions.

Returns the specified information about a slide show or a specific slide in the slide show.

If this function is not available, you must install the Slide Show add-in.

**Syntax**

**SLIDE.GET**(**type\_num**, name\_text, slide\_num)

Type\_num    is a number specifying the type of information you want.

These values of type\_num return information about a slide show.

|  |  |
| --- | --- |
| **Type\_num** | **Type of information** |
| 1 | Number of slides in the slide show |
| 2 | A two-item horizontal array containing the numbers of the first and last slides in the current selection, or the #VALUE error value if the selection is nonadjacent |
| 3 | Version number of the Slide Show add-in that created the slide show sheet |

These values of type\_num return information about a specific slide in the slide show.

|  |  |
| --- | --- |
| **Type\_num** | **Type of information** |
| 4 | Transition effect number |
| 5 | Transition effect name |
| 6 | Transition effect speed |
| 7 | Number of seconds the slide is displayed before advancing |
| 8 | Name of the sound file associated with the slide, or empty text ("") if none is specified (in Microsoft Excel for the Macintosh, this includes the number or name of the sound resource within the sound file) |

Name\_text    is the name of an open slide show sheet for which you want information. If name\_text is omitted, it is assumed to be the active sheet.

Slide\_num    is the number of the slide about which you want information.

* If slide\_num is omitted, it is assumed to be the slide associated with the active cell on the sheet specified by name\_text.
* If type\_num is 1 through 3, slide\_num is ignored.

Return to [top](#Q)

# SLIDE.PASTE

This function should not be used in Microsoft Excel 95 or later because the Slide Show add-in is available only in Microsoft Excel version 5.0 or earlier versions.

Equivalent to clicking the Paste button on a slide show sheet. Pastes the contents of the Clipboard as the next available slide of the active slide show sheet, and gives the slide the attributes you specify.

If this function is not available, you must install the Slide Show add-in.

**Syntax**

**SLIDE.PASTE**(effect\_num, speed\_num, advance\_rate\_num, soundfile\_text)

**SLIDE.PASTE**?(effect\_num, speed\_num, advance\_rate\_num, soundfile\_text)

Effect\_num    is a number specifying the transition effect you want to use when displaying the slide.

* The numbers correspond to the effects in the Effect list in the Edit Slide dialog box. The first effect in the list is 1 (None).
* If effect\_num is omitted, the default setting is used.

Speed\_num    is a number from 1 to 10 specifying the speed of the transition effect.

* If speed\_num is omitted, the default setting is used.
* If speed\_num is greater than 10, Microsoft Excel uses the value 10 anyway.
* If effect\_num is 1 (none), speed\_num is ignored.

Advance\_rate\_num    is a number specifying how long (in seconds) the slide is displayed before advancing to the next one.

* If advance\_rate\_num is omitted, the default setting is used.
* If advance\_rate\_num is 0, you must press a key or click with the mouse to advance to the next slide.

Soundfile\_text    is the name of a file enclosed in quotation marks and specifies sound that will be played when the slide is displayed.

* If soundfile\_text is omitted, Microsoft Excel plays the default sound defined for the slide show sheet, if any.
* If soundfile\_text is empty text (""), no sound is played.
* In Microsoft Excel for the Macintosh, soundfile\_text also includes the number or name of the sound resource to play in the file.

Resource    is the number or name of a sound resource in soundfile\_text.

* This argument applies only to Microsoft Excel for the Macintosh.
* If resource is omitted, Microsoft Excel uses the first resource in the file.
* If the file does not contain a sound resource with the specified name or number, Microsoft Excel halts the macro and displays an error message.

**Remarks**

* SLIDE.PASTE returns TRUE if it successfully pastes the slide, or FALSE if you click the Cancel button when using the dialog-box form. If the active sheet is not a slide show or is protected, SLIDE.PASTE returns the #N/A error value. If the Clipboard format is not compatible with the slide show sheet's format, SLIDE.PASTE returns the #VALUE error value.

**Examples**

In Microsoft Excel for Windows, the following macro formula pastes the contents of the Clipboard into the active slide show sheet. The slide's transition effect is fade, at a speed of 8; it is displayed for five seconds; and Microsoft Excel plays the specified sound file:

SLIDE.PASTE(3, 8, 5, "C:\SLIDES\SOUND\MACHINES.WAV")

In Microsoft Excel for the Macintosh, the formula is:

SLIDE.PASTE(3, 8, 5, "HARD DISK:SLIDES:SOUND:MACHINE SOUNDS")

Return to [top](#Q)

# SLIDE.PASTE.ROW

This function should not be used in Microsoft Excel 95 or later because the Slide Show add-in is available only in Microsoft Excel version 5.0 or earlier versions.

Equivalent to clicking the Paste Row button on a slide show sheet. Pastes previously cut or copied slides onto the current selection. For more information, see SLIDE.COPY.ROW.

If this function is not available, you must install the Slide Show add-in.

**Syntax**

**SLIDE.PASTE.ROW**( )

**Related Function**

SLIDE.COPY.ROW   Copies the selected slides and pastes them onto the Clipboard

Return to [top](#Q)

# SLIDE.SHOW

This function should not be used in Microsoft Excel 95 or later because the Slide Show add-in is available only in Microsoft Excel version 5.0 or earlier versions.

Equivalent to clicking the Start Show button on a slide show sheet. Starts the slide show in the active sheet.

If this function is not available, you must install the Slide Show add-in.

**Syntax**

**SLIDE.SHOW**(initialslide\_num, repeat\_logical, dialogtitle\_text, allownav\_logical, allowcontrol\_logical)

**SLIDE.SHOW**?(initialslide\_num, repeat\_logical, dialogtitle\_text, allownav\_logical, allowcontrol\_logical)

All arguments except dialogtitle\_text correspond to options and settings in the Start Show dialog box.

Initialslide\_num    is a number from 1 to the number of slides in the slide show and specifies which slide to display first. If omitted, it is assumed to be 1.

Repeat\_logical    is a logical value specifying whether to repeat or end the slide show after displaying the last slide. If repeat\_logical is TRUE, the slide show repeats; if FALSE or omitted, the slide show ends.

Dialogtitle\_text    is text enclosed in quotation marks that specifies the title of the dialog boxes displayed during the slide show. If dialogtitle\_text is omitted, it is assumed to be "Slide Show".

Allownav\_logical    is a logical value specifying whether to enable or disable navigational keys (arrow keys, PAGE UP, PAGE DOWN, and so on) or the mouse during the slide show. If allownav\_logical is TRUE or omitted, you can press navigational keys or use the mouse to move between slides; if FALSE, all movement is controlled by the slide show sheet settings.

Allowcontrol\_logical    is a logical value specifying whether to enable or disable the Slide Show Options dialog box during the slide show. If allowcontrol\_logical is TRUE or omitted, you can press ESC to interrupt the slide show and display the dialog box; if FALSE, pressing ESC stops the slide show but does not display the dialog box.

**Tip**If you want to display the last slide in a show but don't know its number, use SLIDE.GET(1) as the initialslide\_num argument.

**Remarks**

SLIDE.SHOW returns the values shown in the following table:

|  |  |
| --- | --- |
| **Situation** | **Returned value** |
| The slide show ends normally. | TRUE |
| You press the Cancel button when using the dialog-box form. | FALSE |
| The active sheet is not a slide show or is protected. | #N/A |
| You interrupt the slide show, and then stop it. | 1 |

Return to [top](#Q)

# SOLVER.ADD

Equivalent to clicking the Solver command on the Tools menu and clicking the Add button in the Solver Parameters dialog box. Adds a constraint to the current problem. For an explanation of constraints, see "Remarks" later in this topic.

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

**Syntax**

**SOLVER.ADD**(**cell\_ref, relation**, formula)

Cell\_ref    is a reference to a cell or range of cells on the active sheet and forms the left side of the constraint.

Relation    specifies the arithmetic relationship between the left and right sides, or whether cell\_ref must be an integer.

|  |  |
| --- | --- |
| **Relation** | **Arithmetic relationship** |
| 1 | <= |
| 2 | = |
| 3 | >= |
| 4 | Int (cell\_ref is an integer) |

Formula    is the right side of the constraint and will often be a single number, but it may be a formula (as text) or a reference to a range of cells.

* If relation is 4, cell\_ref must be a subset of the references in the By Changing cells text box.
* if relation is 4, formula must be either "=integer" or "integer".
* Any cell reference in a formula must use the R1C1 reference style.
* If formula is a reference to a range of cells, the number of cells in the range usually matches the number of cells in cell\_ref, although the shape of the areas need not be the same. For example, cell\_ref could be a row and formula could refer to a column, as long as the number of cells is the same. Formula can also be a single reference, as in the following relationship:  A1:A4 <= B1.

**Remarks**

* The SOLVER.ADD, SOLVER.CHANGE, and SOLVER.DELETE functions correspond to the Add, Change, and Delete buttons in the Solver Parameters dialog box. You use these functions to define constraints. For many macro applications, however, you may find it more convenient to load the problem specifications from the sheet in a single step using the SOLVER.LOAD function.
* Each constraint is uniquely identified by the combination of the cell reference on the left and the relationship (<=, =, or >=) between its left and right sides, or the cell reference may be defined as an integer only. This takes the place of selecting the appropriate constraint in the Tools Solver Parameters dialog box. You can manipulate the constraints with SOLVER.CHANGE or SOLVER.DELETE. The constraints in a Solver problem can refer to a maximum of 400 cells.

**Related Function**

SOLVER.DELETE   Deletes an existing constraint

Return to [top](#Q)

# SOLVER.CHANGE

Equivalent to clicking the Solver command on the Tools menu and clicking the Change button in the Solver Parameters dialog box. Changes the right side of an existing constraint.

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

**Syntax**

**SOLVER.CHANGE**(**cell\_ref, relation**, formula)

For an explanation of the arguments and constraints, see SOLVER.ADD.

**Remarks**

* If the combination of cell\_ref and relation does not match any existing constraint, the function returns the value 4 and no action is taken.
* To change the cell\_ref or relation of an existing constraint, use SOLVER.DELETE to delete the old constraint and then use SOLVER.ADD to add the constraint in the form you want.

**Related Functions**

SOLVER.DELETE   Deletes an existing constraint

SOLVER.ADD   Adds a constraint to the current problem

Return to [top](#Q)

# SOLVER.DELETE

Equivalent to clicking the Solver command on the Tools menu and clicking the Delete button in the Solver Parameters dialog box. Deletes an existing constraint.

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

**Syntax**

**SOLVER.DELETE**(**cell\_ref, relation**, formula)

For an explanation of the arguments and constraints, see SOLVER.ADD.

**Remarks**

If the combination of cell\_ref and relation does not match any existing constraint, the function returns the value 4 and no action is taken. If the constraint is found, it is deleted, and the function returns the value 0.

**Related Function**

SOLVER.ADD   Adds a constraint to the current problem

Return to [top](#Q)

# SOLVER.FINISH

Equivalent to clicking OK in the Solver Results dialog box that appears when the solution process is complete. The dialog-box form displays the dialog box with the arguments that you supply as defaults. This function must be used if you supplied the value TRUE for the userfinish argument to SOLVER.SOLVE.

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

**Syntax**

**SOLVER.FINISH**(keep\_final, report\_array)

**SOLVER.FINISH**?(keep\_final, report\_array)

Keep\_final    is the number 1 or 2 and specifies whether to keep the final solution. If keep\_final is 1 or omitted, the final solution values are kept in the changing cells. If keep\_final is 2, the final solution values are discarded and the former values of the changing cells are restored.

Report\_array    is an array argument specifying what reports to create when Solver is finished.

|  |  |
| --- | --- |
| **If report\_array is** | **Microsoft Excel creates** |
| {1} | An answer report |
| {2} | A sensitivity report |
| {3} | A limit report |

Any combination of these produces multiple reports. For example, if report\_array is {1, 2}, Microsoft Excel creates an answer report and a sensitivity report.

**Related Function**

SOLVER.SOLVE   Equivalent to clicking the Solver command on the Tools menu and clicking the Solve button in the Solver Parameters dialog box

Return to [top](#Q)

# SOLVER.GET

Returns information about current settings for Solver. The settings are specified in the Solver Parameters and Solver Options dialog boxes.

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

**Syntax**

**SOLVER.GET**(**type\_num**, sheet\_name)

Type\_num    is a number specifying the type of information you want.

The following settings are specified in the Solver Parameters dialog box.

|  |  |
| --- | --- |
| **Type\_Num** | **Returns** |
| 2 | A number corresponding to the Equal To option 1 = Max 2 = Min 3 = Value of |
| 3 | The value in the Value Of box |
| 4 | The reference (as a multiple reference if necessary) in the By Changing Cells box |
| 5 | The number of constraints |
| 6 | An array of the left sides of the constraints in the form of text |
| 7 | An array of numbers corresponding to the relationships between the left and right sides of the constraints: 1 = <= 2 = = 3 = >= 4 = int |
| 8 | An array of the right sides of the constraints in the form of text |

The following settings are specified in the Solver Options dialog box:

|  |  |
| --- | --- |
| **Type\_Num** | **Returns** |
| 10 | The maximum number of iterations |
| 11 | The precision |
| 12 | The integer tolerance value |
| 13 | TRUE if the Assume Linear Model check box is selected; FALSE otherwise |
| 14 | TRUE if the Show Iteration Results check box is selected; FALSE otherwise |
| 15 | TRUE if the Use Automatic Scaling check box is selected; FALSE otherwise |
| 16 | A number corresponding to the type of estimates: 1 = Tangent 2 = Quadratic |
| 17 | A number corresponding to the type of derivatives: 1 = Forward 2 = Central |
| 18 | A number corresponding to the type of search: 1 = Quasi-Newton 2 = Conjugate Gradient |

Sheet\_name    is the name of a sheet that contains the scenario for which you want information. If sheet\_name is omitted, it is assumed to be the active sheet.

Return to [top](#Q)

# SOLVER.LOAD

Equivalent to clicking the Solver command on the Tools menu, clicking the Options button in the Solver Parameters dialog box, and clicking the Load Model button in the Solver Options dialog box. Loads Solver problem specifications that you have previously saved on the worksheet.

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

**Syntax**

**SOLVER.LOAD**(**load\_area**)

Load\_area    is a reference on the active sheet to a range of cells from which you want to load a complete problem specification.

* The first cell in load\_area contains a formula for the Set Cell box; the second cell contains a formula for the changing cells; subsequent cells contain constraints in the form of logical formulas. The last cell optionally contains an array of Solver option values. The order of the Solver option values is the same as the top-to-bottom order in the Solver Options dialog box.
* Although load\_area must be on the active sheet, it need not be the current selection.

Return to [top](#Q)

# SOLVER.OK

Equivalent to clicking the Solver command on the Tools menu and specifying options in the Solver Parameters dialog box. Specifies basic Solver options, except that constraints are added via SOLVER.ADD.

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

**Syntax**

**SOLVER.OK**(set\_cell, max\_min\_val, value\_of, **by\_changing**)

**SOLVER.OK**?(set\_cell, max\_min\_val, value\_of, by\_changing)

Set\_cell    corresponds to the Set Target Cell box in the Solver Parameters dialog box.

* Set\_cell must be a reference to a cell on the active worksheet.
* If you enter a cell reference, you must also enter a value for max\_min\_val. If you do not enter a cell, you must include three commas before the by\_changing value.

Max\_min\_val    corresponds to the options Max, Min, and Value Of in the Solver Parameters dialog box. Use this option only if you entered a reference for set\_cell.

|  |  |
| --- | --- |
| **Max\_min\_val** | **Option specified** |
| 1 | Maximize |
| 2 | Minimize |
| 3 | Match specific value |

Value\_of    is a number that becomes the target for the cell in the Set Target Cell box if max\_min\_val is 3. Value\_of is ignored if the cell is being maximized or minimized.

By\_changing    indicates the changing cells, as entered in the By Changing Cells box. By\_changing must refer to a cell or range of cells on the active worksheet, and can be a multiple selection.

**Remarks**

The constraints in a Solver problem can refer to a maximum of 400 cells.

**Related Function**

SOLVER.SOLVE   Returns an integer value indicating the condition that caused Solver to stop

Return to [top](#Q)

# SOLVER.OPTIONS

Equivalent to clicking the Solver command on the Tools menu and then clicking the Options button in the Solver Parameters dialog box. Specifies the available options.

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

**Syntax**

**SOLVER.OPTIONS**(max\_time, iterations, precision, assume\_linear, step\_thru, estimates, derivatives, search, int\_tolerance, scaling)

The arguments correspond to the options in the dialog box. If an argument is omitted, Microsoft Excel uses an appropriate value based on the current situation. If any of the arguments are the wrong type, the function returns the #N/A error value. If an argument has the correct type but an invalid value, the function returns a positive integer corresponding to its position. A zero indicates all options were accepted.

Max\_time    must be an integer greater than zero and less than 32768. It corresponds to the Max Time box.

Iterations    must be an integer greater than zero and less than 32768. It corresponds to the Iterations box.

Precision    must be a number between zero and one, but not equal to zero or one. It corresponds to the Precision box.

Assume\_linear    is a logical value corresponding to the Assume Linear Model check box and allows Solver to arrive at a solution more quickly. If TRUE, Solver assumes that the underlying model is linear; if FALSE, it does not.

Step\_thru    is a logical value corresponding to the Show Iteration Results check box. If you have supplied SOLVER.SOLVE with a valid command macro reference, your macro will be called each time Solver pauses. If TRUE, Solver pauses at each trial solution; if FALSE, it does not.

Estimates    is the number 1 or 2 and corresponds to the Estimates options: 1 for the Tangent option and 2 for the Quadratic option.

Derivatives    is the number 1 or 2 and corresponds to the Derivatives options: 1 for the Forward option and 2 for the Central option.

Search    is the number 1 or 2 and corresponds to the Search options: 1 for the Quasi-Newton option and 2 for the Conjugate Gradient option.

Int\_tolerance    is a decimal number corresponding to the Tolerance box in the Solver Options dialog box, and must be between zero and 1, inclusively. This argument applies only if integer constraints have been defined.

Scaling    is a logical value corresponding to the Use Automatic Scaling check box. If scaling is TRUE, then if two or more constraints differ by several orders of magnitude, Solver scales the constraints to similar orders of magnitude during computation. If scaling is FALSE, Solver calculates normally.

Return to [top](#Q)

# SOLVER.RESET

Equivalent to clicking the Solver command on the Tools menu and clicking the Reset All button in the Solver Parameters dialog box. Erases all cell selections and constraints from the Solver Parameters dialog box and restores all the settings in the Solver Options dialog box to their defaults.

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

**Syntax**

**SOLVER.RESET**( )

Return to [top](#Q)

# SOLVER.SAVE

Equivalent to clicking the Solver command on the Tools menu, clicking the Options button in the Solver Parameters dialog box, and clicking the Save Model button in the Solver Options dialog box. Saves the Solver problem specifications on the worksheet.

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

**Syntax**

**SOLVER.SAVE** (**save\_area**)

Save\_area    is a reference on the active sheet to a range of cells or to the upper-left corner of a range of cells into which you want to paste the current problem specification.

* If you specify only one cell for save\_area, the area is extended downwards for as many cells as are required to hold the problem specifications (3 plus the number of constraints).
* If you specify more than one cell and if the area is too small, the last constraints (in alphabetic order by cell reference) or options will be omitted and the function will return a nonzero value.
* Save\_area must be on the active worksheet, but it need not be the current selection.

Return to [top](#Q)

# SOLVER.SOLVE

Equivalent to clicking the Solver command on the Tools menu and clicking the Solve button in the Solver Parameters dialog box. If successful, returns an integer value indicating the condition that caused Solver to stop as described in "Remarks" later in this topic.

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

**Syntax**

**SOLVER.SOLVE**(user\_finish, show\_ref)

User\_finish    is a logical value specifying whether to display the Solver Results dialog box.

* If user\_finish is TRUE, SOLVER.SOLVE returns its integer value without displaying anything. Your macro should decide what action to take (for example, by examining the return value or presenting its own dialog box); it must call SOLVER.FINISH in any case to restore the sheet to its proper state.
* If user\_finish is FALSE or omitted, Solver displays the Solver Results dialog box, which allows you to keep or discard the final solution and run reports.

Show\_ref    is a macro to be called in place of the Show Trial Solution dialog box. It is used when you want to regain control whenever Solver finds a new intermediate solution value.

* For this argument to have an effect, the Show Iteration Results check box must be selected in the Solver Options dialog box. This can be done manually by selecting the check box, or automatically by calling SOLVER.OPTIONS in your macro.
* The macro you call can inspect the current solution values on the sheet or take other actions such as saving or charting the intermediate values. It must return the value TRUE with a statement such as =RETURN(TRUE) if the solution process is to continue, or FALSE if the solution process should stop at this point.

**Remarks**

If a problem has not been completely defined, SOLVER.SOLVE returns the #N/A error value. Otherwise, the Solver application is started and the problem specifications are passed to it. When the solution process is complete, SOLVER.SOLVE returns an integer value indicating the stopping condition:

|  |  |
| --- | --- |
| **Value** | **Stopping condition** |
| 0 | Solver found a solution. All constraints and optimality conditions are satisfied. |
| 1 | Solver has converged to the current solution. All constraints are satisfied. |
| 2 | Solver cannot improve the current solution. All constraints are satisfied. |
| 3 | Stop chosen when the maximum iteration limit was reached. |
| 4 | The Set Cells values do not converge. |
| 5 | Solver could not find a feasible solution. |
| 6 | Solver stopped at user's request. |
| 7 | The conditions for Assume Linear Model are not satisfied. |
| 8 | The problem is too large for Solver to solve. |
| 9 | Solver encountered an error value in a target or constraint cell. |
| 10 | Stop chosen when the maximum time limit was reached. |
| 11 | There is not enough memory available to solve the problem. |
| 12 | Another Excel instance is using SOLVER.DLL. Try again later. |
| 13 | Error in model. Please verify that all cells and constraints are valid. |

**Related Function**

SOLVER.FINISH   Equivalent to clicking OK in the Solver Results dialog box that appears when the solution process is complete

Return to [top](#Q)

# SORT

Equivalent to clicking the Sort command on the Data menu. Sorts the rows or columns of the selection according to the contents of a key row or column within the selection. Use SORT to rearrange information into ascending or descending order.

**Syntax 1**

For Worksheet and macro sheets

**SORT**(orientation, key1, order1, key2, order2, key3, order3, header, custom, case)

**SORT**?(orientation, key1, order1, key2, order2, key3, order3, header, custom, case)

**Syntax 2**

For PivotTable reports

**SORT**(orientation, key1, order1, type, custom)

**SORT**?(orientation, key1, order1, type, custom)

Orientation    is a number specifying whether to sort by rows or columns. Enter 1 to sort top to bottom or 2 to sort left to right.

Key1    is a reference to the cell or cells you want to use as the first sort key. The sort key identifies which column to sort by when sorting rows or which row to sort by when sorting columns. For a PivotTable report, if type is 1, then key1 is a cell reference which indicates what value to sort by. There are two ways to specify sort keys:

|  |  |
| --- | --- |
| **Type of key** | **Examples** |
| An R1C1-style reference in the form of text. If the reference is relative, it is assumed to be relative to the active cell in the selection. | "C2" or "C[1]" or "Price" |

Order1    specifies whether to sort the row or column containing key1 in ascending or descending order. Enter 1 to sort in ascending order or 2 to sort in descending order.

Key2, order2, key3, and order3    are similar to key1 and order1. Key2 specifies the second sort key, and order2 specifies whether to sort the row or column containing key2 in ascending or descending order. Key3 and order3 work similarly.

Header    is a number indicating how Microsoft Excel is to handle headers on list.

|  |  |
| --- | --- |
| **Header** | **Defined** |
| 0 | Microsoft Excel will guess if there is a header |
| 1 | Forces Microsoft Excel to assume there is a header |
| 2 or omitted | Forces Microsoft Excel to assume there is no header |

Type    is a number specifying whether to sort the field by labels or values. Use one to sort by values or two to sort by labels.

Custom    is a number that specifies what kind of custom sorting you want. This corresponds to the First Key Sort Order drop-down box in the Sort Options dialog box. For a PivotTable report, custom is a number indicating what custom sort order to use when sorting labels.

|  |  |
| --- | --- |
| **Number** | **Type of sort** |
| 1 | Normal |
| 2 | Weekdays in abbreviated form ("Sun", "Mon", and so on) |
| 3 | Weekdays |
| 4 | Months in abbreviated form ("Jan" "Feb", and so on) |
| 5 | Months |

Case    is a logical value that determines whether the sort is case sensitive. If TRUE, the sort is case sensitive. If FALSE or omitted, the sort will not be case sensitive.

**Tip**If you want to sort using more than three keys, then sort the data three keys at a time, starting with the least important group of keys and progressing to the most important group, but listing the most important key first within each group.

**Remarks**

In the dialog box form of this function, if the header argument is omitted, then Microsoft Excel will guess whether or not there are headers.

Return to [top](#Q)

# SOUND.NOTE

This function should not be used in Microsoft Excel 97 or later because sound notes are available only in Microsoft Excel 95 or earlier versions.

Records sound into or erases sound from a cell note or imports sound from another file into a cell note. This function requires that you have recording hardware installed in your computer, and you must be running Microsoft Windows version 3.1 or later, or Apple system software version 6.07 or later.

**Syntax 1**

Recording or erasing sound

**SOUND.NOTE**(cell\_ref, erase\_snd)

**Syntax 2**

Importing sound from another file

**SOUND.NOTE**(cell\_ref, file\_text, resource)

Cell\_ref    is a reference to the cell containing a note into which you want to record or import sounds or from which you want to erase a sound.

Erase\_snd    is a logical value specifying whether to erase the sound in the note. If erase\_snd is TRUE, Microsoft Excel erases only the sound from the note. If FALSE or omitted, Microsoft Excel displays the Record dialog box so that you can record sound into the note.

File\_text    is the name of a file containing sounds.

Resource    is the number or name of a sound resource in file\_text that you want to import into your note.

* This argument applies only to Microsoft Excel for the Macintosh.
* If resource is omitted, Microsoft Excel uses the first resource in the file.
* If the file does not contain a sound resource with the specified name or number, Microsoft Excel halts the macro and displays an error message.

**Remarks**

* To find out if a cell has sound attached to it, use GET.CELL(47).
* Sounds notes are not available in Microsoft Excel 97 or later.

**Examples**

The following macro formula erases the sound, if present, from cell A1 on the active sheet:

SOUND.NOTE(!$A$1, TRUE)

The following macro formula displays the Record dialog box so that you can record sound into a note for cell A1 on the active sheet:

SOUND.NOTE(!$A$1)

In Microsoft Excel for Windows, the following macro formula imports the sound from a file named CHIMES.WAV into a note for the cell named Doorbell on the active sheet:

SOUND.NOTE(!Doorbell, "C:\SOUNDS\CHIMES.WAV")

In Microsoft Excel for the Macintosh, the following macro formula imports a sound called Chimes from a file named SOFT SOUNDS into a note for the cell named Doorbell on the active sheet:

SOUND.NOTE(!Doorbell, "HARD DISK:SOUNDS:SOFT SOUNDS", "Chimes")

**Related Functions**

NOTE   Creates or changes a cell note

SOUND.PLAY   Plays the sound from a cell note or a file

Return to [top](#Q)

# SOUND.PLAY

This function should not be used in Microsoft Excel 97 or later because sound notes are available only in Microsoft Excel 95 or earlier versions.

Plays the sound from a cell note or a file. Equivalent to clicking the Note command on the Insert menu and clicking the Play button, or clicking the Note command on the Insert menu, clicking the Import button, and then opening a file, selecting a sound, and clicking the Play button. To play sounds in Microsoft Excel for Windows, you must have a sound board installed in your computer.

**Syntax**

**SOUND.PLAY**(cell\_ref, file\_text, resource)

Cell\_ref    is a reference to the cell note containing sound that you want to play. If cell\_ref is omitted, Microsoft Excel plays the sound from the active cell, or from a file if you specify one.

File\_text    is the name of a file containing sounds. If cell\_ref is specified, file\_text is ignored.

Resource    is a number or name given as text specifying a sound resource in file\_text that you want to play.

* This argument applies only to Microsoft Excel for the Macintosh.
* If cell\_ref is specified, resource is ignored.
* If resource is omitted, Microsoft Excel uses the first sound resource in the file.
* If the file does not contain a sound resource with the specified name or number, Microsoft Excel halts the macro and displays an error message.

**Related Function**

SOUND.NOTE   Records or imports sound into or erases sound from cell notes

Return to [top](#Q)

# SPELLING

Equivalent to clicking the Spelling command on the Tools menu. Checks the spelling of words in the current selection.

**Syntax**

**SPELLING**(custom\_dic, ignore\_uppercase, always\_suggest)

Custom\_dic    is the filename of the custom dictionary to examine if words are not found in the main dictionary. If custom\_dic is omitted, the currently specified dictionary is used.

Ignore\_uppercase    is a logical value corresponding to the Ignore UPPERCASE check box.

|  |  |
| --- | --- |
| **If ignore\_uppercase is** | **Microsoft Excel will** |
| TRUE | Ignore words in all uppercase letters |
| FALSE | Check words in all uppercase letters |
| Omitted | Use the current setting |

Always\_suggest    is a logical value corresponding to the Always Suggest check box.

|  |  |
| --- | --- |
| **If always\_suggest is** | **Microsoft Excel will** |
| TRUE | Display a list of suggested alternate spellings when an incorrect spelling is found |
| FALSE | Wait for user to input the correct spelling |
| Omitted | Use the current setting |

**Related Function**

SPELLING.CHECK   Checks the spelling of a word

Return to [top](#Q)

# SPELLING.CHECK

Checks the spelling of a word. Returns TRUE if the word is spelled correctly; FALSE otherwise.

**Syntax**

**SPELLING.CHECK**(**word\_text**, custom\_dic, ignore\_uppercase)

Word\_text    is the word whose spelling you want to check. It can be text or a reference to text.

Custom\_dic    is the filename of a custom dictionary to examine if the word is not found in the main dictionary.

Ignore\_uppercase    is a logical value corresponding to the Ignore Words In Uppercase check box. If ignore\_uppercase is TRUE, the check box is selected, and Microsoft Excel ignores words in all uppercase letters; if FALSE, the check box is cleared, and Microsoft Excel checks all words; if omitted, the current setting is used.

**Remarks**

This function does not have a dialog-box form. To display the Spelling dialog box, use SPELLING.

**Related Function**

SPELLING   Checks the spelling of words in the current selection

Return to [top](#Q)

# SPLIT

Equivalent to choosing the Split command from the Window menu or to dragging the split bar in the active window's scroll bar. Splits the active window into panes. Use SPLIT when you want to view different parts of the active sheet at the same time.

**Syntax**

**SPLIT**(col\_split, row\_split)

Col\_split    specifies where to split the window vertically and is measured in columns from the left of the window.

Row\_split    specifies where to split the window horizontally and is measured in rows from the top of the window.

If an argument is 0 and there is a split in that direction, the split is removed. If an argument is omitted, a split in that direction is not changed.

**Related Function**

FREEZE.PANES   Freezes or unfreezes the panes of a window

Return to [top](#Q)

# SQL.BIND

Specifies where results from a SQL query are placed when they are retrieved with SQL.RETRIEVE. If this function is not available, you must install the Microsoft ODBC add-in (XLODBC.XLA).

**Syntax**

**SQL.BIND**(**connection\_num**, column, reference)

Connection\_num    is the unique connection ID of the data source for which you want to define storage.

* Connection\_num was returned by a previously executed SQL.OPEN function.
* If connection\_num is not valid, then SQL.BIND returns the #VALUE! error value.

Column    is the number of the result column that is to be bound. Result columns are numbered from left to right starting with 1. If column is omitted then all bindings for connection\_num are removed. Column number 0 contains row numbers for the result set. If column number 0 is bound then SQL.RETRIEVE will return row numbers in the bound location.

Reference     is a single cell reference on the worksheet where the results should be placed. If reference is omitted, the binding is removed for the column.

When SQL.RETRIEVE is called, the result rows in this column will be placed in the reference cell and the cells immediately below reference. The number of rows that will be retrieved is one of the SQL.RETRIEVE arguments.

**Remarks**

* If SQL.BIND is completed successfully then it will return a vertical array listing the bound columns on the current connection. If SQL.BIND is unable to bind the result column then it will return the error value #N/A. In such a case SQL.BIND will place error information in memory for the SQL.ERROR function, if such information is available.
* SQL.BIND tells the ODBC interface where to place results when they are retrieved using SQL.RETRIEVE. Binding is not necessary but can be useful if you want the results from different columns to be placed in disjoint worksheet locations.
* If bindings are used, SQL.BIND must be called once for each column in the result set. If a result column is not bound then it will not be returned. A binding remains valid for as long as connection\_num is open.
* Call SQL.BIND after calling SQL.OPEN and SQL.EXEC.QUERY, but before calling SQL.RETRIEVE or SQL.RETRIEVE.TO.FILE. Calls to SQL.BIND will not affect results that have already been retrieved.

**Example**

SQL.BIND(conn1,1,"[Book1]Sheet1!C1") stores data obtained from the data source conn1 on Sheet1 from left to right in cell C1, starting with column1.

**Related Functions**

SQL.OPEN   Establishes a connection with a data source

SQL.EXEC.QUERY   Sends a query to a data source

SQL.RETRIEVE.TO.FILE   Retrieves query results and places them in a file

SQL.RETRIEVE   Retrieves query results

SQL.GET.SCHEMA   Gets information about a connected data source.

SQL.CLOSE   Closes a connection to a data source.

SQL.ERROR   Returns detailed error information

Return to [top](#Q)

# SQL.CLOSE

Terminates a connection to an external data source. If this function is not available, you must install the Microsoft ODBC add-in (XLODBC.XLA).

**Syntax**

**SQL.CLOSE**(**connection\_num**)

Connection\_num    is the unique connection ID of the data source from which you wish to disconnect.

* Connection\_num is returned by a previously executed SQL.OPEN function.
* If connection\_num is not valid, SQL.CLOSE returns the #VALUE! error value.

**Remarks**

* If the connection is successfully terminated SQL.CLOSE will return zero and the connection ID number is then no longer valid.
* If SQL.CLOSE is unable to disconnect with the data source then it will return the error value the #N/A error value. In such a case SQL.CLOSE will place error information in memory for the SQL.ERROR function, if such information is available.
* SQL.CLOSE works with data sources in much the same manner as FCLOSE works with files. If the call is successful then SQL.CLOSE will terminate the specified data source connection.

**Example**

SQL.CLOSE(conn1) will close the connection with connection\_num conn1

**Related Functions**

SQL.OPEN   Establishes a connection with a data source

SQL.EXEC.QUERY   Sends a query to a data source

SQL.BIND   Specifies storage for a result column

SQL.RETRIEVE.TO.FILE   Retrieves query results and places them in a file

SQL.RETRIEVE   Retrieves query results

SQL.GET.SCHEMA   Gets information about a connected data source.

SQL.ERROR   Returns detailed error information

Return to [top](#Q)

# SQL.ERROR

Returns detailed error information when it is called after a previous XLODBC.XLA function call has failed. If this function is not available, you must install the Microsoft ODBC add-in (XLODBC.XLA).

**Syntax**

**SQL.ERROR**()

Calling SQL.ERROR returns detailed error information in a two dimensional array. Each row in the array describes exactly one error. If a function call generates multiple errors, a row will be created for each error. When SQL.ERROR is processed successfully, all SQL.ERROR information is cleared. Also, all SQL.ERROR information is automatically removed whenever an ODBC function completes successfully.

Each row will have exactly three fields. The information in these three fields is obtained through the SQLERROR API function call. These fields are:

* A textual message describing the error.
* The ODBC error class and subclass as a character string.
* The data source native error code as a numeric value.

If one or more of these fields is not available for the type of error that was encountered, the field will be left blank. For more information on the meaning of these three fields, refer to Chapter 24, "ODBC Function Reference", in the Microsoft Open Database Connectivity Programmer's Reference for the SQLError API function. See also Appendix A, "ODBC Error Codes" in the same manual.

**Remarks**

* SQL.ERROR cannot provide information on Excel errors.
* If no error information is available when SQL.ERROR is called, then it well return the error value #N/A but does not post any error information to SQL.ERROR.
* SQL.ERROR stores and returns error information by processing SQL.ERROR (in the ODBC API reference) in a loop until SQL\_NO\_DATA\_FOUND is encountered. In the SQL.ERROR function, the error information is automatically defined and stored in memory whenever an XLODBC.XLA function fails. If the call is successful then SQL.ERROR will return the error information available. If SQL.ERROR fails, it will return the error value #N/A.

**Example**

When entered as an array formula, the following example will return error information about each argument in, for example, SQL.GET.QUERY.

SQL.ERROR()

**Related Functions**

SQL.OPEN   Establishes a connection with a data source

SQL.EXEC.QUERY   Sends a query to a data source

SQL.BIND   Specifies storage for a result column

SQL.RETRIEVE.TO.FILE   Retrieves query results and places them in a file

SQL.RETRIEVE   Retrieves query results

SQL.CLOSE   Closes a data source connection

Return to [top](#Q)

# SQL.EXEC.QUERY

Sends a query to a data source using an existing connection. If this function is not available, you must install the Microsoft ODBC add-in (XLODBC.XLA).

**Syntax**

**SQL.EXEC.QUERY**(**connection\_num**, **query\_text**)

Connection\_num    is the unique connection ID of the data source you want to query.

* Connection\_num is returned by a previously executed SQL.OPEN function.
* If connection\_num is not valid, SQL.EXEC.QUERY returns the #VALUE! error value.

Query\_text is the SQL language query that is to be executed on the data source. The query must follow the SQL syntax guidelines in the Appendix of the Microsoft Excel ODBC Developers Guide.

* If SQL.EXEC.QUERY is unable to execute query\_text on the specified data source, SQL.EXEC.QUERY returns the #N/A error value.
* Excel limits strings to a length of 255 characters. If query\_text needs to be longer than 255 characters then query\_text should be a vertical array or vertical range of cells. The values in the array will be joined together to form the complete SQL query.

**Remarks**

* Before calling SQL.EXEC.QUERY a connection must be established with a data source using SQL.OPEN. A successful call to SQL.OPEN returns a unique connection ID number. SQL.EXEC.QUERY uses that connection ID number to send SQL language queries to the data source.
* Any results generated from the query will not be returned immediately-- SQL.EXEC.QUERY only executes the query. Retrieving results is handled by the functions SQL.RETRIEVE and SQL.RETRIEVE.TO.FILE.
* If SQL.EXEC.QUERY is called using a previously used connection ID number, all pending results on that connection will automatically be discarded. The connection ID will then refer to the new query and its results.
* If SQL.EXEC.QUERY is unable to successfully execute the query on the specified data source then an error value will be returned. In such a case SQL.EXEC.QUERY will place error information in memory for the SQL.ERROR function, if such information is available. If SQL.EXEC.QUERY is able to successfully execute the query on the specified connection it will return one of three values depending on the type of SQL statement that was executed.
  + If it was a SELECT statement then SQL.EXEC.QUERY will return the number of result columns available.
  + If it was an UPDATE, INSERT, or DELETE statement then SQL.EXEC.QUERY will return the number of rows affected by the statement.
  + If it was a legal SQL query that is not in one of the categories above, SQL.EXEC.QUERY will return 0 (zero).

**Example**

SQL.EXEC.QUERY(conn1, "SELECT Custmr\_ID, Due\_Date FROM Orders WHERE Order\_Amt > 100") executes a SQL query from a SQL table named "Orders"

**Related Functions**

SQL.OPEN   Establishes a connection with a data source

SQL.BIND   Specifies storage for a result column

SQL.RETRIEVE.TO.FILE   Retrieves query results and places them in a file

SQL.RETRIEVE   Retrieves query results

SQL.GET.SCHEMA   Gets information about a connected data source.

SQL.CLOSE   Closes a data source connection

SQL.ERROR   Returns detailed error information

Return to [top](#Q)

# SQL.GET.SCHEMA

Returns information about the structure of the data source on a particular connection. The return value from a successful call to SQL.GET.SCHEMA depends on the type of information that was requested. A list of the accepted requests and their respective return values is listed in the syntax section below.

If this function is not available, you must install the Microsoft ODBC add-in (XLODBC.XLA).

**Syntax**

**SQL.GET.SCHEMA**(**connection\_num**, **type\_num**, qualifier\_text)

Connection\_num    is the unique connection ID of the data source you want information about.

* Connection\_num is returned by a previously executed SQL.OPEN function.
* If connection\_num is not valid, SQL.GET.SCHEMA returns the #VALUE! error value.

Type\_num specifies the type of information you want returned. The following is a list of valid type\_num values.

|  |  |
| --- | --- |
| **Type\_num** | **Returns** |
| 1 | A list of available data sources, as a vertical array. |
| 2 | A list of databases on the current connection, as a vertical array . |
| 3 | A list of owners in a database on the current connection, as a vertical array. |
| 4 | A list of tables for a given owner and database on the current connection, as a vertical array. |
| 5 | A list of columns in a particular table and their data types, as a two-dimensional array. The returned array will have two fields and will have a row for each column in the table. The first field will be the name of the column. The second field is the data type of the column. The data type will be a number that corresponds to the ODBC C header file data types. These #define numbers are found in Microsoft Excel ODBC Developer Guide. |
| 6 | User ID of the current user |
| 7 | Name of the current database. |
| 8 | The name of the data source as given in the ODBC.INI file. |
| 9 | The name of the data source DBMS (i.e. Oracle, SQL Server, etc.). |
| 10 | The server name for the data source. |
| 11 | The terminology used by the data source to refer to owners ( i.e. "owner", "Authorization ID", "Schema", etc.). |
| 12 | The terminology used by the data source to refer to tables ( i.e. "table", "file", etc.). |
| 13 | The terminology used by the data source to refer to qualifiers (i.e. "database" or "directory"). |
| 14 | The terminology used by the data source to refer to procedures (i.e. "database procedure", "stored procedure", or "procedure"). |

Qualifier\_text    is only included for type\_num values of 3, 4 and 5. It is a text string used to qualify the search for the requested information and should be enclosed by quotation marks.

|  |  |
| --- | --- |
| **Type\_num** | **Qualifier\_text** |
| 3 | The value of qualifier\_text should be the name of a database in the current data source. SQL.GET.SCHEMA will then only return the names of table owners in that database. |
| 4 | The value of qualifier\_text should be both a database name and an owner name. The syntax of qualifier\_text is "DatabaseName.OwnerName". A period is used to separate the two names. SQL.GET.SCHEMA will then return an array of table names that are located in the given database and owned by the given owner. |
| 5 | The value of qualifier\_text should be the name of a table. Information about the columns in that table will be returned. |

**Remarks**

* If SQL.GET.SCHEMA is unable to find the requested information then it will return the error value #N/A. In such a case SQL.GET.SCHEMA will place error information in memory for the SQL.ERROR function, if such information is available.
* SQL.GET.SCHEMA works with the ODBC functions SQLGetInfo and SQLTables to find the requested information. Refer to the Microsoft Excel ODBC Programmer Guide for more information on these two functions.

**Example**

SQL.GET.SCHEMA(conn1,7) returns the name of the current database.

SQL.GET.SCHEMA(conn1,9) returns the name of the DBMS used by the data source.

**Related Functions**

SQL.OPEN   Establishes a connection with a data source

SQL.EXEC.QUERY   Sends a query to a data source

SQL.BIND   Specifies storage for a result column

SQL.RETRIEVE.TO.FILE   Retrieves query results and places them in a file

SQL.RETRIEVE   Retrieves query results

SQL.CLOSE   Closes a data source connection

SQL.ERROR   Returns detailed error information

Return to [top](#Q)

# SQL.OPEN

Establishes a connection with a data source. If the connection is successfully established SQL.OPEN will return a connection ID number. Use the connection ID number with other ODBC macro functions to identify a connection.

If this function is not available, you must install the Microsoft ODBC add-in (XLODBC.XLA).

**Syntax**

**SQL.OPEN**(**connection\_string**, output\_ref, driver\_prompt)

Connection\_string    is a text string that contains the information necessary to establish a connection to a data source. Any data-source-name that is used in connection\_string must be an existing data source name defined with ODBC Setup or the ODBC Administration Utility.

* Connection\_string must follow the format described in Chapter 24, "ODBC Function Reference", of the Microsoft Open Database Connectivity Programmer's Reference for SQLDriverConnect. In this string the user supplies the data source name, one or more user ID's, one or more passwords, and any other information necessary to successfully connect to a DBMS. An example of a SQL.OPEN connection\_string entered would be: "DSN=MyServer; UID=dbayer; PWD=123; Database=pubs"
* Enter the connection\_string as an array when the length exceeds 255 characters. Or enter connection\_string as an array of cells containing the same information. The connection string should be horizontal array.

Output\_ref    is a single cell reference where you want the completed connection string to be placed. Use output\_ref when you want SQL.OPEN to return the completed connection string. If output\_ref is omitted, a completed connection string will not be returned.

Driver\_prompt     is a number from 1 to 4 specifying if and how you want to be prompted by the driver. This sets the fDriverCompletion flag in ODBC's SQLDriverConnect.

|  |  |
| --- | --- |
| **Number** | **Description** |
| 1 | Always brings up a dialog box. |
| 2 | Bring up dialog only if there is not enough information to connect. The driver uses information from the connection string and from the data source specification as defaults. |
| 3 | Same as 2, but the driver grays and disables any prompts for information not needed. |
| 4 | If the connection string is unsuccessful, do not bring up a dialog box. |

**Remarks**

* If SQL.OPEN is unable to connect with the information provided then it will return the error value #N/A. In such a case, SQL.OPEN will place error information in memory for the SQL.ERROR function, if more information is available.
* If the call is successful then SQL.OPEN will return a unique connection ID number that can be used in future function calls to identify the connection.
* If connection\_array does not allow SQL.OPEN to connect to a data source, then the error value #N/A will be returned.

**Example**

conn1=SQL.OPEN('DSN=NWind;DBQ=C:\MSQUERY;FIL=dBASE4',C15, 2) sets the name conn1 to the return value of SQL.OPEN, which connects to the NWind data source, specifies where to place the connection string, and displays the driver dialog box only if additional information is needed.

**Related Functions**

SQL.EXEC.QUERY   Sends a query to a data source

SQL.BIND   Specifies storage for a result column

SQL.RETRIEVE.TO.FILE   Retrieves query results and places them in a file

SQL.RETRIEVE   Retrieves query results

SQL.GET.SCHEMA   Gets information about a connected data source.

SQL.CLOSE   Closes a data source connection

SQL.ERROR   Returns detailed error information

Return to [top](#Q)

# SQL.RETRIEVE

Retrieves all or part of the results from a previously executed query. The connection used must have already been established using the macro function SQL.OPEN. Also, a query must already have been executed using SQL.EXEC.QUERY and results must be pending.

If this function is not available, you must install the Microsoft ODBC add-in (XLODBC.XLA).

**Syntax**

**SQL.RETRIEVE**(**connection\_num**, destination\_ref, max\_columns, max\_rows, col\_names\_logical, row\_nums\_logical, named\_rng\_logical, fetch\_first\_logical)

Connection\_num    is the unique connection ID for a data source. The data source specified must have pending query results. Pending query results are generated by a call to SQL.EXEC.QUERY on the same connection.

* If there are no pending results on the connection SQL.RETRIEVE returns the #N/A error value.
* If connection\_num is not valid, SQL.EXEC.QUERY returns the #VALUE! error value.

Destination\_ref    specifies where the results should be placed. It is either a reference to a single cell or it is omitted.

* If destination\_ref refers to a single cell then SQL.RETRIEVE will return all of the pending results in the cells to the right, below, and including destination\_ref. This is the same convention used in Microsoft Excel when multiple cells are pasted into a single-cell selection. Any previous values contained in the destination cells will be overwritten without confirmation.
* If destination\_ref is omitted then the bindings established by previous calls to SQL.BIND will be used to return results. If no such bindings exist for the current connection then SQL.RETRIEVE will return the #REF! error value. If a particular result column has not been bound then its results will be discarded. Max\_rows specifies the number of rows that will be returned under each bound column. The first row of results will be placed in the bound cell and any additional rows will be placed in the rows immediately under the bound cell.

Max\_columns    is the maximum number of columns to be retrieved. It is only used when destination\_ref is not omitted.

* If max\_columns specifies more columns than are available in the results, SQL.RETRIEVE will place data in the columns for which data is available and clear the additional columns.
* If max\_columns specifies fewer columns than are available in the results, the rightmost result columns will be discarded to fit the chosen size. Column position will be determined by the order in which the data source returned them.
* If max\_columns is omitted then all of the result columns will be returned.

Max\_rows    is the maximum number of rows to be returned.

* If max\_rows specifies more rows than are available in the results, SQL.RETRIEVE will place data in the rows for which data is available and clear the additional rows.
* If max\_rows specifies fewer rows than are available in the results, SQL.RETRIEVE will place data in the selected rows but will not discard the additional rows. These extra rows can be retrieved via additional calls to SQL.RETRIEVE. This process is described in the fetch\_first\_logical argument description.
* If max\_rows is omitted then all rows in the result set will be returned.

Col\_names\_logical    is a logical value which, if TRUE, causes the column names to be returned as the first row of results. It FALSE or omitted, the column names will not be returned.

Row\_nums\_logical    is used only when destination\_ref is included. If row\_nums\_logical is TRUE then the first column in the result set will contain row numbers. If FALSE then row numbers will not be returned. This column of row numbers will not have a column name and the column heading will be left blank. Row numbers can also be retrieved by binding column number 0 with SQL.BIND.

Named\_rng\_logical    is a logical value which, if TRUE, sets each column of results to be declared as a named range on the worksheet. The name of the each range will be the result column name. The named range will only include the rows that were fetched with this SQL.RETRIEVE function call. If FALSE, the results will not be declared as a named range.

Fetch\_first\_logical    is a logical value that allows you to request results from the beginning of the result set.

* If the first call to SQL.RETRIEVE did not return all of the rows in the result set then SQL.RETRIEVE may be called again to return the next set of rows. This process can be repeated until no more result rows are available, at which time SQL.RETRIEVE will return the value 0 (zero). This will not halt the running of the macro. During each of these calls, including the first call, fetch\_first\_logical should be set to FALSE.
* If you want to move the cursor back to the beginning of the result set then fetch\_first\_logical should be set to TRUE. This causes the same SQL query text to be executed again on the data source. The cursor will then be positioned at the top of the result set and SQL.RETRIEVE will fill destination\_ref beginning with the first row of results. Further calls to SQL.RETRIEVE, for the purpose of retrieving additional rows, can then be made with fetch\_first\_logical set to FALSE .

**Remarks**

* Before calling SQL.RETRIEVE a connection must be established with a data source using SQL.OPEN.
* If SQL.RETRIEVE is unable to retrieve the results on the specified data source then an error value will be returned. In such a case SQL.RETRIEVE will place error information in memory for the SQL.ERROR function, if such information is available.
* If SQL.RETRIEVE is able to successfully return rows of results on the specified connection it will return the number of rows that were actually returned. If there were no results pending on the connection then SQL.RETRIEVE will return the #N/A error value.If no data was found then SQL.RETRIEVE returns 0 (zero).
* A successful call to SQL.OPEN returns a unique connection ID number, which is used in a call to SQL.EXEC.QUERY to send a SQL language query. Following this call to SQL.EXEC.QUERY, SQL.RETRIEVE uses the same connection ID number to retrieve query results.

**Example**

SQL.RETRIEVE(conn1,sheet1!C1,1) stores data obtained from the data source conn1 on Sheet1 from left to right in cell C1, using only column 1.

**Related Functions**

SQL.OPEN   Establishes a connection with a data source

SQL.EXEC.QUERY   Sends a query to a data source

SQL.BIND   Specifies storage for a result column

SQL.RETRIEVE.TO.FILE   Retrieves query results and places them in a file

SQL.GET.SCHEMA   Gets information about a connected data source.

SQL.CLOSE   Close a data source connection

SQL.ERROR   Returns detailed error information

Return to [top](#Q)

# SQL.RETRIEVE.TO.FILE

Retrieves all of the results from a previously executed query and places them in a file. The connection used must have already been established using the macro function SQL.OPEN. Also, a query must already have been executed using SQL.EXEC.QUERY and results must be pending.

If this function is not available, you must install the Microsoft ODBC add-in (XLODBC.XLA).

**Syntax**

**SQL.RETRIEVE.TO.FILE**(**connection\_num**, **destination**, col\_names\_logical, column\_delimiter)

Connection\_num    is the unique connection ID for a data source. The data source specified must have query results pending. Pending results were generated by a previous call to SQL.EXEC.QUERY on the same connection.

* If there are no pending results on the connection SQL.RETRIEVE.TO.FILE returns the #N/A error value. The file is not affected.
* If connection\_num is not valid, SQL.RETRIEVE.TO.FILE returns the #VALUE! error value.

Destination    specifies the name and path of the file where the results should be placed. SQL.RETRIEVE.TO.FILE will open the specified file and fill it with the entire result set.

* The format of the data in the file will be compatible with the Microsoft Excel ".CSV" format. The overall format will be that columns will be separated by the value in column\_delimiter (see below) and the individual rows will be separated by a linefeed/carriage-return.
* If the file specified by destination cannot be opened then the error value #N/A will be returned by SQL.RETRIEVE.TO.FILE.
* If the file already exists its previous contents will be overwritten by SQL.RETRIEVE.TO.FILE.

Col\_names\_logical    is a logical value that, if TRUE, allows the column names to be returned as the first row of data. If FALSE or omitted, the column names will not be returned.

Column\_delimiter    is the value that will be used to separate the elements in each row. If column\_delimiter is omitted then a tab will be used. If another value is desired then it should be enclosed in quotation marks. Possible values for column\_delimiter might be: "," or ";" or " ". The string "tab" can also be used to specify a tab separator (even though this is redundant, since a tab is the default).

**Remarks**

* If SQL.RETRIEVE.TO.FILE is unable to retrieve the results on the specified connection then an error value will be returned. In such a case SQL.RETRIEVE.TO.FILE will place error information in memory for the SQL.ERROR function, if such information is available.
* If SQL.RETRIEVE.TO.FILE is able to successfully return rows of results on the specified connection and place them in a file it will return the number of rows that were actually written to the file. If there were no results pending on the connection then SQL.RETRIEVE.TO.FILE will return the #N/A error value and the file will not be created or modified.
* Before calling SQL.RETRIEVE.TO.FILE a connection must be established with a data source using SQL.OPEN.
* A successful call to SQL.OPEN returns a unique connection ID number, which can be used in a call to SQL.EXEC.QUERY to send a SQL language query. Following this call to SQL.EXEC.QUERY, SQL.RETRIEVE.TO.FILE uses the same connection ID number to retrieve query results and place them in a file.

**Example**

SQL.RETRIEVE.TO.FILE(conn1,"C:\MSQUERY\RESULTS1.QRY",TRUE,",") retrieves the results of a previously executed query and places them in the file RESULTS1.QRY, with column names that are comma delimited.

**Related Functions**

SQL.OPEN   Establishes a connection with a data source

SQL.EXEC.QUERY   Sends a query to a data source

SQL.BIND   Specifies storage for a result column

SQL.RETRIEVE   Retrieves query results

SQL.GET.SCHEMA   Gets information about a connected data source.

SQL.CLOSE   Closes a data source connection

SQL.ERROR   Returns detailed error information

Return to [top](#Q)

# STANDARD.FONT

Sets the attributes of the standard font in Microsoft Excel version 2.2 and earlier. This function is included only for macro compatibility. To define and apply a style in Microsoft Excel version 5.0 or later, use the APPLY.STYLE and DEFINE.STYLE functions.

**Syntax**

**STANDARD.FONT**(name\_text, size\_num, bold, italic, underline, strike, color, outline, shadow)

The arguments for this function are the same as those for FORMAT.FONT.

**Related Functions**

APPLY.STYLE   Applies a style to the selection

DEFINE.STYLE   Defines a cell style

FORMAT.FONT   Applies a font to the selection

Return to [top](#Q)

# STANDARD.WIDTH

Sets the default width used for all columns that you have not previously adjusted on the active worksheet.

**Syntax**

**STANDARD.WIDTH**(standard\_num)

Standard\_num    specifies how wide you want the columns to be in units of one character of the font corresponding to the Normal cell style.

Return to [top](#Q)

# STEP

Stops the normal flow of a macro and calculates it one cell at a time. Running a macro one cell at a time is called single-stepping and is very useful when you are debugging a macro. Use the STEP function, instead of clicking the Step Into button in the Macro dialog box when you want to start single-stepping at a specific line in a macro. The Macro dialog box appears when you click the Macros command (Tools menu, Macro submenu).

**Syntax**

**STEP**( )

**Remarks**

* When Microsoft Excel encounters a STEP function, it stops running the macro and displays a dialog box. The dialog box tells you which cell in the macro Microsoft Excel is about to calculate, and what formula is in that cell. You can click Step to carry out the next instruction; click Evaluate to calculate part of the formula; click Halt to interrupt the macro; or click Continue to continue the macro without single-stepping.
* When placed at the beginning of a macro, STEP is equivalent to clicking the Macro command on the Tools menu and clicking the Step Into button in the Macro dialog box.
* To step through the calculation of a custom function, place the STEP function at the start of the custom function.

**Related Functions**

HALT   Stops all macros from running

RUN   Runs a macro

Return to [top](#Q)

# STYLE

Checks the fonts for a bold and/or italic font and applies it to the current selection in Microsoft Excel for the Macintosh version 1.5 or earlier. If no appropriate font is available, Microsoft Excel finds the most similar font available and formats the selection using that font. This function is included only for macro compatibility. If you want to change a font to bold or italic, use the FONT.PROPERTIES function.

**Syntax**

**STYLE**(bold,italic)

**STYLE**?(bold,italic)

**Related Function**

FONT.PROPERTIES   Applies a font to the selection

Return to [top](#Q)

# SUBSCRIBE.TO

Inserts the contents of the edition into the active sheet at the point of the current selection. Use SUBSCRIBE.TO to incorporate editions published from other workbooks into your Microsoft Excel worksheets and macro sheets. SUBSCRIBE.TO returns TRUE if successful.

**Syntax**

**SUBSCRIBE.TO**(**file\_text, format\_num**)

**Important**This function is only available if you are using Microsoft Excel for the Macintosh with system software version 7.0 or later.

File\_text    is the name, as a text string, of the edition you want to insert into the active sheet. Unless file\_text is in the current folder, supply the full path of the workbook. If file\_text cannot be found, SUBSCRIBE.TO returns the #VALUE! error value and interrupts the macro.

**Remarks**

* If a single cell is selected, the data from the edition file is placed into as large a range of cells as is required by the data. Data already present in those cells is replaced. If the data is a picture, it is inserted from the upper-left corner of the selected cell.
* If a range of cells is selected, and the range is not big enough to contain the edition data, Microsoft Excel displays a dialog box asking if you want to clip the data to fit the range.

Format\_num    is the number 1 or 2 and specifies the format type of the file you are subscribing to.

|  |  |
| --- | --- |
| **Format\_num** | **Format type** |
| 1 or omitted | Picture |
| 2 | Text (includes BIFF, VALU, TEXT, and CSV formats) |

**Related Functions**

CREATE.PUBLISHER   Creates a publisher from the selection

EDITION.OPTIONS   Sets publisher and subscriber options

GET.LINK.INFO   Returns information about a link

Return to [top](#Q)

# SUBTOTAL.CREATE

Equivalent to clicking the Subtotals command on the Data menu. Generates a subtotal in a list or database.

**Syntax**

**SUBTOTAL.CREATE**(**at\_change\_in**, **function\_num**, **total**, replace, pagebreaks, summary\_below)

**SUBTOTAL.CREATE**?(at\_change\_in, function\_num, total, replace, pagebreaks, summary\_below)

At\_change\_in    is a column offset corresponding to the At Each Change In text box on the Subtotal dialog box.

Function\_Num    is a number corresponding to the Use Function list box specifying which function you want to use in subtotaling your data.

|  |  |
| --- | --- |
| **Function** | **Function\_Num** |
| SUM | 1 |
| COUNTA | 2 |
| AVERAGE | 3 |
| MAX | 4 |
| MIN | 5 |
| PRODUCT | 6 |
| COUNT | 7 |
| STDEV | 8 |
| STDEVP | 9 |
| VAR | 10 |
| VARP | 11 |

Total    is an array of column offsets corresponding to the Add Subtotal To list box. Indicates which columns you want aggregated according to function\_num; for example, {4,5}

Replace    is a logical value which, if TRUE, causes any previous subtotals to be replaced by new subtotals. If FALSE or omitted, subtotals will not be replaced.

PageBreaks    is a logical value corresponding to the Page Break Between Groups check box which, if TRUE, creates a page break below each subtotal. If FALSE or omitted, does not create a page break below each subtotal.

Summary\_Below    is a logical value corresponding to the Summary Below Data check box which, if TRUE, places subtotal rows below the data they refer to, and a grand total at the bottom of the list. If FALSE, places subtotal rows above the data they refer to, and a grand total just below the header.

**Related Function**

SUBTOTAL.REMOVE   Removes all previously existing subtotals and grand totals in a list

Return to [top](#Q)

# SUBTOTAL.REMOVE

Equivalent to clicking the Subtotal command on the Data menu, and then clicking the Remove All button in the Subtotal dialog box. Removes all previously existing subtotals and grand totals in a list. Any page breaks and outlines will also be removed.

**Syntax**

**SUBTOTAL.REMOVE**()

**Related Function**

SUBTOTAL.CREATE   Generates a subtotal in a list or database

Return to [top](#Q)

# SUMMARY.INFO

Equivalent to clicking the Properties command on the File menu. Generates the summary information for the active workbook.

**Syntax**

**SUMMARY.INFO**(title, subject, author, keywords, comments)

**SUMMARY.INFO**?(title, subject, author, keywords, comments)

The arguments correspond to the text boxes on the Summary tab of the Properties dialog box. If any arguments are omitted, that text box is left empty.

Title    specifies a title for the file, not necessarily a file name. Long names can be entered, up to 255 characters.

Subject     is information pertaining to the subject matter of the workbook.

Author     is initially the name specified in the User Name box on the General tab in the Options dialog box, which appears from the Options command from the Tools menu. If this is omitted, the registered user of the copy of Microsoft Excel will be used.

Keywords    are keywords that can be later used in searching for the contents in the file.

Comments     is a comment that can be entered to help a user learn more about the contents or subject matter of the workbook.

**Related Functions**

FIND.FILE   Lets you search for files based on criteria such as author or creation date

GET.WORKBOOK   Returns information about a workbook sheet

Return to [top](#Q)