Excel® for Chemists: A Comprehensive Guide, Third Edition
by E. Joseph Billo

Appendix F

Alphabetical List of Selected VBA Keywords

This listing of VBA objects, properties, methods, functions and other keywords will be useful when creating your own VBA procedures. The list is not exhaustive, but contains mainly those keywords that are used in the procedures shown in this book.

For each VBA keyword, the required syntax is given, along with some comments on the required and optional arguments, one or more examples and a list of related keywords. See Excel's On-Line Help for further information.

Abs Function
Returns the absolute value of a number.
Syntax: **Abs(number)**
Example: **Abs(-7.3)** returns 7.3
See also: **Sgn**

Activate Method
Activates an object.
Syntax: *object*.**Activate**
Object can be **Chart, Worksheet** or **Window**.
Example: **Workbooks**("BOOK1.XLS").**Worksheets**("Sheet1").**Activate**
See also: **Select**

ActiveCell Property
Returns the active cell of the active window. Read-only.
Syntax: **ActiveCell** and **Application.ActiveCell** are equivalent.
See also: **Activate, Select**

ActiveSheet Property
Returns the active sheet of the active workbook. Read-only.
Syntax: *object*.**ActiveSheet**
Object can be **Application, Window** or **Workbook**.
Example: **Application.ActiveSheet.Name** returns the name of the active sheet of the active workbook. Returns **None** if no sheet is active.
See also: **Activate, Select**

Address Property
Returns a reference, as text
Syntax: *object*.**Address** *(rowAbsolute,columnAbsolute, referenceStyle, external, relativeTo)*
All arguments are optional. If *rowAbsolute* or *columnAbsolute* are **True** or omitted, returns that part of the address as an absolute reference. *ReferenceStyle* can be xlA1 or xlR1C1. If external is **True**, returns an external reference. See On-Line Help for information about the *relativeTo* argument.
See also: **Offset**

And Operator
Logical operator. (expression1 **And** expression2) evaluates to **True** if both expression1 and expression2 are **True**. Also can be used to perform bitwise comparison of two numerical values: (13 **And** 6) evaluates to 4. (13 = 00001101, 6 = 00000110, 4 = 00000100).
See also: **Or, Not, Xor**

Application Object
Represents the Microsoft Excel application.

Array Function
Returns a **Variant** containing an array.
Syntax: **Array** (*arglist*)
Example: **Array** (*31,28,31,30,31,30,31,31,30,31,30,31*)
See also: **Dim**

As Keyword
Used with **Dim** to specify the data type of a variable.

Asc Function
Returns the numeric code for the first character of text.
Syntax: **Asc**(*character*)
Example: **Asc** ("A") returns 65.
See also: **Chr**

Atn Function
Returns the angle corresponding to a tangent value.
Syntax: **Atn**(*number*)
Number can be in the range $-\infty$ to $+\infty$. The returned angle is in radians, in the range $-\pi/2$ to $+\pi/2$ (-90° to 90°). To convert the result to degrees, multiply by $180/\pi$.
Example: **Atn**(1) returns 0.785388573 or 45 degrees.
See also: **Cos, Sin, Tan**

Bold Property
Returns **True** if the font is Bold. Sets the Bold font. Read-write.
Syntax: *object*.**Bold**
Object must be **Font**.
Example: **Range**("A1:E1").**Font.Bold = True** makes the cells bold.
See also: **Italic**

Boolean Data Type
Use to declare a variable's type as **Boolean** (**True** or **False**), either in a **Dim** statement, or in a **Sub** or **Function** statement. Two bytes required per variable. When number values are converted to Boolean values, 0 becomes **False** and all other values become **True**. When Boolean values are converted to numbers, **False** becomes 0 and **True** becomes –1.
See also: **Dim, As, Double, Integer, String, Variant**

Call Command
Transfers control to a **Sub** procedure.
Syntax: **Call** *name (argument1, ...)*
Name is the name of the procedure. *Argument1*, etc., are the names assigned to the arguments passed to the procedure. **Call** is optional; if omitted, the parentheses around the argument list must also be omitted.
Example: **Call** Task1(argument1,argument2)
See also: **Sub, Function**

Case Keyword
See: **Select Case**

Cells Method
Returns a single cell by specifying the row and column.
Syntax: *object*.**Cells**(*row, column*)
Object is optional; if not specified, **Cells** refers to the active sheet.
Example: **Cells**(2,1).**Value** = 5 enters the value 5 in cell A2.
See also: **Range**

Characters Object
Represents characters in any object containing text. Use the **Characters** object to format characters within a text string.
Syntax: *expression*.**Characters**(*start, length*)
Example: **Selection.Characters(Start:=x, Length:=1).Font.Subscript = True**

Clear Method
Clears formulas and formatting from a range of cells.
Syntax: *object.***Clear**
Object can be **Range** (or **ChartArea**).
Example: **Range**("A1:C10").**Clear**
See also: **ClearContents, ClearFormats** in Excel's On-Line Help.

Close Method
Closes a window, workbook or workbooks.
Syntax: For workbooks, use *object.***Close**. For a workbook or window, use *object.***Close**(*SaveChangesLogical, FileName*).
Object can be **Window, Workbook** or **Workbooks**. If *SaveChangesLogical* is **False**, does not save changes; if omitted, displays a "Save Changes?" dialog box.
Example: **Workbooks**("BOOK1.XLS").**Close**
See also: **Open, Save, SaveAs**

Column Property
Returns a number corresponding to the first column in the range. Read-only.
Syntax: *object.***Column**
Object must be **Range**.
See also: **Columns, Row, Rows**
Columns Method
Returns a **Range** object that represents a single column or multiple columns
Syntax: *object.***Columns**(*index*)
Object can be **Worksheet** or **Range**. *Index* is the name or number (column A = 1, etc.) of the column.
Example: **Selection.Columns.Count** returns the number of columns in the selection.
See also: **Range, Rows**

ColumnWidth Property
Returns or sets the width of all columns in the range. If columns in the range have different widths, returns **Null**.
Example: **Worksheets**("Sheet1").**Columns**("C").**ColumnWidth** = 30
See also: **RowHeight**

ConvertFormula Method
Converts cell references between A1-style and R1C1-style, and between absolute and relative. On-Line Help states that *Formula* must begin with an equal sign, but references in a string that does not begin with an equal sign are also converted.
Syntax: **expression.ConvertFormula(*Formula, FromReferenceStyle, ToReferenceStyle, ToAbsolute, RelativeTo*)**
Example:

FormulaString = **Application.ConvertFormula**(FormulaString, **xlA1, xlA1, xlAbsolute**)
See also: **Address**

Copy Method
Copies the selected object to the Clipboard or to another location.
Syntax: *object*.**Copy**(*destination*)
Object can be **Range, Worksheet, Chart** and many other objects. *Destination* specifies the range where the copy will be pasted. If omitted, copy goes to the Clipboard.
Example: **Worksheets**("Sheet1").**Range**("A1:C50").**Copy**
See also: **Cut, Paste**

Cos Function
Returns the cosine of an angle.
Syntax: **Cos**(*number*)
Number is the angle in radians; it can be in the range $-\infty$ to $+\infty$. To convert an angle in degrees to one in radians, multiply by $\pi/180$. Returns a value between −1 and 1.
See also: **Atn, Sin, Tan**

Count Property
Returns the number of items in the collection. Read-only.
Syntax: *object*.**Count**
Object can be any collection.
Example: The statement N = array.**Count** counts the number of values in the range array.

Cut Method
Cuts the selected object and pastes to the Clipboard or to another location.
Syntax: *object*.**Cut**(*destination*)
Object can be **Range, Worksheet, Chart** or one of many other objects. *Destination* specifies the range where the copy will be pasted. If omitted, copy goes to the Clipboard.
Example: **Worksheets**("Sheet1").**Range**("A1:C50").**Cut**
See also: **Copy, Paste**

CVErr Function
Returns a Variant containing an error value specified by the user.
Syntax: **CVErr**(*number*)
CVErr can return either Excel's built-in worksheet error values, or a user-defined error value. The values of *number* for built-in worksheet error values are xlErrDiv0, xlErrNA, xlErrName, xlErrNull, xlErrNum, xlErrRef, xlErrValue.
See also: **IsError**

Delete Method
Deletes the selected object.
Syntax: *object*.**Delete***(shift)*
Object can be **Range**, **Worksheet, Chart** and many other objects. *Shift* specifies how to shift cells when a range is deleted from a worksheet (xlToLeft or xlUp). Can also use *shift* = 1 or 2, respectively. If *shift* is omitted, Excel moves the cells without displaying the "Shift Cells?" dialog box.
Example: **Worksheets**("Sheet12").**Range**("A1:A10").**Delete** (xlToLeft) deletes the indicated range and shifts cells to left.

Dim Keyword
Declares an array and allocates storage for it.
Syntax: **Dim** ***variable*** *(subscripts)*
Variable is the name assigned to the array. *Subscripts* are the size dimensions of the array; an array can have up to 60 size dimensions. Each size dimension has a default lower value of zero; a single number for a size dimension is taken as the upper limit. Use *lower* **To** *upper* to specify a range that does not begin at zero. Use **Dim** with empty parentheses to specify an array whose size dimensions are defined within a procedure by means of the **ReDim** statement.
Example: **Dim** Matrix (5,5) **As Double** creates a 6 × 6 array of double-precision variables.
See also: **ReDim**

Do...Loop Command
Delineates a block of statements to be repeated.
Syntax: The beginning of the loop is delineated by **Do** or **Do Until** *condition* or **Do While** *condition*. The end of the loop is delineated by **Loop** or **Loop Until** *condition* or **Loop While** *condition*. *Condition* must evaluate to **True** or **False.**
Example: See examples of **Do...Loop** structures in Chapter 16.
See also: **Exit, For, Next, Wend, While**

Double Data Type
Use to declare a variable's type as double-precision floating-point (15 significant digits), either in a **Dim** statement, or in a **Sub** or **Function** statement. Eight bytes required per variable.
Example: **Dim** tolerance **As Double**
See also: **Dim, As, Boolean, Integer, String, Variant**

Else Keyword
Optional part of **If...Then** structure.

ElseIf Keyword
Optional part of **If...Then** structure.

End Command
Terminates a procedure or block.
Syntax: **End** terminates a procedure. **End Function** is required to terminate a **Function** procedure. **End If** is required to terminate a block **If** structure. **End Select** is required to terminate a **Select Case** structure. **End Sub** is required to terminate a **Sub** procedure. **End With** is required to terminate a **With** structure.
Example: See examples under **Select Case**.
See also: **Exit, Function, If, Then, Else, Select Case, Sub, With**

EndIf Keyword
Optional part of **If...Then** structure.

Err Function
Returns a run-time error number. Use in error-handling routine to determine the error and take appropriate corrective action.
Example:
```
If Err.Number = 13 Then
(code for corrective action here)
Resume pt1
End If
```
See also: **Error, On Error, Resume**

Evaluate Method
Converts a name or formula to a value.
Syntax: **Evaluate**(expression)
Expression must be a string, maximum length 255 characters. An initial equal sign is not necessary.
Example:
```
F$ = "2*3"
MsgBox Evaluate(F$)
```
See also: **Formula**

Exit Command
Exits a **Do**..., **For**..., **Function**... or **Sub**... structure.
Syntax: **Exit Do, Exit For, Exit Function, Exit Sub**
From a **Do** or **For** loop, control is transferred to the statement following the **Loop** or **Next** statement, or, in the case of nested loops, to the loop that is one level above the loop containing the **Exit** statement. From a **Function** or **Sub** procedure, control is transferred to the statement following the one that called the procedure.
Example: See examples of **Exit** procedures in Chapter 16.
See also: **Do, For...Next, Function, Stop, Sub**

Exp Function
Returns *e* raised to a power.
Syntax: **Exp**(*number*)
Returns the value of *e* raised to the power *number*.
See also: **Log**

False Keyword
Use the keywords **True** or **False** to assign the value **True** or **False** to Boolean (logical) variables.
When other numeric data types are converted to Boolean values, 0 becomes **False** while all other values become **True**. When Boolean values are converted to other data types, **False** becomes 0 while **True** becomes –1.
Example: **If** SubFlag = **False Then**...
See also: **True**

FillDown Method
Copies the contents and format(s) of the top cell(s) of a specified range into the remaining rows.
Syntax: *object*.**FillDown**
Object must be **Range**.
Example: **Worksheets**("Sheet12").**Range**("A1:A10").**FillDown**
See also: **FillLeft, FillRight, FillUp** in Excel's On-Line Help.

FillRight Method
Copies the contents and format(s) of the leftmost cell(s) of a specified range into the remaining columns.
Syntax: *object*.**FillDown**
Object must be **Range**.
Example: **Worksheets**("Sheet12").**Range**("A1:A10").**FillRight**
See also: **FillDown, FillLeft, FillUp** in Excel's On-Line Help.

Fix Function
Truncates a number to an integer.
Syntax: **Fix**(*number*)
If *number* is negative, **Fix** returns the first negative integer greater than or equal to *number*.
Example: **Fix**(-2.5) returns –2.
See also: **Int**

Font Property
Returns the font of the object. Read-only.
Syntax: *object*.**Font**
Example: ActiveCell.Font.Bold = **True** makes characters in the active cell bold.
See also: **FontStyle**

FontStyle Property
Returns or sets the font of the object. Read-write.
Syntax: *object*.**FontStyle**
Example: **Range**("A1:E1").**Font.FontStyle** = "Bold"
See also: **Font**

For...Next Command
Delineates a block of statements to be repeated.
Syntax: **For** *counter* = *start* **To** *end* **Step** *increment*
(statements)
Next *counter*
Step *increment* is optional; if not included, the default value 1 is used. *Increment* can be negative, in which case *start* should be greater than *end*.
Example: See examples of **For...Next** procedures in Chapter 16.
See also: **Do...Loop, Exit, For Each...Next, While...Wend**

For Each...Next Command
Delineates a block of statements to be repeated.
Syntax: **For Each** *element* **In** *group*
(statements)
Next *element*
Group must be a collection or array. *Element* is the name assigned to the variable used to step through the collection or array. *Group* must be a collection or array.
Example: See examples of **For Each...Next** procedures in Chapter 16.
See also: **Do...Loop, Exit, For...Next, While...Wend**

Format Function
Formats a value according to a formatting code expression.
Syntax: **Format**(***expression,formattext***)
Expression is usually a number, although strings can also be formatted. *Formattext* is a built-in or custom format. Additional information can be found in *Microsoft Excel/Visual Basic Reference*, or VBA On-Line Help.
Example: **Format**(TelNumber,"(###) ###-####") formats the value TelNumber in the form of a telephone number.

Formula Property
Returns or sets the formula in a cell.
If a cell contains a value, returns the value; if the cell contains the formula, returns the formula as a string.
See also: **Text, Value**

Function Keyword
Marks the beginning of a **Function** procedure.
Syntax: **Function** *name argument1, ...*
Name is the name of the variable whose value is passed back to the caller. *Argument1*, etc., are the names assigned to the arguments passed from the caller to the procedure.
Example: See examples of **Function** procedures in Chapter 19.
See also: **Call, Sub**

GoTo Command
Unconditional branch within a procedure.
Syntax: **GoTo** ***label***
Label can be a name or a line number.

If...Then...Else...End If Command
Delineates a block of conditional statements.
Syntax: **If** *condition* **Then** ... **Else** ... **End If**
The statement can be all on one line (e.g., **If** *condition* **Then** *statement*). Alternatively, a block **If** structure can be used, in which case the first line consists of **If** *condition* **Then**; the end of the structure is delineated by **End If**. *Condition* must evaluate to **True** or **False.** The ellipsis following **Then** and **Else** can represent a single statement or several statements separated by colons; these are executed if *condition* is **True** or **False,** respectively.

Examples:
```
If Char = "." Then GoTo 2000
If (Char >= "0" And Char <= "9") Then
(statements)
End If
```

See also: **ElseIf, End**

InputBox Function
Displays an input dialog box and waits for user input.
Syntax: **InputBox**(*prompt,title,default,xpos,ypos,helpfile,context*)
See *Microsoft Excel/Visual Basic Reference* or On-Line Help for details.
See also: **InputBox** Method, **MsgBox**

InputBox Method
Displays an input dialog box and waits for user input.
Syntax: *object*.**InputBox**(*prompt,title,default,left,top,helpfile,context, type*)
Object must be **Application**. The **InputBox** method has the additional *type* argument that allows the input of a reference. See *Microsoft Excel/Visual Basic Reference* or On-Line Help for details.
See also: **InputBox** Function, **MsgBox**

Insert Method
Inserts a range of cells in a worksheet.
Syntax: *object*.**Insert***(shift)*
Object is a **Range** object. *Shift* specifies how to shift cells when a range is inserted in a worksheet (xlToRight or xlDown). Can also use *shift* = 1 or 2, respectively. If *shift* is omitted, the "Shift Cells?" dialog box is not displayed.
Examples: **Worksheets**("Sheet12").**Range**("A1:A10").**Insert** (1) inserts the indicated range and shifts cells to right.
Worksheets("Sheet1").**Columns**(4).**Insert** inserts a new column to the left of column D.
See also: **Delete**

Instr Function
Returns a number specifying the position of the first occurrence of one string within another. Returns zero if the search string is not found.
Syntax: **InStr(***start, string_to_search, string_to_look_for, compare)*
Optional *start* specifies the start position for the search. If omitted, search begins at position 1. Optional *compare* determines the type of comparison. See On-Line Help for details.
Example: **InStr(**1,NameText,"!"***)*** finds the first occurrence of the "!" character within the string contained in the variable NameText.

Int Function
Rounds a number to an integer.
Syntax: **Int**(*number*)
If *number* is negative, **Int** returns the first negative integer less than or equal to *number*.
Example: **Int**(-2.5) returns –3.
See also: **Fix**

Integer Data Type
Use to declare a variable's type as **Integer**, either in a **Dim** statement, or in a **Sub** or **Function** statement. Two bytes required per variable.
Example: **Dim J As Integer**
See also: **Dim, As, Boolean, Double, String, Variant**

Intersect Method
Returns a **Range** object that represents the intersection of two ranges.
Syntax: **Intersect** (***range1, range2***)
See also: **Union, Areas, Caller**

IsArray Function
Returns **True** if the variable is an array.
Syntax: **IsArray**(*name*)
See also: other **Is** functions

IsDate Function
Returns **True** if the expression can be converted to a date.
Syntax: **IsDate**(*expression*)
See also: other **Is** functions

IsEmpty Function
Returns **True** if the variable has been initialized.
Syntax: **IsEmpty**(*expression*)
See also: other **Is** functions

IsMissing Function
Returns **True** if an optional argument has not been passed to a procedure.
Syntax: **IsMissing**(*name*)
See also: other **Is** functions

IsNull Function
Returns **True** if the expression is null (i.e., contains no valid data).
Syntax: **IsNull**(*expression*)
See also: other **Is** functions

IsNumeric Function
Returns **True** if the expression can be evaluated to a number.
Syntax: **IsNumeric**(*expression*)
See also: other **Is** functions

Italic Property
Returns **True** if the font is Italic. Sets the Italic font. Read-write.
Syntax: *object*.**Italic**
Object must be **Font**.
Example: **Range**("A1:E1").**Font.Italic = True** makes the cells italic.
See also: **Bold**

LBound Function
Returns the lower limit of an array dimension.
Syntax: **LBound**(*array,dimension*)
Array is the name of the array. *Dimension* is an integer (1, 2, 3, etc.) specifying the dimension to be returned; if omitted, the value 1 is used.
Example: If the array table was dimensioned using the statement **Dim** table (1 **To** 3, 1000), **LBound**(table,1) returns 1, **LBound**(table,2) returns 0.
See also: **Dim, UBound**

LCase Function
Converts a string into lowercase letters.
Syntax: **LCase**(*string*)
See also: **UCase**

LTrim Function
Returns a string without leading spaces.
Syntax: **LTrim**(*string*)
See also: **RTrim**

Left Function
Returns the leftmost characters of a string.
Syntax: **Left**(*string,number*)
If *number* is zero, a null string is returned. If *number* is greater than the number of characters in *string*, the entire string is returned.
Example: **Left**("CHEMISTRY",4) returns CHEM
See also: **Len, Mid, Right**

Len Function
Returns the length (number of characters) in a string.
Syntax: **Len**(*string*)
Example: **Len**("CHEMISTRY") returns 9.
See also: **Left, Mid, Right**

Log Function
Returns the natural (base-*e*) logarithm of a number.
Syntax: **Log**(*number*)
Number must be a value or expression greater than zero. VBA does not provide base-10 logarithms; use **Log**(value)/**Log**(10).
See also: **Exp**

MacroOptions Method
Sets options in the Macro Options dialog box.
Syntax: **Application.MacroOptions**(*macro, description, hasMenu, menuText, hasShortcutKey, shortcutKey, category, statusbar, helpContext, helpFile*)

macro is the name of the macro. *description* is the description that appears in the dialog box. *category* is the function category that the macro appears in: Financial, 1; Date & Time, 2; Math & Trig, 3; Statistical, 4; Lookup & Reference, 5; Database, 6; Text, 7; Logical, 8; Information, 9; User Defined, 14; Engineering, 15.
Example: **Application.MacroOptions** macro:="FtoC", Description:= "Converts Fahrenheit temperature to Celsius", Category:=3
provides a description for the macro FtoC and assigns it to the Math & Trig category.

Mid Function
Returns the specified number of characters from a text string, beginning at the specified position.
Syntax: **Mid**(*string,start,number*)

If *start* is greater than the number of characters in *string*, returns a null string. If *number* is omitted, all characters from *start* to the end of the string are returned.
Example: **Mid**("H2SO4",2,1) returns 2.
See also: **Left, Len, Right**

Mod Operator
Returns the remainder resulting from the division of two numbers.
Syntax: *result* = *number1* **Mod** *number2*

MsgBox Function
Displays a message box.
Syntax: **MsgBox**(***prompt,****buttons,title,helpfile,context*)
See *Microsoft Excel/Visual Basic Reference* or On-Line Help for details.
See also: **InputBox**

Name Property
Returns or sets the name of an object.
Example: SeriesName = **Selection.Name** assigns the name of the selected chart series to the variable SeriesName.
See also: **NameLocal, Names**

Next Keyword
Delineates the end of a **For...Next** or **For Each...Next** block of statements.
Not Operator
Logical operator. Performs logical negation: **True** becomes **False**, **False** becomes **True**.
See also: **And, Or**

Now Function
Returns the current date and time.
Syntax: **Now**
See also: other date and time functions.

NumberFormat Property
Returns or sets the number format code of a cell.
Example: **Range**("A1:A10").**NumberFormat=** "0.00" sets the number format of the specified range of cells.
See also: **GoSub, GoTo, Return, Select Case**

On...GoTo Command
Branches to one of several specified lines, depending on the value of an expression.
Syntax: **On** *expression* **GoTo** *label1, ...*
See explanation under **On...GoSub** command.

Example: See examples of **On...GoTo** procedures in Chapter 16.
See also: **GoSub, GoTo, Return, Select Case**

On Error GoTo Command
Enables an error-handling routine and specifies the action to be taken in event of an error.
Examples: **On Error GoTo** *line* (enables the error-handling routine at the specified location in the procedure)
On Error Resume Next (execution resumes with the statement immediately following the statement that caused the error)
On Error GoTo 0 (disables any enabled error handler in the current procedure)

Open Method
Opens a workbook.
Syntax: *object*.**Open**(*filename, ...*)
Object must be **Workbooks**. *Filename* is required. See On-Line Help for the remaining arguments.
Example: **Workbooks.Open**("SOLVSTAT.XLS")
See also: **Close, Save, SaveAs**

Option Base Keyword
Use at module level to declare lower bound for an array.
Can be **Option Base** 0 or 1. The statement can appear only once in a module and must precede all **Dim** or equivalent declaration.
See also: **Dim, LBound, ReDim**

Option Explicit Statement
Use at module level to force explicit declaration of all variables in that module.
See also: **Option Base, Option Compare**

Optional Keyword
Indicates that an argument in a function is not required. All arguments following the **Optional** keyword must be optional. All optional arguments are **Variant**.
Syntax: **Function** name(*argument1*,... **Optional** *argument*)
See also: **Function, ParamArray**

Or Operator
Logical operator. (expression1 **Or** expression2) evaluates to **True** if either expression1 or expression2 is **True**. Also can be used to perform bitwise comparison of two numerical values: (13 **Or** 6) evaluates to 15. (13 = 00001101, 6 = 00000110, 15 = 00001111).
See also: **Or, Not, Xor**

ParamArray Keyword
Allows the use of an indefinite number of arguments for a function. The argument becomes an array of **Variant** elements. The array has lower array index of zero, even if **Option Base 1** is declared.
Syntax: **Function** name(*argument1*,... **ParamArray** *argument*() **As Variant)**
Example: **Function** test **(ParamArray** rng() **As Variant)**
See also: **Dim, Function, Variant**

Paste Method
Pastes the contents of the Clipboard onto a worksheet.
Syntax: *object*.**Paste**(*destination*)
Object must be **Worksheet.** There are other **Paste** methods, with different syntax, for **Chart** and many other objects. *Destination* specifies the range where the copy will be pasted. If omitted, copy is pasted to the current selection.
Example: **Worksheets**("Sheet1").**Range**("A1:C50").**Copy**
ActiveSheet.Paste
See also: **Copy, Cut**

Preserve Command
Preserves data in an existing array when using **ReDim**.

Private Command
Indicates that the procedure is available only to procedures in the same module.

Public Command
Indicates that the procedure is available to all other procedures.

Quit Method
Quits Microsoft Excel.
Syntax: *object*.**Quit**
Object must be **Application**.
Example: **Application.Quit**
See also: **Close, Save**
Range Method
Returns a **Range** object that represents a cell or range of cells.
Syntax: *object*.**Range**(*reference*)
Object is required if it is **Worksheet**. *Reference* must be an A1-style reference, in quotes, or the name of the reference.
Example: **Worksheets**("Sheet12").**Range**("A1").**Value** = 5
See also: **Cells**

ReDim Keyword
Allocates or re-allocates dynamic array storage.
Syntax: **ReDim** ***variable*** *(subscripts)*
For discussion of *variable* and *subscripts*, see comments under the entry for **Dim**.

You can use **ReDim** repeatedly to change the number of elements in an array, or the number or dimensions.

Example: **Dim** Matrix()
(statements)
ReDim Matrix (5,5)
(statements)
ReDim Matrix (15,25)

See also: **Dim**

Resume Command

Resumes execution after an error-handling routine is finished.

Examples: **Resume 0**
Resume Next (execution resumes with the statement immediately following the statement that caused the error)
Resume *label* (Execution resumes at the specified location in the procedure)

See ***also:*** **On Error GoTo**

Return Command

Delineates the end of a subroutine within a procedure.

Right Function

Returns the rightmost characters of a string.

Syntax: **Right**(*string,number*)

If *number* is zero, a null string is returned. If *number* is greater than the number of characters in *string*, the entire string is returned.

Example: **Right**(303585842,4) returns 5842.

See also: **Left, Len, Mid**

Rnd Function

Returns a random number between 0 and 1.

Syntax: **Rnd**

Row Property

Returns a number corresponding to the first row in the range. Read-only.

Syntax: *object*.**Row**

Object must be **Range**.

Example: **If ActiveCell.Row** = 10 **Then ActiveCell.Interior.ColorIndex** = 27 changes the interior color of the active cell to yellow if it is in row 10.

See also: **Column, Columns, Rows**

RowHeight Property

Returns or sets the height of all rows in the range.

Example: **Worksheets**("Sheet1").**Rows**(1).**RowHeight** = 15

See also: **ColumnWidth**

Rows Method
Returns a **Range** object that represents a single row or multiple rows.
Syntax: *object*.**Rows**(*index*)
Object can be **Worksheet** or **Range**. *Index* is the name or number of the row.
Example: **Selection.Rows.Count** returns the number of rows in the selection.
See also: **Columns, Range**

RTrim Function
Returns a string without trailing spaces.
Syntax: **RTrim**(*string*)
See also: **LTrim, Trim**

Save Method
Saves changes to active workbook.
Syntax: *object*.**Save**(*filename*)
Object must be **Workbook**. If *filename* is omitted, uses a default name.
Example: **ActiveWorkbook.Save**
See also: **Close, Open, SaveAs**

SaveAs Method
Saves changes to active workbook or other document with a different filename.
Syntax: *object*.**SaveAs**(*filename, ...*)
Object can be **Worksheet, Workbook, Chart** or other document types. See *Microsoft Excel/Visual Basic Reference* or On-Line Help for details.
Example: NewChart.**SaveAs**("New Chart")
See also: **Close, Open, Save**

Select Method
Selects an object.
Syntax: *object*.**Select**
Object can be **Chart, Worksheet** or one of many other objects.
Example: **Range**("A1:C50").**Select**
See also: **Activate**

Select Case Command
Executes one of several blocks of statements, depending on the value of an expression.

```
Syntax:   Select Case expression
          Case expression1
          (statements)
          Case expression2
          (statements)
          End Select
```

You can also use the **To** keyword in *expression*, e.g., **Case** "A" **To** "M". *Expression* can also be a logical expression. Use **Case Else** (not required) to

handle all cases not covered by the preceding **Case** statements.
Example: See examples of **Select Case** procedures in Chapter 16.
See also: **If...Then...Else, On...GoSub, On...GoTo**

Selection Property
Returns the selected object. The object returned depends on the type of selection.
See also: **Activate, ActiveCell, Select**

Set Command
Assigns an object reference to a variable.
See also: **Dim, ReDim**

Sgn Function
Returns the sign of a number.
Syntax: **Sgn**(*number*)
Returns 1, 0 or –1 if *number* is positive, zero or negative, respectively.
Example: **Sgn**(-7.3) returns –1.
See also: **Abs**

Sin Function
Returns the sine of an angle.
Syntax: **Sin**(*number*)
Number is the angle in radians; it can be in the range $-\infty$ to $+\infty$. To convert an angle in degrees to one in radians, multiply by $\pi/180$. Returns a value between –1 and 1.
See also: **Atn, Cos, Tan**

Sort Method
Sorts a range of cells.
Syntax: *object*.**Sort**(*sortkey1,order1,sortkey2,order2, ...*)
Object must be **Range**. See *Microsoft Excel/Visual Basic Reference* or On-Line Help for details.

Sqr Function
Returns the square root of a number.
Syntax: **Sqr**(*number*)
Number must be greater than or equal to zero.

Step Keyword
Stops execution, but does not close files or clear variables.
See also: **End**
Stop Command
Stops execution, but does not close files or clear variables.
See also: **End**

Str Function
Converts a number to a string.
Syntax: **Str**(*number*)
A leading space is reserved for the sign of the number; if the number is positive, the string will contain a leading space.
See also: **Format**

String Data Type
Use to declare a variable's type as **String**, either in a **Dim** statement, or in a **Sub** or **Function** statement. One byte/character required per variable.
Example: **Dim** J **As Integer**
See also: **Dim, As, Boolean, Double, String, Variant**

Sub Keyword
Marks the beginning of a **Sub** procedure.
Syntax: **Sub** *name* (*argument1, ...*)
Name is the name of the procedure. *Argument1*, etc., are the names assigned to the arguments passed from the caller to the procedure. The end of the procedure is delineated by **End Sub**
Example: See examples of **Sub** procedures in Chapter 18.
See also: **Call, Function**

Tan Function
Returns the tangent of an angle.
Syntax: **Tan**(*number*)
Number is the angle in radians; it can be in the range $-\infty$ to $+\infty$. To convert an angle in degrees to one in radians, multiply by $\pi/180$. Returns a value between $-\infty$ and $+\infty$.
See also: **Atn, Cos, Sin**

Text Property
Returns or sets the text associated with an object.
The text can be associated with a chart, button, textbox, control or range. For all except range, this property is read-write, but for a range, it is read-only.
Example: **Worksheets**("Sheet1").**Buttons**(1).**Text** = "Undo"
See also: **Formula, Value**

Trim Function
Returns a string without leading or trailing spaces.
Syntax: **Trim**(*string*)
See also: **LTrim, RTrim**

True Keyword
Use the keywords **True** or **False** to assign the value **True** or **False** to Boolean

(logical) variables.
When other numeric data types are converted to Boolean values, 0 becomes **False** while all other values become **True**. When Boolean values are converted to other data types, **False** becomes 0 while **True** becomes –1.
Example: **If** FirstFlag = **True Then GoTo** 2000

UBound Function
Returns the upper limit of an array dimension.
Syntax: **UBound**(***array***, *dimension*)
Array is the name of the array. *Dimension* is an integer (1, 2, 3, etc.) specifying the dimension to be returned; if omitted, the value 1 is used.
Example: If the array table was dimensioned using the statement **Dim** table (1 **To** 3, 1000), **UBound**(table,3) returns 1, **UBound**(table,2) returns 1000.
See also: **Dim, LBound**

UCase Function
Converts a string into uppercase letters.
Syntax: **UCase**(***string***)
See also: **LCase**

Union Method
Returns a **Range** object that represents the union of two or more ranges, i.e., performs the same function as the comma character in the worksheet expression SUM(A1, B2, C3).
Syntax: **Union** (***range1, range2***)
See also: **Intersect, Areas, Caller**

Until Command
Optional part of **Do...Loop** structure.
Syntax: See explanation under **Do...Loop.**

Val Function
Converts a string to a number.
Syntax: **Val**(*string*)
Val stops at the first non-numeric character other than the period.
Example: **Val**("21 Lawrence Avenue") returns 21.
See also: **Str**

Value Property
Returns the value of an object.
Syntax: *object*.**Value**
If *object* is **Range**, returns or sets the value(s) of the cell(s). Read-write.
If **Range** contains more than one cell, returns an array of values.
Example: **Worksheets**("Sheet12").**Range**("A1").**Value** = "Volume, mL"

Variant Data Type
Use to declare a variable's type as **Variant**, either in a **Dim** statement, or in a **Sub** or **Function** statement. **Variant** is the default data type, so usually not required. It is required when using the **ParamArray** keyword. Sixteen bytes + one byte/character required per variable.
Example: **Function** test **(ParamArray** rng() **As Variant)**
See also: **Dim, As, Boolean, Double, Integer, String**

Wend Command
Delineates the end of a **While...Wend** procedure.
Syntax: See explanation under **Do...Loop.**
See also: **Do...Loop, While...Wend**

While...Wend Command
Executes a series of statements as long as a specified condition is true.
Syntax: See explanation under **Do...Loop.**
See also: **Do...Loop, Wend**

With...End With command
Delineates a block of statements to be executed on a single object.
Syntax: **With** *object*
(statements)
End With
See also: **Do...Loop, While...Wend**

XOr Operator
Exclusive Or operator.
Use to perform bitwise comparison of two numerical values: (13 **XOr** 6) evaluates to 11. (13 = 00001101, 6 = 00000110, 11 = 00001011).
See also: **Or, Not, Or**