B

# VBA Statements and Functions Reference

This appendix contains a complete listing of all Visual Basic for Applications (VBA) statements and built-in functions. For details, consult Excel's online help.

Note

**There are no new VBA statements in Excel 2010.**

**Table B-1:** Summary of VBA Statements

| Statement | Action |
|---|---|
| `AppActivate` | Activates an application window |
| `Beep` | Sounds a tone via the computer's speaker |
| `Call` | Transfers control to another procedure |
| `ChDir` | Changes the current directory |
| `ChDrive` | Changes the current drive |
| `Close` | Closes a text file |
| `Const` | Declares a constant value |
| `Date` | Sets the current system date |
| `Declare` | Declares a reference to an external procedure in a Dynamic Link Library (DLL) |
| `DefBool` | Sets the default data type to `Boolean` for variables that begin with specified letters |
| `DefByte` | Sets the default data type to `Byte` for variables that begin with specified letters |
| `DefCur` | Sets the default data type to `Currency` for variables that begin with specified letters |
| `DefDate` | Sets the default data type to `Date` for variables that begin with specified letters |
| `DefDec` | Sets the default data type to `Decimal` for variables that begin with specified letters |

*continued*

**Table B-1:** Summary of VBA Statements *(continued)*

| Statement | Action |
|---|---|
| `DefDbl` | Sets the default data type to `Double` for variables that begin with specified letters |
| `DefInt` | Sets the default data type to `Integer` for variables that begin with specified letters |
| `DefLng` | Sets the default data type to `Long` for variables that begin with specified letters |
| `DefObj` | Sets the default data type to `Object` for variables that begin with specified letters |
| `DefSng` | Sets the default data type to `Single` for variables that begin with specified letters |
| `DefStr` | Sets the default data type to `String` for variables that begin with specified letters |
| `DefVar` | Sets the default data type to `Variant` for variables that begin with specified letters |
| `DeleteSetting` | Deletes a section or key setting from an application's entry in the Windows Registry |
| `Dim` | Declares variables and (optionally) their data types |
| `Do-Loop` | Loops through a set of instructions |
| `End` | Used by itself, exits the program; also used to end a block of statements that begin with `If`, `With`, `Sub`, `Function`, `Property`, `Type`, or `Select` |
| `Enum` | Declares a type for enumeration |
| `Erase` | Re-initializes an array |
| `Error` | Simulates a specific error condition |
| `Event` | Declares a user-defined event |
| `Exit Do` | Exits a block of `Do-Loop` code |
| `Exit For` | Exits a block of `For-Next` code |
| `Exit Function` | Exits a `Function` procedure |
| `Exit Property` | Exits a property procedure |
| `Exit Sub` | Exits a subroutine procedure |
| `FileCopy` | Copies a file |
| `For Each-Next` | Loops through a set of instructions for each member of a series |
| `For-Next` | Loops through a set of instructions a specific number of times |
| `Function` | Declares the name and arguments for a `Function` procedure |
| `Get` | Reads data from a text file |
| `GoSub...Return` | Branches to and returns from a procedure |
| `GoTo` | Branches to a specified statement within a procedure |
| `If-Then-Else` | Processes statements conditionally |
| `Implements` | Specifies an interface or class that will be implemented in a class module |
| `Input #` | Reads data from a sequential text file |
| `Kill` | Deletes a file from a disk |
| `Let` | Assigns the value of an expression to a variable or property |
| `Line Input #` | Reads a line of data from a sequential text file |
| `Load` | Loads an object but doesn't show it |

| Statement | Action |
|---|---|
| `Lock...Unlock` | Controls access to a text file |
| `Lset` | Left-aligns a string within a string variable |
| `Mid` | Replaces characters in a string with other characters |
| `MkDir` | Creates a new directory |
| `Name` | Renames a file or directory |
| `On Error` | Gives specific instructions for what to do in the case of an error |
| `On...GoSub` | Branches, based on a condition |
| `On...GoTo` | Branches, based on a condition |
| `Open` | Opens a text file |
| `Option Base` | Changes the default lower limit for arrays |
| `Option Compare` | Declares the default comparison mode when comparing strings |
| `Option Explicit` | Forces declaration of all variables in a module |
| `Option Private` | Indicates that an entire module is `Private` |
| `Print #` | Writes data to a sequential file |
| `Private` | Declares a local array or variable |
| `Property Get` | Declares the name and arguments of a `Property Get` procedure |
| `Property Let` | Declares the name and arguments of a `Property Let` procedure |
| `Property Set` | Declares the name and arguments of a `Property Set` procedure |
| `Public` | Declares a public array or variable |
| `Put` | Writes a variable to a text file |
| `RaiseEvent` | Fires a user-defined event |
| `Randomize` | Initializes the random number generator |
| `ReDim` | Changes the dimensions of an array |
| `Rem` | Specifies a line of comments (same as an apostrophe [`'`]) |
| `Reset` | Closes all open text files |
| `Resume` | Resumes execution when an error-handling routine finishes |
| `RmDir` | Removes an empty directory |
| `RSet` | Right-aligns a string within a string variable |
| `SaveSetting` | Saves or creates an application entry in the Windows Registry |
| `Seek` | Sets the position for the next access in a text file |
| `Select Case` | Processes statements conditionally |
| `SendKeys` | Sends keystrokes to the active window |
| `Set` | Assigns an object reference to a variable or property |
| `SetAttr` | Changes attribute information for a file |
| `Static` | Declares variables at the procedure level so that the variables retain their values as long as the code is running |

*continued*

**Table B-1:** Summary of VBA Statements *(continued)*

| Statement | Action |
|---|---|
| `Stop` | Pauses the program |
| `Sub` | Declares the name and arguments of a `Sub` procedure |
| `Time` | Sets the system time |
| `Type` | Defines a custom data type |
| `Unload` | Removes an object from memory |
| `While...Wend` | Loops through a set of instructions as long as a certain condition remains true |
| `Width #` | Sets the output line width of a text file |
| `With` | Sets a series of properties for an object |
| `Write #` | Writes data to a sequential text file |

# Invoking Excel functions in VBA instructions

If a VBA function that's equivalent to one you use in Excel isn't available, you can use Excel's worksheet functions directly in your VBA code. Just precede the function with a reference to the `WorksheetFunction` object. For example, VBA doesn't have a function to convert radians to degrees. Because Excel has a worksheet function for this procedure, you can use a VBA instruction such as the following:

```
Deg = Application.WorksheetFunction.Degrees(3.14)
```

The `WorksheetFunction` object was introduced in Excel 97. For compatibility with earlier versions of Excel, you can omit the reference to the `WorksheetFunction` object and write an instruction such as the following:

```
Deg = Application.Degrees(3.14)
```

Note

**There are no new VBA functions in Excel 2010.**

**Table B-2:** Summary of VBA Functions

| Function | Action |
|---|---|
| `Abs` | Returns the absolute value of a number |
| `Array` | Returns a variant containing an array |
| `Asc` | Converts the first character of a string to its ASCII value |
| `Atn` | Returns the arctangent of a number |

| **Function** | **Action** |
|---|---|
| CallByName | Executes a method, or sets or returns a property of an object |
| CBool | Converts an expression to a Boolean data type |
| CByte | Converts an expression to a Byte data type |
| CCur | Converts an expression to a Currency data type |
| CDate | Converts an expression to a Date data type |
| CDbl | Converts an expression to a Double data type |
| CDec | Converts an expression to a Decimal data type |
| Choose | Selects and returns a value from a list of arguments |
| Chr | Converts a character code to a string |
| CInt | Converts an expression to an Integer data type |
| CLng | Converts an expression to a Long data type |
| Cos | Returns the cosine of a number |
| CreateObject | Creates an Object Linking and Embedding (OLE) Automation object |
| CSng | Converts an expression to a Single data type |
| CStr | Converts an expression to a String data type |
| CurDir | Returns the current path |
| CVar | Converts an expression to a variant data type |
| CVDate | Converts an expression to a Date data type (for compatibility, not recommended) |
| CVErr | Returns a user-defined error value that corresponds to an error number |
| Date | Returns the current system date |
| DateAdd | Adds a time interval to a date |
| DateDiff | Returns the time interval between two dates |
| DatePart | Returns a specified part of a date |
| DateSerial | Converts a date to a serial number |
| DateValue | Converts a string to a date |
| Day | Returns the day of the month of a date |
| DDB | Returns the depreciation of an asset |
| Dir | Returns the name of a file or directory that matches a pattern |
| DoEvents | Yields execution so the operating system can process other events |
| Environ | Returns an operating environment string |
| EOF | Returns True if the end of a text file has been reached |
| Error | Returns the error message that corresponds to an error number |
| Exp | Returns the base of natural logarithms (e) raised to a power |
| FileAttr | Returns the file mode for a text file |
| FileDateTime | Returns the date and time when a file was last modified |

*continued*

**Table B-2:** Summary of VBA Functions *(continued)*

| Function | Action |
|---|---|
| `FileLen` | Returns the number of bytes in a file |
| `Filter` | Returns a subset of a string array, filtered |
| `Fix` | Returns the integer portion of a number |
| `Format` | Displays an expression in a particular format |
| `FormatCurrency` | Returns an expression formatted with the system currency symbol |
| `FormatDateTime` | Returns an expression formatted as a date or time |
| `FormatNumber` | Returns an expression formatted as a number |
| `FormatPercent` | Returns an expression formatted as a percentage |
| `FreeFile` | Returns the next available file number when working with text files |
| `FV` | Returns the future value of an annuity |
| `GetAllSettings` | Returns a list of settings and values from the Windows Registry |
| `GetAttr` | Returns a code representing a file attribute |
| `GetObject` | Retrieves an OLE Automation object from a file |
| `GetSetting` | Returns a specific setting from the application's entry in the Windows Registry |
| `Hex` | Converts from decimal to hexadecimal |
| `Hour` | Returns the hour of a time |
| `IIf` | Evaluates an expression and returns one of two parts |
| `Input` | Returns characters from a sequential text file |
| `InputBox` | Displays a box to prompt a user for input |
| `InStr` | Returns the position of a string within another string |
| `InStrRev` | Returns the position of a string within another string from the end of the string |
| `Int` | Returns the integer portion of a number |
| `IPmt` | Returns the interest payment for a given period of an annuity |
| `IRR` | Returns the internal rate of return for a series of cash flows |
| `IsArray` | Returns `True` if a variable is an array |
| `IsDate` | Returns `True` if a variable is a date |
| `IsEmpty` | Returns `True` if a variable has not been initialized |
| `IsError` | Returns `True` if an expression is an error value |
| `IsMissing` | Returns `True` if an optional argument was not passed to a procedure |
| `IsNull` | Returns `True` if an expression contains a `Null` value |
| `IsNumeric` | Returns `True` if an expression can be evaluated as a number |
| `IsObject` | Returns `True` if an expression references an OLE Automation object |
| `Join` | Combines strings contained in an array |
| `LBound` | Returns the smallest subscript for a dimension of an array |

| Function | Action |
|---|---|
| LCase | Returns a string converted to lowercase |
| Left | Returns a specified number of characters from the left of a string |
| Len | Returns the number of characters in a string |
| Loc | Returns the current read or write position of a text file |
| LOF | Returns the number of bytes in an open text file |
| Log | Returns the natural logarithm of a number |
| LTrim | Returns a copy of a string with no leading spaces |
| Mid | Returns a specified number of characters from a string |
| Minute | Returns the minute of a time |
| MIRR | Returns the modified internal rate of return for a series of periodic cash flows |
| Month | Returns the month of a date as a number |
| MonthName | Returns the month of a date as a string |
| MsgBox | Displays a modal message box |
| Now | Returns the current system date and time |
| NPer | Returns the number of periods for an annuity |
| NPV | Returns the net present value of an investment |
| Oct | Converts from decimal to octal |
| Partition | Returns a string representing a range in which a value falls |
| Pmt | Returns a payment amount for an annuity |
| Ppmt | Returns the principal payment amount for an annuity |
| PV | Returns the present value of an annuity |
| QBColor | Returns a red/green/blue (RGB) color code |
| Rate | Returns the interest rate per period for an annuity |
| Replace | Returns a string in which a substring is replaced with another string |
| RGB | Returns a number representing an RGB color value |
| Right | Returns a specified number of characters from the right of a string |
| Rnd | Returns a random number between 0 and 1 |
| Round | Returns a rounded number |
| RTrim | Returns a copy of a string with no trailing spaces |
| Second | Returns the seconds portion of a specified time |
| Seek | Returns the current position in a text file |
| Sgn | Returns an integer that indicates the sign of a number |
| Shell | Runs an executable program |
| Sin | Returns the sine of a number |
| SLN | Returns the straight-line depreciation for an asset for a period |

*continued*

**Table B-2:** Summary of VBA Functions *(continued)*

| Function | Action |
|---|---|
| `Space` | Returns a string with a specified number of spaces |
| `Spc` | Positions output when printing to a file |
| `Split` | Returns a one-dimensional array containing a number of substrings |
| `Sqr` | Returns the square root of a number |
| `Str` | Returns a string representation of a number |
| `StrComp` | Returns a value indicating the result of a string comparison |
| `StrConv` | Returns a converted string |
| `String` | Returns a repeating character or string |
| `StrReverse` | Returns a string, reversed |
| `Switch` | Evaluates a list of Boolean expressions and returns a value associated with the first `True` expression |
| `SYD` | Returns the sum-of-years' digits depreciation of an asset for a period |
| `Tab` | Positions output when printing to a file |
| `Tan` | Returns the tangent of a number |
| `Time` | Returns the current system time |
| `Timer` | Returns the number of seconds since midnight |
| `TimeSerial` | Returns the time for a specified hour, minute, and second |
| `TimeValue` | Converts a string to a time serial number |
| `Trim` | Returns a string without leading spaces and/or trailing spaces |
| `TypeName` | Returns a string that describes the data type of a variable |
| `UBound` | Returns the largest available subscript for a dimension of an array |
| `UCase` | Converts a string to uppercase |
| `Val` | Returns the number formed from any initial numeric characters of a string |
| `VarType` | Returns a value indicating the subtype of a variable |
| `Weekday` | Returns a number indicating a day of the week |
| `WeekdayName` | Returns a string indicating a day of the week |
| `Year` | Returns the year of a date |