

VBA and Excel Object Model Quick Reference

VBA Statements Quick Reference

Legend

Plain courier text = required

[] = optional

| = or

Italics = user-defined

. . . = list of items

File and Folder Handling

STATEMENT	DESCRIPTION
ChDir <i>path</i>	Changes to the specified folder location.
ChDrive <i>drive</i>	Changes to the specified drive.
Close [<i>filenumber</i>]	Closes a file opened by using an Open statement.
FileCopy <i>source, destination</i>	Copies a file from the source to the specified destination.
Kill <i>pathname</i>	Deletes files from a disk. Use wildcards * for multiple characters and ? for single characters.
Lock [#] <i>filenumber</i> [, <i>recordrange</i>]	Locks all or a portion of an open file to prevent access by other processes.
MkDir <i>path</i>	Creates a new directory or folder.
Open <i>pathname</i> For mode [Access <i>access</i>][lock] As [#] <i>filenumber</i> [Len= <i>reclength</i>]	Opens the specified file to allow input/output operations.
Print # <i>filenumber</i> [, <i>outputlist</i>]	Writes display-formatted data sequentially to a file.
Put [#] <i>filenumber</i> , [<i>recnumber</i> ,] <i>varname</i>	Writes data contained in a variable to a disk file.
Reset	Closes all files opened using the Open statement.
Rmdir <i>path</i>	Removes the specified folder.
SetAttr <i>pathname, attributes</i>	Sets the attribute information for the specified file.
Unlock [#] <i>filenumber</i> [, <i>recordrange</i>]	Unlocks a file to allow access by other processes.
Width # <i>filenumber</i> , <i>width</i>	Assigns the output line width for a file opened using the Open statement.
Write # <i>filenumber</i> [, <i>outputlist</i>]	Writes data to a sequential text file.

Interaction

STATEMENT	DESCRIPTION
AppActivate <i>title</i> [, <i>wait</i>]	Activates an application window.
DeleteSetting <i>appname</i> , <i>section</i> [, <i>key</i>]	Deletes a section or key setting from an application's entry in the Windows Registry.
SaveSetting <i>appname</i> , <i>section</i> , <i>key</i> , <i>setting</i>	Saves an application entry in the application's entry in the Windows Registry.
SendKeys <i>string</i> [, <i>wait</i>]	Sends one or more keystrokes to the active window as if they were typed on the keyboard.

VBA Statements Quick Reference *(continued)*

Program Flow

STATEMENT	DESCRIPTION
[Public Private] Declare Sub name Lib "libname" [Alias "aliasname"] [(arglist)]	Declares a reference to an external DLL library function.
Do [{While Until} condition] [statements] Loop	Repeats a block of statements while or until a condition is true. The condition is checked at the beginning of the loop.
Do [statements] Loop [{While Until} condition]	Repeats a block of statements while or until a condition is true. Because the condition is checked at the end of the loop, the block of statements always executes at least once.
Exit Do For Function Property Sub	Exits the specified Do Loop, For Next, Function, Sub, or Property code.
For Each element In group [statements] Next [element]	Repeats a block of statements for each element in an array or collection.
For counter = start To end [Step step] [statements] Next [counter]	Repeats a section of code the specified number of times.
[Public Private Friend] [Static] Function name [(arglist)] [As type] [statements] [name = expression] End Function	Defines a procedure that returns a value.
If condition Then [statements] [Elseif condition-n Then] [elseifstatements]] [Else [elsestatements]] End If	Conditionally executes a block of statements based upon the value of an expression.
[Public Private Friend] [Static] Property Get name [(arglist)] [As type] [statements] [name = expression] End Property	Declares the name and arguments associated with a procedure.
[Public Private Friend] [Static] Property Let name ([arglist,] value) [statements] End Property	Declares the name and arguments of a procedure that assigns a value to a property.
[Public Private Friend] [Static] Property Set name ([arglist,] reference) [statements] End Property	Declares the name and arguments of a procedure that sets a reference to an object.

*continued**continued*

VBA and Excel Object Model Quick Reference (continued)

VBA Statements Quick Reference *(continued)*

Program Flow (continued)

STATEMENT	DESCRIPTION
Select Case <i>testexpression</i> [Case <i>expressionlist-n</i> [<i>statements-n</i>]] [Case Else [<i>elstatements</i>]] End Select	Executes one block out of a series of statement blocks depending upon the value of an expression.
[Private Public Friend] [Static] Sub name [(<i>arglist</i>)] [<i>statements</i>] End Sub	Declares the name, arguments, and code that form a Sub procedure.
While <i>condition</i> [<i>statements</i>] Wend	Executes a block of statements as long as the specified condition is true.
With <i>object</i> [<i>statements</i>] End With	Executes a block of statements on a single object or on a user-defined data type.

Variable Declaration

STATEMENT	DESCRIPTION
[Public Private] Const <i>constname</i> [As type] = <i>expression</i>	Declares a constant value.
Dim [WithEvents] <i>varname</i> [(<i>subscripts</i>)] [As [New] type]	Declares variables and allocates the appropriate storage space.
Friend [WithEvents] <i>varname</i> [(<i>subscripts</i>)] [As [New] type]	Declares a procedure or variable to only have scope in the project where it is defined.
Option Compare {Binary Text Database}	Specifies the default comparison method to use when comparing strings.
Option Explicit	Forces declaration of all variables within the module.
Option Private	Indicates that all code within the entire module is Private. This option is used by default. You can overwrite the effects of this option by declaring a specific procedure Public.
Private [WithEvents] <i>varname</i> [(<i>subscripts</i>)] [As [New] type]	Declares variables and procedures to only have scope within the current module.
Public [WithEvents] <i>varname</i> [(<i>subscripts</i>)] [As [New] type]	Declares variables and procedures to have scope within the entire project.

VBA Statements Quick Reference *(continued)*

Variable Declaration *(continued)*

STATEMENT	DESCRIPTION
ReDim [Preserve] <i>varname(subscripts)</i> [As <i>type</i>]	Changes the dimensions of a dynamic array.
[Private Public] Type <i>varname</i> <i>elementname</i> [(<i>subscripts</i>)] As <i>type</i> <i>elementname</i> [(<i>subscripts</i>)] As <i>type</i> ... End Type	Defines a custom data type.

VBA Function Quick Reference

Legend

Plain courier text = required [] = optional | = or
Italics = user-defined . . . = list of items

Array Functions

FUNCTION	DESCRIPTION	RETURNS
Array(<i>arg1, arg2, arg3, . . .</i>)	Creates a variant array containing the specified elements.	Variant
LBound(<i>arrayname</i> [, <i>dimension</i>])	Returns the smallest subscript for the specified array.	Long
UBound(<i>arrayname</i> [, <i>dimension</i>])	Returns the largest subscript for the specified array.	Long

Data Type Conversion Functions

FUNCTION	DESCRIPTION	RETURNS
Asc(<i>string</i>)	Returns the character code of the first letter in a string.	Integer
CBool(<i>expression</i>)	Converts an expression to Boolean data type (True or False)	Boolean
CByte(<i>expression</i>)	Converts an expression to Byte data type.	Byte
CCur(<i>expression</i>)	Converts an expression to Currency data type.	Currency
CDate(<i>expression</i>)	Converts an expression to a Date data type.	Date
CDBl(<i>expression</i>)	Converts an expression to Double data type.	Double
CDec(<i>expression</i>) (Decimal)	Converts an expression to a decimal value.	Variant
Chr(<i>charactercode</i>)	Converts the character code to the corresponding character. Chr(9) returns a tab, Chr(34) returns quotation marks, etc.	Variant
CInt(<i>expression</i>)	Converts an expression to an Integer data type, rounding any fractional parts.	Integer

continued



VBA and Excel Object Model Quick Reference (continued)

VBA Function Quick Reference *(continued)*

Data Type Conversion Functions (continued)

FUNCTION	DESCRIPTION	RETURNS
CLng(<i>expression</i>)	Converts an expression to the Long data type.	Long
CSng(<i>expression</i>)	Converts an expression to the Single data type.	Single
CStr(<i>expression</i>)	Returns a string containing the specified expression.	String
CVar(<i>expression</i>)	Converts any data type to a Variant data type. All numeric values are treated as Double data types and string expressions are treated as String data types.	Variant
Format(<i>expression</i> [, <i>format</i> [, <i>firstdayofweek</i> [, <i>firstweekofyear</i>]]])	Formats the expression using either predefined or user-defined formats.	Variant
FormatCurrency(<i>Expression</i> [, <i>NumDigitsAfterDecimal</i> [, <i>IncludeLeadingDigit</i> [, <i>UseParensForNegativeNumbers</i> [, <i>GroupDigits</i>]]]])	Formats the expression as a currency value using the system-defined currency symbol.	Currency
FormatDateTime(<i>Date</i> [, <i>NamedFormat</i>])	Formats an expression as a date and time.	Date
FormatNumber(<i>Expression</i> [, <i>NumDigitsAfterDecimal</i> [, <i>IncludeLeadingDigit</i> [, <i>UseParensForNegativeNumbers</i> [, <i>GroupDigits</i>]]]])	Formats the expression as a number.	Mixed
FormatPercent(<i>Expression</i> [, <i>NumDigitsAfterDecimal</i> [, <i>IncludeLeadingDigit</i> [, <i>UseParensForNegativeNumbers</i> [, <i>GroupDigits</i>]]]])	Returns the expression formatted as a percentage with a trailing % character.	String
Hex(<i>number</i>)	Converts a number to a hexadecimal value. Rounds numbers to nearest whole number before converting.	String
Oct(<i>number</i>)	Converts a number to an octal value. Rounds numbers to nearest whole number before converting.	Variant (String)
Str(<i>number</i>)	Converts a number to a string using the Variant data type.	Variant (String)
Val(<i>string</i>)	Returns the numeric portion of a string formatted as a number of the appropriate data type.	Mixed

VBA Function Quick Reference *(continued)*

Date and Time Functions

FUNCTION	DESCRIPTION	RETURNS
Date	Returns the current system date.	Date
DateAdd(<i>interval</i> , <i>number</i> , <i>date</i>)	Returns a date that is the specified interval of time from the original date.	Date
DateDiff(<i>interval</i> , <i>date1</i> , <i>date2</i> [, <i>firstdayofweek</i> [, <i>firstweekofyear</i>]])	Determines the time interval between two dates.	Long
DatePart(<i>interval</i> , <i>date</i> [, <i>firstdayofweek</i> [, <i>firstweekofyear</i>]])	Returns the specified part of a date.	Integer
DateSerial(<i>year</i> , <i>month</i> , <i>day</i>)	Converts the specified date to a serial number.	Date
DateValue(<i>date</i>)	Converts a string to a date.	Date
Day(<i>date</i>)	Returns a whole number between 1 and 31 representing the day of the month.	Integer
Hour(<i>time</i>)	Returns a whole number between 0 and 23 representing the hour of the day.	Integer
Minute(<i>time</i>)	Returns a whole number between 0 and 59 representing the minute of the hour.	Integer
Month(<i>date</i>)	Returns a whole number between 1 and 12 representing the month of the year.	Integer
Now	Returns the current system date and time.	Date
Second(<i>time</i>)	Returns a whole number between 0 and 59 representing the second of the minute.	Integer
Time	Returns the current system time.	Date
Timer	Indicates the number of seconds that have elapsed since midnight	Single
TimeSerial(<i>hour</i> , <i>minute</i> , <i>second</i>)	Creates a time using the specified hour, minute, and second values.	Date
TimeValue(<i>time</i>)	Converts a time to the serial number used to store time.	Date
WeekDay(<i>date</i> , [<i>firstdayofweek</i>])	Returns a whole number representing the first day of the week.	Integer
Year(<i>date</i>)	Returns a whole number representing the year portion of a date	Integer



VBA and Excel Object Model Quick Reference (continued)

VBA Function Quick Reference *(continued)*

File and Folder Handling Functions

FUNCTION	DESCRIPTION	RETURNS
CurDir(<i>drive</i>)	Returns the current path.	String
Dir[(<i>pathname</i> [, <i>attributes</i>])]	Returns the name of the file, directory, or folder that matches the specified pattern.	String
EOF(<i>filenumber</i>)	Returns -1 when the end of a file has been reached.	Integer
FileAttr(<i>filenumber</i> , <i>returntype</i>)	Indicates the file mode used for files opened with the Open statement.	Long
FileDateTime(<i>pathname</i>)	Indicates the date and time when a file was last modified.	Date
FileLen(<i>pathname</i>)	Indicates the length of a file in bytes.	Long
FreeFile(<i>rangenum</i>)	Returns the next file number available for use by the Open statement.	Integer
GetAttr(<i>pathname</i>)	Returns a whole number representing the attributes of a file, directory, or folder.	Integer
Input(<i>number</i> , [#] <i>filenumber</i>)	Returns a string containing the indicated number of characters from the specified file.	String
Loc(<i>filenumber</i>)	Indicates the current read/write position in an open file.	Long
LOF(<i>filenumber</i>)	Returns the size in bytes of a file opened using the Long Open statement.	Long
Seek(<i>filenumber</i>)	Specifies the current read/write position with a file opened with the Open statement.	Long

Financial Functions

FUNCTION	DESCRIPTION	RETURNS
DDB(<i>cost</i> , <i>salvage</i> , <i>life</i> , <i>period</i> [, <i>factor</i>])	Specifies the depreciation value for an asset during a specific time frame.	Double
FV(<i>rate</i> , <i>nper</i> , <i>pmt</i> [, <i>pv</i> [, <i>type</i>]])	Determines the future value of an annuity based on periodic fixed payments.	Double
IPmt(<i>rate</i> , <i>per</i> , <i>nper</i> , <i>pv</i> [, <i>fV</i> [, <i>type</i>]])	Determines the interest payment on an annuity for a specific period of time.	Double
IRR(<i>values</i> ()[, <i>guess</i>])	Determines the internal rate of returns for a series of cash flows.	Double
MIRR(<i>values</i> ()[, <i>finance_rate</i> , <i>reinvest_rate</i>])	Returns the modified interest rate of returns for a series of periodic cash flows	Double

VBA Function Quick Reference *(continued)*

Financial Functions (continued)

FUNCTION	DESCRIPTION	RETURNS
NPer(<i>rate</i> , <i>pmt</i> , <i>pv</i> [, <i>f</i> <i>v</i> [, <i>type</i>]])	Returns the number of periods for an annuity.	Double
NPV(<i>rate</i> , <i>values</i> ())	Returns the net present value of an investment.	Double
Pmt(<i>rate</i> , <i>nper</i> , <i>p</i> <i>v</i> [, <i>f</i> <i>v</i> [, <i>type</i>]])	Returns the payment amount for an annuity based on fixed payments.	Double
PPmt(<i>rate</i> , <i>per</i> , <i>nper</i> , <i>p</i> <i>v</i> [, <i>f</i> <i>v</i> [, <i>type</i>]])	Returns the principal payment amount for an annuity.	Double
PV(<i>rate</i> , <i>nper</i> , <i>p</i> <i>mt</i> [, <i>f</i> <i>v</i> [, <i>type</i>]])	Returns the present value of an annuity.	Double
Rate(<i>nper</i> , <i>p</i> <i>mt</i> , <i>p</i> <i>v</i> [, <i>f</i> <i>v</i> [, <i>type</i> [, <i>guess</i>]]])	Returns the interest rate per period for an annuity.	Double
SLN(<i>cost</i> , <i>salvage</i> , <i>life</i>)	Determines the straight-line depreciation of an asset for a single period.	Double
SYD(<i>cost</i> , <i>salvage</i> , <i>life</i> , <i>period</i>)	Determines the sum-of-years' digits depreciation of an asset for a specified period.	Double

Information Functions

FUNCTION	DESCRIPTION	RETURNS
CVErr(<i>errornumber</i>)	Returns a user-defined error number.	Variant
Error[(<i>errornumber</i>)]	Returns the error message for the specified error number.	String
IsArray(<i>varname</i>)	Indicates whether a variable contains an array.	Boolean
IsDate(<i>expression</i>)	Indicates whether an expression contains a date.	Boolean
IsEmpty(<i>expression</i>)	Indicates whether a variable has been initialized.	Boolean
IsError(<i>expression</i>)	Indicates whether an expression is an error value.	Boolean
IsMissing(<i>argname</i>)	Indicates whether an optional argument was passed to a procedure.	Boolean
IsNull(<i>expression</i>)	Indicates whether an expression contains no valid data.	Boolean
IsNumeric(<i>expression</i>)	Indicates whether an expression is a number.	Boolean
IsObject(<i>identifier</i>)	Indicates whether a variable references an object.	Boolean
TypeName(<i>varname</i>)	Specifies the variable type.	String
VarType(<i>varname</i>)	Specifies the subtype of a variable.	Integer



VBA and Excel Object Model Quick Reference (continued)

VBA Function Quick Reference *(continued)*

Interaction Functions

FUNCTION	DESCRIPTION	RETURNS
Choose(<i>index</i> , <i>choice-1</i> , [<i>choice-2</i> , ...])	Selects and returns a value from a list of arguments.	Mixed
DoEvents()	Yields execution so the operating system can process other events.	Integer
IIf(<i>expr</i> , <i>truepart</i> , <i>falsepart</i>)	Evaluates the expression and returns either the truepart or falsepart parameter value.	Mixed
InputBox(<i>prompt</i> [, <i>title</i>][, <i>default</i>][, <i>xpos</i>][, <i>ypos</i>][, <i>helpfile</i> , <i>context</i>])	Displays a dialog box prompting the user for input.	String
GetAllSettings(<i>appname</i> , <i>section</i>)	Returns a list of key settings and their values from the Windows Registry.	Variant
GetObject([<i>pathname</i>][, <i>class</i>])	Returns a reference to an object provided by an ActiveX Component.	Variant
GetSetting(<i>appname</i> , <i>section</i> , <i>key</i> [, <i>default</i>])	Returns a key setting value from an application's entry in the Windows registry.	Variant
MsgBox(<i>prompt</i> [, <i>buttons</i>][, <i>title</i>][, <i>helpfile</i> , <i>context</i>])	Displays a message box and returns a value representing the button pressed by the user	Integer
Partition(<i>number</i> , <i>start</i> , <i>stop</i> , <i>interval</i>)	Indicates where a number occurs within a series of ranges.	String
QBColor(<i>color</i>)	Returns the RGB color code for the specified color.	Long
Switch(<i>expr-1</i> , <i>value-1</i> [, <i>expr-2</i> , <i>value-2</i> ...])	Evaluates a list of expressions and returns the value associated with the first True expression.	Variant
RGB(<i>red</i> , <i>green</i> , <i>blue</i>)	Returns a number representing the RGB color value.	Long

Mathematical Functions

FUNCTION	DESCRIPTION	RETURNS
Abs(<i>number</i>)	Returns the absolute value of a number.	Mixed
Atn(<i>number</i>)	Returns the arctangent of a number.	Double
Cos(<i>number</i>)	Returns the cosine of an angle.	Double

VBA Function Quick Reference *(continued)*

Mathematical Functions (continued)

FUNCTION	DESCRIPTION	RETURNS
Exp(<i>number</i>)	Returns the base of the natural logarithms raised to a power.	Double
Fix(<i>number</i>)	Returns the integer portion of a number. With negative values, returns the first negative value greater than or equal to the number.	Integer
Int(<i>number</i>)	Returns the integer portion of a number. With negative values, returns the first negative number less than or equal to the number.	Integer
Log(<i>number</i>)	Returns the natural logarithm of a number.	Double
Round(<i>expression</i> [, <i>numdecimalplaces</i>])	Rounds a number to the specified number of decimal places.	Mixed
Rnd[(<i>number</i>)]	Returns a random number between 0 and 1.	Single
Sgn(<i>number</i>)	Returns 1 for a number greater than 0, 0 for a value of 0, and -1 for a number less than zero.	Integer
Sin(<i>number</i>)	Specifies the sine of an angle.	Double
Sqr(<i>number</i>)	Specifies the square root of a number.	Double
Tan(<i>number</i>)	Specifies the tangent of an angle.	Double

String Manipulation Functions

FUNCTION	DESCRIPTION	RETURNS
nStr([<i>start</i> ,] <i>string1</i> , <i>string2</i> [, <i>compare</i>])	Specifies the position of one string within another string.	Long
InStrRev(<i>stringcheck</i> , <i>stringmatch</i> [, <i>start</i> [, <i>compare</i>]])	Specifies the position of one string within another starting at the end of the string.	Long
LCase(<i>string</i>)	Converts a string to lowercase.	String
Left(<i>string</i> , <i>length</i>)	Returns the specified number of characters from the left side of a string.	String
Len(<i>string</i> <i>varname</i>)	Determines the number of characters in a string.	Long
Trim(<i>string</i>)	Trims spaces from the left side of a string.	String
Mid(<i>string</i> , <i>start</i> [, <i>length</i>])	Returns the specified number of characters from the center of a string.	String
Right(<i>string</i> , <i>length</i>)	Returns the specified number of characters from the right side of a string.	String
RTrim(<i>string</i>)	Trims spaces from the right side of a string.	String
Space(<i>number</i>)	Creates a string with the specified number of spaces.	String

continued

continued



VBA and Excel Object Model Quick Reference (continued)

VBA Function Quick Reference *(continued)*

String Manipulation Functions (continued)

FUNCTION	DESCRIPTION	RETURNS
<code>Spc(<i>n</i>)</code>	Positions output when printing to a file.	String
<code>Str(<i>number</i>)</code>	Returns a string representation of a number.	String
<code>StrComp(<i>string1</i>, <i>string2</i>[, <i>compare</i>])</code>	Returns a value indicating the result of a string comparison.	Integer
<code>StrConv(<i>string</i>, <i>conversion</i>, <i>LCID</i>)</code>	Converts a string to the specified format.	String
<code>String(<i>number</i>, <i>character</i>)</code>	Creates a string by repeating a character the specified number of times.	String
<code>Tab[(<i>n</i>)]</code>	Positions output when printing to a file.	String
<code>Trim(<i>string</i>)</code>	Trims spaces from left and right of a string.	String
<code>UCase(<i>string</i>)</code>	Converts a string to uppercase.	String

VBA Function Constants and Characters

vbMsgBoxStyle Constants (MsgBox Function)

CONSTANT	VALUE	DESCRIPTION
<code>vbAbortRetryIgnore</code>	2	Displays Abort, Retry, and Ignore buttons.
<code>vbApplicationModal</code>	0	Creates application modal message box.
<code>vbCritical</code>	16	Displays Critical Message icon.
<code>vbDefaultButton1</code>	0	Makes first button default.
<code>vbDefaultButton2</code>	256	Makes second button default.
<code>vbDefaultButton3</code>	512	Makes third button default.
<code>vbDefaultButton4</code>	768	Makes fourth button default.
<code>vbExclamation</code>	48	Displays Warning Message icon.
<code>vbInformation</code>	64	Displays Information Message icon.
<code>vbMsgBoxHelpButton</code>	16384	Adds a Help button.

VBA Function Constants and Characters *(continued)*

vbMsgBoxStyle Constants (MsgBox Function) (continued)

CONSTANT	VALUE	DESCRIPTION
vbMsgBoxRight	524288	Right aligns text in the box.
vbMsgBoxRtlReading	1048576	Used only with Hebrew and Arabic systems for right-to-left reading.
vbMsgBoxSetForeground	65536	Makes message box the foreground window.
vbOKCancel	1	Displays OK and Cancel buttons.
vbOKOnly	0	Displays only the OK button.
vbQuestion	32	Displays Warning Query icon.
vbRetryCancel	5	Displays Retry and Cancel buttons.
vbSystemModal	4096	Creates a system modal message box.
vbYesNo	4	Displays Yes and No buttons.
vbYesNoCancel	3	Displays Yes, No, and Cancel buttons.

vbDayOfWeek Constants

CONSTANT	VALUE	DESCRIPTION
vbUseSystemDayOfWeek	0	Uses the system defined first day of week.
vbSunday	1	Sunday (default).
vbMonday	2	Monday.
vbTuesday	3	Tuesday.
vbWednesday	4	Wednesday.
vbThursday	5	Thursday.
vbFriday	6	Friday.
vbSaturday	7	Saturday.

vbFirstWeekOfYear Constants

CONSTANT	VALUE	DESCRIPTION
vbUseSystem	0	Uses system defined first week of year.
vbFirstJan1	1	Starts with week in which January 1 occurs (default).
vbFirstFourDays	2	Starts with the first week that has at least four days in the new year.
vbFirstFullWeek	3	Starts with first full week of the year.



VBA and Excel Object Model Quick Reference (continued)

VBA Function Constants and Characters *(continued)*

Format Function Characters

DATE/TIME CHARACTERS	DISPLAYS
d	Day with no leading zero.
ddd	Three-letter abbreviation of day (Sun. – Sat.).
dddd	Full day name (Sunday).
dddddd	Complete date using short date format.
dddddd	Complete date using long date format.
w	Day of week as number (1 for Sunday).
ww	Week of year as number.
m	Month with no leading zero.
mmm	Three letter abbreviation of month (Jan.-Dec.).
mmmm	Complete month name.
q	Quarter of year.
y	Day of year as number.
yy	Year as 2-digit number.
yyyy	Year as 4-digit number.
h	Hour with no leading zero.
n	Minutes with no leading zero.
s	Seconds with no leading zero.
ttttt	Complete time using system time format.
c	Date as dddddd and time as ttttt .

Format Function Predefined Formats

FORMAT	DESCRIPTION
General Date	Uses general date format.
Long Date	Uses system-defined long date, such as Tuesday, August 7, 2007.
Medium Date	Uses the medium date format, such as 07-Aug-07.
Short Date	Uses system-defined short date, such as 8/7/2007.
Long Time	Uses system-defined long time, such as 5:45:30 P.M.
Medium Time	Uses the medium time format, such as 05:45 P.M.
Short Time	Uses the short time format, such as 17:45.

VBA Function Constants and Characters *(continued)*

Format Function Predefined Formats *(continued)*

FORMAT	DESCRIPTION
General Number	Uses the general number format.
Currency	Places the appropriate currency symbol in front of the number.
Fixed	Uses a fixed decimal format.
Standard	Uses standard formatting.
Percent	Converts the expression to a percentage.
Scientific	Displays the expression using scientific notation.
Yes/No	Converts the expression to a Yes or No value.
True/False	Converts the expression to a True or False value.
On/Off	Converts the expression to an On or Off value.

Excel Object Model Constants

XlColumnDataType Constants

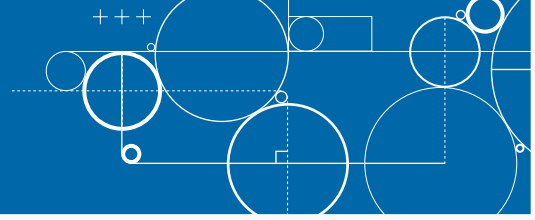
CONSTANT	VALUE	DESCRIPTION
xlDMYFormat	4	DMY format date.
xlDYMFormat	7	DYM format date.
xlEMDFormat	10	EMD format date.
xlGeneralFormat	1	General format.
xlMDYFormat	3	MDY format date.
xlMYDFormat	6	MYD format date.
xlSkipColumn	9	Skip Column.
xlTextFormat	2	Text format.
xlYDMFormat	8	YDM format date.
xlYMDFormat	5	YMD format date.

XlFileFormat Constants

CONSTANT	VALUE	DESCRIPTION
xlAddIn	18	Excel add-in.
xlAddIn8	18	Excel 2007 Add-In
xlCSV	6	Comma-separated values format.
xlCSVMac	22	Macintosh comma-separated values format.

*continued**continued* →

VBA and Excel Object Model Quick Reference (continued)



Excel Object Model Constants (continued)

XlFileFormat Constants (continued)

CONSTANT	VALUE	DESCRIPTION
xlCSVMSDOS	24	MSDOS comma-separated values format.
xlCSVWindows	23	MS Windows comma-separated values format.
xlCurrentPlatformText	-4158	Text file based on current operating system.
xlDBF2	7	DBase II format.
xlDBF3	8	DBase III format.
xlDBF4	11	DBase IV format.
xlDIF	9	Data interchange format.
xlExcel12	50	Excel 12 format.
xlExcel2	16	Excel 2
xlExcel2FarEast	27	Excel 2.0 format – Far East version.
xlExcel3	29	Excel 3.0 format.
xlExcel4	33	Excel 4.0 format.
xlExcel4Workbook	35	Excel 4.0 workbook format.
xlExcel5	39	Excel 5.0 format.
xlExcel7	39	Excel 97 format.
xlExcel9597	43	Excel 95 – 97 format.
xlHtml	44	HTML format.
xlIntlAddIn	26	Excel international Add-in.
xlIntlMacro	25	Excel international macro.
xlOpenXMLAddin	55	Open XML Add-In.
xlOpenXMLTemplate	54	Open XML Template.
xlOpenXMLTemplateMacroEnabled	53	Open XML Template Macro Enabled.
xlOpenXMLWorkbook	51	Open XML Workbook.
xlOpenXMLWorkbookzMacroEnabled	52	Open XML Workbook Enabled.
xlSYLK	2	Symbolic link format.
xlTemplate	17	Template file format.
xlTemplate8	17	Template.
xlTextMac	19	Macintosh text file format.
xlTextMSDOS	21	MSDOS text file format.
xlTextPrinter	36	Text file created for a printer (.prn).
xlTextWindows	20	MS Window text file format.

Excel Object Model Constants (continued)

XlFileFormat Constants (continued)

CONSTANT	VALUE	DESCRIPTION
xlUnicodeText	42	Unicode text file format.
xlWebArchive	45	Web archive format (.mht).
xlWJ2WD1	14	WJ2WD1
xlWJ3	40	WJ3
xlWJ3FM3	41	WJ3FJ3
xlWK1	5	Lotus 2.x format.
xlWK1ALL	31	Lotus 2.x .all format.
xlWK1FMT	30	Lotus 2.x .fmt format.
xlWK3	15	Lotus 3.x format.
xlWK3FM3	32	Lotus 3.x and Lotus 1-2-3 for Windows format.
xlWK4	38	Lotus 4.0 format.
xlWKS	4	MS Works file format.
xlWorkBookDefault	51	Workbook default
xlWorkbookNormal	-4143	Excel workbook format.
xlWorks2FarEast	28	MS Works file – Far East format.
xlWQ1	34	Quattro Pro for MSDOS format.
xlXMLSpreadsheet	46	XML format.

MsoFileType Constants

CONSTANT	VALUE	DESCRIPTION
msoFileTypeAllFiles	1	All file types.
msoFileTypeBinders	6	Microsoft Office Binder file.
msoFileTypeCalendarItem	11	Microsoft Outlook Calendar item.
msoFileTypeContactItem	12	Microsoft Outlook Contact item.
msoFileTypeDatabases	7	Database files.
msoFileTypeDataConnectionFiles	17	Database connection files.
msoFileTypeDesignerFiles	22	Designer files.
msoFileTypeDocumentImagingFiles	20	Document imaging files.
msoFileTypeExcelWorkbooks	4	Microsoft Excel Workbooks.
msoFileTypeJournalItem	14	Journal items.
msoFileTypeMailItem	10	Microsoft Outlook Mail message.
msoFileTypeNoteItem	13	Microsoft Outlook Note item.
msoFileTypeOfficeFiles	2	All Microsoft Office file types.

continued

continued →

VBA and Excel Object Model Quick Reference (continued)



Excel Object Model Constants (continued)

MsoFileTypeConstant (continued)

CONSTANT	VALUE	DESCRIPTION
msoFileTypeOutlookItems	9	Microsoft Outlook files.
msoFileTypePhotoDrawFiles	16	Microsoft PhotoDraw files.
msoFileTypePowerPointPresentations	5	Microsoft PowerPoint files.
msoFileTypeProjectFiles	19	Microsoft Project files.
msoFileTypePublisherFiles	18	Microsoft Publisher files.
msoFileTypeTaskItem	15	Microsoft Outlook Task item.
msoFileTypeTemplates	8	Template files.
msoFileTypeVisioFiles	21	Visio files.
msoFileTypeWebPages	23	Web pages including .htm, .asp, and .mht files.
msoFileTypeWordDocuments	3	Microsoft Word documents.

XlChartType Constants

CONSTANT	VALUE	CHART TYPE
xl3DArea	-4098	3D Area.
xl3DAreaStacked	78	3D Stacked Area.
xl3DAreaStacked100	79	100% Stacked Area.
xl3DBarClustered	60	3D Clustered Bar.
xl3DBarStacked	61	3D Stacked Bar.
xl3DBarStacked100	62	3D 100% Stacked Bar.
xl3DColumn	-4100	3D Column.
xl3DColumnClustered	54	3D Clustered Column.
xl3DColumnStacked	55	3D Stacked Column.
xl3DColumnStacked100	56	3D 100% Stacked Column.
xl3DLine	-4101	3D Line.
xl3DPie	-4102	3D Pie.
xl3DPieExploded	70	Exploded 3D Pie.
xlArea	1	Area.
xlAreaStacked	76	Stacked Area.
xlAreaStacked100	77	100% Stacked Area.
xlBarClustered	57	Clustered Bar.

Excel Object Model Constants (continued)

XlChartType Constants (continued)

CONSTANT	VALUE	CHART TYPE
xlBarOfPie	71	Bar of Pie.
xlBarStacked	58	Stacked Bar.
xlBarStacked100	59	100% Stacked Bar.
xlBubble	15	Bubble.
xlBubble3DEffec	87	Bubble with 3D effects.
xlColumnClustered	51	Clustered Column.
xlColumnStacked	52	Stacked Column.
xlColumnStacked100	53	100% Stacked Column.
xlConeBarClustered	102	Clustered Cone Bar.
xlConeBarStacked	103	Stacked Cone Bar.
xlConeBarStacked100	104	100% Stacked Cone Bar.
xlConeCol	105	3D Cone Column.
xlConeColClustered	99	Clustered Cone Column.
xlConeColStacked	100	Stacked Cone Column.
xlConeColStacked100	101	100% Stacked Cone Column.
xlCylinderBarClustered	95	Clustered Cylinder Bar.
xlCylinderBarStacked	96	Stacked Cylinder Bar.
xlCylinderBarStacked100	97	100% Stacked Cylinder Bar.
xlCylinderCol	98	3D Cylinder Column.
xlCylinderColClustered	92	Clustered Cone Column.
xlCylinderColStacked	93	Stacked Cone Column.
xlCylinderColStacked100	94	100% Stacked Cylinder Column.
xlDoughnut	-4120	Doughnut.
xlDoughnutExploded	80	Exploded Doughnut.
xlLine	4	Line.
xlLineMarkers	65	Line with Markers.
xlLineMarkersStacked	66	Stacked Line with Markers.
xlLineMarkersStacked100	67	100% Stacked Line with Markers.
xlLineStacked	63	Stacked Line.
xlLineStacked100	64	100% Stacked Line.
xlPie	5	Pie.
xlPieExploded	69	Exploded Pie.
xlPieOfPie	68	Pie of Pie.
xlPyramidBarClustered	109	Clustered Pyramid Bar.

continued

continued →

VBA and Excel Object Model Quick Reference (continued)

Excel Object Model Constants (continued)

XlChartType Constants (continued)

CONSTANT	VALUE	CHART TYPE
xlPyramidBarStacked	110	Stacked Pyramid Bar.
xlPyramidBarStacked100	111	100% Stacked Pyramid Bar.
xlPyramidCol	112	3D Pyramid Column.
xlPyramidColClustered	106	Clustered Pyramid Column.
xlPyramidColStacked	107	Stacked Pyramid Column.
xlPyramidColStacked100	108	100% Stacked Pyramid Column.
xlRadar	-4151	Radar.
xlRadarFilled	82	Filled Radar.
xlRadarMarkers	81	Radar with Data Markers.
xlStockHLC	88	High-Low-Close.
xlStockOHLC	89	Open-High-Low-Close.
xlStockVHLC	90	Volume-High-Low-Close.
xlStockVOHLC	91	Volume-Open-High-Low-Close.
xlSurface	83	3D Surface.
xlSurfaceTopView	85	Top View Surface.
xlSurfaceTopViewWireframe	86	Top View Wireframe Surface.
xlSurfaceWireframe	84	3D Surface Wireframe.
xlXYScatter	-4169	Scatter.
xlXYScatterLines	74	Scatter with Lines.
xlXYScatterLinesNoMarkers	75	Scatter with Lines and No Data Markers.
xlXYScatterSmooth	72	Scatter with Smoothed Lines.
xlXYScatterSmoothNoMarkers	73	Scatter with Smoothed Lines and No Data Markers.

XlLineStyle Constants

CONSTANT	VALUE	DESCRIPTION
xlContinuous	1	Continuous solid line.
xlDash	-4155	Dashed line.
xlDashDot	4	Line with the pattern dash dot.
xlDashDotDot	5	Line with the pattern dash dot dot.
xlDot	-4118	Dotted line.

Excel Object Model Constants (continued)

XLineStyle Constants (continued)

CONSTANT	VALUE	DESCRIPTION
xlDouble	-4119	Double solid line.
xlSlantDashDot	13	Slanted line with the pattern dash dot.
xlLineStyleNone	-4142	No line.

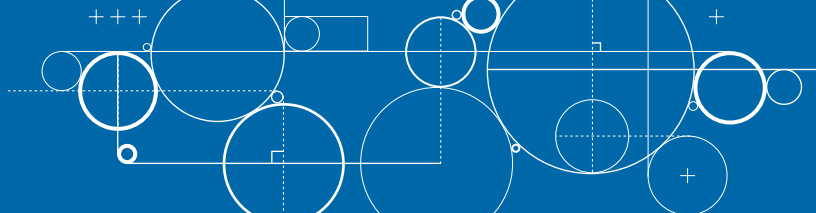
XlBorderWeight Constants

CONSTANT	VALUE	DESCRIPTION
xlHairline	1	Creates a very thin line.
xlMedium	-4138	Creates a medium width line.
xlThick	4	Creates a thick line.
xlThin	2	Creates a thin line.

XlPattern Constants

CONSTANT	VALUE	DESCRIPTION
xlPatternAutomatic	-4105	System default.
xlPatternChecker	9	Checkered pattern.
xlPatternCrissCross	16	Criss-cross pattern.
xlPatternDown	-4121	Downward pattern.
xlPatternGray25	-4124	25% gray pattern.
xlPatternGray50	-4125	50% gray pattern.
xlPatternGray75	-4126	75% gray pattern.
xlPatternGrid	15	Grid pattern.
xlPatternHorizontal	-4128	Horizontal pattern.
xlPatternLightHorizontal	11	Light horizontal pattern.
xlPatternLightVertical	12	Light vertical pattern.
xlPatternLightDown	13	Light downward pattern.
xlPatternLightUp	14	Light upward pattern.
xlPatternNone	-4142	No pattern.
xlPatternSemiGray75	10	75% semi-gray pattern.
xlPatternSolid	1	Solid color, no pattern.
xlPatternUp	-4162	Upward pattern.
xlPatternVertical	-4166	Vertical pattern.

Ribbon Controls Quick Reference



XML Controls

XML controls specific to the Ribbon.

CONTROL	DESCRIPTION	COMMON ATTRIBUTES	CHILDREN
customUI	The root tag for Ribbon customizations.	None	commands, ribbon
commands	Globally repurposed commands.	None	command
command	Represents the command that you are repurposing.	enabled, getEnabled, idMso (required), onAction	contextualTabs, officeMenu, qat, tabs
contextualTabs	The contextual tabs that display in Excel. For example, the Chart tools.	None	tabSet
tabSet	A collection of tab controls.	getVisible, idMso (required), visible	tab
qat	The Quick Access Toolbar. Used only in the start from scratch mode.	None	documentControls, sharedControls
sharedControls	Controls shared across documents. In general, you should use documentControls, not sharedControls.	None	button, control, separator
documentControls	Controls specific to a document.	None	button, control, separator
officeMenu	Microsoft Office menu controls.	None	button, checkbox, control, dynamicMenu, gallery, menu, menuSeparator, splitButton, toggleButton
tabs	Container for tab controls.	None	tab
tab	A tab on the Ribbon.	getKeytip, getLabel, getVisible, id, idMso, idQ, insertAfterMso, insertAfterQ, insertBeforeMso, insertBeforeQ, keytip, label, tag, visible	group
group	A group on a tab on the Ribbon.	getImage, getImageMso, getKeytip, getLabel, getScreentip, getSupertip, getVisible, id, idMso, idQ, image, imageMso, insertAfterMso, insertAfterQ, insertBeforeMso, insertBeforeQ, keytip, label, screentip, supertip, visible	box, button, buttonGroup, checkBox, comboBox, control, dialogBoxLauncher, dropDown, editBox, gallery, labelControl, menu, separator, splitButton, toggleButton

CONTROL	DESCRIPTION	COMMON ATTRIBUTES	CHILDREN
box	Use to arrange controls within a group.	getVisible, id, idQ, insertAfterMso, insertAfterQ, insertBeforeMso, insertBeforeQ, visible	box, button, buttonGroup, checkBox, comboBox, control, dropdown, dynamicMenu, editBox, gallery, labelControl, menu, splitButton, toggleButton
button	Use to represent a button control.	description, enabled, getDescription, getEnabled, getImage, getImageMso, getKeytip, getLabel, getScreentip, getShowImage, getShowLabel, getSize, getSupertip, getVisible, id, idMso, idQ, image, imageMso, insertAfterMso, insertAfterQ, insertBeforeMso, insertBeforeQ, keytip, label, onAction, screentip, showImage, showLabel, size, supertip, tag, visible	None
buttonGroup	Use to create a grouping of buttons.	getVisible, id, idQ, insertAfterMso, insertAfterQ, insertBeforeMso, insertBeforeQ, visible	button, control, dynamicMenu, gallery, menu, splitButton, toggleButton
checkbox	Use to create a check box control.	description, enabled, getDescription, getEnabled, getKeytip, getLabel, getScreentip, getSupertip, getVisible, id, idMso, idQ, insertAfterMso, insertAfterQ, insertBeforeMso, insertBeforeQ, keytip, label, onAction, screentip, supertip, tag, visible	None
comboBox	Use to create a combo box control.	enabled, getEnabled, getImage, getImageMso, getKeytip, getLabel, getScreentip, getShowImage, getShowLabel, getSupertip, getVisible, id, idMso, idQ, image, imageMso, insertAfterMso, insertAfterQ, insertBeforeMso, insertBeforeQ, label, screentip, showImage, showLabel, supertip, tag, visible, Shared with editBox: getText, maxLength, onChange, sizeString	item
dialogBox Launcher	Use to create a dialog box launcher for a group. A group control can only have one dialog box launcher, the control must contain a button control, and must be the final element in the group element.	None	button (required)

*continued**continued*

Ribbon Controls Quick Reference (continued)

XML Basic Controls *(continued)*

CONTROL	DESCRIPTION	COMMON ATTRIBUTES	CHILDREN
dropdown	Use to create a drop-down list box.	enabled, getEnabled, getImage, getImageMso, getKeytip, getLabel, getScreenTip, getShowImage, getShowLabel, getSupertip, getVisible, id, idMso, idQ, image, imageMso, insertAfterMso, insertAfterQ, insertBeforeMso, insertBeforeQ, keytip, label, onAction, screenTip, showImage, showLabel, supertip, tag, visible Shared with comboBox: getItemCount, getItemID, getItemImage, getItemLabel, getItemScreenTip, getItemSupertip, showItemImage, Shared with editBox: sizeString	item
dynamicMenu	Use to create a menu at run time.	description, enabled, getDescription, getEnabled, getImage, getImageMso, getKeytip, getLabel, getScreenTip, getShowImage, getShowLabel, getSize, getSupertip, getVisible, id, idMso, idQ, image, imageMso, insertAfterMso, insertAfterQ, insertBeforeMso, insertBeforeQ, keytip, label, screenTip, showImage, showLabel, supertip, tag, visible	Same as a menu, but is populated by using the getContent callback.
editBox	Use to create an edit box control.	enabled, getEnabled, getImage, getImageMso, getKeytip, getLabel, getScreenTip, getShowImage, getShowLabel, getSupertip, getVisible, id, idMso, idQ, image, imageMso, insertAfterMso, insertAfterQ, insertBeforeMso, insertBeforeQ, keytip, label, screenTip, showImage, showLabel, supertip, tag, visible	None
gallery	Use to create a gallery control.	description, enabled, getDescription, getEnabled, getImage, getImageMso, getKeytip, getLabel, getScreenTip, getShowImage, getShowLabel, getSize, getSupertip, getVisible, id, idMso, idQ, image, imageMso, insertAfterMso, insertAfterQ, insertBeforeMso, insertBeforeQ, keytip, label, onAction, screenTip, showImage, showLabel, size, supertip, tag, visible, Shared with comboBox: getItemCount, getItemID, getItemImage, getItemLabel, getItemScreenTip, getItemSupertip, showItemImage, showItemLabel, Shared with dropDown: getSelectedItemID, getSelectedItem Index Shared with editBox: sizeString	item, button. Buttons must be listed after the items, and all buttons appear at the bottom of the gallery.
item	A static gallery, dropDown, or comboBox item. If you specify static items, you cannot also specify dynamic items.	id, image, imageMso, label, screenTip, supertip	None

XML Basic Controls *(continued)*

CONTROL	DESCRIPTION	COMMON ATTRIBUTES	CHILDREN
labelControl	Use to create a label control.	enabled, getEnabled, getLabel, getScreentip, getShowLabel, getSupertip, getVisible, id, idMso, idQ, insertAfterMso, insertAfterQ, insertBeforeMso, insertBeforeQ, label, screentip, showLabel, supertip, tag, visible	None
menu	Use to create a menu control.	description, enabled, getDescription, getEnabled, getImage, getImageMso, getKeytip, getLabel, getScreentip, getShowImage, getShowLabel, getSize, getSupertip, getVisible, id, idMso, idQ, image, imageMso, insertAfterMso, insertAfterQ, insertBeforeMso, insertBeforeQ, keytip, label, screentip, showImage, showLabel, size, supertip, tag, visible	button, checkBox, control, dynamicMenu, gallery, menu, menuSeparator, splitButton, toggleButton
menu Separator	Use to create a separator line (which can optionally include a text label) between menu items.	id, idQ, insertAfterMso, insertAfterQ, insertBeforeMso, insertBeforeQ	None
separator	Use to create a separator line between controls.	getVisible, id, idQ, insertAfterMso, insertAfterQ, insertBeforeMso, insertBeforeQ, visible	None
splitButton	Use to create a split button control.	enabled, getEnabled, getKeytip, getShowLabel, getSize, getSupertip, getVisible, id, idMso, idQ, insertAfterMso, insertAfterQ, insertBeforeMso, insertBeforeQ, keytip, showLabel (determines whether the button or toggle button control displays its label), size, supertip, tag, visible	button or toggleButton (required, only one permitted, and must appear before the menu): The main button for the split button control menu (required, and only one permitted): The menu of a split button control.
toggleButton	Use to create a toggle button control.	description, enabled, getDescription, getEnabled, getImage, getImageMso, getKeytip, getLabel, getPressed, getScreentip, getShowImage, getShowLabel, getSize, getSupertip, getVisible, id, idMso, idQ, image, imageMso, insertAfterMso, insertAfterQ, insertBeforeMso, insertBeforeQ, keytip, label, onAction, screentip, showImage, showLabel, size, supertip, tag, visible	None

Ribbon Controls Quick Reference (continued)

Attributes and Methods of Ribbon Controls

The following tables list the attributes and methods related to specific ribbon controls.

CONTROL	ATTRIBUTE OR METHOD	TYPE OR ACTION	DESCRIPTION
customUI	xmlns	String	You must set xmlns to http://schemas.microsoft.com/office/2006/01/customui
customUI	onLoad	callback	As the Ribbon load passes a Ribbon parameter to the callback procedure. This enables the associated code to store a reference to the Ribbon for later use.
customUI	loadImage	callback	Use to create a procedure to load all of the images required by the Ribbon.
CONTROL	ATTRIBUTE	VALUES	DESCRIPTION
ribbon	startFromScratch	True, False, 1, 0	Set to True, to hide built-in Ribbon tabs and display a minimal File menu.
CONTROL	ATTRIBUTE	VALUES	DESCRIPTION
box	boxStyle	Horizontal, Vertical	Sets the flow of the controls inside a box.
CONTROL	ATTRIBUTE	TYPE OR ACTION	DESCRIPTION
checkBox	getPressed	callback	Use to specify whether the checkBox control is pressed.
CONTROL	METHOD	ATTRIBUTE OR METHOD	DESCRIPTION
comboBox	getItemCount	callback	Returns the number of items in a comboBox.
comboBox	getItemID	callback	Returns the ID of for the item.
comboBox	getItemImage	callback	Returns the image for the item.
comboBox	getItemLabel	callback	Returns the label of for the item.
comboBox	getItemScreenTip	callback	Returns the ScreenTip of for the item.
comboBox	getItemSupertip	callback	Returns the Enhanced ScreenTip for the item.
comboBox	showItemImage	True, False, 1, 0	Specifies whether to display the item image.
CONTROL	METHOD	ACTION	DESCRIPTION
dropdown	getSelectedItemID	callback	Asks for the item that should be selected by ID. Specify either this attribute or the getSelectedItemIndex attribute, but not both.
dropdown	getSelectedItemIndex	callback	Asks for the item that should be selected by index. Specify either this attribute or the getItemID attribute, but not both.
dropdown	showItemLabel	True, False, 1, 0	Indicates whether items should display labels.

Attributes and Methods of Ribbon Controls *(continued)*

CONTROL	METHOD	ACTION	DESCRIPTION
dynamicMenu	getContent	callback	Returns an XML string that contains the contents of the dynamic menu.
CONTROL	ATTRIBUTE OR METHOD	TYPE OR ACTION	DESCRIPTION
editBox	getText	callback	Returns the text that displays in the edit box.
editBox	maxLength	Integer	The maximum number of characters that a user can type in a edit box.
editBox	onChange	callback	Called when the value in the edit box changes.
editBox	sizeString	String	A string, such as “xxxxxxxxxx”. Determines the size of an edit box.
CONTROL	ATTRIBUTE OR METHOD	TYPE OR ACTION	DESCRIPTION
gallery	columns	Integer	The number of columns in a gallery.
gallery	getItemHeight	callback	Requests the height of items, in pixels.
gallery	getItemWidth	callback	Requests the width of items, in pixels.
gallery	itemHeight	Integer	The height of items, in pixels.
gallery	itemWidth	Integer	The width of items, in pixels.
gallery	rows	Integer	The number of rows in a gallery.
CONTROL	ATTRIBUTE	VALUES	DESCRIPTION
menu	itemSize	Normal, Large	The size of an item. The Description property shows for Large menu items.
CONTROL	ATTRIBUTE	TYPE OR ACTION	DESCRIPTION
menuSeparator	title	String	The text for this separator.
menuSeparator	getTitle	callback	Callback for this separator’s text.
CONTROL	ATTRIBUTE	TYPE OR ACTION	DESCRIPTION
toggleButton	getPressed	callback	Enables you to specify whether the toggle button control is pressed.

Ribbon Controls Quick Reference (continued)

Callbacks

The following table lists all of the callbacks used by RibbonX.

CONTROL	CALLBACK NAME	SIGNATURES
(several controls)	getDescription	Sub GetDescription (control As IRibbonControl, ByRef description)
(several controls)	getEnabled	Sub GetEnabled (control As IRibbonControl, ByRef enabled)
(several controls)	getImage	Sub GetImage (control As IRibbonControl, ByRef image)
(several controls)	getImageMso	Sub GetImageMso (control As IRibbonControl, ByRef imageMso)
(several controls)	getLabel	Sub GetLabel (control As IRibbonControl, ByRef label)
(several controls)	getKeytip	Sub GetKeytip (control As IRibbonControl, ByRef label)
(several controls)	getSize	Sub GetSize (control As IRibbonControl, ByRef size)
(several controls)	getScreentip	Sub GetScreentip (control As IRibbonControl, ByRef screentip)
(several controls)	getSupertip	Sub GetSupertip (control As IRibbonControl, ByRef screentip)
(several controls)	getVisible	Sub GetVisible (control As IRibbonControl, ByRef visible)
button	getShowImage	Sub GetShowImage (control As IRibbonControl, ByRef showImage)
button	getShowLabel	Sub GetShowLabel (control As IRibbonControl, ByRef showLabel)
button	onAction – repurposed	Sub OnAction (control As IRibbonControl, byRef CancelDefault)
button	onAction	Sub OnAction (control As IRibbonControl)
checkBox	getPressed	Sub GetPressed (control As IRibbonControl, ByRef returnValue)
checkBox	onAction	Sub OnAction (control As IRibbonControl, pressed As Boolean)(pvarfPressed)
comboBox	getItemCount	Sub GetItemCount (control As IRibbonControl, ByRef count)
comboBox	getItemID	Sub GetItemID (control As IRibbonControl, index As Integer, ByRef id)
comboBox	getItemImage	Sub GetItemImage (control As IRibbonControl, index As Integer, ByRef image)
comboBox	getItemLabel	Sub GetItemLabel (control As IRibbonControl, index As Integer, ByRef label)
comboBox	getItemScreenTip	Sub GetItemScreenTip (control As IRibbonControl, index As Integer, ByRef screentip)
comboBox	getItemSuperTip	Sub GetItemSuperTip (control As IRibbonControl, index As Integer, ByRef supertip)
comboBox	getText	Sub GetText (control As IRibbonControl, ByRef text)
comboBox	onChange	Sub OnChange (control As IRibbonControl, text As String)
customUI	loadImage	Sub LoadImage (imageId As string, ByRef image)
customUI	onLoad	Sub OnLoad (ribbon As IRibbonUI)
dropDown	getItemCount	Sub GetItemCount (control As IRibbonControl, ByRef count)

CONTROL	CALLBACK NAME	SIGNATURES
dropDown	getItemID	Sub GetItemID (control As IRibbonControl, index As Integer, ByRef id)
dropDown	getItemImage	Sub GetItemImage (control As IRibbonControl, index As Integer, ByRef image)
dropDown	getItemLabel	Sub GetItemLabel (control As IRibbonControl, index As Integer, ByRef label)
dropDown	getItemScreenTip	Sub GetItemScreenTip (control As IRibbonControl, index As Integer, ByRef screenTip)
dropDown	getItemSuperTip	Sub GetItemSuperTip (control As IRibbonControl, index As Integer, ByRef superTip)
dropDown	getSelectedItemID	Sub GetSelectedItemID (control As IRibbonControl, ByRef index)
dropDown	getSelectedItemIndex	Sub GetSelectedItemIndex (control As IRibbonControl, ByRef index)
dropDown	onAction	Sub OnAction (control As IRibbonControl, selectedId As String, selectedIndex As Integer)
dynamicMenu	getContent	Sub GetContent (control As IRibbonControl, ByRef content)
editBox	getText	Sub GetText (control As IRibbonControl, ByRef text)
editBox	onChange	Sub OnChange (control As IRibbonControl, text As String)
gallery	getItemCount	Sub GetItemCount (control As IRibbonControl, ByRef count)
gallery	getItemHeight	Sub getItemHeight (control As IRibbonControl, ByRef height)
gallery	getItemID	Sub GetItemID (control As IRibbonControl, index As Integer, ByRef id)
gallery	getItemImage	Sub GetItemImage (control As IRibbonControl, index As Integer, ByRef image)
gallery	getItemLabel	Sub GetItemLabel (control As IRibbonControl, index As Integer, ByRef label)
gallery	getItemScreenTip	Sub GetItemScreenTip (control As IRibbonControl, index as Integer, ByRef screen)
gallery	getItemSuperTip	Sub GetItemSuperTip (control As IRibbonControl, index as Integer, ByRef screen)
gallery	getItemWidth	Sub getItemWidth (control As IRibbonControl, ByRef width)
gallery	getSelectedItemID	Sub GetSelectedItemID (control As IRibbonControl, ByRef index)
gallery	getSelectedItemIndex	Sub GetSelectedItemIndex (control As IRibbonControl, ByRef index)
gallery	onAction	Sub OnAction (control As IRibbonControl, selectedId As String, selectedIndex As Integer)
menuSeparator	getTitle	Sub GetTitle (control As IRibbonControl, ByRef title)
toggleButton	getPressed	Sub GetPressed (control As IRibbonControl, ByRef returnValue)
toggleButton	onAction	Sub OnAction (control As IRibbonControl, pressed As Boolean, byRef cancelDefault)
toggleButton	onAction	Sub OnAction (control As IRibbonControl, pressed As Boolean)

Ribbon Controls Quick Reference (continued)

Attributes

The following table lists all of the Ribbon attributes used by RibbonX.

ATTRIBUTE	TYPE OR VALUE	DESCRIPTION
description	String	When the itemSize attribute is set to large, sets the description text that displays in menus.
enabled	true, false, 0, 1	Enables controls.
getContent	callback	Retrieves XML content that describes the menu. Used with a dynamic menu.
getDescription	callback	Returns the control description.
getEnabled	callback	Returns the control enabled state.
getImage	callback	Returns the image.
getImageMso	callback	Uses a control ID to return a built-in control icon.
getItemCount	callback	Returns the number of items in a combo box, drop-down list, or gallery.
getItemID	callback	Returns the ID for a specific item in a combo box, drop-down list, or gallery.
getItemImage	callback	Returns the image for a specific item in a combo box, drop-down list, or gallery.
getItemLabel	callback	Returns the label for a specific item in a combo box, drop-down list, or gallery.
getItemScreenTip	callback	Returns the ScreenTip for a specific item in a combo box, drop-down list, or gallery.
getItemSupertip	callback	Returns the Enhanced ScreenTip for a specific item in a combo box, drop-down list, or gallery.
getKeytip	callback	Returns the KeyTip.
getLabel	callback	Returns the label.
getPressed	callback	When used with a toggle button, gets a value that indicates whether the state is pressed or not pressed. When used with a checkbox, gets a value that indicates whether the state is selected or cleared.
getScreenTip	callback	Returns the ScreenTip.
getSelectedItemID	callback	For a drop-down list or gallery, gets the ID of the selected item.
getSelectedItemIndex	callback	For a drop-down list or gallery, gets the index of the selected item.
getShowImage	callback	Returns a value that sets whether to display the control image.
getShowLabel	callback	Returns a value that sets whether to display the control label.
getSize	callback	Returns a value that sets the size of a control (normal or large).
getSupertip	callback	Returns a value that sets the Enhanced ScreenTip for a control.
getText	callback	For a text box or edit box, gets the text to display in the edit portion of the control.

Attributes (*continued*)

ATTRIBUTE	TYPE OR VALUE	DESCRIPTION
getTitle	callback	For a menu separator, sets the text to display (rather than a horizontal line).
getVisible	callback	Returns the value that determines whether the control is visible.
id	String	A user-defined unique identifier for a control. If you define an id, do not assign an idMso or an idQ.
idMso	control id	Built-in control ID. If you define an idMso, do not assign an id or an idQ.
idQ	qualified id	Qualified control ID, prefixed with a namespace identifier. If you define an idQ, do not assign an idMso or an id.
image	String	Sets the image for a control.
imageMso	control id	Sets the identifier for a built-in image.
insertAfterMso	control id	Specifies the identifier for the built-in control after which the control is positioned.
insertAfterQ	qualified id	Specifies the identifier of a qualified control (that is, the control whose idQ property was specified) after which the control is positioned.
insertBeforeMso	control id	Specifies the identifier for the built-in control before the control is positioned.
insertBeforeQ	qualified id	Specifies the identifier of a qualified control (that is, a control whose idQ property was specified) before which the control is positioned.
itemSize	large, normal	Sets the size for the items in the menu.
keytip	String	Sets the KeyTip for the control. KeyTips display when the user presses the ALT key plus a letter.
label	String	Sets the label for the control.
onAction	callback	Called when the user clicks the control.
onChange	callback	Called when the user commits text in an edit box or combo box.
screentip	String	Sets the control's ScreenTip.
showImage	true, false, 0, 1	Specified whether the control's image displays.
showItemImage	true, false, 0, 1	In a combo box, drop-down list, or gallery, specifies whether each item's image shows.
showItemLabel	true, false, 0, 1	In a combo box, drop-down list, or gallery, specifies whether to show each item's label.
showLabel	true, false, 0, 1	Specifies whether the control's label shows.
size	large, normal	Sets the size of the control.
sizeString	String	Sets a string, such as "MMMMM". The string sets the width of the control.
supertip	String	Sets the Enhanced ScreenTip for the control. An EnhancedScreenTip is a longer screen tip.
tag	String	Sets user-defined text that enables you to store information about the control that is not pertinent to any other specific property.
title	String	Used with a menu separator. Sets the text displayed (rather than a horizontal line).
visible	true, false, 0, 1	Determines whether the control is visible.