

Word OCX is a powerful ActiveX Control written in Visual Basic 6.0 (ensuring maximum compatibility and effectiveness) to exchange information between Word and Visual Basic 6.0 easily and efficiently. Our professional, talented staff devoted hundreds of hours to create Word OCX and I think you'll agree that by using Word OCX in your own application you can save lots of time and money. All functions have been thoroughly tested and optimized to perform data transfer as quickly as possible. If you discover any problems or have an idea for a new function please email a detailed description to Frank immediately at kuslusi@mail.ic.net

The cost of Word OCX is only \$49.95 per developer. If you want to customize the Word OCX functions to your liking the complete Visual Basic 6.0 source code can be purchased for only \$99.95. All upgrades are free and Word OCX may be distributed with your application royalty free.

To add the Word OCX component to your Visual Basic 6.0 project:

- 1) Load Visual Basic 6.0.
- 2) From the Project Menu select "Components..."
- 3) Select component "Word_OCX" and click the OK Button. The component should appear in your toolbar.
- 4) Double-click on the component in the toolbar to add to a Visual Basic form.

Distributing Word OCX with your application

Word OCX may be distributed with your application royalty free. When creating the installation program for your application you should include file MSINET.OCX (Microsoft Internet Transfer Control) if you are using the functions UploadFile() and/or DownloadFile(). The file MSINET.OCX can be found in your Window's System directory which is C:\WINDOWS\SYSTEM32 on most machines. In addition, you should also include Microsoft Data Access Components (MDAC) 2.7 or greater in your installation program if you are using any functions that utilize ADO recordsets and/or databases (RecordsetToWordTable(), DatabaseTableToWordTable(),etc). If the MDAC installation file MDAC_TYP.EXE can't be found on your machine you can download it free from Microsoft's web site at <http://www.microsoft.com/data>

Using Word OCX in your Visual Basic Application

If you have an existing document then the you should use the OpenDocument function. Otherwise, you can use function CreateDocument to create a document.

Below is an example of how to open two documents, make changes to them, save the changes, close the documents, and finally close Word.

Dim i As Integer

'Create first document

i = WordOCX1.CreateDocument(App.Path & "\test1.doc")

'Create second document

i = WordOCX1.CreateDocument(App.Path & "\test2.doc")

'Open first document

i = WordOCX1.OpenDocument(App.Path & "\test1.doc")

'Open second document

i = WordOCX1.OpenDocument(App.Path & "\test2.doc")

'Activate the first workbook

i = WordOCX1.ActivateDocument("test1.doc")

'Add some text to it

i = WordOCX1.TextToWord("This is some text.")

'Done making changes to this document - save and close
WordOCX1.CloseDocument True

```
'Activate the second document
i = WordOCX1.ActivateDocument("test2.doc")
'i = WordOCX1.ActivateDocument(1) 'can also refer to document by number
```

```
'add some text to it
i = WordOCX1.TextToWord("This is some text.")
```

```
'Done making changes to this document - save and close
WordOCX1.CloseDocument True
```

```
'Done, close Word
WordOCX1.CloseWord False
```

Alternatively, you can also use the ActivateDocument() function to activate any open documents:

```
Dim arr() As Variant
Dim s As String
Dim i As Integer
Dim n As Integer
```

```
'Store list of all open documents to a one dimensional array
i = WordOCX1.GetDocumentNames(arr)
If i = 1 Then
    For n = 0 To UBound(arr)
        s = s & arr(n) & vbCrLf
    Next
    MsgBox s, vbOKOnly, "List of all open documents"
    'Modify each document
    For n = 0 To UBound(arr)
        'activate document
        i = WordOCX1.ActivateDocument(arr(n))
        'i = WordOCX1.ActivateDocument(n + 1) 'can also refer to document by number
        If i = 1 Then
            'add some text to the active document
            i = WordOCX1.TextToWord("This is some text.")
            'save changes and close
            WordOCX1.CloseDocument True
        End If
    Next
    'close Word
    WordOCX1.CloseWord False
ElseIf i = -1 Then
    MsgBox "Word is not loaded.", vbOKOnly
ElseIf i = -2 Then
    MsgBox "No documents found.", vbOKOnly
Else
    MsgBox "An error has occurred.", vbOKOnly
End If
```

After a Word document has been opened you're ready to harness the power of Word OCX!
Below are all the functions available to Word OCX divided into four categories:

- A) Functions that move data from Visual Basic to Word
- B) Functions for spell checking Visual Basic objects
- C) Functions for creating labels in Word
- D) Miscellaneous functions

A. Functions that move data from Visual Basic to Word

Note: Before moving data to Word you may want to temporarily hide Word to improve speed using the HideWord function. Hiding Word minimizes screen re-drawing.

ClipboardTextToWord() - Function to copy Clipboard text to a document.

Arguments:

Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As ParagraphAlignType
Optional lngColor As ColorType
Optional blnItalic As Boolean
Optional blnUnderline As Boolean
Optional lngAnimation As Long
Optional blnEmboss As Boolean
Optional blnEngrave As Boolean
Optional blnShadow As Boolean
Optional blnStrikeThrough As Boolean
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer
Clipboard.Clear
Clipboard.SetText "Hello World!"
i = WordOCX1.ClipboardTextToWord
```

CollectionToWord() - Function to copy Collection items to a document.

Arguments:

cl As Collection
strDelimiter As String
Optional strEndOfDataChar As String
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional lngColor As ColorType
Optional blnItalic As Boolean
Optional blnUnderline As Boolean
Optional lngAnimation As Long
Optional blnEmboss As Boolean
Optional blnEngrave As Boolean
Optional blnShadow As Boolean
Optional blnStrikeThrough As Boolean
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Object is not a Collection

-4 = Unknown error

Example:

```
Dim Temperatures As Collection
Dim i As Integer
Set Temperatures = New Collection 'create new instance of collection object
Temperatures.Add 76, "Atlanta" 'add items to collection object. City name is the key.
Temperatures.Add 85, "Miami"
Temperatures.Add 66, "Seattle"
i = WordOCX1.TextToWord("CollectionToWord() example:", True)
i = WordOCX1.CollectionToWord(Temperatures, vbCrLf)
Set Temperatures = Nothing 'Destroy Collection Object
```

ComboBoxToWord() - Function to copy ComboBox items to a document.

Arguments:

```
cb As Object
strDelimiter As String
Optional strEndOfDataChar As String
Optional blnSelectedOnly As Boolean
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional lngColor As ColorType
Optional blnItalic As Boolean
Optional blnUnderline As Boolean
Optional lngAnimation As Long
Optional blnEmboss As Boolean
Optional blnEngrave As Boolean
Optional blnShadow As Boolean
Optional blnStrikeThrough As Boolean
Optional lngInsertionPoint As Long
```

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Object is not a ComboBox
-4 = Unknown error

Example:

```
'This example assumes a ComboBox control is on your form
Dim i As Integer
I = WordOCX1.ComboBoxToWord(Combo1, ", ", ".")
```

DatabaseTableToWordTable() - Function to copy database table records to a document table.

Arguments:

```
strDSN As String
strSQL As String
Optional strUserID As String
Optional strPassword As String
```

Optional blnIncludeHeadings As Boolean
Optional lngTableType As Long
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional intColumnWidth As Integer
Optional blnBorders As Boolean
Optional blnShading As Boolean
Optional blnFont As Boolean
Optional blnColor As Boolean
Optional blnHeadingRows As Boolean
Optional blnLastRow As Boolean
Optional blnFirstColumn As Boolean
Optional blnLastColumn As Boolean
Optional blnAutoFit As Boolean
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unable to connect to database/Problem with DSN
-4 = Unable to open recordset
-5 = Unknown error

Examples:

Dim i As Integer
i = WordOCX1.DatabaseTableToWordTable("MyDSN", "SELECT * FROM Authors ORDER BY Author", , ,
True, wdoTableFormatColorful2, False, "Arial", 12, 15, wdoAlignParagraphLeft, , True, True, True, True, True,
False, False, False, True)

DataGridToWordTable() - Function to copy DataGrid to a document table.

Arguments:

dg As Object
Optional blnIncludeHeadings As Boolean
Optional lngTableType As Long
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional intColumnWidth As Integer
Optional blnBorders As Boolean
Optional blnShading As Boolean
Optional blnFont As Boolean
Optional blnColor As Boolean
Optional blnHeadingRows As Boolean
Optional blnLastRow As Boolean
Optional blnFirstColumn As Boolean
Optional blnLastColumn As Boolean
Optional blnAutoFit As Boolean
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

'This example assumes a populated DataGrid control is on your form

Dim i As Integer

i = WordOCX1.DataGridToWordTable(DataGrid1, True, wdoTableFormat3DEffects3, False, "Arial", , , , True, True, True, True, True, False, False, False, True)

Notes:

Before invoking this function make sure that the record pointer is positioned on the first record in the Data Grid otherwise undesired results may occur.

DictionaryToWord() - Function to copy Dictionary items to a document.

Arguments:

dt As Object

strDelimiter As String

Optional blnIncludeKeys As Boolean

Optional strEndOfDataChar As String

Optional blnBold As Boolean

Optional strFont As String = "Times New Roman"

Optional intSize As Integer = 12

Optional intSpaceAfter As Integer = 10

Optional lngParagraphAlign As Long

Optional lngColor As Long

Optional blnItalic As Boolean

Optional blnUnderline As Boolean

Optional lngAnimation As Long

Optional blnEmboss As Boolean

Optional blnEngrave As Boolean

Optional blnShadow As Boolean

Optional blnStrikeThrough As Boolean

Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

Dim i As Integer

Dim vItem As Variant

Dim vKey As Variant

Dim objDict As Scripting.Dictionary

Set objDict = New Scripting.Dictionary 'create new instance of objDictionary Object

objDict.CompareMode = BinaryCompare 'set compare mode. Also DatabaseCompare & TextCompare

objDict.Add "Key1", "Item1"

objDict.Add "Key2", "Item2"

objDict.Add "Key3", "Item3"

objDict.Add "Key4", "Item4"

i = WordOCX1.TextToWord("DictionaryToWord() example:", True)

i = WordOCX1.DictionaryToWord(objDict, "", True)

Set objDict = Nothing 'Destroy objDictionary Object

DirListBoxToWord() - Function to copy DirListBox items to a document.

Arguments:

dlb As Object
strDelimiter As String
Optional strEndOfDataChar As String
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional lngColor As Long
Optional blnItalic As Boolean
Optional blnUnderline As Boolean
Optional lngAnimation As Long
Optional blnEmboss As Boolean
Optional blnEngrave As Boolean
Optional blnShadow As Boolean
Optional blnStrikeThrough As Boolean
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Object is not a DirListBox
-4 = Unknown error

Example:

'This example assumes a DirListBox control is on your form

Dim i As Integer

i = WordOCX1.DirListBoxToWord(Dir1, ", ", ". ", False, , , wdoAlignParagraphLeft)

DriveListBoxToWord() - Function to copy DriveListBox items to a document.

Arguments:

dlb As Object
strDelimiter As String
Optional strEndOfDataChar As String
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional lngColor As Long
Optional blnItalic As Boolean
Optional blnUnderline As Boolean
Optional lngAnimation As Long
Optional blnEmboss As Boolean
Optional blnEngrave As Boolean
Optional blnShadow As Boolean
Optional blnStrikeThrough As Boolean
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Object is not a DriveListBox
-4 = Unknown error

Example:

```
Dim i As Integer
i = WordOCX1.DriveListBoxToWord(Drive1, vbCrLf, , False, , , wdoAlignParagraphLeft)
```

FileListBoxToWord() - Function to copy FileListBox items to a document.

Arguments:

flb As Object
strDelimiter As String
Optional blnSelectedOnly As Boolean
Optional strEndOfDataChar As String
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional lngColor As Long
Optional blnItalic As Boolean
Optional blnUnderline As Boolean
Optional lngAnimation As Long
Optional blnEmboss As Boolean
Optional blnEngrave As Boolean
Optional blnShadow As Boolean
Optional blnStrikeThrough As Boolean
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Object is not a FileListBox
-4 = Unknown error

Example:

‘This example assumes a FileListBox control is on your form

```
Dim i As Integer
i = WordOCX1.FileListBoxToWord(File1, ", ", True, ".", False, , , wdoAlignParagraphLeft)
```

FlexGridToWordTable() - Function to copy FlexGrid to a document table.

Arguments:

fg As Object
Optional lngStartRow As Long
Optional lngStartCol As Long
Optional ByVal lngEndRow As Long

Optional ByVal lngEndCol As Long
 Optional lngTableType As Long
 Optional blnBold As Boolean
 Optional strFont As String = "Times New Roman"
 Optional intSize As Integer = 12
 Optional intSpaceAfter As Integer = 10
 Optional lngParagraphAlign As Long
 Optional intColumnWidth As Integer
 Optional blnBorders As Boolean
 Optional blnShading As Boolean
 Optional blnFont As Boolean
 Optional blnColor As Boolean
 Optional blnHeadingRows As Boolean
 Optional blnLastRow As Boolean
 Optional blnFirstColumn As Boolean
 Optional blnLastColumn As Boolean
 Optional blnAutoFit As Boolean
 Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
 -1 = Word is not active
 -2 = Word document not active
 -3 = Unknown error

Example:

'This example assumes a MSFlexGrid control is on your form

Dim i As Integer

I = WordOCX1.FlexGridToWordTable(MSFlexGrid1, , , , wdoTableFormatColorful1, False, , , ,
 wdoAlignParagraphCenter, , True, True, True, True, True, False, True, False, True)

HFlexGridToWordTable() - Function to copy HFlexGrid to a document table.

Arguments:

fg As Object
 Optional lngStartRow As Long
 Optional lngStartCol As Long
 Optional ByVal lngEndRow As Long
 Optional ByVal lngEndCol As Long
 Optional lngTableType As Long
 Optional blnBold As Boolean
 Optional strFont As String = "Times New Roman"
 Optional intSize As Integer = 12
 Optional intSpaceAfter As Integer = 10
 Optional lngParagraphAlign As Long
 Optional intColumnWidth As Integer
 Optional blnBorders As Boolean
 Optional blnShading As Boolean
 Optional blnFont As Boolean
 Optional blnColor As Boolean
 Optional blnHeadingRows As Boolean
 Optional blnLastRow As Boolean
 Optional blnFirstColumn As Boolean
 Optional blnLastColumn As Boolean
 Optional blnAutoFit As Boolean
 Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

'This example assumes a MSHFlexGrid control is on your form

Dim i As Integer

i = WordOCX1.HFlexGridToWordTable(MSHFlexGrid1, , , , wdoTableFormatColorful1, False, , ,
wdoAlignParagraphCenter, , True, True, True, True, True, False, True, False, True)

LabelToWord() - Function to copy label caption to a document.

Arguments:

lbl As Object

Optional blnBold As Boolean

Optional strFont As String = "Times New Roman"

Optional intSize As Integer = 12

Optional intSpaceAfter As Integer = 10

Optional lngParagraphAlign As Long

Optional lngColor As Long

Optional blnItalic As Boolean

Optional blnUnderline As Boolean

Optional lngAnimation As Long

Optional blnEmboss As Boolean

Optional blnEngrave As Boolean

Optional blnShadow As Boolean

Optional blnStrikeThrough As Boolean

Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Object is not a Label
-4 = Unknown error

Example:

'This example assumes a Label control is on your form

Dim i As Integer

i = WordOCX1.LabelToWord(Label1, True, , 16, , wdoAlignParagraphCenter, wdoColorDarkGreen, False, False,
wdoAnimationNone, True, True, True, False, wdoEndofDoc)

ListBoxToWord() - Function to copy ListBox items to a document.

Arguments:

lb As Object

strDelimiter As String

Optional strEndOfDataChar As String

Optional blnSelectedOnly As Boolean

Optional blnBold As Boolean

Optional strFont As String = "Times New Roman"

Optional intSize As Integer = 12

Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional lngColor As Long
Optional blnItalic As Boolean
Optional blnUnderline As Boolean
Optional lngAnimation As Long
Optional blnEmboss As Boolean
Optional blnEngrave As Boolean
Optional blnShadow As Boolean
Optional blnStrikeThrough As Boolean
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Object is not a ListBox
-4 = Unknown error

Example:

```
Dim i As Integer  
i = WordOCX1.ListBoxToWord(List1, vbCrLf, , True)
```

ListViewToWordTable() - Function to copy ListView items to a document table.

Arguments:

lv As Object
Optional blnSelectedOnly As Boolean
Optional blnIncludeHeadings As Boolean
Optional lngTableType As Long
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional intColumnWidth As Integer
Optional blnBorders As Boolean
Optional blnShading As Boolean
Optional blnFont As Boolean
Optional blnColor As Boolean
Optional blnHeadingRows As Boolean
Optional blnLastRow As Boolean
Optional blnFirstColumn As Boolean
Optional blnLastColumn As Boolean
Optional blnAutoFit As Boolean
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word Document not active
-3 = Unknown error

Example:

'This example assumes a ListView control is on your form
Dim i As Integer

i = WordOCX1.ListViewToWordTable(ListView1, False, True, wdoTableFormatColorful2, False, "Arial", 12, 15, wdoAlignParagraphLeft, , True, True, True, True, True, False, False, False, True)

MaskedTextBoxToWord() - Function to copy masked edit textbox text to a document.

Arguments:

tb As Object
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional lngColor As Long
Optional blnItalic As Boolean
Optional blnUnderline As Boolean
Optional lngAnimation As Long
Optional blnEmboss As Boolean
Optional blnEngrave As Boolean
Optional blnShadow As Boolean
Optional blnStrikeThrough As Boolean
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

'This example assumes a MaskedTextBox control is on your form
Dim i As Integer
i = WordOCX1.MaskedTextBoxToWord(MaskedText1)

OneDArrayToWord() - Function to copy 1D array to a document.

Arguments:

varArray As Variant
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional lngColor As Long
Optional blnItalic As Boolean
Optional blnUnderline As Boolean
Optional lngAnimation As Long
Optional blnEmboss As Boolean
Optional blnEngrave As Boolean
Optional blnShadow As Boolean
Optional blnStrikeThrough As Boolean
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success

-1 = Word is not active
-2 = Word document not active
-3 = varArray is not an array
-4 = Unknown error

Example:

```
Dim i As Integer
Dim arr(0 To 3) As Variant
arr(0) = "orange"
arr(1) = "apple"
arr(2) = "pear"
arr(3) = "peach"
i = WordOCX1.OneDArrayToWord(arr)
```

OneDArrayToWordTable() - Function to copy 1D array to a document table.

Arguments:

```
varArray As Variant
Optional lngTableType As Long
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional intColumnWidth As Integer
Optional blnBorders As Boolean
Optional blnShading As Boolean
Optional blnFont As Boolean
Optional blnColor As Boolean
Optional blnHeadingRows As Boolean
Optional blnLastRow As Boolean
Optional blnFirstColumn As Boolean
Optional blnLastColumn As Boolean
Optional blnAutoFit As Boolean
Optional lngInsertionPoint As Long
```

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = varArray is not an array
-4 = Unknown error

Example:

```
Dim i As Integer
Dim arr(0 To 3) As Variant
arr(0) = "orange"
arr(1) = "apple"
arr(2) = "pear"
arr(3) = "peach"
i = WordOCX1.OneDArrayToWordTable(arr)
```

RecordsetToWordTable() - Function to copy a recordset to a document table.

Arguments:

objRecd As Object
Optional blnIncludeHeadings As Boolean
Optional lngTableType As Long
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional intColumnWidth As Integer
Optional blnBorders As Boolean
Optional blnShading As Boolean
Optional blnFont As Boolean
Optional blnColor As Boolean
Optional blnHeadingRows As Boolean
Optional blnLastRow As Boolean
Optional blnFirstColumn As Boolean
Optional blnLastColumn As Boolean
Optional blnAutoFit As Boolean
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer
Dim strConn As String
Dim cn As ADODB.Connection
Dim rs As ADODB.Recordset
On Error Resume Next
Screen.MousePointer = vbHourglass
strConn = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path & _
"\author.mdb;Persist Security Info=False"
Set cn = New ADODB.Connection
cn.CursorLocation = adUseClient
cn.Open strConn
Set rs = New ADODB.Recordset
rs.CursorType = adOpenStatic
rs.CursorLocation = adUseClient
rs.LockType = adLockPessimistic
rs.Source = "Select AU_ID As [ID Number], [Year Born] As [Birth Date], Author As [Author Name] From Authors"
rs.ActiveConnection = cn
rs.Open
i = WordOCX1.TextToWord("RecordsetToWordTable() example:", True, "Arial", 12, 15, wdoAlignParagraphLeft)
i = WordOCX1.RecordsetToWordTable(rs, True, wdoTableFormatColorful2, False, "Arial", 12, 15,
wdoAlignParagraphLeft, , True, True, True, True, True, False, False, False, True)
rs.Close
cn.Close
Set rs = Nothing
Set cn = Nothing
Screen.MousePointer = vbDefault
```

RichTextBoxToWord() - Function to copy rich textbox text to a document.

Arguments:

tb As Object
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional lngColor As Long
Optional blnItalic As Boolean
Optional blnUnderline As Boolean
Optional lngAnimation As Long
Optional blnEmboss As Boolean
Optional blnEngrave As Boolean
Optional blnShadow As Boolean
Optional blnStrikeThrough As Boolean
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

'This example assumes a RichTextBox control is on your form

Dim i As Integer

i = WordOCX1.RichTextBoxToWord(RichTextBox1)

TextBoxToWord() - Function to copy textbox text to a document

Arguments:

tb As Object
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional lngColor As Long
Optional blnItalic As Boolean
Optional blnUnderline As Boolean
Optional lngAnimation As Long
Optional blnEmboss As Boolean
Optional blnEngrave As Boolean
Optional blnShadow As Boolean
Optional blnStrikeThrough As Boolean
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

'This example assumes a TextBox control is on your form

```
Dim i As Integer
i = WordOCX1.TextBoxToWord(Text1)
```

TextFileToWord() - Function to copy text file contents to a document.

Arguments:

```
strTextFile As String
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional lngColor As Long
Optional blnItalic As Boolean
Optional blnUnderline As Boolean
Optional lngAnimation As Long
Optional blnEmboss As Boolean
Optional blnEngrave As Boolean
Optional blnShadow As Boolean
Optional blnStrikeThrough As Boolean
Optional lngInsertionPoint As Long
```

Returns: Integer

```
1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error
```

Example:

```
Dim i As Integer
i = WordOCX1.TextFileToWord(App.Path & "\file.txt")
```

TextFileToWordTable() - Function to copy text file contents to a document table.

Arguments:

```
strTextFile As String
Optional ByVal strDelimiter As String
Optional blnRemoveQuotes As Boolean
Optional lngTableType As Long
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional intColumnWidth As Integer
Optional blnBorders As Boolean
Optional blnShading As Boolean
Optional blnFont As Boolean
Optional blnColor As Boolean
Optional blnHeadingRows As Boolean
Optional blnLastRow As Boolean
Optional blnFirstColumn As Boolean
Optional blnLastColumn As Boolean
Optional blnAutoFit As Boolean
Optional lngInsertionPoint As Long
```

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Invalid file
-4 = Unknown error

Example:

```
Dim i as integer
I = WordOCX1.TextFileToWordTable(App.Path & "\file.txt", ",", False, wdoTableFormatElegant, False, , , ,
wdoAlignParagraphLeft, , True, True, True, True, False, False, False, True)
```

TextToWord() - Function to copy a string to a document.

Arguments:

strText As String
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional lngColor As Long
Optional blnItalic As Boolean
Optional blnUnderline As Boolean
Optional lngAnimation As Long
Optional blnEmboss As Boolean
Optional blnEngrave As Boolean
Optional blnShadow As Boolean
Optional blnStrikeThrough As Boolean
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer
i = WordOCX1.TextToWord("Hello World!", True, "Arial", 12, 15, wdoAlignParagraphLeft)
```

TextToWordTable() - Function to copy a string to a document table.

Arguments:

strText As String
Optional lngTableType As Long
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long

Optional intColumnWidth As Integer
Optional blnBorders As Boolean
Optional blnShading As Boolean
Optional blnFont As Boolean
Optional blnColor As Boolean
Optional blnHeadingRows As Boolean
Optional blnLastRow As Boolean
Optional blnFirstColumn As Boolean
Optional blnLastColumn As Boolean
Optional blnAutoFit As Boolean
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer
Dim s As String
s = "This is some" & vbTab & "text seperated" & vbTab & "by tabs"
i = WordOCX1.TextToWordTable(s)
```

TwoArrayToWordTable() - Function to copy a two dimensional array to a document table.

Arguments:

varArray As Variant
Optional lngTableType As Long
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional intColumnWidth As Integer
Optional blnBorders As Boolean
Optional blnShading As Boolean
Optional blnFont As Boolean
Optional blnColor As Boolean
Optional blnHeadingRows As Boolean
Optional blnLastRow As Boolean
Optional blnFirstColumn As Boolean
Optional blnLastColumn As Boolean
Optional blnAutoFit As Boolean
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = varArray is not an array
-4 = Unknown error

Example:

'create a Word table from a two dimensional array
Dim i As Integer

```

Dim arr(0 To 3, 0 To 3) As Variant
Screen.MousePointer = vbHourglass
arr(0, 0) = "Employee"
arr(1, 0) = "John Stevens"
arr(2, 0) = "Kevin Jackson"
arr(3, 0) = "Susan Smith"
arr(0, 1) = "Age"
arr(1, 1) = 28
arr(2, 1) = 33
arr(3, 1) = 52
arr(0, 2) = "Sex"
arr(1, 2) = "M"
arr(2, 2) = "M"
arr(3, 2) = "F"
arr(0, 3) = "Profession"
arr(1, 3) = "Forensic Scientist"
arr(2, 3) = "Software Engineer"
arr(3, 3) = "Clinical Research Nurse"
i = WordOCX1.InsertBlankLines(1) 'insert a blank line
i = WordOCX1.TextToWord("TwoDArrayToWordTable() example:", True, "Arial", 12, 15,
wdoAlignParagraphCenter, RGB(0, 255, 0), True, True, wdoAnimationMarchingRedAnts)
i = WordOCX1.TwoDArrayToWordTable(arr, wdoTableFormatColorful2, False, "Arial", 12, 0,
wdoAlignParagraphLeft, , True, True, True, True, False, False, False, True, wdoEndofDoc)
Screen.MousePointer = vbDefault

```

XMLFileToWordTable() - Function to copy an XML table file's contents to a document table.

Arguments:

```

strXMLFile As String
Optional blnIncludeHeadings As Boolean
Optional lngTableType As Long
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional intColumnWidth As Integer
Optional blnBorders As Boolean
Optional blnShading As Boolean
Optional blnFont As Boolean
Optional blnColor As Boolean
Optional blnHeadingRows As Boolean
Optional blnLastRow As Boolean
Optional blnFirstColumn As Boolean
Optional blnLastColumn As Boolean
Optional blnAutoFit As Boolean
Optional lngInsertionPoint As Long

```

Returns: Integer

```

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unable to access XML DLL
-4 = Unable to open XML file
-5 = Unknown error

```

Example:

```

Dim i As Integer

```

i = WordOCX1.XMLFileToExcel(App.Path & "\\portfolio.xml", 1, xlo_A, xloNormal, True)

B. Functions for spell checking Visual Basic objects

SpellCheckCollection () - Function to spell check a collection.

Arguments:

cl As Collection

Optional blnNoChanges As Boolean

Optional strCustomDictionary As String

Returns: Integer

1 = Success

-1 = cl is not a Collection Object

-2 = Unknown error

Example:

```
Dim i As Integer
```

```
Dim Cities As Collection
```

```
Dim vItem As Variant
```

```
Dim s As String
```

```
Set Cities = New Collection 'create new instance of collection object
```

```
Cities.Add "Atlana" 'add items to collection object. City name is the key.
```

```
Cities.Add "Miami"
```

```
Cities.Add "Seattle"
```

```
Cities.Add "New York"
```

```
For Each vItem In Cities
```

```
    s = s & vItem & vbCrLf
```

```
Next
```

```
MsgBox s, vbOKOnly, "Collection items before spell check"
```

```
i = WordOCX1.SpellCheckCollection(Cities)
```

```
s = ""
```

```
For Each vItem In Cities
```

```
    s = s & vItem & vbCrLf
```

```
Next
```

```
MsgBox s, vbOKOnly, "Collection items after spell check"
```

```
Set Cities = Nothing 'Destroy Collection Object
```

SpellCheckComboBox () - Function to spell check a ComboBox control.

Arguments:

cb As Object

Optional blnNoChanges As Boolean

Optional strCustomDictionary As String

Returns: Integer

1 = Success

-1 = cb is not a ComboBox control

-2 = Unknown error

Example:

```
Dim i As Integer
```

```
i = WordOCX1.SpellCheckComboBox(Combo1, False)
```

SpellCheckDatabaseTable () - Function to spell check a database table.

Arguments:

strDSN As String
strSQL As String
Optional strUserID As String
Optional strPassword As String
Optional blnNoChanges As Boolean
Optional strCustomDictionary As String

Returns: Integer

1 = Success
-1 = Can't open database
-2 = Can't open table
-3 = Unknown error

Example:

```
Dim i As Integer  
i = WordOCX1.SpellCheckDatabaseTable("MyDSN", "Select * from authors", , , True)
```

SpellCheckDataGrid () - Function to spell check a DataGrid control.

Arguments:

dg As Object
Optional lngStartCol As Long
Optional ByVal lngEndCol As Long
Optional blnNoChanges As Boolean
Optional strCustomDictionary As String

Returns: Integer

1 = Success
-2 = Unknown error

Example:

```
Dim i As Integer  
i = WordOCX1.SpellCheckDataGrid(DataGrid1, , , True)
```

SpellCheckDictionary () - Function to spell check a dictionary.

Arguments:

dt As Object
Optional blnNoChanges As Boolean
Optional strCustomDictionary As String

Returns: Integer

1 = Success
-1 = Unknown error

Example:

```

Dim i As Integer
Dim vItem As Variant
Dim vKey As Variant
Dim objDict As Scripting.Dictionary
Dim s As String
Set objDict = New Scripting.Dictionary 'create new instance of objDictionary Object
objDict.CompareMode = BinaryCompare 'set compare mode. Also DatabaseCompare & TextCompare
objDict.Add "Key1", "appel"
objDict.Add "Key2", "orange"
objDict.Add "Key3", "pineappel"
objDict.Add "Key4", "lemonn"
For Each vItem In objDict.Items
    s = s & vItem & vbCrLf
Next
MsgBox s, vbOKOnly, "Dictionary items before spell check"
i = WordOCX1.SpellCheckDictionary(objDict, False)
s = ""
For Each vItem In objDict.Items
    s = s & vItem & vbCrLf
Next
MsgBox s, vbOKOnly, "Dictionary items after spell check"
Set objDict = Nothing 'Destroy objDictionary Object

```

SpellCheckFlexGrid () - Function to spell check a FlexGrid control.

Arguments:

```

fg As Object
Optional lngStartRow As Long
Optional lngStartCol As Long
Optional ByVal lngEndRow As Long
Optional ByVal lngEndCol As Long
Optional blnNoChanges As Boolean
Optional strCustomDictionary As String

```

Returns: Integer

1 = Success
-1 = Unknown error

Example:

```

Dim i As Integer
i = WordOCX1.SpellCheckFlexGrid(MSFlexGrid1, 1, 1, , , False)

```

SpellCheckHFlexGrid () - Function to spell check a HFlexGrid control.

Arguments:

```

fg As Object
Optional lngStartRow As Long
Optional lngStartCol As Long
Optional ByVal lngEndRow As Long
Optional ByVal lngEndCol As Long
Optional blnNoChanges As Boolean
Optional strCustomDictionary As String

```

Returns: Integer

1 = Success
-1 = Unknown error

Example:

```
Dim i As Integer  
i = WordOCX1.SpellCheckHFlexGrid(MSHFlexGrid1, 1, 1, , , False)
```

SpellCheckLabel () - Function to spell check a Label control.

Arguments:

lbl As Object
Optional blnNoChanges As Boolean
Optional strCustomDictionary As String

Returns: Integer

1 = Success
-1 = lbl is not a Label control
-2 = Unknown error

Example:

‘This example assumes a Label control is on your form
Dim i As Integer
i = WordOCX1.SpellCheckLabel(Label1)

SpellCheckListBox () - Function to spell check a ListBox control.

Arguments:

lb As Object
Optional blnSelectedOnly As Boolean
Optional blnNoChanges As Boolean
Optional strCustomDictionary As String

Returns: Integer

1 = Success
-1 = lbl is not a ListBox control
-2 = Unknown error

Example:

‘This example assumes a ListBox control is on your form
Dim i As Integer
i = WordOCX1.SpellCheckListBox(List1, False)

SpellCheckListView () - Function to spell check a ListView control.

Arguments:

lv As Object
Optional blnSelectedOnly As Boolean
Optional blnIncludeHeadings As Boolean
Optional blnNoChanges As Boolean

Optional strCustomDictionary As String

Returns: Integer

1 = Success
-1 = Unknown error

Example:

'This example assumes a ListView control is on your form
Dim i As Integer
i = WordOCX1.SpellCheckListView(ListView1, False, True)

SpellCheckOneDArray () - Function to spell check a one dimensional array.

Arguments:

varArray As Variant
Optional blnNoChanges As Boolean
Optional strCustomDictionary As String

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

```
Dim arr(3) As Variant
Dim i As Integer
Dim s As String
arr(0) = "Everybuddy"
arr(1) = "hav"
arr(2) = "funn"
arr(3) = "tonite"
s = ""
For i = 0 To UBound(arr)
    s = s & arr(i) & " "
Next
s = s & "!"
MsgBox s, vbOKOnly, "One dimensional array before spell check"
i = WordOCX1.SpellCheckOneDArray(arr)
s = ""
For i = 0 To UBound(arr)
    s = s & arr(i) & " "
Next
s = s & "!"
MsgBox s, vbOKOnly, "One dimensional array after spell check"
```

SpellCheckRecordset () - Function to spell check a recordset.

Arguments:

objRecd As Object
Optional lngStartCol As Long
Optional ByVal lngEndCol As Long
Optional blnNoChanges As Boolean
Optional strCustomDictionary As String

Returns: Integer

1 = Success
-1 = Unknown error

Example:

```
Dim i As Integer
Dim strConn As String
Dim cn As ADODB.Connection
Dim rs As ADODB.Recordset
On Error Resume Next
Screen.MousePointer = vbHourglass
strConn = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path & _
"\author.mdb;Persist Security Info=False"
Set cn = New ADODB.Connection
cn.CursorLocation = adUseClient
cn.Open strConn
Set rs = New ADODB.Recordset
rs.CursorType = adOpenStatic
rs.CursorLocation = adUseClient
rs.LockType = adLockPessimistic
rs.Source = "Select * From SpellCheck"
rs.ActiveConnection = cn
rs.Open
i = WordOCX1.SpellCheckRecordset(rs, , , True)
rs.Close
cn.Close
Set rs = Nothing
Set cn = Nothing
Screen.MousePointer = vbDefault
```

SpellCheckRichTextBox () - Function to spell check a RichTextBox control.

Arguments:

tb As Object
Optional blnNoChanges As Boolean
Optional strCustomDictionary As String

Returns: Integer

1 = Success
-1 = Unknown error

Example:

```
Dim i As Integer
i = WordOCX1.SpellCheckRichTextBox(RichText1, False)
```

SpellCheckText() - Function to spell check a string variable.

Arguments:

strText As String
Optional blnNoChanges As Boolean
Optional strCustomDictionary As String

Returns: Integer

1 = Success
-1 = Unknown error

Example:

```
Dim i As Integer
Dim s As String
s = "Mary had a litle lamb whoz fleez wuz white as snow." & _
  " Everywher that Mary went that lamb was sur to go!"
MsgBox s, vbOKOnly, "Text before spell check"
i = WordOCX1.SpellCheckText(s)
MsgBox s, vbOKOnly, "Text after spell check"
```

SpellCheckTextBox () - Function to spell check a TextBox control.

Arguments:

tb As Object
Optional blnNoChanges As Boolean
Optional strCustomDictionary As String

Returns: Integer

1 = Success
-1 = tb is not a TextBox control
-2 = Unknown error

Example:

```
Dim i As Integer
i = WordOCX1.SpellCheckTextBox(Text1, False)
```

SpellCheckThreeDArray () - Function to spell check a three dimensional array.

Arguments:

varArray As Variant
Optional blnNoChanges As Boolean
Optional strCustomDictionary As String

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

```
Dim arr(0 To 1, 0 To 1, 0 To 1) As Variant
Dim s As String
Dim i As Integer
Dim n As Integer
Dim k As Integer
'first row, first level
arr(0, 0, 0) = "appel"
arr(0, 1, 0) = "orang"
'second row, first level
```

```

arr(1, 0, 0) = "lemonn"
arr(1, 1, 0) = "plumm"
'first row, second level
arr(0, 0, 1) = "graifruit"
arr(0, 1, 1) = "peech"
'second row, second level
arr(1, 0, 1) = "grai"
arr(1, 1, 1) = "mangoe"
For i = LBound(arr, 3) To UBound(arr, 3) 'level
    For n = LBound(arr, 1) To UBound(arr, 1) 'row
        For k = LBound(arr, 2) To UBound(arr, 2) 'column
            s = s & arr(n, k, i) & " "
        Next
        s = s & vbCrLf
    Next
Next
Next
MsgBox s, vbOKOnly, "Three dimensional array items before spell check"
i = WordOCX1.SpellCheckThreeDArray(arr, False)
s = ""
For i = LBound(arr, 3) To UBound(arr, 3) 'level
    For n = LBound(arr, 1) To UBound(arr, 1) 'row
        For k = LBound(arr, 2) To UBound(arr, 2) 'column
            s = s & arr(n, k, i) & " "
        Next
        s = s & vbCrLf
    Next
Next
Next
MsgBox s, vbOKOnly, "Three dimensional array items after spell check"

```

SpellCheckTwoDArray () - Function to spell check a two dimensional array.

Arguments:

varArray As Variant
Optional blnNoChanges As Boolean
Optional strCustomDictionary As String

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

```

Dim arr(0 To 3, 0 To 3) As Variant
Dim i As Integer
Dim n As Integer
Dim s As String
arr(0, 0) = "Mary"
arr(1, 0) = "lamb"
arr(2, 0) = "wite"
arr(3, 0) = "thet"
arr(0, 1) = "hadd"
arr(1, 1) = "whoz"
arr(2, 1) = "az"
arr(3, 1) = "Mary"
arr(0, 2) = "a"
arr(1, 2) = "flees"
arr(2, 2) = "snow"
arr(3, 2) = "wint"

```

```

arr(0, 3) = "littel"
arr(1, 3) = "wuz"
arr(2, 3) = "everywher"
arr(3, 3) = "that lamb was sure to go!"
s = ""
For i = 0 To UBound(arr, 1)
    For n = 0 To UBound(arr, 2)
        s = s & arr(i, n) & " "
    Next
    s = s & vbCrLf
Next
MsgBox s, vbOKOnly, "Two dimensional array before spell check"
i = WordOCX1.SpellCheckTwoDArray(arr, False)
s = ""
For i = 0 To UBound(arr, 1)
    For n = 0 To UBound(arr, 2)
        s = s & arr(i, n) & " "
    Next
    s = s & vbCrLf
Next
MsgBox s, vbOKOnly, "Two dimensional array after spell check"

```

C. Functions for creating labels in Word

CreateLabelsFromDatabase() - Function to create labels from a database.

Note: Addressee name contained in one field.

Arguments:

strDatabase As String
strSQL As String

Returns: Integer

1 = Success
-1 = Unknown error

Example:

```

Dim i As Integer
'create labels from a database - Addressee name contained in one field
i = WordOCX1.CreateLabelsFromDatabase(App.Path & "\author.mdb", "SELECT Name AS
Contact_Name,Address1,Address2,City,State,Zip AS Postal_Code,Country FROM Addresses WHERE Region = 1
ORDER BY Name")
MsgBox "Labels created", vbOKOnly, "CreateLabelsFromDatabase()"
'Note: If the field names in your table are not named
'Contact_Name, Address1, Address2, City, State, Postal_Code, Country
'then you can use the "AS" statement in the SQL statement to
'give the field name an alias. For example:
'SELECT Name AS Contact_Name,Addr1 AS Address1,Addr2 AS Address2, etc.
'You can also create labels from any ODBC compliant database such as
'Oracle, SQL Server, Sybase, dBase, Foxpro, etc by passing a connection
'string rather than a database path. Some examples:
'i = WordOCX1.CreateLabelsFromDatabase("DSN=MyDSN;", "SELECT Name AS
Contact_Name,Address1,Address2,City,State,Zip AS Postal_Code,Country FROM Addresses WHERE Region = 1
ORDER BY Name")
'i = WordOCX1.CreateLabelsFromDatabase("DSN=MySQLServerDSN;DATABASE=pubs;uid=sa;pwd=;",
"SELECT Name AS Contact_Name,Address1,Address2,City,State,Zip AS Postal_Code,Country FROM Addresses
WHERE Region = 1 ORDER BY Name")

```

CreateLabelsFromDatabase2() - Function to create labels from a database.

Note: Addressee name contained in two fields.

Arguments:

strDatabase As String

strSQL As String

Returns: Integer

1 = Success

-1 = Unknown error

Example:

Dim i As Integer

'create labels from a database - Addressee name contained in two fields

i = WordOCX1.CreateLabelsFromDatabase2(App.Path & "\author.mdb", "SELECT First AS First_Name, Last AS Last_Name, Address1, Address2, City, State, Zip AS Postal_Code, Country FROM Addresses2 WHERE Region = 1 ORDER BY Last")

MsgBox "Labels created", vbOKOnly, "CreateLabelsFromDatabase2()"

'Note: If the field names in your table are not named

'First_Name, Last_Name, Address1, Address2, City, State, Postal_Code, Country

'then you can use the "AS" statement in the SQL statement to

'give the field name an alias. For example:

'SELECT First AS First_Name, Last AS Last_Name, Addr1 AS Address1, Addr2 AS Address2, etc.

CreateLabelsFromDatabase3() - Function to create labels from a database.

Note: Addressee name contained in four fields.

Arguments:

strDatabase As String

strSQL As String

Returns: Integer

1 = Success

-1 = Unknown error

Example:

Dim i As Integer

'create labels from a database - Two addressee names contained in four fields

i = WordOCX1.CreateLabelsFromDatabase3(App.Path & "\author.mdb", "SELECT First1 AS First_Name1, Last1 AS Last_Name1, First2 AS First_Name2, Last2 AS Last_Name2, Address1, Address2, City, State, Zip AS Postal_Code, Country FROM Addresses3 WHERE Region = 1 ORDER BY Last1")

MsgBox "Labels created", vbOKOnly, "CreateLabelsFromDatabase3()"

'Note: If the field names in your table are not named

'First_Name1, Last_Name1, First_Name2, Last_Name2, Address1, Address2, City, State, Postal_Code, Country

'then you can use the "AS" statement in the SQL statement to

'give the field name an alias. For example:

'SELECT First1 AS First_Name1, Last1 AS Last_Name1, First2 AS First_Name2, Last2 AS Last_Name2, Addr1 AS Address1, Addr2 AS Address2, etc.

CreateLabelsFromTextFile() - Function to create labels from a text file.

Note: Addressee name contained in one field.

Arguments:

strTextFile As String

Returns: Integer

1 = Success
-1 = Unknown error

Example:

```
Dim i As Integer
'create labels from a tab-delimited text file
i = WordOCX1.CreateLabelsFromTextFile(App.Path & "\data.txt")
MsgBox "Labels created", vbOKOnly, "CreateLabelsFromTextFile()"
```

D. Miscellaneous functions

AboutBox () - Function to display a message box containing information about Word OCX.

Arguments: None

Returns: None

Example:

Call AboutBox

ActivateDocument() - Function to set a Word document as active.

Arguments:

Optional varDocument As Variant
Optional blnSwitchToWord As Boolean

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Unknown error

Example:

```
Dim i As Integer
i = WordOCX1.ActivateDocument("test.doc", True)
```

ActiveDocumentName() - Function to get active Word document name.

Arguments: None

Returns: String, returns empty string if there is no active workbook.

Example:

MsgBox WordOCX1.ActiveDocumentName

ActiveDocumentPath() - Function to get active Word document path.

Arguments: None

Returns: String, returns empty string if there is no active document.

Example:

```
MsgBox WordOCX1.ActiveDocumentPath
```

AddDocument() - Function to add a Word document.

Arguments:

Optional varTemplate As Variant

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Unknown error

Example:

```
Dim i As Integer  
i = WordOCX1.AddDocument("c:\MyTemplate.dot")
```

Notes:

After the document is added it becomes the active document.

CheckGrammar() - Function to check the grammar of the active Word document.

Arguments: None

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer  
i = WordOCX1.CheckGrammar()
```

CheckSpelling() - Function to spell check the active Word document.

Arguments: None

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active

-3 = Unknown error

Example:

```
Dim i As Integer
i = WordOCX1.CheckSpelling()
```

CloseDocument() - Function to close the active Word document.

Arguments:

blnSave As Boolean

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Unknown error

Example:

```
Dim i As Integer
i = WordOCX1.CloseDocument(True)
```

CloseWord() - Function to close Word.

Arguments:

blnSave As Boolean

Returns: None

Example:

```
WordOCX1.CloseWord True
```

CollectionToOneDArray() - Function to copy Collection items to a one dimensional array.

```
cl As Collection
varArray As Variant
```

Returns: Integer

1 = Success
-1 = cl is not a Collection
-2 = varArray is not an array
-3 = Unknown error

Example:

```
Dim i As Integer
dim arr() as Variant
Dim Temperatures As Collection
Set Temperatures = New Collection 'create new instance of collection object
Temperatures.Add 76, "Atlanta" 'add items to collection object. City name is the key.
Temperatures.Add 85, "Miami"
Temperatures.Add 66, "Seattle"
```

```
i = WordOCX1.CollectionToOneDArray(Temperatures, arr)
Set Temperatures = Nothing 'Destroy Collection Object
```

ComboBoxToOneDArray() - Function to copy ComboBox items to a one dimensional array.

```
cb As Object
varArray As Variant
```

Returns: Integer

```
1 = Success
-1 = cb is not a ComboBox
-2 = varArray is not an array
-3 = Unknown error
```

Example:

```
Dim i As Integer
dim arr() as Variant
i = WordOCX1.ComboBoxToOneDArray(Combo1, arr)
```

ComputeOneDArray() - Function to add/subtract/multiply/divide all items in a one dimensional array by specified number.

Arguments:

```
varArray As Variant
lngCompute As Long
dblNumber As Double
```

Returns: Integer

```
1 = Success
-1 = varArray is not an array
-2 = Invalid value passed for variable lngCompute (valid values are 0,1,2,3)
-3 = Unknown error
```

Example:

```
Dim i As Integer
Dim arr(10) as Variant
For i = LBound(arr) To UBound(arr)
    arr(i) = i
Next
i = WordOCX1.ComputeOneDArray(arr, wdoAdd, 2.5) 'Add 2.5 to each element in array
```

ComputeThreeDArray() - Function to add/subtract/multiply/divide all items in a three dimensional array by specified number.

Arguments:

```
varArray As Variant
lngCompute As Long
dblNumber As Double
```

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Invalid value passed for variable lngCompute (valid values are 0,1,2,3)
-3 = Unknown error

Example:

```
Dim arr(0 To 4, 0 To 4, 0 To 2) As Variant
Dim i As Integer
Dim n As Integer
Dim j As Integer
Dim k As Integer
Dim s As String
s = ""
For j = LBound(arr, 3) To UBound(arr, 3)
    k = 0
    For i = LBound(arr, 1) To UBound(arr, 1)
        For n = LBound(arr, 2) To UBound(arr, 2)
            arr(i, n, j) = k
            k = k + 1
            s = s & arr(i, n, j) & ", "
        Next
    Next
    s = s & vbCrLf & vbCrLf
Next
MsgBox s, vbOKOnly, "Array values before ComputeThreeDArray()"
i = WordOCX1.ComputeThreeDArray(arr, wdoAdd, 1.5)
s = ""
For j = LBound(arr, 3) To UBound(arr, 3)
    For i = LBound(arr, 1) To UBound(arr, 1)
        For n = LBound(arr, 2) To UBound(arr, 2)
            s = s & arr(i, n, j) & ", "
        Next
    Next
    s = s & vbCrLf & vbCrLf
Next
MsgBox s, vbOKOnly, "Array values after ComputeThreeDArray()"
```

ComputeTwoDArray() - Function to add/subtract/multiply/divide all items in a two dimensional array by specified number.

Arguments:

varArray As Variant
lngCompute As Long
dblNumber As Double

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Invalid value passed for variable lngCompute (valid values are 0,1,2,3)
-3 = Unknown error

Example:

```
Dim arr(0 To 9, 0 To 9) As Variant
Dim i As Integer
Dim n As Integer
Dim k As Integer
Dim s As String
```

```

s = ""
k = 0
For i = LBound(arr, 1) To UBound(arr, 1)
    For n = LBound(arr, 2) To UBound(arr, 2)
        arr(i, n) = k
        k = k + 1
        s = s & arr(i, n) & ","
    Next
    s = s & vbCrLf
Next
MsgBox s, vbOKOnly, "Array values before ComputeTwoDArray()"
i = WordOCX1.ComputeTwoDArray(arr, wdoAdd, 1.5)
s = ""
For i = LBound(arr, 1) To UBound(arr, 1)
    For n = LBound(arr, 2) To UBound(arr, 2)
        s = s & arr(i, n) & ","
    Next
    s = s & vbCrLf
Next
MsgBox s, vbOKOnly, "Array values after ComputeTwoDArray()"

*****

```

CreateChart() - Function to create a chart in a Word document.

Arguments:

```

varArray As Variant
Optional lngChart As Long = wdoLine
Optional strTitle As String
Optional strAxisTitle As String
Optional intLeft As Integer = 100
Optional intTop As Integer = 100
Optional intWidth As Integer = 350
Optional intHeight As Integer = 200
Optional intChartFontSize As Integer = 8
Optional intChartTitleFontSize As Integer = 12
Optional blnAutoScaleFont As Boolean

```

Returns: Integer

```

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = varArray is not an array
-4 = Unknown error

```

Example:

```

Dim i As Integer
Dim arr(3, 2) As Variant
'populate X-Axis with products
arr(0, 0) = ""
arr(1, 0) = "Widgets"
arr(2, 0) = "Gadgets"
arr(3, 0) = "Gizmos"
'populate Y-Axis with years
arr(0, 1) = "1999"
arr(0, 2) = "2000"
'now add the sales data
arr(1, 1) = 75323
arr(1, 2) = 34902
arr(2, 1) = 57357

```

```
arr(2, 2) = 28643
arr(3, 1) = 30577
arr(3, 2) = 46981
i = WordOCX1.CreateChart(arr, wdo3DBarClustered, "Annual Sales", "Dollars ($) ", 100, 100, 350, 200, 8, 10, False)
```

CreateDocument() - Function to create a Word document.

Arguments:

strDocument As String
Optional strPassword As String
Optional strWritePassword As String

Returns: Integer

1 = Success
-1 = Unknown error

Example:

```
Dim i As Integer
i = WordOCX1.CreateDocument(App.Path & "\MyFile.doc", "password", "password")
```

CreateHTML() - Function to create a HTML file from the active Word document.

Arguments:

strHTMLFile As String

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer
'Create a HTML file from the active document
i = WordOCX1.CreateHTML("C:\MyHTML.htm")
```

CreatePDF() - Function to create a PDF file from the active Word document.

Arguments:

strPDFFile As String
Optional blnCommaDelimited As Boolean
Optional strFontName As String = "Courier"
Optional intFontSize As Integer = 10
Optional intRotation As Integer
Optional sngWidth As Single = 8.5
Optional sngHeight As Single = 11
Optional strAuthor As String
Optional strCreator As String
Optional strKeywords As String

Optional strSubject As String

Optional strTitle As String

Returns: Integer

1 = Success

-1 = Word is not active

-2 = Word document not active

-3 = Unknown error

Example:

Dim i As Integer

'Create a PDF file from the active document

i = WordOCX1.CreatePDF("C:\MyPDF.pdf", False, "Times-Roman", 10, 0, 8.5, 11, "Frank", "Frank", "Add keywords here", "Add subject here", "Add title here")

CreatePDFFromFile() - Function to create a PDF file from a text file

Arguments:

strTextFile As String

strPDFFile As String

Optional strFontName As String = "Courier"

Optional intFontSize As Integer = 10

Optional intRotation As Integer

Optional sngWidth As Single = 8.5

Optional sngHeight As Single = 11

Optional strAuthor As String

Optional strCreator As String

Optional strKeywords As String

Optional strSubject As String

Optional strTitle As String

Returns: Integer

1 = Success

-1 = Text file does not exist

-2 = Unknown error

Example:

Dim i As Integer

i = WordOCX1.CreatePDFFromFile(App.Path & "\file.txt", App.Path & "\file.pdf")

CreatePDFUsingDistiller() - Function to create a PDF file from the active Word document using the Distiller.

Note: Adobe Acrobat Distiller must be installed before using this function

Arguments:

strPDFFile As String

Optional strDistiller As String = "Acrobat Distiller"

Returns: Integer

1 = Success

-1 = Word is not active

-2 = Word document not active

-3 = Unknown error

Example:

Dim i As Integer

'Create a PDF file from the active document - requires that Adobe Acrobat Distiller is installed

MsgBox "You must have Adobe Acrobat Distiller installed before this function will work." & vbCrLf & "The Distiller is not installed by default when installing Adobe Acrobat." & vbCrLf & "You must perform a custom install rather than a typical install then select the Distiller.", vbOKOnly, "Warning Message"

i = WordOCX1.CreatePDFUsingDistiller(App.Path & "\test.pdf")

'Note: If you are having trouble with this function you may need to

'adjust your Adobe Distiller settings

'To do this:

'1) Select "Printers" from Window's Start Menu

'2) Right click on the Acrobat Distiller printer to see menu

'2) Select "Printing Preferences..." from the menu

'3) Select tab "Adobe PDF Settings"

'4) Uncheck option "Do not send fonts to Distiller"

'5) Click the OK BUtton to save your changes

'6) Try running function CreatePDFUsingDistiller again

DatabaseTableToArray() - Function to move database table records to an array.

Arguments:

strDSN As String

strSQL As String

varArray As Variant

Optional strUserID As String

Optional strPassword As String

Optional blnIncludeHeadings As Boolean

Returns: Integer

1 = Success

-1 = varArray is not an array

-2 = Unable to connect to database/Problem with DSN

-3 = Unable to open recordset

-4 = Unknown error

Example:

Dim i As Integer

dim arr() As Variant

i = WordOCX1.DatabaseTableToArray("MyDSN","SELECT * FROM MyTable",arr,"",",",False)

DatabaseTableToTextFile() - Function to move database table records to a text file.

strDSN As String

strSQL As String

strTextFile As String

Optional strUserID As String

Optional strPassword As String

Optional blnIncludeHeadings As Boolean

Optional ByVal strDelimiter As String

Optional blnQuotes As Boolean

Returns: Integer

1 = Success
-1 = Unable to connect to database/Problem with DSN
-2 = Unable to open recordset
-3 = Unknown error

Example:

```
Dim i As Integer
'Create a comma-delimited text file from a recordset
i = WordOCX1.DatabaseTableToTextFile("MyDSN", "Select * From Authors", c:\Authors.csv", "", "", True, ",")
```

DataGridToArray() - Function to copy DataGrid to one dimensional array.

Arguments:

dg As Object
varArray as Variant
Optional blnIncludeHeadings As Boolean

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

```
Dim i As Integer
dim arr() As Variant
i = WordOCX1.DataGridToArray(DataGrid1, arr, False)
```

Notes:

Before invoking this function make sure that the record pointer is positioned on the first record in the Data Grid otherwise undesired results may occur.

DictionaryToArray() - Function to copy Dictionary items to an array.

Arguments:

dt As Object
varArray As Variant
Optional blnIncludeKeys As Boolean

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

```
Dim i As Integer
Dim vItem As Variant
Dim vKey As Variant
Dim Dict As Scripting.Dictionary
Set Dict = New Scripting.Dictionary 'create new instance of Dictionary Object
Dict.CompareMode = BinaryCompare 'set compare mode. Also DatabaseCompare & TextCompare
Dict.Add "Key1", "Item1"
```

```
Dict.Add "Key2", "Item2"
Dict.Add "Key3", "Item3"
Dict.Add "Key4", "Item4"
i = WordOCX1.DictionaryToArray(Dict, arr, True)
Set Dict = Nothing 'Destroy Dictionary Object
```

DirListBoxToOneDArray() - Function to copy DirListBox items to a one dimensional array.

Arguments: None

dlb As Object
varArray As Variant

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

```
Dim i As Integer
Dim arr() As Variant
i = WordOCX1.DirListBoxToOneDArray(Dir1, arr)
```

DocumentCount() - Function to get the number of active documents.

Arguments: None

Returns: Long

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Long
i = WordOCX1.DocumentCount
MsgBox "There are " & CStr(i) & " active documents."
```

DownloadFile() - Function to download a file from a remote web server.

Note: An internet connection must be established before using this function.

Arguments:

inet As Object
strRemoteHost As String
strRemoteDirectory As String
strLocalFile As String
strRemoteFile As String
Optional strUsername As String
Optional strUserPassword As String

Returns: Integer

1 = Success
-1 = Unknown error

Example:

```
Dim i As Integer
'Note: you must modify the remote host settings before this will work
i = WordOCX1.DownloadFile(Inet1, "microsoft.com", "/mydirectory", "c:\file.txt", "file.txt")
If i = 1 Then
    MsgBox "File downloaded successfully!", vbOKOnly
Else
    MsgBox "Unable to download file!", vbOKOnly
End If
```

DriveListBoxToOneDArray() - Function to copy DriveListBox items to a one dimensional array.

Arguments: None

dlb As Object
varArray As Variant

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

```
Dim i As Integer
Dim arr() As Variant
i = WordOCX1.DriveListBoxToOneDArray(Drive1, arr)
```

EmailDocument() - Function to email the active document.

Note: Microsoft Mail must be installed.

Arguments:

Optional blnSendAsAttachment As Boolean

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Microsoft Mail not installed
-4 = Unknown error

Example:

```
Dim i As Integer
i = WordOCX1.EmailDocument(True)
```

ExcelToOneDArray() - Function to copy a an Excel workbook worksheet range to a one dimensional array.

Arguments: None

strWorkbook As String
varWorksheet As Variant
lngStartRow As Long
lngStartCol As Long
lngEndRow As Long
lngEndCol As Long
varArray As Variant
Optional blnCellValue As Boolean
Optional strPassword As String
Optional strWriteResPassword As String

Returns: Integer

1 = Success
-1 = Unable to create an Excel object
-2 = varArray is not an array
-3 = Unable to open workbook
-4 = Unable to open worksheet
-5 = Unknown error

Example:

```
Dim i As Integer
Dim arr() As Variant
i = WordOCX1.ExcelToOneDArray(App.Path & "\test.xls", 1, 1, 1, 6, 1, arr, True)
```

ExcelToTwoDArray() - Function to copy a an Excel workbook worksheet range to a two dimensional array.

Arguments: None

strWorkbook As String
varWorksheet As Variant
lngStartRow As Long
lngStartCol As Long
lngEndRow As Long
lngEndCol As Long
varArray As Variant
Optional blnCellValue As Boolean
Optional strPassword As String
Optional strWriteResPassword As String

Returns: Integer

1 = Success
-1 = Unable to create an Excel object
-2 = varArray is not an array
-3 = Unable to open workbook
-4 = Unable to open worksheet
-5 = Unknown error

Example:

```
Dim i As Integer
Dim arr As Variant
i = WordOCX1.ExcelToTwoDArray(App.Path & "\test.xls", 1, 1, 1, 10, 5, arr, True)
```

FileListBoxToOneDArray() - Function to copy FileListBox items to a one dimensional array.

Arguments: None

flb As Object
varArray As Variant
Optional blnSelectedOnly As Boolean

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

```
Dim i As Integer
Dim arr() As Variant
i = WordOCX1.FileListBoxToOneDArray(File1, arr, True)
```

FlexGridToArray() - Function to copy FlexGrid items to an array.

Arguments:

fg As Object
varArray As Variant
Optional lngStartRow As Long
Optional lngStartCol As Long
Optional ByVal lngEndRow As Long
Optional ByVal lngEndCol As Long

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

```
Dim arr() As Variant
Dim i As Integer
Dim n As Integer
Dim c As String
Screen.MousePointer = vbHourglass
i = WordOCX1.FlexGridToArray(MSFlexGrid1, arr, 0, 0)
Screen.MousePointer = vbDefault
c = ""
For i = 0 To UBound(arr, 1)
    For n = 0 To UBound(arr, 2)
        c = c & arr(i, n) & "  "
    Next
    c = c & vbCrLf
Next
MsgBox c, vbOKOnly, "Array Items"
```

GetDocumentNames() - Function to store names of all the active documents in a one dimensional array.

Arguments:

varArray As Variant

Returns: Integer

1 = Success
-1 = Word is not active
-2 = No active documents
-3 = Unknown error

Example:

```
Dim arr() As Variant
Dim s As String
Dim i As Integer
i = WordOCX1.GetDocumentNames(arr)
For i = 0 To UBound(arr)
    s = s & arr(i) & vbCrLf
Next
MsgBox s, vbOKOnly, "Active Documents"
```

GetDocumentObject() - Function to get the active Word document object.

Arguments:

objWordDocument As Object

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document is not active
-3 = Unknown error

Example:

```
Dim i As Integer
Dim objDoc As Object
i = WordOCX1.GetDocumentObject(objDoc)
MsgBox objDoc.Name
```

GetWordObject() - Function to get the active Word application object.

Arguments:

objWordApp As Object

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Unknown error

Example:

```
Dim i As Integer
Dim objWord As Object
i = WordOCX1.GetWordObject(objWord)
MsgBox objWord.Name
```

HFlexGridToArray() - Function to copy HFlexGrid items to an array.

Arguments:

fg As Object
varArray As Variant
Optional lngStartRow As Long
Optional lngStartCol As Long
Optional ByVal lngEndRow As Long
Optional ByVal lngEndCol As Long

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

```
Dim arr() As Variant
Dim i As Integer
Dim n As Integer
Dim c As String
Screen.MousePointer = vbHourglass
i = WordOCX1.HFlexGridToArray(MSHFlexGrid1, arr, 0, 0)
Screen.MousePointer = vbDefault
c = ""
For i = 0 To UBound(arr, 1)
    For n = 0 To UBound(arr, 2)
        c = c & arr(i, n) & "  "
    Next
    c = c & vbCrLf
Next
MsgBox c, vbOKOnly, "Array Items"
```

HideWord() - Function to hide Word.

Arguments: None

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Unknown error

Example:

```
Dim i As Integer
i = WordOCX1.HideWord
```

InsertBlankLines () - Function to insert blank lines into the active document.

Arguments:

intLines As Integer
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = intLines is invalid
-4 = Unknown error

Example:

```
Dim i As Integer
Dim n As Long
n = 4
i = WordOCX1.InsertBlankLines(n)
```

InsertHyperlink () - Function to insert a hyperlink into the active document.

Arguments:

strText As String
strHTTP As String
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long
Optional lngColor As Long
Optional blnItalic As Boolean
Optional blnUnderline As Boolean
Optional lngAnimation As Long
Optional blnEmboss As Boolean
Optional blnEngrave As Boolean
Optional blnShadow As Boolean
Optional blnStrikeThrough As Boolean
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer
i = WordOCX1.InsertHyperlink("Visit SkySof Software's web page by clicking here", "http://ic.net/~kusluski",
False, , 12, 15, wdoAlignParagraphCenter, wdoColorDarkBlue, False, False, wdoAnimationLasVegasLights, False,
False, False, False, wdoEndofDoc)
```

InsertHyperlinks () - Function to insert one or more hyperlinks into the active document.

Arguments:

varArray As Variant
Optional blnBold As Boolean
Optional strFont As String = "Times New Roman"
Optional intSize As Integer = 12
Optional intSpaceAfter As Integer = 10
Optional lngParagraphAlign As Long

Optional lngColor As Long
Optional blnItalic As Boolean
Optional blnUnderline As Boolean
Optional lngAnimation As Long
Optional blnEmboss As Boolean
Optional blnEngrave As Boolean
Optional blnShadow As Boolean
Optional blnStrikeThrough As Boolean
Optional lngInsertionPoint As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = varArray is not an array
-4 = Unknown error

Example:

```
Dim i As Integer
Dim arr(2, 1) As String
arr(0, 0) = "Visit SkySof Software's Web Page"
arr(0, 1) = "http://ic.net/~kuslusi"
arr(1, 0) = "Visit Microsoft's Web Page"
arr(1, 1) = "http://www.microsoft.com"
arr(2, 0) = "Visit USA Today's Web Page"
arr(2, 1) = "http://www.usatoday.com"
i = WordOCX1.TextToWord("InsertHyperlinks() example:", True, "Arial", 12, 15, wdoAlignParagraphCenter,
RGB(0, 255, 0), True, True)
i = WordOCX1.InsertHyperlinks(arr, False, "Arial", 12, 5, wdoAlignParagraphCenter, wdoColorDarkBlue, False,
False, wdoAnimationLasVegasLights, False, False, False, False, wdoEndofDoc)
```

InsertLinkedPicture() – Function to insert a linked picture file into the active document.

Arguments:

strFile As String
Optional sngLeft As Single
Optional sngTop As Single
Optional sngWidth As Single
Optional sngHeight As Single

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer
'Note: Passed numbers for Left, Top, Width, and Height indicate pixels
i = WordOCX1.InsertLinkedPicture(App.Path & "\file.jpg", 100, 50, 200, 150)
'To insert picture at original size omit optional arguments:
i = WordOCX1.InsertLinkedPicture(App.Path & "\file.jpg")
```

InsertPageBreak() – Function to insert a page break into the active document.

Arguments: None

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer
i = WordOCX1.InsertPageBreak
```

InsertPicture() – Function to insert a picture file into the active document.

Arguments:

strFile As String
Optional sngLeft As Single
Optional sngTop As Single
Optional sngWidth As Single
Optional sngHeight As Single

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer
'Note: Passed numbers for Left, Top, Width, and Height indicate pixels
i = WordOCX1.InsertPicture(App.Path & "\file.jpg", 100, 50, 200, 150)
'To insert picture at original size omit optional arguments:
i = WordOCX1.InsertPicture(App.Path & "\file.jpg")
```

IsWordAvailable() - Function to determine if Word is installed on a machine.

Arguments: None

Returns: String

Empty string = Word is not installed
Populated String = Word version number

Example:

```
Dim s As String
s = WordOCX1.IsWordAvailable
If Len(s) Then
    MsgBox "Word version " & s & " is installed on this machine."
Else
    MsgBox "Word is not installed on this machine."
End If
```

IsWordUp() - Function to determine if Word is currently running.

Arguments: None

Returns: Boolean

True = Word is running

False = Word is not running

Example:

If WordOCX1.IsWordUp Then

 MsgBox "Word lives!"

Else

 MsgBox "Word has not been loaded."

End If

ListBoxToOneDArray() - Function to copy ListBox items to a one dimensional array.

Arguments:

lb As Object

varArray As Variant

Optional blnSelectedOnly As Boolean

Returns: Integer

1 = Success

-1 = Object is not a ListBox

-2 = varArray is not an array

-3 = Unknown error

Example:

Dim i As Integer

Dim arr() As Variant

i = WordOCX1.ListBoxToOneDArray(List1, arr, True)

ListViewToArray() - Function to copy ListView items to an array.

Arguments:

lv As Object

varArray As Variant

Optional blnSelectedOnly As Boolean

Optional blnIncludeHeadings As Boolean

Returns: Integer

1 = Success

-1 = varArray is not an array

-2 = Unknown error

Example:

Dim i As Integer

Dim arr() As Variant

```
i = WordOCX1.ListViewToArray(ListView1, arr, False, True)
```

MultiLineTextToOneDArray() - Function to copy multi-line text to a one dimensional array.

Arguments:

```
ByVal strText As String  
varArray As Variant  
Optional ByVal strDelimiter As String
```

Returns: Integer

```
1 = Success  
-1 = varArray is not an array  
-2 = Unknown error
```

Example:

```
Dim i As Integer  
Dim arr() As Variant  
i = WordOCX1.MultiLineTextToOneDArray ("4.5^12/12/2001^string3", arr, "^")
```

NumberOfPages () - Function to calculate the number of pages in the current/passed document.

Arguments:

```
lngPages As Long  
Optional strDocument As String
```

Returns: Integer

```
1 = Success  
-1 = Word is not active  
-2 = Word document not active  
-3 = Unknown error
```

Example:

```
Dim i As Integer  
Dim lngNoPages As Long  
'calculate the number of pages in the active document  
i = WordOCX1.NumberOfPages(lngNoPages)  
MsgBox lngNoPages, vbOKOnly, "Number of pages"
```

OneDArrayToHTMLFile() - Function to copy 1D array to an HTML file.

Arguments:

```
varArray As Variant  
strHTMLFile As String  
Optional strPageHeading As String  
Optional strPageTitle As String  
Optional strPageBackground As String  
Optional strPageBackgroundColor As String  
Optional lngTableAlign As Long  
Optional strBackground As String  
Optional strBackgroundColor As String
```

Optional intBorder As Integer
Optional strBorderColor As String
Optional intCellSpacing As Integer
Optional intCellPadding As Integer
Optional intWidthPct As Integer = 100
Optional lngTableCellAlign As Long
Optional blnWrapCellData As Boolean
Optional blnBoldFirstRow As Boolean

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

```
Dim i As Integer
Dim arr(0 To 5) As Variant
Dim strTable As String
arr(0) = "apple"
arr(1) = "orange"
arr(2) = "mango"
arr(3) = "plum"
arr(4) = "lemon"
arr(5) = "pineapple"
'sort the array in ascending order
i = WordOCX1.SortOneDArray(arr, wdoAscending)
'create a HTML file
i = WordOCX1.OneDArrayToHTMLFile(arr, App.Path & "\table.htm", CStr(UBound(arr) + 1) & " items",
"My Web Page", App.Path & "image.jpg", "", wdoTableAlignCenter, "", "cyan", 1, "blue", 2, 2, 15,
wdoTableAlignLeft, False, False)
```

OneDArrayToHTMLTable() - Function to copy 1D array to an HTML table.

Arguments:

varArray As Variant
strTable As String
Optional lngTableAlign As Long
Optional strBackground As String
Optional strBackgroundColor As String
Optional intBorder As Integer
Optional strBorderColor As String
Optional intCellSpacing As Integer
Optional intCellPadding As Integer
Optional intWidthPct As Integer = 100
Optional lngTableCellAlign As Long
Optional blnWrapCellData As Boolean
Optional blnBoldFirstRow As Boolean

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

```
Dim i As Integer
Dim arr(0 To 5) As Variant
```

```

Dim strTable As String
arr(0) = "apple"
arr(1) = "orange"
arr(2) = "mango"
arr(3) = "plum"
arr(4) = "lemon"
arr(5) = "pineapple"
'sort the array in ascending order
i = WordOCX1.SortOneDArray(arr, wdoAscending)
'create a HTML table - store in variable strTable
i = WordOCX1.OneDArrayToHTMLTable(arr, strTable, wdoTableAlignCenter, "", "cyan", 1, "blue", 2, 15, wdoTableAlignRight, False, False)
MsgBox strTable, vbOKOnly, "Contents of variable strTable"
'now create a web page to display the table
strTable = "<HTML><HEAD><CENTER><STRONG>Table Example</STRONG></CENTER><BR><TITLE>Table Example</TITLE></HEAD>" & _
"<BODY BGCOLOR='White'>" & vbCrLf & strTable & vbCrLf & "</BODY></HTML>"
MsgBox strTable, vbOKOnly, "Contents of variable strTable"
'save contents of variable strTable to a HTML file
Open App.Path & "\table.htm" For Output As #1
Print #1, strTable
Close #1

```

OneDArrayToText() - Function to copy a one dimensional array to a string variable.

Arguments:

```

varArray As Variant
strText As String
strEndOfRowChar As String
Optional strEndOfArrayChar As String
Optional blnQuotes As Boolean

```

Returns: Integer

```

1 = Success
-1 = varArray is not an array
-2 = Unknown error

```

Example:

```

Dim arr(0 To 4) As Variant
Dim s As String
Dim i As Integer
arr(0) = "The Wizard of Oz"
arr(1) = "Dirty Harry"
arr(2) = "Terminator"
arr(3) = "Star Wars"
arr(4) = "The Cable Guy"
i = WordOCX1.OneDArrayToText(arr, s, ", ", ".", True)
MsgBox s

```

OneDArrayToTextFile() - Function to copy a one dimensional array to a text file.

Arguments:

```

varArray As Variant
strTextFile As String
Optional blnQuotes As Boolean

```

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

```
Dim i As Integer
Dim arr(3) As Integer
arr(0) = 5
arr(1) = 2
arr(2) = 9
arr(3) = 4
i = WordOCX1.OneDArrayToTextFile(arr, "C:\MyFile.csv", True)
```

OpenDocument() - Function to open a Word document.

Arguments:

Optional strDocument As String
Optional blnSwitchToWord As Boolean
Optional strPassword As String
Optional strWriteResPassword As String
Optional strFormCaption As String
Optional blnHideWord As Boolean
Optional lngPageOrientation As Long
Optional sngTopMargin As Single
Optional sngLeftMargin As Single
Optional sngRightMargin As Single
Optional sngBottomMargin As Single
Optional sngPageHeight As Single
Optional sngPageWidth As Single

Returns: Integer

1 = Success
-1 = Can't create Word object
-2 = Can't open Word document
-3 = Unknown error

Example:

```
Dim i As Integer
i = WordOCX1.OpenDocument(App.Path & "\test.doc", False, , , Me.Caption, False)
```

PrintDocument() - Function to print a document.

Arguments:

Optional varDocument As Variant
Optional blnSwitchToWord As Boolean
Optional strFormCaption As String

Returns: Integer

1 = Success
-1 = Word is not active

-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer  
i = WordOCX1.PrintDocument(, True)
```

PrintPreviewDocument() - Function to print preview a Word document.

Arguments:

Optional varDocument As Variant
Optional blnSwitchToWord As Boolean
Optional strFormCaption As String

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer  
i = WordOCX1.PrintPreviewDocument(, True)
```

RecordsetToArray() - Function to copy a recordset to an array.

Arguments:

objRecd As Object
varArray As Variant
Optional blnIncludeHeadings As Boolean

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

```
Dim i As Integer  
Dim strConn As String  
Dim cn As ADODB.Connection  
Dim rs As ADODB.Recordset  
Dim arr() As Variant  
On Error Resume Next  
strConn = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path & _  
"\author.mdb;Persist Security Info=False"  
Set cn = New ADODB.Connection  
cn.CursorLocation = adUseClient  
cn.Open strConn  
Set rs = New ADODB.Recordset  
rs.CursorType = adOpenStatic  
rs.CursorLocation = adUseClient  
rs.LockType = adLockPessimistic
```

```
rs.Source = "Select * From Authors"
rs.ActiveConnection = cn
rs.Open
i = WordOCX1.RecordsetToArray(rs, arr, False)
rs.Close
cn.Close
Set rs = Nothing
Set cn = Nothing
```

RecordsetToTextFile() - Function to move a recordset to a text file.

Arguments:

```
objRecd As Object
strTextFile As String
Optional blnIncludeHeadings As Boolean
Optional ByVal strDelimiter As String
Optional blnQuotes As Boolean
```

Returns: Integer

```
1 = Success
-1 = Unknown error
```

Example:

```
Dim strConn As String
Dim cn As ADODB.Connection
Dim rs As ADODB.Recordset
Dim i As Integer
Dim n As Integer
Dim c As String
On Error Resume Next
Screen.MousePointer = vbHourglass
strConn = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path & _
"\author.mdb;Persist Security Info=False"
Set cn = New ADODB.Connection
cn.CursorLocation = adUseClient
cn.Open strConn
Set rs = New ADODB.Recordset
rs.CursorType = adOpenStatic
rs.CursorLocation = adUseClient
rs.LockType = adLockPessimistic
rs.Source = "Select * From Authors"
rs.ActiveConnection = cn
rs.Open
'Create comma-delimited text file from a recordset
i = WordOCX1.RecordsetToTextFile(rs, App.Path & "\authors.txt", True, ",", True)
rs.Close
cn.Close
Set rs = Nothing
Set cn = Nothing
Screen.MousePointer = vbDefault
```

RichTextBoxToOneDArray() - Function to copy rich textbox text to a one dimensional array.

Arguments:

```
tb As Object
```

varArray As Variant

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

```
Dim i As Integer
Dim arr() As Variant
i = WordOCX1.RichTextBoxToOneDArray(RichText1, arr)
```

SaveDocument() - Function to save the active Word document.

Arguments:

Optional strDocument As String
Optional strPassword As String
Optional strWritePassword As String

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Notes:

You must specify a document name when setting passwords.

Example:

This example saves the workbook with the same name

```
Dim i As Integer
i = WordOCX1.SaveDocument
If i = 1 Then
    MsgBox "Document saved."
End If
```

This example saves a copy of the active document with a different name

```
Dim i As Integer
i = WordOCX1.SaveDocument("c:\files\NewDocument.doc", "Password", "WritePassword")
If i = 1 Then
    MsgBox "Document saved."
End If
```

SetPageBottomMargin () - Function to set the bottom margin of the document's page.

Arguments:

sngBottomMargin As Single

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer
i = WordOCX1.SetPageBottomMargin(1.5)
```

SetPageHeight () - Function to set the document's page height.

Arguments:

sngPageHeight As Single

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer
i = WordOCX1.SetPageHeight(300.5)
```

SetPageLeftMargin () - Function to set the left margin of the document's page.

Arguments:

SngLeftMargin As Single

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer
i = WordOCX1.SetPageLeftMargin(1.5)
```

SetPageNumberInFooter () - Function to set the page number in the document's footer.

Arguments:

lngPageNumberAlign As Long
Optional blnFirstPage As Boolean

Returns: Integer

1 = Success
-1 = Word is not active

-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer  
i = WordOCX1.SetPageNumberInFooter(wdoPageAlignRight, True)
```

SetPageNumberInHeader () - Function to set the page number in the document's header.

Arguments:

lngPageNumberAlign As Long
Optional blnFirstPage As Boolean

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer  
i = WordOCX1.SetPageNumberInHeader(wdoPageAlignRight, True)
```

SetPageOrientation () - Function to set the page orientation mode to portrait or landscape.

Arguments:

lngPageOrientation As Long

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown Error

Example:

```
Dim i As Integer  
i = WordOCX1.SetPageOrientation(wdoOrientPortrait)
```

SetPageRightMargin () - Function to set the right margin of the document's page.

Arguments:

sngRightMargin As Single

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer
i = WordOCX1.SetPageRightMargin(1.5)
```

SetPageTopMargin () - Function to set the top margin of the document's page.

Arguments:

sngTopMargin As Single

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer
i = WordOCX1.SetPageTopMargin(1.5)
```

SetPageWidth () - Function to set the document's page width.

Arguments:

sngPageWidth As Single

Returns: Integer

1 = Success
-1 = Word is not active
-2 = Word document not active
-3 = Unknown error

Example:

```
Dim i As Integer
i = WordOCX1.SetPageWidth(200.5)
```

SortOneDArray() - Function to sort a one dimensional array.

Arguments:

varArray As Variant
Optional ByVal lngSortOrder As Long

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

```
Dim i As Integer
dim arr(0 To 3) As String
arr(0) = "fish"
arr(1) = "dog"
arr(2) = "cat"
arr(3) = "bird"
i = WordOCX1.SortOneDArray(arr, wdoAscending)
```

SwitchToWorld () - Function to activate Word.

Arguments: None

Returns: Integer

```
1 = Success
-1 = Word is not active
-2 = Word document not active
```

Example:

```
Dim i As Integer
i = WordOCX1.SwitchToWorld
```

TextBoxToOneDArray() - Function to copy textbox text to a one dimensional array.

Arguments:

```
tb As Object
varArray As Variant
```

Returns: Integer

```
1 = Success
-1 = Object is not a TextBox
-2 = varArray is not an array
-3 = Unknown error
```

Example:

```
Dim i As Integer
Dim arr() As Variant
i = WordOCX1.TextBoxToOneDArray(Text1, arr)
```

TextFileToArray() - Function to copy text file contents to a two dimensional array.

Arguments:

```
strTextFile As String
varArray As Variant
Optional ByVal strDelimiter As String
Optional blnRemoveQuotes As Boolean
```

Returns: Integer

```
1 = Success
-1 = Text file not found
-2 = varArray is not an array
```

-3 = Unknown error

Example:

```
Dim i As Integer
Dim arr() As Variant
i = WordOCX1.TextFileToArray("c:\file.txt", arr, ",")
```

TextFileToOneDArray () - Function to copy text file contents to a one dimensional array.

Arguments:

```
strTextFile As String
varArray As Variant
```

Returns: Integer

1 = Success
-1 = Text file not found
-2 = varArray is not an array
-3 = Unknown error

Example:

```
Dim i As Integer
Dim arr() As Variant
i = WordOCX1.TextFileToArray("c:\file.txt", arr, ",")
```

TreeViewToTwoDArray() – Function to copy a two dimensional array to a TreeView object.

Arguments

```
tv As Object
varArray As Variant
Optional blnSuppressRows As Boolean
```

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

‘This example assumes a TreeView control is on your form

```
Dim i As Integer
Dim n As Integer
Dim arr() As Variant
Dim s As String
i = WordOCX1.TreeViewToTwoDArray(TreeView1, arr, True)
For i = 0 To UBound(arr, 1)
    For n = 0 To UBound(arr, 2)
        s = s & arr(i, n) & "    "
    Next
    s = s & vbCrLf
Next
MsgBox s, vbOKOnly, "Array Items"
```

TwoDArrayToCollection() - Function to copy a two dimensional array to a collection object.

Arguments

varArray As Variant
cl As Collection

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = cl is not a collection object
-3 = Unknown error

Example:

```
Dim objCol As Collection
Dim i As Integer
Dim s As String
Dim arr(0 To 2, 0 To 1) As Variant
'Create collection object
Set objCol = New Collection
'Add elements to 2D array
'Items
arr(0, 0) = "Apple"
arr(1, 0) = "Orange"
arr(2, 0) = "Mango"
'Indexes
arr(0, 1) = 1
arr(1, 1) = 2
arr(2, 1) = 3
i = WordOCX1.TwoDArrayToCollection(arr, objCol)
If i = 0 Then Exit Sub
For i = 1 To objCol.Count
    s = s & objCol.Item(i) & vbCrLf
Next
Set objCol = Nothing 'Destroy Collection Object
MsgBox s, vbOKOnly, "Collection Items"
```

TwoDArrayToDictionary() - Function to copy a two dimensional array to a dictionary object.

varArray As Variant
dt As Object

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

```
Dim arr() As String
Dim arr2(0 To 2, 0 To 1) As Variant
Dim i As Integer
Dim objDict As Scripting.Dictionary
Dim s As String
Dim vItem As Variant
Dim vKey As Variant
'Create dictionary object
```

```

Set objDict = New Scripting.Dictionary 'create new instance of objDictionary Object
objDict.CompareMode = BinaryCompare 'set compare mode. Also DatabaseCompare & TextCompare
'Add elements to 2D array
'Keys
arr2(0, 0) = 1
arr2(1, 0) = 2
arr2(2, 0) = 3
'Items
arr2(0, 1) = "Apple"
arr2(1, 1) = "Orange"
arr2(2, 1) = "Mango"
i = WordOCX1.TwoDArrayToDictionary(arr2, objDict)
If i <> 1 Then Exit Sub
'Get keys
i = 0
For Each vKey In objDict.Keys
    ReDim Preserve arr(i)
    arr(i) = "Key: " & vKey
    i = i + 1
Next
i = 0
'Get items
For Each vItem In objDict.Items
    arr(i) = arr(i) & ", Item: " & vItem
    i = i + 1
Next
For i = 0 To objDict.Count - 1
    s = s & arr(i) & vbCrLf
Next
Set objDict = Nothing 'Destroy Dictionary Object
MsgBox s, vbOKOnly, "Dictionary Items"

```

TwoDArrayToFlexGrid() - Function to copy a two dimensional array to a FlexGrid control.

```

varArray As Variant
fg As Object

```

Returns: Integer

```

1 = Success
-1 = varArray is not an array
-2 = Unknown error

```

Example:

'Note: This example assumes a FlexGrid control (named MSFlexGrid) is on your form.

```

Dim arr(0 To 10, 0 To 9) As Variant
Dim i As Integer
Dim n As Integer
Dim j As Integer
j = 1
For i = LBound(arr, 1) To UBound(arr, 1)
    For n = LBound(arr, 2) To UBound(arr, 2)
        arr(i, n) = j
        j = j + 1
    Next
Next
i = WordOCX1.TwoDArrayToFlexGrid(arr, MSFlexGrid1)

```

TwoDArrayToHTMLFile() - Function to copy 2D array to an HTML file.

Arguments:

varArray As Variant
strHTMLFile As String
Optional strPageHeading As String
Optional strPageTitle As String
Optional strPageBackground As String
Optional strPageBackgroundColor As String
Optional lngTableAlign As Long
Optional strBackground As String
Optional strBackgroundColor As String
Optional intBorder As Integer
Optional strBorderColor As String
Optional intCellSpacing As Integer
Optional intCellPadding As Integer
Optional intWidthPct As Integer = 100
Optional lngTableCellAlign As Long
Optional blnWrapCellData As Boolean
Optional blnBoldFirstRow As Boolean
Optional blnBoldFirstColumn As Boolean

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

```
Dim i As Integer
Dim arr(0 To 3, 0 To 3) As Variant
Dim strTable As String
arr(0, 0) = "Mary"
arr(1, 0) = "lamb"
arr(2, 0) = "white"
arr(3, 0) = "that"
arr(0, 1) = "had"
arr(1, 1) = "whose"
arr(2, 1) = "as"
arr(3, 1) = "Mary"
arr(0, 2) = "a"
arr(1, 2) = "fleece"
arr(2, 2) = "snow"
arr(3, 2) = "went"
arr(0, 3) = "little"
arr(1, 3) = "was"
arr(2, 3) = "everywhere"
arr(3, 3) = "that lamb was sure to go!"
'create a HTML file
i = Form1.WordOCX1.TwoDArrayToHTMLFile(arr, App.Path & "\table.htm", CStr(UBound(arr)) & " rows", "My
Web Page", App.Path & "\image.jpg", "", wdoTableAlignCenter, "", "#FFD700", 1, "black", 2, 2, 30,
wdoTableAlignRight, False, True, True)
```

TwoDArrayToHTMLTable() - Function to copy 2D array to an HTML table.

Arguments:

varArray As Variant
strTable As String

Optional lngTableAlign As Long
 Optional strBackground As String
 Optional strBackgroundColor As String
 Optional intBorder As Integer
 Optional strBorderColor As String
 Optional intCellSpacing As Integer
 Optional intCellPadding As Integer
 Optional intWidthPct As Integer = 100
 Optional lngTableCellAlign As Long
 Optional blnWrapCellData As Boolean
 Optional blnBoldFirstRow As Boolean
 Optional blnBoldFirstColumn As Boolean

Returns: Integer

1 = Success
 -1 = varArray is not an array
 -2 = Unknown error

Example:

```
Dim strTable As String
Dim arr(0 To 3, 0 To 3) As Variant
Dim i As Integer
Dim n As Integer
Dim s As String
arr(0, 0) = "Mary"
arr(1, 0) = "lamb"
arr(2, 0) = "white"
arr(3, 0) = "that"
arr(0, 1) = "had"
arr(1, 1) = "whose"
arr(2, 1) = "as"
arr(3, 1) = "Mary"
arr(0, 2) = "a"
arr(1, 2) = "fleece"
arr(2, 2) = "snow"
arr(3, 2) = "went"
arr(0, 3) = "little"
arr(1, 3) = "was"
arr(2, 3) = "everywhere"
arr(3, 3) = "that lamb was sure to go!"
'create a HTML table - store in variable strTable
i = Form1.WordOCX1.TwoDArrayToHTMLTable(arr, strTable, wdoTableAlignCenter, App.Path & "\image.jpg",
"", 1, "black", 2, 2, 30, wdoTableAlignRight, False, True, True)
MsgBox strTable, vbOKOnly, "Contents of variable strTable"
'now create a web page to display the table
strTable = "<HTML><HEAD><CENTER><STRONG>Table
Example</STRONG></CENTER><BR><TITLE>Table Example</TITLE></HEAD>" & _
"<BODY BGCOLOR='White'>" & vbCrLf & strTable & vbCrLf & "</BODY></HTML>"
MsgBox strTable, vbOKOnly, "Contents of variable strTable"
'save contents of variable strTable to a HTML file
Open App.Path & "\table.htm" For Output As #1
Print #1, strTable
Close #1
```

TwoDArrayToListView() - Function to copy a two dimensional array to a ListView control.

varArray As Variant
 lv As Object
 Optional varSelectItems As Variant

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

'Note: This example assumes a ListView control (named ListView1) with three columns is on your form.

Dim i As Integer

Dim arr(0 To 3, 0 To 2) As Variant

'Names

arr(0, 0) = "Billy Bob"

arr(1, 0) = "Kelly Jones"

arr(2, 0) = "Wayne Cohen"

arr(3, 0) = "Barbara Lee"

'Ages

arr(0, 1) = 24

arr(1, 1) = 17

arr(2, 1) = 35

arr(3, 1) = 53

'Sex

arr(0, 2) = "Male"

arr(1, 2) = "Female"

arr(2, 2) = "Male"

arr(3, 2) = "Female"

i = WordOCX1.TwoDArrayToListView(arr, ListView1, Array("Billy Bob", "Wayne Cohen"))

TwoDArrayToText() - Function to copy a two dimensional array to a string variable.

Arguments:

varArray As Variant

strText As String

strEndOfRowChar As String

Optional strEndOfArrayChar As String

Optional blnQuotes As Boolean

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

Example:

Dim arr(0 To 3, 0 To 2) As Variant

Dim s As String

Dim i As Integer

arr(0, 0) = "Employee"

arr(1, 0) = "John Stevens"

arr(2, 0) = "Kevin Jackson"

arr(3, 0) = "Susan Smith"

arr(0, 1) = "Age"

arr(1, 1) = 28

arr(2, 1) = 33

arr(3, 1) = 52

arr(0, 2) = "Sex"

arr(1, 2) = "M"

arr(2, 2) = "M"

```
arr(3, 2) = "F"
```

```
i = WordOCX1.TwoDArrayToText(arr, s, vbTab, vbCrLf, True)
MsgBox s
```

```
*****
```

TwoDArrayToTextFile() - Function to copy a two dimensional array to a text file.

Arguments:

```
varArray As Variant
strTextFile As String
Optional ByVal strDelimiter As String
Optional blnQuotes As Boolean
```

Returns: Integer

```
1 = Success
-1 = varArray is not an array
-2 = Unknown error
```

Example:

```
Dim arr(0 To 10, 0 To 9) As Variant
Dim i As Integer
Dim n As Integer
Dim j As Integer
Screen.MousePointer = vbHourglass
j = 1
For i = LBound(arr, 1) To UBound(arr, 1)
    For n = LBound(arr, 2) To UBound(arr, 2)
        arr(i, n) = j
        j = j + 1
    Next
Next
'Create comma-delimited text file from a two dimensional array
i = WordOCX1.TwoDArrayToTextFile(arr, App.Path & "\TwoD.txt", ",", True)
Screen.MousePointer = vbDefault
'Open the comma-delimited text file in Notepad
Shell "notepad.exe " & App.Path & "\TwoD.txt", vbNormalFocus
```

```
*****
```

TwoDArrayToTreeView() – Function to copy a two dimensional array to a TreeView control.

Arguments:

```
varArray As Variant
tv As Object
Optional blnExpandAll As Boolean
```

Returns: Integer

```
1 = Success
-1 = varArray is not an array
-2 = Unknown error
```

Example:

```
'This example assumes a TreeView control is on your form
Dim arr(15, 3) As String
Dim i As Integer
```

```

arr(0, 0) = "Food"
arr(1, 1) = "Fruit"
arr(2, 2) = "Apples"
arr(3, 3) = "Granny Smith"
arr(4, 3) = "Macintosh"
arr(5, 2) = "Oranges"
arr(6, 3) = "Navel"
arr(7, 3) = "Valencia"
arr(8, 1) = "Veggies"
arr(9, 2) = "Beans"
arr(10, 3) = "Green"
arr(11, 3) = "Lima"
arr(12, 2) = "Lettuce"
arr(13, 3) = "Romaine"
arr(14, 3) = "Iceburg"
arr(15, 3) = "Green Leaf"
i = WordOCX1.TwoDArrayToTreeView(arr, TreeView1, True)

```

UnhideWord() - Function to unhide Word.

Arguments: None

Returns: Integer

```

1 = Success
-1 = Word is not active
-2 = Unknown error

```

Example:

```

Dim i As Integer
i = WordOCX1.UnhideWord

```

UploadFile() - Function to copy a local file to a remote web server.

Note: An internet connection must be established before using this function.

Arguments:

```

inet As Object
strRemoteHost As String
strRemoteDirectory As String
strLocalFile As String
strRemoteFile As String
Optional strUsername As String
Optional strUserPassword As String

```

Returns: Integer

```

1 = Success
-1 = Unknown error

```

Example:

```

Dim i As Integer
Dim arr(0 To 5) As Variant
Dim strTable As String
arr(0) = "apple"
arr(1) = "orange"

```

```

arr(2) = "mango"
arr(3) = "plum"
arr(4) = "lemon"
arr(5) = "pineapple"
'sort the array in ascending order
i = Form1.WordOCX1.SortOneDArray(arr, wdoAscending)
'create a HTML file
i = Form1.WordOCX1.OneDArrayToHTMLFile(arr, App.Path & "\table.htm", CStr(UBound(arr) + 1) & " items",
"My Web Page", App.Path & "\image.jpg", "", wdoTableAlignCenter, "", "cyan", 1, "blue", 2, 2, 15,
wdoTableAlignLeft, False, False)
If MsgBox("HTML file created. Do you want to upload it to a web server?", vbQuestion + vbYesNo, "HTML
Table") = vbYes Then
    'Note: you must modify the remote host settings before this will work
    i = WordOCX1.UploadFile(Inet1, "www.microsoft.com", "/mydirectory", App.Path & "\table.htm", "table.htm",
"username", "password")
    If i = 1 Then
        MsgBox "File uploaded successfully!", vbOKOnly
    Else
        MsgBox "Unable to upload file!", vbOKOnly
    End If
End If

```

XMLFileToArray() - Function to copy an XML table file's contents to a two dimensional array.

Arguments:

```

strXMLFile As String
varArray As Variant
Optional blnIncludeHeadings As Boolean

```

Returns: Integer

```

1 = Success
-1 = varArray is not an array
-2 = Unable to access XML DLL
-3 = unable to load XML file
-4 = Unknown error

```

Example:

```

Dim arr() As Variant
Dim i As Integer
Dim n As Integer
Dim s As String
i = WordOCX1.XMLFileToArray(App.Path & "\portfolio.xml", arr, True)
For i = 0 To UBound(arr, 1)
    For n = 0 To UBound(arr, 2)
        s = s & arr(i, n) & ", "
    Next
    s = s & vbCrLf
Next
MsgBox s, vbOKOnly, "Array Items"

```

XMLFileToRecordset() - Function to copy an XML table file's contents to a recordset.

Arguments:

```

strXMLFile As String
objRecd As Object

```

Returns: Integer

1 = Success
-1 = Unable to access XML DLL
-2 = unable to load XML file/Unknown error

Example:

```
Dim i As Integer
Dim rs As ADODB.Recordset
Dim fld As ADODB.Field
Dim s As String
Set rs = New ADODB.Recordset
i = WordOCX1.XMLFileToRecordset(App.Path & "\portfolio.xml", rs)
For Each fld In rs.Fields
    If fld.Name <> "$Text" Then
        s = s & fld.Name & ", "
    End If
Next
s = s & vbCrLf
Do While Not rs.EOF
    For Each fld In rs.Fields
        If fld.Name <> "$Text" Then
            s = s & fld.Value & ", "
        End If
    Next
    s = s & vbCrLf
    rs.MoveNext
Loop
MsgBox s, vbOKOnly, "Recordset Items"
```

XMLFileToTreeView() - Function to copy items from a XML file to a TreeView control.

Arguments:

strXMLFile As String
tv As Object
Optional blnExpandAll As Boolean

Returns: Integer

1 = Success
-1 = unable to create the XML ActiveX object
-2 = unable to load XML file
-3 = Unknown error

Example:

'This example assumes a TreeView control is on your form

```
Dim i As Integer
TreeView1.Nodes.Clear
i = WordOCX1.XMLFileToTreeView(App.Path & "\file.xml", TreeView1, False)
'To use an XML file located on a web server use the URL
'i = WordOCX1.XMLFileToTreeView("http://www.mypage.com/file.xml", TreeView1, True)
```