Filter by title

create files

Automate Word using Visual C# to create files

Call Excel macros

Count cells number with cell color using VBA

Create a RealTimeData server for Excel

Create script for Outlook Rules Wizard

Compile Error in VBA macro

Declare the return type explicitly in 64-bit macro

Display progress bar with user form in Excel

Emails move to the Junk E-mail folder when SCL value is -1

Embed and automate documents with VB

How to Automate Microsoft Word by using Visual Basic to create a new document

Article • 10/22/2021 • Applies to: Microsoft Word

In this article

Summary

References

Summary

This step-by-step article describes how to create a new document in Word by using Automation from Visual Basic.

Sample Code

The sample code in this article demonstrates how to do the following:

- Insert paragraphs with text and formatting.
- Browse and modify various ranges within a document.
- Insert tables, format tables, and populate the tables with data.
- Add a chart.

To create a new Word document by using Automation from Visual Basic, follow these steps:

1. In Visual Basic, create a new Standard EXE project. Form1 is created by default.

2. On the **Project** menu, click**References**, click one of the following options, and then click **OK**:

   - For Office Word 2007, click **Microsoft Word 12.0 Object Library**.
   - For Word 2003, click **Microsoft Word 11.0 Object Library**.
   - For Word 2002, click **Microsoft Word 10.0 Object Library**.
   - For Word 2000, click **Microsoft Word 9.0 Object Library**.

3. Add a CommandButton control to Form1.

4. Add the following code to the Click event for Command1:

VB Copy

```
Dim oWord As Word.Application
Dim oDoc As Word.Document
Dim oTable As Word.Table
Dim oPara1 As Word.Paragraph, oPara2 As Word.Paragraph
Dim oPara3 As Word.Paragraph, oPara4 As Word.Paragraph
Dim oRng As Word.Range
Dim oShape As Word.InlineShape
Dim oChart As Object
Dim Pos as Double
```

Use Office Web Components using ASP.NET

Use VB ActiveX for Word from Internet

```
'Start Word and open the document template.
Set oWord = CreateObject("Word.Application")
oWord.Visible = True
Set oDoc = oWord.Documents.Add

'Insert a paragraph at the beginning of the document.
Set oPara1 = oDoc.Content.Paragraphs.Add
oPara1.Range.Text = "Heading 1"
oPara1.Range.Font.Bold = True
oPara1.Format.SpaceAfter = 24    '24 pt spacing after paragraph.
oPara1.Range.InsertParagraphAfter

'Insert a paragraph at the end of the document.
'** \endofdoc is a predefined bookmark.
Set oPara2 = oDoc.Content.Paragraphs.Add(oDoc.Bookmarks("\endofdoc").Range
oPara2.Range.Text = "Heading 2"
oPara2.Format.SpaceAfter = 6
oPara2.Range.InsertParagraphAfter

'Insert another paragraph.
Set oPara3 = oDoc.Content.Paragraphs.Add(oDoc.Bookmarks("\endofdoc").Range
oPara3.Range.Text = "This is a sentence of normal text. Now here is a tabl
oPara3.Range.Font.Bold = False
oPara3.Format.SpaceAfter = 24
oPara3.Range.InsertParagraphAfter

'Insert a 3 x 5 table, fill it with data and make the first row
'bold,italic.
Dim r As Integer, c As Integer
Set oTable = oDoc.Tables.Add(oDoc.Bookmarks("\endofdoc").Range, 3, 5)
oTable.Range.ParagraphFormat.SpaceAfter = 6
For r = 1 To 3
    For c = 1 To 5
        oTable.Cell(r, c).Range.Text = "r" & r & "c" & c
    Next
Next
oTable.Rows(1).Range.Font.Bold = True
oTable.Rows(1).Range.Font.Italic = True

'Add some text after the table.
'oTable.Range.InsertParagraphAfter
Set oPara4 = oDoc.Content.Paragraphs.Add(oDoc.Bookmarks("\endofdoc").Range
oPara4.Range.InsertParagraphBefore
oPara4.Range.Text = "And here's another table:"
oPara4.Format.SpaceAfter = 24
oPara4.Range.InsertParagraphAfter

'Insert a 5 x 2 table, fill it with data and change the column widths.
Set oTable = oDoc.Tables.Add(oDoc.Bookmarks("\endofdoc").Range, 5, 2)
oTable.Range.ParagraphFormat.SpaceAfter = 6
For r = 1 To 5
    For c = 1 To 2
        oTable.Cell(r, c).Range.Text = "r" & r & "c" & c
    Next
Next
oTable.Columns(1).Width = oWord.InchesToPoints(2)   'Change width of colum
oTable.Columns(2).Width = oWord.InchesToPoints(3)

'Keep inserting text. When you get to 7 inches from top of the
'document, insert a hard page break.
Pos = oWord.InchesToPoints(7)
oDoc.Bookmarks("\endofdoc").Range.InsertParagraphAfter
Do
    Set oRng = oDoc.Bookmarks("\endofdoc").Range
    oRng.ParagraphFormat.SpaceAfter = 6
    oRng.InsertAfter "A line of text"
    oRng.InsertParagraphAfter
Loop While Pos >= oRng.Information(wdVerticalPositionRelativeToPage)
oRng.Collapse (wdCollapseEnd)
oRng.InsertBreak wdPageBreak
oRng.Collapse wdCollapseEnd
oRng.InsertAfter "We're now on page 2. Here's my chart:"
oRng.InsertParagraphAfter

'Insert a chart and change the chart.
Set oShape = oDoc.Bookmarks("\endofdoc").Range.InlineShapes.AddOLEObject(
    ClassType:="MSGraph.Chart.8", FileName _
    :="", LinkToFile:=False, DisplayAsIcon:=False)
```

```
Set oChart = oShape.OLEFormat.Object
oChart.charttype = 4 'xlLine = 4
oChart.Application.Update
oChart.Application.Quit
'... If desired, you can proceed from here using the Microsoft Graph
'Object model on the oChart object to make additional changes to the
'chart.
oShape.Width = oWord.InchesToPoints(6.25)
oShape.Height = oWord.InchesToPoints(3.57)

'Add text after the chart.
Set oRng = oDoc.Bookmarks("\endofdoc").Range
oRng.InsertParagraphAfter
oRng.InsertAfter "THE END."

'All done. Unload this form.
Unload Me
```

5. Press F5 to run the program and then click Command1.

After the code completes, examine the document that was created for you. The document contains two pages of formatted paragraphs, tables, and a chart.

Use a Template

If you are using Automation to build documents that are all in a common format, you can benefit from starting the process with a new document that is based on a preformatted template. Using a template with your Word Automation client has two significant advantages over building a document from nothing:

- You can have greater control over the formatting and placement of objects throughout your documents.
- You can build your documents with less code.

By using a template, you can fine-tune the placement of tables, paragraphs, and other objects within the document, as well as include formatting on those objects. By using Automation, you can create a new document based on your template with code such as the following:

VB Copy

```
oWord.Documents.Add "<Path to your template>\MyTemplate.dot"
```

In your template, you can define bookmarks so that your Automation client can fill in variable text at a specific location in the document, as follows:

VB Copy

```
oDoc.Bookmarks("MyBookmark").Range.Text = "Some Text Here"
```

Another advantage to using a template is that you can create and store formatting styles that you wish to apply at run time, as follows:

VB Copy

```
oDoc.Bookmarks("MyBookmark").Range.Style = "MyStyle"
```

or

VB Copy

```
oWord.Selection.Style = "MyStyle"
```

References

For additional information, click the article numbers below to view the articles in the Microsoft Knowledge Base:

285332 How To Automate Word 2002 with Visual Basic to Create a Mail Merge

Microsoft Office Development with Visual Studio