How to: Programmatically Hide Text in Documents

You can hide text in a document by setting the P:Microsoft.Office.Interop.Word._Font.Hidden property of the P:Microsoft.Office.Interop.Word.Range.Font for a particular range of text.

For example, you can temporarily hide the text within a Microsoft.Office.Tools.Word.Bookmark (in a document-level customization) or a Microsoft.Office.Interop.Word.Bookmark (in an VSTO Add-in) before sending a document to a printer.

Applies to: The information in this topic applies to document-level projects and VSTO add-in projects for Word. For more information, see Features Available by Office Application and Project Type.

To hide text in a Bookmark control while printing the document

1. Create a procedure that hides all text that is in a specified range.

   C#
   ```csharp
           static void HideText(Word.Range rng)
           {
               rng.Font.Hidden = 1;   // 1 = True
           }
   ```

2. Create a procedure that unhides all text that is in a specified range.

   C#
   ```csharp
           static void UnhideText(Word.Range rng)
           {
               rng.Font.Hidden = 0;   // 0 = False
           }
   ```

3. Pass the range of a bookmark to the HideText method, print the document, and then pass the same range to the UnhideText method.

   The following code example can be used in a document-level customization. To use this example, run it from the ThisDocument class in your project.

   C#
   ```csharp
           HideText(bookmark1.Range);
   ```

```csharp
            object oTrue = true;
            object oFalse = false;
            object range = Word.WdPrintOutRange.wdPrintAllDocument;
            object items = Word.WdPrintOutItem.wdPrintDocumentContent;
            object copies = "1";
            object pages = "";
            object pageType = Word.WdPrintOutPages.wdPrintAllPages;

            this.PrintOut(
                ref oTrue, ref oFalse, ref range, ref missing, ref missing, ref missing,
                ref items, ref copies, ref pages, ref pageType, ref oFalse, ref oTrue,
                ref missing, ref oFalse, ref missing, ref missing, ref missing, ref
        missing);

            UnhideText(bookmark1.Range);
```

The following code example can be used in an VSTO Add-in. This example uses the active document. To use the example, run it from the `ThisAddIn` class in your project.

C#

```csharp
            HideText(bookmark1.Range);

            this.Application.ActiveDocument.PrintOut(true, false,
        Word.WdPrintOutRange.wdPrintAllDocument,
                Item:Word.WdPrintOutItem.wdPrintDocumentContent, Copies:"1", Pages:"",
                PageType:Word.WdPrintOutPages.wdPrintAllPages, PrintToFile:false,
        Collate:true,
                ManualDuplexPrint:false);

            UnhideText(bookmark1.Range);
```

Compiling the Code

This code example assumes that the document contains a Microsoft.Office.Tools.Word.Bookmark control (in a document-level customization) or Microsoft.Office.Interop.Word.Bookmark control (in an VSTO Add-in) that is named `bookmark1`.

See Also

How to: Programmatically Print Documents
How to: Programmatically Define and Select Ranges in Documents
How to: Programmatically Reset Ranges in Word Documents
How to: Programmatically Update Bookmark Text
Optional Parameters in Office Solutions

© 2017 Microsoft