**Finding Text Not Using a Particular Font**

There are a couple of ways that you can go about finding what you need and, potentially, fixing the problem. First, you could use Find and Replace in this manner:

1. Press **Ctrl+H**. Word displays the Find tab of the Find and Replace dialog box. (If you are using Word 2010, you need display the Home tab of the ribbon, click the down-arrow at the right side of the Find tool, and then choose Advanced Find.)
2. Click the More button, if it is available.
3. Make sure the Find What box is empty.
4. Click Format and then choose Font. Word displays the Find Font dialog box. (See Figure 1.)



**Figure 1.** The Find Font dialog box.

1. Use the controls in the dialog box to specify that you want to find the Times Roman font you are using. (Just choose the font. You don't have to specify any other settings unless you want to.)
2. Click on OK. The font specifications you are looking for appear just below the Find What box.
3. Click Reading Highlight and choose Highlight All. Word highlights in the document all instances of words that match the font you chose.
4. Press **Esc**. **Or**

8. Clear the Find & Replace boxes.  
 9. Place the cursor in the Find box & click Format/Highlight twice. It will now show "Not highlight" in the Find box.  
  
Then, instead of scrolling through the document, you can now press "Find Next" & whatever isn't highlighted (the "anomolies") will be found.

At this point, all instances of Times Roman in your document are highlighted. You can easily scroll through the document to see what isn't highlighted—these are the "anomalies" you are seeking.

You could also use Find and Replace to apply some stand-out formatting to text that isn't in Times Roman font. Just figure out some formatting that you are not using in the document, such as double underlines or red text. Select the whole document (**Ctrl+A**) and apply that formatting to the whole document. Then use Find and Replace to find all instances to Times Roman (see the steps above) and replace it with Times Roman without the stand-out formatting you previously applied. What you end up with is only the text that is not Times Roman formatted with the stand-out formatting.

If you prefer, you can use a macro to modify the font used in the document. The following is a short little macro that looks at the font used for each word in your document:

Sub FixFormatAnomalies()

Dim aWord

For Each aWord In ActiveDocument.Words

If aWord.Font.Name <> "Times Roman" Then

aWord.Font.Name = "Times Roman"

End If

Next aWord

End Sub

If the font used for a word doesn't match what you want, then the routine changes the font. The only thing you'll need to do to make the macro work in your case is to change the desired font name on two lines. (You'll want to use the font name exactly as it appears in the list of fonts usable by Word. Pay particular attention to capitalization.)

Another approach is to simply check each character in the document and highlight the character if it doesn't use your desired font. The following macro highlights the offending characters in yellow:

Sub HighlightOtherFonts()

Dim iCounter As Integer

Dim sFontName As String

Dim sPrompt As String

Dim sTitle As String

Dim sDefault As String

Dim c As Range

' Gets the name of the font as typed by the user

sPrompt = "Type the name of the font that is OK to "

sPrompt = sPrompt & "have in the document."

sTitle = "Acceptable Font Name"

sDefault = ActiveDocument.Styles(wdStyleNormal).Font.Name

sFontName = InputBox(sPrompt, sTitle, sDefault)

' Verifies that the name of the font is valid

For Each sFont In Application.FontNames

If UCase(sFontName) = UCase(sFont) Then

' Changes the user-typed name of the font to

' the version recognized by the application

' Example: 'times new roman' (user-typed) is

' changed to 'Times New Roman' (application version)

sFontName = sFont

Exit For

Else

' Terminates the loop if the name of the font is invalid

iCounter = iCounter + 1

If iCounter = FontNames.Count Then

sPrompt = "The font name as typed does not match "

sPrompt = sPrompt & "any fonts available to the "

sPrompt = sPrompt & "application."

sTitle = "Font Name Not Found"

MsgBox sPrompt, vbOKOnly, sTitle

Exit Sub

End If

End If

Next sFont

' Checks each character in the document, highlighting

' if the character's font doesn't match the OK font

For Each c In ActiveDocument.Characters

If c.Font.Name <> sFontName Then

' Highlight the selected range of text in yellow

c.FormattedText.HighlightColorIndex = wdYellow

End If

Next c

End Sub

One of the nice features of this macro is that it prompts you for the font that you find acceptable. It then checks to make sure that what you enter matches one of the fonts available in the system. Because the macro checks each character in the document individually, you may need to be patient while it is running. The longer the document, the longer the macro takes to complete its work.

If you simply want to find the next occurrence of a font change, then the following very short macro is quite handy:

Sub FindDifferentFont()

Selection.SelectCurrentFont

Selection.Collapse wdCollapseEnd

End sub

The SelectCurrentFont method extends the current selection until there is a change in either the font (typeface) or the font size. So if you start at the beginning of the document and run the macro (perhaps you could assign it to a shortcut key), then the insertion point is moved to where the current font ends and a new font begins.