

Re: Spelling Error Fequency

Source:

<http://www.tech-archive.net/Archive/Word/microsoft.public.word.vba.general/2005-04/msg00142.html>

- *From:* "Jezebel" <warcimes@xxxxxxxxxxxxxxxx>
 - *Date:* Sat, 2 Apr 2005 15:25:06 +1000
-

Aren't you having fun!

Class modules are very easy. In this case, the entire code for the class module would be

```
Option Explicit
Public Name as string
Public Count as long
```

But if you really want to get your teeth into some interesting coding, the 'sort' section of your code has a lot of possibilities. What you currently have is a rudimentary bubble sort. Do a Google on sorting and searching algorithms for some of the other options. QuickSort is fun to code (although you'd have to be a seriously appalling speller for it to make any appreciable difference in this case...)

And a collection is yet another option. If you had a collection of class objects, you could, on completion, add them to a second collection using a key constructed from the Count property. Then iterate the collection to retrieve the words in order of frequency: thus a total $2n$ operations to sort and output, as opposed to, I think, $n + n \log(n)$ using your current method.

"Greg Maxey" <gmaxey@xxxxxxxxxxxxxxxx> wrote in message news:ecxSVHONFHA.1268@xxxxxxxxxxxxxxxx

> Jezebel,

>

> I have been off adapting the Word Frequency macro to achieve my objective
> and it works great. Thanks for your suggestions. I have never ventured
> in the Class module arena and may not be able to follow you directions. I
> will give it a shot though.

Re: Spelling Error Fequency

```
>
> In the meantime. Here is the code I have adapted:
>
> Sub spErrorFrequency()
>
> Dim SingleSpError As String 'Raw spelling error pulled from
> doc
> Const maxSpErrors = 9000 'Maximum unique spelling errors
> allowed
> Dim arrSpErrors(maxSpErrors) As String 'Array to hold unique misspelled
> words
> Dim Freq(maxSpErrors) As Integer 'Frequency counter for unique
> misspelled Words
> Dim spError As Range 'The spelling error object
> Dim spErrorNum As Integer 'Number of unique misspelled words
> Dim oSpErrorCnt As Long 'Numbe of total misspelled words
> Dim bolSortByFreq As Boolean 'Flag for sorting order
> Dim Found As Boolean 'Temporary flag
> Dim j As Integer 'Temporary variable
> Dim k As Integer 'Temporary variable
> Dim l As Integer 'Temporary variable
> Dim tempCount As Integer 'Temporary variable
> Dim tempString As String
> Dim oRng As Range
>
> 'Set sort order
> bolSortByFreq = True
> If MsgBox("The default sort order is error frequency." _
> & vbCr & "Do you want to sort errors" _
> & " alphabetically instead?", vbYesNo) = vbYes Then
> bolSortByFreq = False
> End If
>
> Selection.HomeKey Unit:=wdStory
> System.Cursor = wdCursorWait
> spErrorNum = 0
>
> 'Count total errors
> oSpErrorCnt = ActiveDocument.Range.SpellingErrors.Count
>
>
> For Each spError In ActiveDocument.Range.SpellingErrors
> SingleSpError = spError
> 'If Len(SingleSpError) > 0 Then
> Found = False
> For j = 1 To spErrorNum
> If arrSpErrors(j) = SingleSpError Then
> Freq(j) = Freq(j) + 1
> Found = True
> Exit For
> End If
```

Re: Spelling Error Fequency

```
> Next j
> If Not Found Then
> spErrorNum = spErrorNum + 1
> arrSpErrors(spErrorNum) = SingleSpError
> Freq(spErrorNum) = 1
> End If
> If spErrorNum > maxSpErrors - 1 Then
> j = MsgBox("The maximum array size has been exceeded. Increase
> maxSpErrors.", vbOKOnly)
> Exit For
> End If
> 'End If
> Next spError
>
> 'Sort
> For j = 1 To spErrorNum - 1
> k = j
> For l = j + 1 To spErrorNum
> If (Not bolSortByFreq And arrSpErrors(l) < arrSpErrors(k)) _
> Or (bolSortByFreq And Freq(l) > Freq(k)) Then k = l
> Next l
> If k <> j Then
> tempString = arrSpErrors(j)
> arrSpErrors(j) = arrSpErrors(k)
> arrSpErrors(k) = tempString
> tempCount = Freq(j)
> Freq(j) = Freq(k)
> Freq(k) = tempCount
> End If
> Next j
>
> 'Now write out the results
> Set oRng = ActiveDocument.Range
> oRng.Move
> oRng.InsertBreak wdSectionBreakNextPage
> oRng.Select
> Selection.ParagraphFormat.TabStops.ClearAll
> With Selection
> For j = 1 To spErrorNum
> .TypeText Text:=arrSpErrors(j) & vbTab & Trim(Str(Freq(j))) & vbCrLf
> Next j
> End With
> Selection.Sections(1).Range.Select
> Selection.ConvertToTable
> Selection.Collapse wdCollapseStart
> ActiveDocument.Tables(1).Rows.Add BeforeRow:=Selection.Rows(1)
> ActiveDocument.Tables(1).Cell(1, 1).Range.InsertBefore "Spelling Error"
> ActiveDocument.Tables(1).Cell(1, 2).Range.InsertBefore "Number of
> Occurrences"
> ActiveDocument.Tables(1).Columns(2).Select
> Selection.ParagraphFormat.Alignment = wdAlignParagraphRight
```

Re: Spelling Error Fequency

```
> Selection.Collapse wdCollapseStart
> ActiveDocument.Tables(1).Rows(1).Shading.BackgroundPatternColor =
> wdColorGray20
> ActiveDocument.Tables(1).Columns(1).PreferredWidth = InchesToPoints(4.75)
> ActiveDocument.Tables(1).Columns(2).PreferredWidth = InchesToPoints(1.9)
>
> ActiveDocument.Tables(1).Rows.Add
> ActiveDocument.Tables(1).Cell(ActiveDocument.Tables(1).Rows.Count,
> 1).Range.InsertBefore "Summary"
> ActiveDocument.Tables(1).Cell(ActiveDocument.Tables(1).Rows.Count,
> 2).Range.InsertBefore "Total"
>
ActiveDocument.Tables(1).Rows(ActiveDocument.Tables(1).Rows.Count).Shading.BackgroundPatternColor
> = wdColorGray20
>
> ActiveDocument.Tables(1).Rows.Add
> ActiveDocument.Tables(1).Cell(ActiveDocument.Tables(1).Rows.Count,
> 1).Range.InsertBefore "Number of Unique Spelling Errors"
> ActiveDocument.Tables(1).Cell(ActiveDocument.Tables(1).Rows.Count,
> 2).Range.InsertBefore Trim(Str(spErrorNum))
>
ActiveDocument.Tables(1).Rows(ActiveDocument.Tables(1).Rows.Count).Shading.BackgroundPatternColor
> = wdColorAutomatic
>
> ActiveDocument.Tables(1).Rows.Add
> ActiveDocument.Tables(1).Cell(ActiveDocument.Tables(1).Rows.Count,
> 1).Range.InsertBefore "Number of Spelling Errors"
> ActiveDocument.Tables(1).Cell(ActiveDocument.Tables(1).Rows.Count,
> 2).Range.InsertBefore (oSpErrorCnt)
>
> Selection.HomeKey wdStory
>
> End Sub
>
> --
> Greg Maxey/Word MVP
> See:
> http://gregmaxey.mvps.org/word\_tips.htm
> For some helpful tips using Word.
>
> Jezebel wrote:
>> Instead of using an array, try using a collection. Create an class
>> module with properties 'Name' and 'Count'. and build a collection of
>> these for the errors you find.
>>
>> Dim oError as clsError
>> Dim colErrors as collection
>>
>> set colErrors = new collection
>>
>> For each oSPError in oSpErrors
```

Re: Spelling Error Fequency

```
>>
>> 'Already in the collection?
>> on error resume next
>> set oError = colErrors(oSPErrors.Text)
>> on error goto 0
>>
>> 'Not in the collection – new error
>> If oError is nothing then
>> set oError = New clsError
>> oError.Name = oSPErrors.Text
>> colErrors.Add oError, oError.Name
>> end if
>>
>> 'Increment the count
>> oError.Count = oError.Count + 1
>> set oError = nothing
>>
>> end if
>>
>> 'List the results
>> For each oError in colErrors
>> Debug.Print oError.Name, oError.Count
>> Next
>>
>>
>>
>>
>>
>> "Greg Maxey" <gmaxey@xxxxxxxxxxxxxxxxxxxx> wrote in message
>> news:uUPa8ayNFHA.2392@xxxxxxxxxxxxxxxxxxxxxxxx
>>> For those who frequent these NGs regularly you know that I can't
>>> spell. Actually I can spell, but I am careless.
>>>
>>> I have been monkeying around with some code that will search a
>>> document then list alphabetically all misspelled words. I am using
>>> my budding but limited knowledge of arrays to sort the list and
>>> purge out duplicates so words that are misspelled more than once are
>>> only listed once. It works. I was thinking it would be helpful to
>>> determine and list how many
>>> times each word was misspelled. I think it would be possible, but I
>>> am not sure how I would proceed.
>>>
>>> Currently I am comparing each error to the contents of the existing
>>> array and if a match occurs I am skipping that error. This way
>>> identical misspellings the array only contains one instance of the
>>> misspelling. I don't see how I could work a counter into this
>>> process. Maybe if all words were put into the array initially, sorted
>>> in then
>>> compare the first to the second and if a match occurs delete the
>>> first and compare the second to the third until a match doesn't
```

Re: Spelling Error Fequency

```
>>> occur then jump up one error and proceed on. This sounds
>>> achievable, but I don't know how to do it.
>>>
>>> Any thoughts?
>>>
>>> Here is the current code:
>>> Sub printSpellingErrors()
>>> Dim arrSpArray() As String
>>> Dim oSpErrors As ProofreadingErrors
>>> Dim oSpError As Object
>>> Dim i As Integer
>>> Dim oRng As Range
>>>
>>> Set oSpErrors = ActiveDocument.Range.SpellingErrors
>>> If oSpErrors.Count = 0 Then
>>> MsgBox "The document contains no spelling errors."
>>> End
>>> End If
>>>
>>> ReDim arrSpArray(0)
>>>
>>> 'Add each error to the array if not a duplicate
>>> For Each oSpError In oSpErrors
>>> 'Compare to each exist element in the array
>>> For i = LBound(arrSpArray) To UBound(arrSpArray)
>>> If oSpError = arrSpArray(i) Then
>>> 'Skip if already in array
>>> GoTo SkipToNext
>>> End If
>>> Next i
>>> 'Otherwise add to array
>>> arrSpArray(UBound(arrSpArray)) = oSpError
>>> 'Preserve and resize array for next element
>>> ReDim Preserve arrSpArray(UBound(arrSpArray) + 1)
>>> SkipToNext:
>>> Next oSpError
>>>
>>> 'Remove last empty element
>>> If UBound(arrSpArray) > 0 Then ReDim Preserve
>>> arrSpArray(UBound(arrSpArray) - 1)
>>>
>>> 'Pass array to sort
>>> BubbleSort arrSpArray
>>>
>>> 'Prepare for display
>>> Set oRng = ActiveDocument.Range
>>> oRng.Move
>>> 'oRng.Text = vbCr
>>> oRng.InsertBreak wdSectionBreakNextPage
>>> oRng.Move
>>> oRng.Text = "List of Misspelled Words" & vbCr
```

Re: Spelling Error Fequency

```
>>> oRng.Move
>>>
>>> i = 0
>>> For i = LBound(arrSpArray) To UBound(arrSpArray)
>>> oRng.Text = arrSpArray(i) & vbCr
>>> oRng.Collapse Direction:=wdCollapseEnd
>>> Next i
>>> 'Clip empty paragraph
>>> oRng.Characters.First.Previous.Delete
>>> End Sub
>>> Sub BubbleSort(TempArray As Variant)
>>>
>>> Dim Temp As Variant
>>> Dim i As Integer
>>> Dim bolExchange As Integer
>>>
>>> Do
>>> bolExchange = False
>>> 'Loop through each element in the array.
>>> For i = LBound(TempArray) To UBound(TempArray) - 1
>>> 'If element > next element then exchange the two elements.
>>> If LCase(TempArray(i)) > LCase(TempArray(i + 1)) Then
>>> bolExchange = True
>>> Temp = TempArray(i)
>>> TempArray(i) = TempArray(i + 1)
>>> TempArray(i + 1) = Temp
>>> End If
>>> Next i
>>> Loop While bolExchange
>>>
>>> End Sub
>>>
>>>
>>>
>>>
>>> ---
>>> Greg Maxey/Word MVP
>>> See:
>>> http://gregmaxey.mvps.org/word\_tips.htm
>>> For some helpful tips using Word.
>
>
```

-
- *Follow-Ups:*
 - ◆ *[Re: Spelling Error Fequency](#)*

Re: Spelling Error Fequency

◇ *From:* Howard Kaikow

• *References:*

◆ *Spelling Error Fequency*

◇ *From:* Greg Maxey

◆ *Re: Spelling Error Fequency*

◇ *From:* Jezebel

◆ *Re: Spelling Error Fequency*

◇ *From:* Greg Maxey

- Prev by Date: *Re: Spelling Error Fequency*
- Next by Date: *Re: Acrobat 7 blocks Global Template Save*
- Previous by thread: *Re: Spelling Error Fequency*
- Next by thread: *Re: Spelling Error Fequency*
- Index(es):
 - ◆ *Date*
 - ◆ *Thread*