

Re: Creating AutoText

Source:

<http://www.tech-archive.net/Archive/Word/microsoft.public.word.vba.general/2005-06/msg00216.html>

- *From:* Jay Freedman <jay.freedman@xxxxxxxxxxxx>
 - *Date:* Mon, 06 Jun 2005 20:29:10 -0400
-

First on the matter of saving Personal.dot: I'm afraid you got the wrong end of the handle there. The `..Saved` property is just a "flag" (in the sense of the little red flag on a mailbox) that tells Word whether there are any changes that need to be saved. When you set it to `True`, you're telling Word "don't bother to save this, there aren't any changes in it". Instead, what you should include in your code is

```
objTemplate.Save
```

which does save the changes and, as a side-effect, also sets the `..Saved` property to `True`.

Regarding the category in which the autotexts are created: What I found in my testing was that the `ReLine` style needs to exist in the `Personal.dot` template before the macro runs, but it doesn't have to be in `Normal.dot`. That shouldn't be a problem if you're supplying the empty `Personal.dot` as part of your setup.

Finally, about where variables should be declared: This is a general programming idea called "scope", or the lifetime of the variable. This is explained at more length in the VBA help topics on "Understanding the Lifetime of Variables" and "Understanding Scope and Visibility", but here's the short version:

– If you declare a variable between a `Sub` statement and its matching `End Sub`, that variable exists only while that subroutine is executing (it's a "local" variable). As soon as execution exits from the subroutine, the memory corresponding to the variable is discarded and the variable name has no meaning. You can declare local variables in different subroutines, all using the same variable name, but they will refer to different values in memory and won't have any effect on each other.

– If you declare a variable outside any subroutine and use the `Private` keyword, the scope of the variable is the current module. Different subroutines in the module share the same piece of memory, and one subroutine can set a value that will be available for use in another subroutine. Sometimes that's what you want, but often it's just a trap

Re: Creating AutoText

that will lead you into hard-to-diagnose logic errors.

– If you declare a variable outside any subroutine and use the Public keyword, the scope of the variable is the current project (usually the template). This has the same danger of logic errors as Private declaration, but worse because there are more places you can make a wrong assignment.

Note that by declaring a single Public variable instead of separate local variables, you are **not** "saving effort" or "saving memory" -- you're making a mistake.

In order to know whether you really need the same value of the same variable in two or more subroutines, I'd have to look at all of the code and know how you're using the variable. In this case, though, I doubt that `bmrange` had any usable value -- it would have had to correspond to the location of the inserted text in a document that didn't exist until this subroutine created it, and that no longer exists after this subroutine exits.

--
Regards,
Jay Freedman
Microsoft Word MVP FAQ: <http://word.mvps.org>

On Sun, 5 Jun 2005 22:39:00 -0400, "Anne P."
<apontillo@xxxxxxxxxxxxxxxx> wrote:

>I just found out how to save the changes to `Personal.dot` after an AutoText
>entry is created there. Remembering that I have a variable name `objTemplate`
>and `Personal.dot` is assigned to that variable, the line of code is:
>
>`objTemplate.Saved = True`
>
>Anne P.
>
>"Anne P." <apontillo@xxxxxxxxxxxxxxxx> wrote in message
>news:OKzbgjaFHA.2884@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
>> Jay,
>>
>> Thank you so much for your response. This worked very well except for two
>> things:
>>
>> 1. The AutoText entry was saved in `Personal.dot`, but under the category
>> Normal, not under the category `ReLine`. However, I fixed that little
>> problem for now by creating a style in `Normal.dot` named `ReLine`. That will
>> do for now, however, if I give the users another type of macro to create
>> other AutoText entries, they would have to create a style in `Normal.dot`
>> before it would work for them. Although that might not be a problem. The
>> reason that I want it under this category is so that I can fill a list box
>> or combo box with all AutoText entries that are under that category.

Re: Creating AutoText

Re: Creating AutoText

>>
>> 2. After this macro has been run, when the user exits Word, they get
>> prompted to save changes to Personal.dot. While I know (and you probably
>> do also) that this should not "freak out" users, the reality is that it
>> does. In a large firm, the Help Desk would be inundated with calls about
>> why they are being asked to save changes to Personal.dot. Is there anyway
>> that I can save the changes to this template programmatically?
>>
>> By the way, I do have Option Explicit at the top of the module for the
>> Userform. Also, all of the variables used in the cmdOK portion that I
>> sent you were declared (including bmrangle). When I saw your modified
>> code, at first I thought that even though I had declared bmrangle, I
>> thought that it didn't work because of how I declared it. As a matter of
>> fact, all of the variables that you have declared underneath cmdOK, I had
>> declared also. I had them declared at the top of the module right below
>> Option Explicit and declared as follows: Public bmrangle as Range. I did
>> it that way because several of the variables I need to use in other parts
>> of the module (such as txtReLine_Change). I thought that if I didn't do
>> the declarations at the top as public, that each portion of the userform
>> code I would have to re-declare the variable to use it again.
>>
>> Sometimes, I am a little unsure of the proper way and/or place to declare
>> my variables.
>>
>> Once again, thank you so much for your help with this. You are the best.
>>
>> Anne P.
>>
>>
>> "Jay Freedman" <jay.freedman@xxxxxxxxxxxx> wrote in message
>> news:ouh4a1579m16h32jd37phuv5143klbdrbl@xxxxxxxxxxx
>>> Hi Anne,
>>>
>>> I finally had some time to look at this. It took a fair amount of
>>> adjustment to get the details right, but the overall plan is correct.
>>> Compare this version of cmdOK_Click() to the one you posted, and feel
>>> free to ask any questions about the differences.
>>>
>>> One thing that makes a big difference is to include the statement
>>> Option Explicit at the top of each module. This forces you to declare
>>> each variable you use, and it pops up an error if you try to use one
>>> that hasn't been defined. In your code, bmrangle is never defined
>>> before you try to assign a style to it, which is one reason the
>>> category never appears. The article at
>>> <http://www.word.mvps.org/FAQs/MacrosVBA/MaintainableCode.htm> explains
>>> the use of the option.
>>>
>>> Private Sub cmdOK_Click()
>>> Dim strATCategory As String
>>> Dim strPath As String
>>> Dim docNew As Document

Re: Creating AutoText

```
>>> Dim objTemplate As Template
>>> Dim oNewStyle As Style
>>> Dim ATrange As Range
>>>
>>> ' validate both text boxes
>>> If txtReLine.Text = "" Then
>>> MsgBox "Please supply text"
>>> txtReLine.SetFocus
>>> Exit Sub
>>> End If
>>>
>>> If txtATName.Text = "" Then
>>> MsgBox "Please supply name"
>>> txtATName.SetFocus
>>> Exit Sub
>>> End If
>>>
>>> strATCategory = "ReLine"
>>>
>>> ' Create a new document.
>>> Set docNew = Documents.Add(Visible:=False)
>>> ' assign range in docNew
>>> Set ATrange = docNew.Range
>>>
>>> ' Add text to document.
>>> ATrange.Text = txtReLine.Text
>>>
>>> 'Create autotext in personal.dot
>>>
>>> ' assign the global template to an object
>>> strPath = Options.DefaultFilePath(wdStartupPath) & _
>>> "\\Personal.dot"
>>> Set objTemplate = Templates(strPath)
>>>
>>> On Error Resume Next
>>> ' if the style already exists, this succeeds
>>> Set oNewStyle = docNew.Styles(strATCategory)
>>> If Err.Number <> 0 Then
>>> ' the style didn't exist, so create it
>>> Err.Clear
>>> Set oNewStyle = _
>>> docNew.Styles.Add(Name:=strATCategory)
>>> End If
>>>
>>> ' change style of selection's paragraph to new category
>>> ATrange.Style = oNewStyle
>>>
>>> ' exclude the paragraph mark
>>> ATrange.MoveEnd unit:=wdCharacter, Count:=-1
>>>
>>> ' add AT entry to global template
```

Re: Creating AutoText

```
>>> objTemplate.AutoTextEntries.Add Name:=txtATName.Text, _
>>> Range:=ATrange
>>>
>>> 'Close document without saving changes.
>>> docNew.Close wdDoNotSaveChanges
>>>
>>> Unload Me
>>> End Sub
>>>
>>> --
>>> Regards,
>>> Jay Freedman
>>> Microsoft Word MVP FAQ: http://word.mvps.org
>>>
>>> On Sat, 4 Jun 2005 15:44:09 -0400, "Anne P."
>>> <apontillo@xxxxxxxxxxxxxxxx> wrote:
>>>
>>>>Hi,
>>>>
>>>>I have a userform with two textboxes: txtReLine and txtATName. After
>>>>the
>>>>user fills out these two textboxes, I am trying to create an AutoText
>>>>entry
>>>>in a template named Personal.dot in the Word startup directory. I am
>>>>creating a hidden instance of Word and opening a new document to insert
>>>>the
>>>>text from txtReLine, so that I can then either select it or assign it to
>>>>a
>>>>range and then create an AutoText entry.
>>>>
>>>>If I run the macro to open the userform, I get an error message "Object
>>>>variable or with block variable not set" and nothing happens. If I step
>>>>through the code in VB Editor, an AutoText entry is created in the
>>>>Personal.dot template, but is created under the category Normal and the
>>>>value is empty. On the line "selection.typetext" the text is typed into
>>>>the
>>>>active document, not the new document that is created. I believe from
>>>>this
>>>>point, I need to somehow assign the text that is entered into the
>>>>document
>>>>to a range object, but I am not sure how to accomplish this.
>>>>
>>>>Below is the relevant code from the cmdOK button. I appreciate any help
>>>>with this.
>>>>
>>>>>strATCategory = "ReLine"
>>>>>' Create new hidden instance of Word.
>>>>>Set wdApp = New Word.Application
>>>>>' Create a new document.
>>>>>Set docNew = wdApp.Documents.Add
>>>>>' Add text to document.
```

Re: Creating AutoText

```
>>>>strReLine = txtReLine.Text
>>>>Selection.TypeText strReLine
>>>>'Create autotext in personal.dot
>>>>
>>>>' assign the global template to an object
>>>>strPath = Options.DefaultFilePath(wdStartupPath) & _
>>>> "\\Personal.dot"
>>>>Set objTemplate = Templates(strPath)
>>>>
>>>>On Error Resume Next
>>>>bRemoveStyle = False
>>>>' if the style already exists, this succeeds
>>>>Set oNewStyle = docNew.Styles(strATCategory)
>>>>If Err.Number < 0 Then
>>>> ' the style didn't exist, so create it
>>>> Err.Clear
>>>> Set oNewStyle = _
>>>> docNew.Styles.Add(Name:=strATCategory)
>>>> bRemoveStyle = True
>>>>End If
>>>>
>>>>' change style of selection's paragraph to new category
>>>>bmrange.Style = docNew.Styles(strATCategory)
>>>>
>>>>' add AT entry to global template
>>>>objTemplate.AutoTextEntries.Add Name:=strATName, _
>>>> Range:=Selection.Range
>>>>
>>>>' reverse style assignment
>>>>ActiveDocument.Undo
>>>>
>>>>' remove new style if it was created by this code
>>>>If bRemoveStyle Then
>>>> ActiveDocument.Styles(strATCategory).Delete
>>>>End If
>>>>
>>>>'Close document without saving changes.
>>>>With docNew
>>>>
>>>> .Close wdDoNotSaveChanges
>>>>End With
>>>>wdApp.Quit
>>>>Set wdApp = Nothing
>>>> Unload Me
>>>>
>>>>Thanks,
>>>>Anne P.
>>>>
>>>>
>>>>
>>>>
```

>

.

• **References:**

- ◆ **Creating AutoText**
◇ From: Anne P.
- ◆ **Re: Creating AutoText**
◇ From: Jay Freedman
- ◆ **Re: Creating AutoText**
◇ From: Anne P.
- ◆ **Re: Creating AutoText**
◇ From: Anne P.

- Prev by Date: **Re: Add-Ins**
- Next by Date: **Re: Multiple selection**
- Previous by thread: **Re: Creating AutoText**
- Next by thread: **Can we enable/disable undo manually/programmatically in Word?**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**