

Re: Enable/Disable Toolbar button

Source:

<http://www.tech-archive.net/Archive/Word/microsoft.public.word.vba.general/2007-10/msg00324.html>

- *From:* NZ VBA Developer <gordon(dot)bentleymix(at)gmail(dot)com>
 - *Date:* Mon, 15 Oct 2007 20:22:00 -0700
-

Shauna,

I'm just re-reading your post to make sure I understand. Are you saying that the WindowSelectionChange event isn't triggered when the cursor is repositioned using the arrow keys? Or clicks through a cross-reference, such as a TOC entry? Cuz it seems to be working OK for me - well at least the arrow keys are. I haven't tried using the Document Map yet to see what happens, but since my 'target audience' isn't especially Word-savvy, I may just leave it alone; I'm not sure anyone at the client site even knows what the Document Map is.

The only time I noticed the buttons not behaving as expected was when I first applied a numbered style to a selection - altho they may have been behaving oddly when I was typing as well. And even then I'm not sure; I'll have to experiment a bit more when I have some time later on.

In any case, I do appreciate the information, and if it turns out that I really do need it, I'm sure I'll appreciate it even more!

--

Cheers!
Gordon

"Shauna Kelly" wrote:

Hi Gordon

The WindowSelectionChange event won't fire when the user negotiates with the cursor instead of the mouse, and it doesn't fire if the user clicks the Document Map to move around. To get around that, you can use the built-in functionality that Word has in its various buttons.

Find a built-in button that has the functionality you want. That is, find a built-in button that turns on and off when you want yours to turn on and off. (For example, if you want a button that is only enabled when the insertion point is in a table, then choose one of the buttons on the Table

Re: Enable/Disable Toolbar button

menu that operates as you want.)

Then **copy** the button you found to your own toolbar. You can then change the icon and the displayed text to suit yourself. If you need more than one button to be enabled and unenabled at the same time, but do different things, then copy the **same** button (and change the icon and displayed text to suit yourself).

Use the Immediate Pane to determine the ID of the button (let's say it's 296). Using the Immediate Pane, give your button(s) a .Tag (something meaningful to you that identifies it as belonging to your project, let's say "Gordon"). You may have several buttons with the same .ID and .Tag combination. Give each button a separate .Parameter, which identifies each one uniquely (let's say "Gordon_RestartNumbering" and "Gordon_ContinueNumbering"). In my example here I used the useless names of "Param1" and "Param 2".

So, in my example document I now have a custom toolbar named "Custom 1" with two buttons. They both have the same .ID of 296. They both have a .Tag of "Gordon". One has .Parameter "Param1" and the other is "Param2". And if I play with Word, they turn on and off as I need them to.

Now use the following code:

1. In an ordinary Module:

Option Explicit

Public gMyButton As CMyButtons

2. In the ThisDocument class:

Option Explicit

```
Private Sub Document_New()  
Set gMyButton = New CMyButtons  
End Sub
```

```
Private Sub Document_Open()  
Set gMyButton = New CMyButtons  
End Sub
```

3. In a Class module named CMyButtons

Option Explicit

'A variable to hold a reference to any button with your combination

Re: Enable/Disable Toolbar button

```
'of ID and Tag
Private WithEvents mMyButton As Office.CommandBarButton

Private Sub Class_Initialize()
'Set mMyButton to refer to a button with your combination of ID
'and Tag. Even though you may have many buttons with the same ID
'and Tag combination, you only need to 'hook' one of them here.
```

```
'This way is fast
Set mMyButton = CommandBars("Custom 1").Controls(1)
```

```
'OR....
```

```
'This way is safe
Set mMyButton = CommandBars.FindControl(Type:=msoControlButton, ID:=296,
Tag:="Gordon")
```

```
End Sub
```

```
Private Sub mMyButton_Click(ByVal Ctrl As Office.CommandBarButton,
CancelDefault As Boolean)
```

```
'This fires when the user clicks one of your buttons with your combination
'of ID and Tag
```

```
If Ctrl.Tag = "Gordon" And Ctrl.ID = 296 Then
'Strictly speaking you don't need this whole If ... .Tag *and* .ID
thing here
'but I find it enormously helpful documentation!
```

```
Select Case Ctrl.Parameter
Case "Param1"
MsgBox "You clicked my Param1 button"
'Run the appropriate code here
```

```
Case "Param2"
MsgBox "You clicked my Param2 button"
'Run the appropriate code here
End Select
```

```
'prevent Word from doing the default action
'for the button
CancelDefault = True
```

```
End If
```

```
End Sub
```

Create a new document from your template and watch the magic happen!

Re: Enable/Disable Toolbar button

For the record, any .OnAction value of the button(s) is ignored when you hook buttons like this.

Now, go buy a copy of Professional Excel Development by Bullen, Bovey and Green where you'll read a much better description of this than I have given. Ignore the early chapters about Excel, if you want to. The book is worth it for the latter chapters and it applies equally well to Word as to Excel.

Hope this helps.

Shauna Kelly. Microsoft MVP.
<http://www.shaunakelly.com/word>

(England v the Springboks. Who would have thought?)

"NZ VBA Developer" <gordon(dot)bentleymix(at)gmail(dot)com> wrote in message news:E11A9728-C8F3-4729-BC7B-BF2DF8766A6C@xxxxxxxxxxxxxxxxxxxx

Hmm... Not quite Jay...

Naming the macros as suggested does make it so they run from the 'native' toolbar button or the right-click menu, but it still doesn't get around the original problem. The native toolbar buttons and the right-click menu items are still disabled when the document is protected.

What I need is a combination of the behaviour provided by the buttons for the native functionality and the behaviour of buttons with macros assigned to them: enabled even though the document is protected but only when the context is correct. In fact, I'd be happy to use the native functionality rather than my macro since all I'm trying to do is to give the users the ability to restart/continue numbering, and the native functionality does this just fine – just not when the doc is protected. All my macro does is change an argument for the ApplyListTemplate method as follows:

```
Sub RestartNumbering()  
Dim LT As ListTemplate  
Set LT = Selection.Style.ListTemplate  
Selection.Range.ListFormat.ApplyListTemplate ListTemplate:=LT,  
ContinuePreviousList:=False  
End Sub
```

Re: Enable/Disable Toolbar button

```
Sub ContinueNumbering()  
Dim LT As ListTemplate  
Set LT = Selection.Style.ListTemplate  
Selection.Range.ListFormat.ApplyListTemplate ListTemplate:=LT,  
ContinuePreviousList:=True  
End Sub
```

Is there a way to capture/monitor the 'selection change' event using VBA?
Seems to me that this is what Word is doing, and I've seen COM add-ins do
the
same...

—
Cheers!
The Kiwi Koder

"Jay Freedman" wrote:

If you name your macros RestartNumbering and ContinueNumbering, then the built-in buttons (with their automatic enable/disable behavior) will run the macros instead of the built-in commands. The macros will also intercept the commands on the context (right-click) menu, which appear only when the appropriate text is clicked.
<http://www.word.mvps.org/FAQs/MacrosVBA/InterceptSavePrint.htm>

On Sun, 14 Oct 2007 18:33:00 -0700, NZ VBA Developer <gordon(dot)bentleymix(at)gmail(dot)com> wrote:

Word does it, so how can I do it?

I want to enable/disable a toolbar button depending on cursor location/selection. For example, if I put the 'Restart Numbering' and 'Continue Numbering' buttons on the 'Formatting' toolbar, Word "knows" to enable these buttons _only_ if the current selection is numbered. Any ideas on how to replicate this functionality for the buttons on my own custom toolbar so the macro assigned to the button can be run only in the correct

Re: Enable/Disable Toolbar button

context?

(In fact, the 'Restart/Continue Numbering' functionality is the functionality that I want to replicate, but I can't just add the native Word buttons to my custom toolbar because this functionality doesn't work with protected documents. So if somebody has a clever idea for getting around this...)

—

Regards,

Jay Freedman

Microsoft Word MVP FAQ: <http://word.mvps.org>

Email cannot be acknowledged; please post all follow-ups to the newsgroup so all may benefit.