Excel Macros for Logging Sheet Access and for Sheet Delete

Ananth Chellappa

Summary of Deliverable(s)

- Macro_A
  - Runs automatically when another (or new) workbook is opened
  - Gets confirmation from user and "sets up" to start logging

- Macro_B
  - Runs (automatically) when a Sheet has been renamed

- Macro_C
  - An intelligent delete function

- Names are just for reference. You may choose appropriate names

Macro_A

- Runs when a (**any**) workbook (xlsx OR xlsm, i.e, workbook where this will add value need not be macro enabled) is opened and..

- prompts user with a dialog to confirm Yes/No to start logging .. (same dialog also has a "Time check" field which accepts duration with a default value filled in of 5 (min)) if Yes :
  - If non-existent, creates a "journal" hidden sheet .. where..
  - All other sheets are listed and the most recent time of access is listed
  - Sets up to start "logging" - that is, every time a new sheet is created, the create-time will be listed with the sheet-name in the "journal" sheet

- The user can make the "journal" sheet Unhidden if she desires and the logging function should continue unaffected

Macro_B

- Runs automatically when a sheet is renamed and updates the name in the "journal" sheet

Macro_C

- When run by user :
    - If the currently Active sheet was created less than "Time check" minutes ago, then will delete the Active sheet **without** asking for **confirmation** (needs to lookup the "journal" sheet for creation time. No entry means False - go to Else condition - below)
    - Else, it will Delete **with** user confirmation
- Updates the "journal" sheet after deleting
- If Logging is not enabled, then always ask for confirmation

Ideas

- The attached PERSONAL.XLSB shows (view Code for ThisWorkbook) a Workbook_Open Sub that runs whenever Excel is started. Something along these lines might be available for the case of "when any workbook (other than PERSONAL.XLSB) is activated". However, the Macro needs to exist in the PERSONAL.XLSB

- The workSheetMonitor class Module along with the CrossHairs module shows how highlighting upon active cell change in any (other) workbook is achieved

- You can test and see that they work (run startCrossHair and see the highlighting in a test.xlsx and then run StopCrossHair)

Ideas .. contd.

- Testing will show that the highlighting utility only works correctly for one workbook
  - Open a workbook, run startCrossHair and now select different cells
  - Open another workbook (or new one) and, now run startCrossHair again from this one and you see the same function operating to highlight the row/column of active cell
  - Go back to previous notebook and you see the function is no longer available
  - -- Just pointing this out because
- The "logging" utility you deliver **must work correctly regardless** how many workbooks are open. Each newly created sheet must be logged to the "journal" sheet in the correct workbook

Checklist

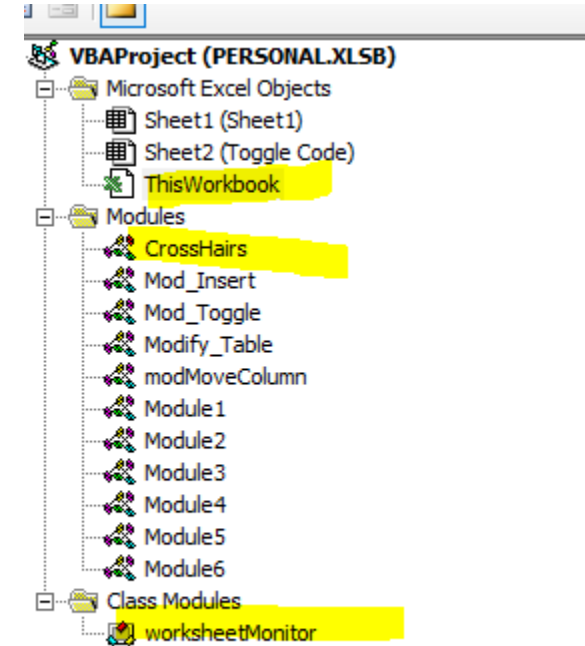| #	Action	Check
1	Open a new Workbook	Are you presented with a dialog?
2		Does it ask you if you want to start logging with a Yes and a No button?
3		Is there "Time check" field with an entry box?
4		On the right side of the box, is there the text "(min)"?
5		Is the default value in the entry box "5"?
6		Can user change the value?
7	Click Yes on the dialog	Has a hidden sheet been created with name "journal"?
8		If you unhide it, does it show that Sheet1 was created when you opened the workbook?
9	Rename Sheet1 to "Start"	Do you see that the name has been updated in the "journal" sheet?
10	Create a new Sheet and call it "Test"	Do you see that the "journal" sheet now has an entry for "Test"?
11	With sheet "Test" active, run Macro_C to delete, within 5 minutes of creation	Do you see that the "jounal" sheet no longer has an entry for 'Test"? was it deleted without asking for confirmation?
12	Make a *copy* of "Start" sheet	Do you see an entry for "Start (2)" in the "journal" sheet?
12	Wait 5 minutes	
13	Go to "Start" sheet and run Macro_C	Do you get a confirmation dialog for deleting?

Motivation

- Based on existing worksheets, we sometimes want to do some filtering and custom highlighting and then select and paste into a PPT or email

- So, we make a copy and make changes and, when done, we want an easy way to delete

- So, Macro_C will be tied to a keyboard shortcut

- But, we don't want to delete the wrong Sheet ☺

Attached PERSONAL.XLSB

- After saving, you can rename to REF.XLSB or any other name

- When opened, you have the option of not enabling macros in case you are concerned

- The code of interest is also pasted here if you want to avoid interacting with the PERSONAL.XLSB altogether

ThisWorkbook code

```
Option Explicit


Private Sub Workbook_Open()

    Application.OnKey "^+{]}", "PERSONAL.XLSB!startCrossHair"   '  start

    Application.OnKey "^+{[}", "PERSONAL.XLSB!stopCrossHair"    '  stop highlighting

    Application.OnKey "^{RIGHT}", "PERSONAL.XLSB!MoveColumnToRight"

    Application.OnKey "^{LEFT}", "PERSONAL.XLSB!MoveColumnToLeft"

    Application.OnKey "^%f", "PERSONAL.XLSB!clear_filter"


End Sub
```

CrossHairs Module Code

```vba
Option Explicit


Private Monitor As worksheetMonitor

Sub startCrossHair()
'

    Set Monitor = New worksheetMonitor


End Sub


Sub StopCrossHair()


    Set Monitor = Nothing


End Sub
```

worksheetMonitor Class Module code

```vba
Option Explicit


Private WithEvents Worksheet As Worksheet


Private Sub Class_Initialize()

    Set Worksheet = ActiveSheet

End Sub



Private Sub Class_Terminate()

    Cells.Interior.ColorIndex = 0

End Sub



Private Sub Worksheet_SelectionChange(ByVal Target As Range)

    If Target.Cells.CountLarge > 1 Then Exit Sub

    Application.ScreenUpdating = False

    ' Clear the color of all the cells

    Cells.Interior.ColorIndex = 0

    With Target

        ' Highlight the entire row and column that contain the active cell

        .EntireRow.Interior.ColorIndex = 4

        .EntireColumn.Interior.ColorIndex = 4

    End With

    Application.ScreenUpdating = True

End Sub
```

Existing Delete Function (May be Useful)

```
Sub DeleteThis()
        Application.DisplayAlerts = False
                ActiveSheet.Delete
        Application.DisplayAlerts = True
End Sub
' this one is dangerous because it doesn't ask for
confirmation - so it is only used by a button
```