

Class: MSc

Subject : Application of IT- Basics and Advance Excel

Chapter: Unit 4 Chapter 3

Chapter Name: Speeding up VBA

Screen Updating

- *As cool as it looks watching your VBA macro manipulate the screen, you can help your Macro run faster if you turn off (disable) ScreenUpdating.*
- *To disable ScreenUpdating, at the beginning of your code put this line:*
Application.ScreenUpdating = False
- *To re-enable ScreenUpdating, At the end of your code put this line:*
Application.ScreenUpdating = True
- *Disabling ScreenUpdating will make your VBA code run MUCH faster, but it will also make your work appear more professional. End-users typically don't want to see the behind the scenes actions of your procedures*

ScreenUpdating Example

```
Sub ScreenUpdating_Example()  
    Application.ScreenUpdating = False  
  
    'Do Something  
    Range("a1").Copy Range("b1")  
    Range("a2").Copy Range("b2")  
    Range("a3").Copy Range("b3")  
  
    Application.ScreenUpdating = True  
End Sub
```

Turn Automatic Calculations Off (or On)

- *Whenever you update a cell value, Excel goes through a process to recalculate the workbook. When working directly within Excel you want this to happen 99.9% of the time. However, this can really slow down your VBA code. It's a good practice to set your calculations to manual at the beginning of macros and restore calculations at the end of macros. If you need to recalculate the workbook you can manually tell Excel to calculate.*
- *You can turn off automatic calculation with a macro by setting it to xlmanual. Use the following piece of VBA code:*

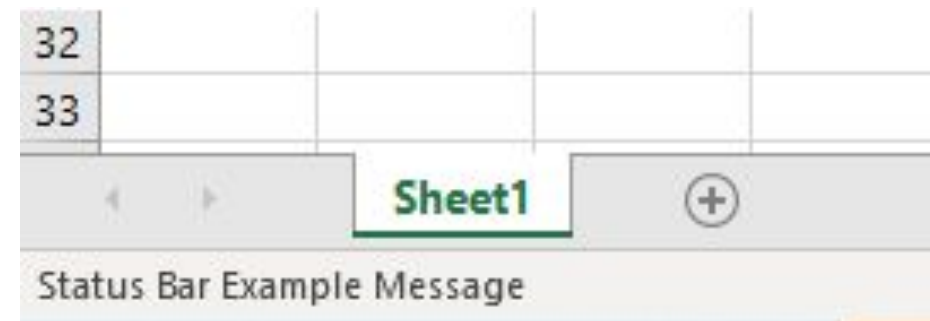
Application.Calculation = xlManual

- *To turn back on automatic calculation with the setting xlAutomatic:*

Application.Calculation = xlAutomatic

Status Bar

- *At the bottom-left corner of Excel, you'll find a Status Bar:*
- *Excel uses this status bar to communicate messages to you. However the StatusBar Property can also be adjusted using VBA, allowing you to display your own messages.*
- *You can update the status bar with your own custom message in Excel while a macro is running, using the StatusBar property.*



Status Bar

- Place this code in a macro whenever you want to update the user with a custom status bar message: **`Application.StatusBar = "I'm working Now!!!"`**
- And at the end of your macro place this code to clear the status bar and return control back to Excel: **`Application.StatusBar = FALSE`**
- Instead of displaying a message while your procedure runs, you can disable the Status Bar. This will increase the speed of your VBA code as Excel can skip processing what Status Bar message to display.
- To disable Status Bar updating while your code is running set the `DisplayStatusBar` property to false: **`Application.DisplayStatusBar = False`**
- At the end of your code, restore Status Bar updating: **`Application.DisplayStatusBar = True`**

Enable Events

- *Application.EnableEvents is a property in Excel VBA where you can specify whether you want events to take place when the VBA code is running or not.*
- *An event is usually attached to an object. VBA Events could include things such as activating a worksheet or saving a workbook or double-clicking on a cell.*
- *Now, these events are useful, but sometimes you may not want them to work.*
- *For example, if you're running a code that will activate and loop through all the open worksheets one by one, you may not want to execute the event code (in case you have it) when each sheet is activated.*
- *In such cases, you can set the Application.EnableEvents property to false, making sure the events are turned off when the code is running.*

Enable Events Example

- Below is a code where the main code is flanked by the `Application.EnableEvents` property.
- When you run this code, in case there are any events associated with the workbook getting saved, it will not be run. And since you have set the property back to `TRUE` in the last line, this will not impact other macros in your workbook.

```
Sub ExampleCode()  
Application.EnableEvents = False  
ThisWorkbook.Save  
Application.EnableEvents = True  
End Sub
```

Speed Up VBA Code

□ For optimal processing speed try using this code:

```
sub RunFast()  
  
Application.ScreenUpdating = False  
Application.DisplayStatusBar = False  
Application.EnableEvents = False  
ActiveSheet.DisplayPageBreaks = False  
Application.Calculation = xlCalculationManual  
  
'Your Code Here  
  
Application.ScreenUpdating = True  
Application.DisplayStatusBar = True  
Application.EnableEvents = True  
ActiveSheet.DisplayPageBreaks = True  
Application.Calculation = xlCalculationAutomatic  
end sub
```