**Macros Run Slower in Newer Excel?**

*by* [*Allen Wyatt*](http://www.tips.net/A0001_Allen_Wyatt.html) (last updated April 2, 2013)

Dave has a Windows XP system that still has an old version of Excel 97 on it. That system has a macro that is used to process information in a worksheet. It is able to process 500 rows of information in about 12 seconds. On a newer machine, running Excel 2010 under Windows 7, the same macro operating on the same data takes approximately 3 to 4 minutes to run. Both machines have the same amount of memory (4 Gb) and comparable processors. Dave wonders why the macro runs so much slower on the newer version of Excel.

It's very hard to make anything but the most general of comments without actually seeing the code, but that doesn't mean there isn't anything to say. :>) The first thing you'll want to do is check to see how your macro is doing its work. For instance, does it step through all the rows of the worksheet to do its processing? (Even though the macro is processing 500 rows of information doesn't mean it isn't stepping through every row on the worksheet.) The reason this might be critical is because earlier versions of Excel only used 65,000+ rows in a worksheet, whereas the latest versions can handle 16 times as many rows (over 1,000,000). If your macro was taking 12 seconds to run before, 16 times as long would be 192 seconds, which is between 3 and 4 minutes.

There are also reports that Excel 2010 communicates rather regularly with the printer driver, in a way not done in previous versions of Excel. You can figure out if this is slowing down your macro by turning off the communication at the start of your macro:

Application.PrintCommunication = False

At the end of your macro, remember to set the property back to True so that Excel can again communicate with the printer.

It could also be that there is something different with the newer machine, even if the difference is subtle. For instance, the newer machine could have different background programs running that the older machine—anything from an anti-virus program to voice recognition software. In addition it might have some different Excel add-ins loading that aren't on the earlier system. You wouldn't think that such things would affect macro performance, but they really could. You won't know, of course, until you track down what is loading, disable it, and then try running your macro again.

Also, it could be that the newer machine has some external devices that it needs to poll periodically which could be slowing things down. I have a system in my office that has an external hard drive, connected through a USB port. At times, the system itself (Windows) needs to go out and power-up the external drive, and things essentially stop on the computer while this happens. Guilty devices could be USB attachments, scanners, network drives, etc. It can be very frustrating, especially when it slows down something that shouldn't take long at all.

A good suggestion in this instance is to not rely solely on what you find with a single Excel 2010 system. If you have access to several machines—perhaps even some out of the office—try the macro on them to see what happens. If it runs faster, then you know that there is something unseen at play on the slow 2010 system.

If you find that everything really is "equal," then you may need to get into the VB Editor on the system and start doing some timing of the various parts of your macro. This is tedious, but it can help you narrow down where, exactly, the macro is bogging down.

If you decide to go this latter route, you'll find it worth your while to invest in a good Excel VBA programmer's reference book. There are several on the market, so shop around a bit. (You can't go wrong with anything authored by John Walkenbach, and I've heard good things about *Excel 2007 VBA Programmer's Reference*.)

*ExcelTips* is your source for cost-effective Microsoft Excel training. This tip (12440) applies to Excel versions: 2010

***Related Tips:***

* [Relative References when Recording Macros](http://excelribbon.tips.net/T003331_Relative_References_when_Recording_Macros.html)
* [Using Macros in Protected Workbooks](http://excelribbon.tips.net/T012570_Using_Macros_in_Protected_Workbooks.html)
* [Workbook Events](http://excelribbon.tips.net/T009546_Workbook_Events.html)
* [Deleting a File in a Macro](http://excelribbon.tips.net/T010001_Deleting_a_File_in_a_Macro.html)

[http://cdni.tn-services.com/S9997/images/books/98px/book_14.gif](http://store.tips.net/T010154_ExcelTips_Times_and_Dates.html)**Got the Time?** Understanding the ins and outs of working with times and dates can be confusing. Remove the confusion--ExcelTips: Times and Dates is an invaluable resource for learning how best to work with times and dates. Check out [*ExcelTips: Times and Dates*](http://store.tips.net/T010154_ExcelTips_Times_and_Dates.html) today!

***Comments for this tip:***

**Dave K**    15 Jan 2013, 07:13

Thanks for the ideas there. I'll try them out. A couple of comments...  
  
Presumably to output to the status bar less frequently implies a progress calculator to decide when to output the message. Does this not exchange one bit of processing for another?  
  
Mu macro runs sequential processing of a number of rows of data. There is no search involved. While there may be 500 rows to process, they are copied individually to a second workbook into a sheet which has only one record or row of data. Five values calculated in a second sheet are copied back to the original.  
  
At the end of the day, my question is not so much about how to make the macro more efficient (although that always helps) but more about why the exact same code on the exact same files should run so much slower on the later version of Excel running on a later version of Windows.  
  
This is becoming academic at this point as my colleague is so pleased at having reduced the processing time from a couple of days to a couple of minutes.   
  
I will say this though. Another colleague has a particularly troublesome workbook that derives data using VLOOKUP and other interesting functions. On his PC (later s/w and o/s versions) it takes up to 30 minutes to open and recalculate. On mine... a few seconds. It definitely seems to me as if some system overhead is getting in the way.  
  
Anyway, thanks for your contributions, everyone. They are much appreciated.  
  
Dave

[**Aldo Santolla**](mailto:a.santolla@shaw.ca)    14 Jan 2013, 11:31

What I find to help speed up searches through large data lists is to set a range variable for the list and use that variable to do the searching. It's like puting the entire list into memory.  
  
For example...  
  
Dim MyDataList as Range  
  
Set MyDataList = WorkSheetName.Range("Where\_My\_Data\_Is")  
  
This may help. I used it on a data list with over 2200 rows and over 30 columns of data, and I don't even notice it working.

**Jeff C**    14 Jan 2013, 08:55

It sounds like you don't want to get rid of the status bar, so perhaps you can update the status less frequently (at 10% intervals, for example).

**Dave K**    14 Jan 2013, 04:32

Hi guys. Thanks for the tips. The structure of my code is as follows:  
  
"Dim" statements for 2 integer counters and 3 worksheet variables;  
turn screen updating off (already there, thanks Aldo);  
assign source and target worksheets to variables;  
count the number of rows to process using the following code:  
iRow = ActiveSheet.Cells(Rows.Count, 1).End(xlUp).Row ;  
  
within the loop of the program...  
output the number of the current row being processed to the status bar (this is very useful for watching the speed and progress) with the code  
Application.StatusBar = "Processing row " & i  
directly assign values, for example:  
wsTarget.Range("E6").Value = wsSource.Range("C" & i).Value  
(17 cell values are assigned);  
followed by 5 similar assignments to return the results to the Source sheet;  
exit program loop;  
select one cell and display a "processing complete" message;  
clear the status bar;  
turn on screen updating;  
  
That's it! No rocket science there.  
I'll try running the process on other PC's and see what happens.  
  
Thanks for your help,  
Dave.

**Thomas Papavasiliou**    13 Jan 2013, 12:29

Also try to avoid the "Select" command

[**Aldo Santolla**](mailto:a.santolla@shaw.ca)    13 Jan 2013, 03:32

One thing I do if the macro I create runs into some heavy work is to turn off Screen Updating in the macro...  
  
Application.ScreenUpdating = False  
  
This will suspending the worksheet screen from updating with data until you turn on screen updating again...  
  
Application.ScreenUpdating = True