

Part 1: What's slow, whats buggy and whats faster!

Posted on [October 9, 2013](#) by [fastexcel](#)

Patrick wanted to know if I had any information on Conditional Format calculation and performance, and I have not looked at it for several years, so here goes!
I have done a series of experiments, using Excel 2007, 2010 and 2013, to try and get some insight on what Excel is doing under the covers. Because there is a lot to cover I have split the post into 3 parts.

This first part covers a simple experiment to see when Conditional formats get executed.

Formatting versus Calculation.

What Excel shows you on the screen or in a printout is the formatted (rendered) version of the results of a calculation.

And because formatting/rendering is such a cpu-intensive process Excel has a lot of tricks to try and minimise the time used (and thats why using `Application.Screenupdating=False` should be used everywhere to speedup your VBA).

Conditional Formats often do both calculation and formatting, so you have got double the chance of things being slow!

Excel does not generally allow formatting to be part of the calculation chain because **formatting occurs after the calculation has finished.**

This is also true for conditional formatting, although it its not clear to what extent there is a separate calculation-of-conditional-formats step before the formatting step.

Excel dynamically formats (re-paints) only what you see on the screen.

To save time Excel only does final formatting for the part of the results you can see on the screen. (so large screens are slower than small ones, and zooming out a long way is slower!).

When you have a lot of conditional formats this can **cause very noticeable delays in scrolling a page up or down,**

Conditional Formats can be Super-Volatile

Because of this dynamic repainting conditional formats are often executed even when no calculation occurs (for instance when you scoll up or down). So its not usually a good idea to embed a heavy calculation into a Conditional Format formula!

Lets start by looking at a very simple example that allows you to track when a condntional format gets executed. You can download the workbook `FormatConditionsA.xlsm` from [SkyDrive](#). Note it contains VBA so will not run properly in the Excel Web App.

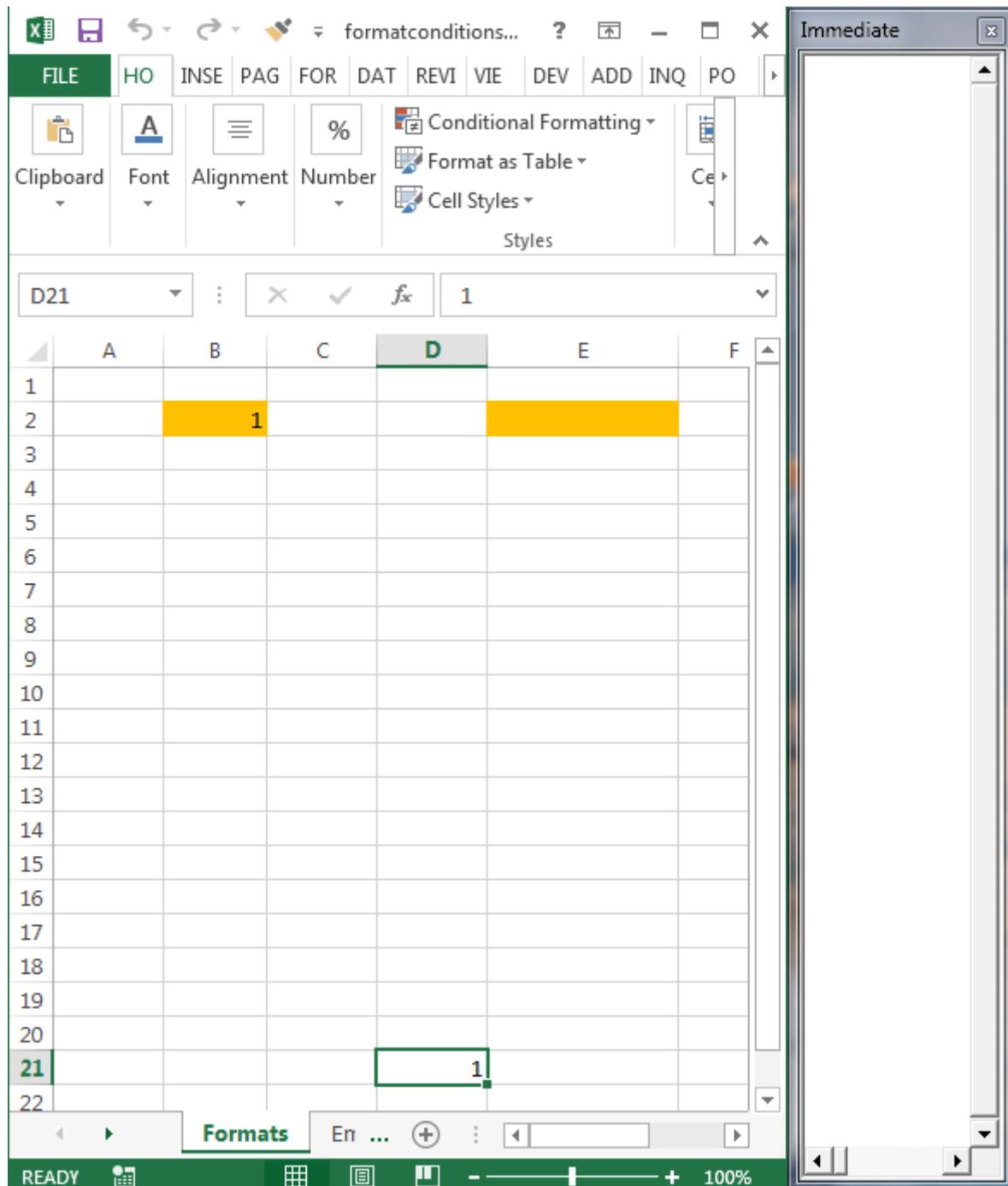
Test workbook `FormatCondntionsA.xlsm`

The workbook uses 3 cells and 2 VBA UDFs:

- Cell B2 contains a formula `=D21` and has two conditional format rules – colour orange if `=signal1(b2)` and colour green if `=signal2(b2)`.
Signal1 and Signal2 are VBA UDFs that increment a calculation counter and show it in the immediate window. Signal1 returns TRUE if B2 is an odd number and Signal2 returns TRUE if B2 is an even number.
- Cell E2 contains 2 conditional format rules that directly check cell D21 for odd (orange) or even (green).
- Cell D21 contains a number which you can change to either odd or even to see the effect on the conditional formats.

To run the experiments open the workbook and press F11 to see the VBIDE, then press Ctrl G to View the immediate window.

Then arrange the Excel window and the VBE window so that you can see both of them, and make sure that you can see Row 2 through 21 of the Excel window.



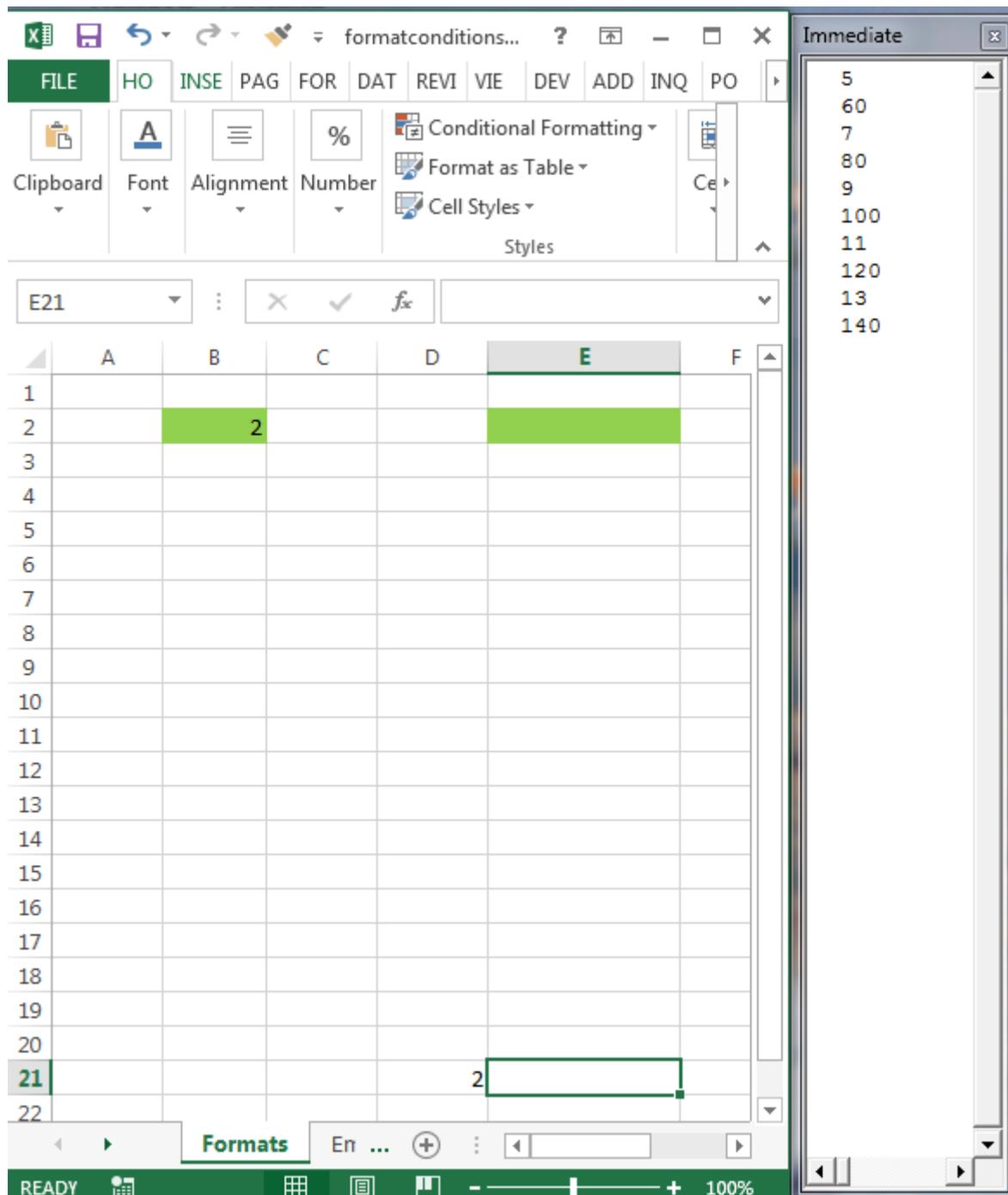
Experiment 1: Automatic calculation mode, User-interface driven

Switch to Automatic Calculation mode.

Clear the VBE immediate window.

Select Cell D21 and increment the number by 1.

Both cell E2 and B2 should change colour, and the Immediate Window shows how many times the UDFs have been calculated.



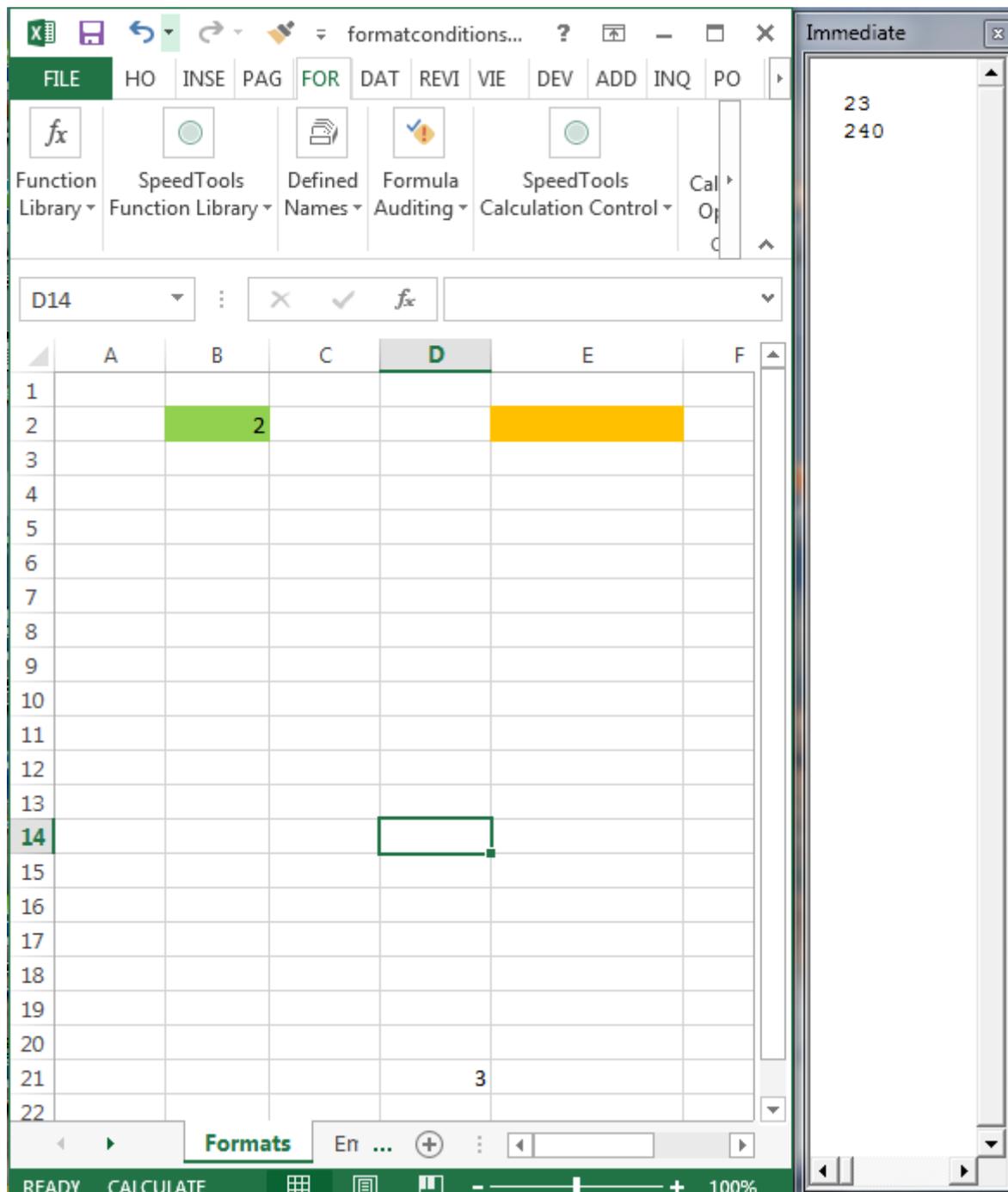
With Excel 2013 I get a total of 10 executions of the UDFs! (5 of each) !!! (No, I have absolutely no idea why, thats got to be a bug.)
Excel 2007 and 2010 only do 4 executions (2 for each UDF).

Experiment 2: Manual calculation mode, User-interface driven

Now switch to Manual calculation mode, clear the immediate window and select D21.

Increment D21 by 1: the result is

- The UDFs are not executed (nothing in the Immediate Window).
- B2 and E2 stay the same colour.

Page Down and then Page Up (to refresh the Excel window):

- **The UDFs are executed once.**
- Cell E2 changes colour because it directly refers to cell D21 which is now Odd.
- Cell B2 has correctly NOT changed colour because the conditional format is driven by cell B2 itself, which has not yet changed because it has not yet recalculated.

Now press F9:

The UDFs are executed once and cell B2 changes colour.

Experiment 3: The effect of refreshing the screen with Page Up and Page Down

Now increment Cell D21 again so that the status bar shows Calculate.

Press page Up Page Down repeatedly: the immediate window shows that the UDFs execute each time the screen gets refreshed with Page Up.

Now Press F9 to recalculate:

- Excel 2013 Page Down Page Up does not execute the UDFs
- Excel 2010 and 2007 does execute the UDFs once for each Page Down Page Up, even though it does not need to.

Experiment 4: Recalculating but with conditional formats scrolled out of sight.

- Clear the immediate window.
- Scroll the Excel window so that row 15 is the first row showing
- Increment cell D21 by 1
- Press F9 to recalculate, or Ctrl/Alt/F9 to Full Calculate

The immediate window shows nothing: the conditional formats have NOT been executed and will not be until you Page Up to make them visible.

(Note: if you only scroll so that the first row is row 3 the conditional formats DO get executed: looks like Excel is using about a 12 row buffer!)

Conclusions from Experiment 1.

- **Conditional formats are executed when the cell containing the conditional format gets repainted.**
- **Conditional Formats are not executed at a calculation unless they are on the visible portion of the screen.**
- **Excel 2013 looks a bit over-enthusiastic in Automatic Calculation mode, but smarter in Manual Calculation mode than Excel 2007/2010.**

In the next post I will explore what the performance impact of conditional formats is, and what is the impact of setting `Application.ScreenUpdating=False` and `Worksheet.EnableFormatConditionsCalculation=False`.

Part 2: What's slow, whats buggy and whats faster!

Posted on [October 10, 2013](#) by [fastexcel](#)

This is the second in a series of Posts on Conditional Formats ([see part 1](#)).

This post looks at the effects (**and the resulting bugs!**) on Conditional Formats of:

- `Application.ScreenUpdating`
- `Application.EnableConditionalFormatsCalculation`
- `Application.Calculation`
- Whether the cells containing the conditional formats are visible or not
- Screen Refresh
- Excel 2007, Excel 2010 and Excel 2013

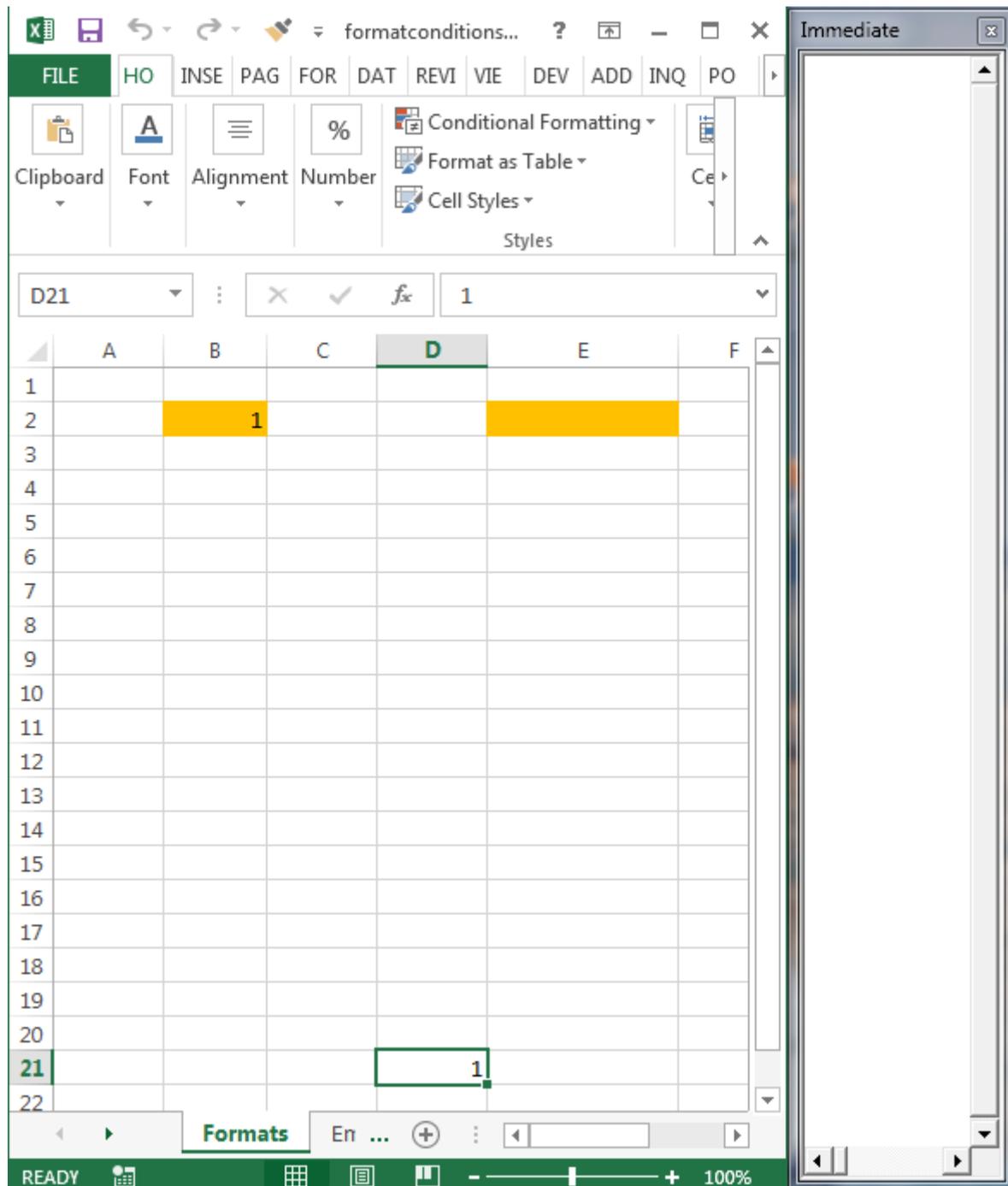
I am using the same (but slightly updated) test workbook as in Part 1: you can download it from [SkyDrive](#).

Running the Tests

The `FormatConditionsA.xlsb` workbook contains 12 VBA subroutines to do the testing (Test1 through Test3C).

Cell B2 uses 2 UDFs (`Signal1` and `Signal2`) to determine whether B2 is even or odd, and the B2 formula refers to D21

Cell E2 has 2 conditional format formulas that test directly whether D21 is even or odd.



You need to run the tests with the VBE window open and the immediate window visible.

For a more detailed explanation of this example workbook see [part 1](#).

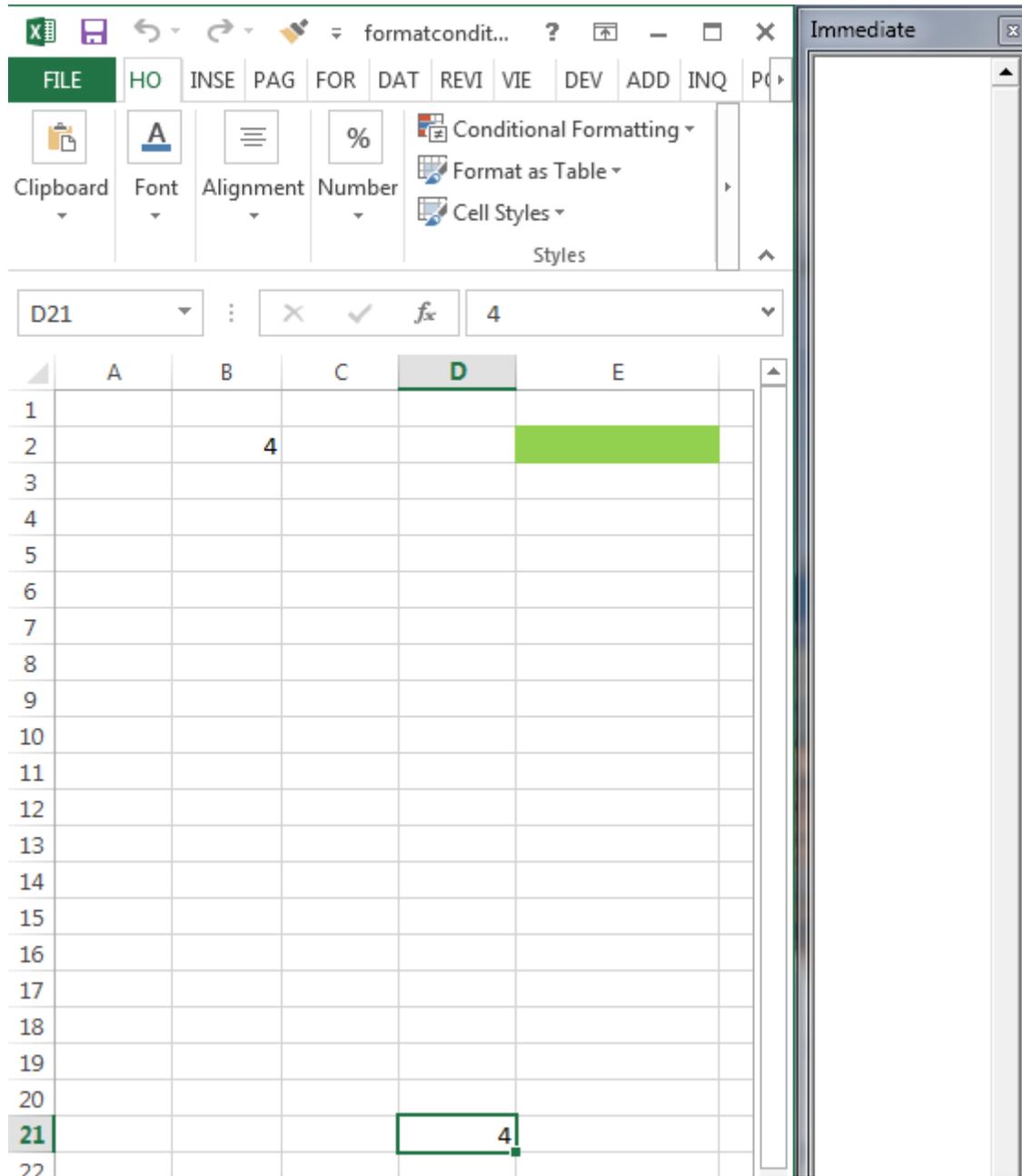
If you run the tests using Excel 2007, Excel 2010 and Excel 2013 you will see that a lot of work has been done by the Excel team to minimise the number of times the conditional formats get executed. But (as always when doing optimisations) this has tended to introduce bugs.

Test1: Screenupdating=True, Enable=True, Calc=Auto

- Excel 2013: OK, 1 call to each UDF, **large pause of a second or two before cell B2 refreshes its colour.**
- Excel 2010: OK, 2 calls to each UDF, no noticeable pause.
- Excel 2007: OK, 4 calls to each UDF, no noticeable pause.

Test2: Screenupdating=true, Enable=true, Calc=manual

- Excel 2007 & Excel 2010: OK
- Excel 2013: **Bug in cell B2 – neither of the conditional formats is applied to B2 and neither of the UDFs are executed. Scrolling down and up to refresh the screen does not fix this**, but pressing F9 does.



Test3: ScreenUpdating=False, Enable=True, Calc=Manual

- Excel 2013: OK
- Excel 2010: **Bug in B2. Neither of conditional formats are applied to B2 and the UDFs are not executed. Page Down Page Up does not fix** but F9 does.
- Excel 2007: **Bug in B2.** Page Down Page Up fixes.

Test1A: Screen=True, EnableFormatConditionsCalculation=False, Calc=Auto

**So what does setting EnableFormatConditionsCalculation to False actually do?
I am not sure, but what it does NOT do is to permanently switch off the evaluation of conditional formats!**

- Excel 2007: OK – **the pause in Test 1 has disappeared!**
- Excel 2010: **Bug in Cell E2**. the left-most vertical border is coloured correctly but the rest of the cell is not! Page Down Page Up fixes it.

	A	B	C	D	E	F
1						
2		3				
3						
4						
5						

- Excel 2013: **Bug in cell B2 and E2**. Page Down Page up fixes it.

	A	B	C	D	E
1					
2		3			
3					

Test2A: Screen=True, EnableFormatConditionsCalculation=False, Calc=Manual

- Excel 2007: OK
- Excel 2010: **Bug in cell E2**. Page Down Page Up fixes it.
- Excel 2013: **Bug in cell B2. Neither Page Down Page Up nor F9 fix it**, but Ctrl/Alt/F9 does.

Test3A: Screen=False, EnableFormatConditionsCalculation=False, Calc=Manual

- Excel 2013: OK
- Excel 2010: **Bug in cell B2: Neither Page Down Page Up nor F9 fix it**, but Ctrl/Alt/F9 does
- Excel 2007: **Bug in cell B2**. Page Down Page Up fixes it.

Tests 1B to 3C: switching to another sheet, run the tests, switch back

Its magic: all these tests run correctly in all versions!

Conclusions

- **Looks like using UDFs in conditional format formulas is rather buggy: avoid.**
- **EnableFormatConditionsCalculation does not look useful. But there were many reports of a problem importing Excel 2003 files with conditional formats into later versions that could be fixed by setting it to True: I don't know if this problem still exists.**
- **The safest way is to activate a sheet that does not contain any conditional formats.**

The next post will focus on the performance of conditional formats.

Part 3: What's slow, whats buggy and whats faster!

Posted on [October 11, 2013](#) by [fastexcel](#)

This is the third in a series of Posts on Conditional Formats ([see part 1](#) and [Part2](#)).

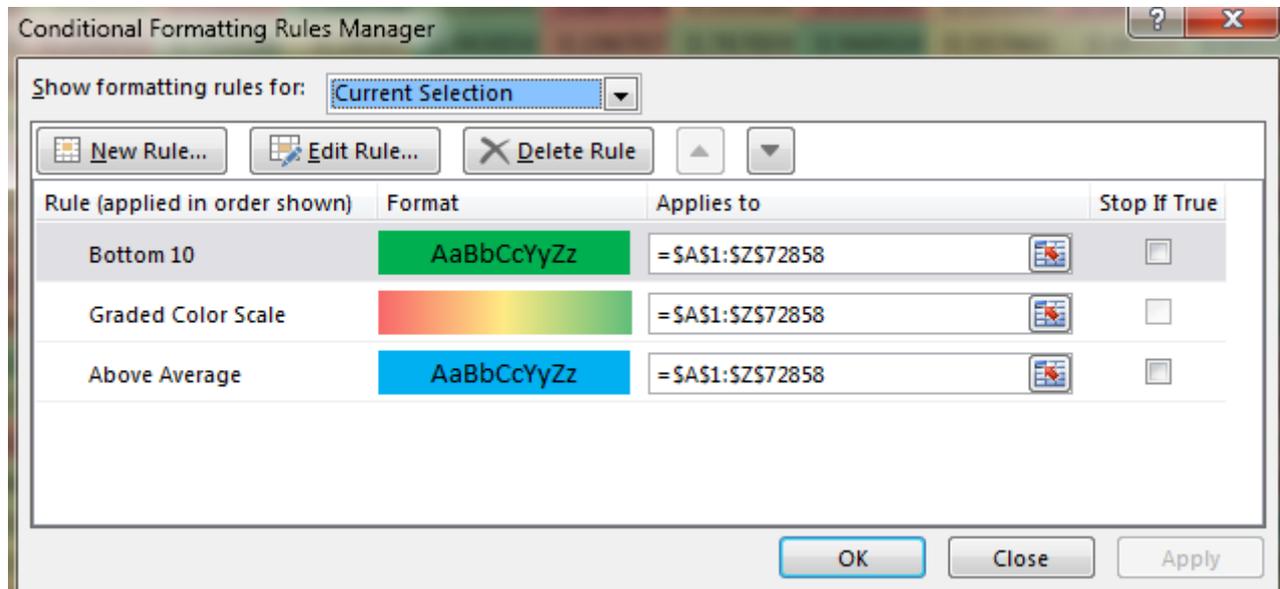
This post looks at the effects on the performance of Conditional Formats of:

- Application.Screenupdating
- Application.EnableConditionalFormatsCalculation
- Application.Calculation
- Whether the cells containing the conditional formats are visible or not

- Screen Refresh
- Excel 2007, Excel 2010 and Excel 2013

The workbook I am using is called (with stunning originality) FormatConditionsB.xlsm, and you can download it from my [Skydrive](#).

It contains 1.9 million Rand() formula in A1:Z72858, and each of these cells has 3 conditional format rules:



So that's 5.7 million conditional format rules.

There are 2 worksheets: Formats and Empty

The workbook also contains the MicroTimer api code for high resolution timing and 5 subs, Testing1 through 4 and testscroll1.

The subs typically set calculation mode, screenupdating and enableformatconditionscalculation, time a calculation and then time a screen update.

For example here is the code for Testing1:

```

1      Sub testing1()
2          Dim osht As Worksheet
3          Dim dtime As Double
4          Application.Calculation = xlCalculationManual
5          Worksheets("Formats").Activate
6          Set osht = Worksheets("Formats")
7          Application.ScreenUpdating = True
8          osht.EnableFormatConditionsCalculation = True
9          dtime = MicroTimer
10         Application.Calculate
11         dtime = MicroTimer - dtime
12         Debug.Print dtime
13         dtime = MicroTimer
14         Application.ScreenUpdating = True
15         dtime = MicroTimer - dtime
16         Debug.Print dtime
17     End Sub

```

Timings with different sheets visible.

The workbook opens with the formats sheet visible.

If you click the Empty tab you instantly see the empty sheet.

But if you then switch back to the Formats sheet **there is a noticeable delay of about a second before the screen refreshes**. Similarly pressing Page Up takes just over a second before the screen refreshes.

This is because Excel re-evaluates the conditional formats for the visible cells on the active sheet at each screen refresh.

Pressing F9 to recalculate the 1.9 million RAND() formulas with the Formats sheet visible takes 2.8 seconds, but with the Empty sheet visible it takes 0.2 seconds- again its the evaluation of the visible conditional formats that takes the time.

Conditional Formats are not directly evaluated by a calculation.

Running the Testing Subroutines

Here are the timings in seconds for running Test1 through Test4, **with the Formats sheet visible.**

Screen Updating	Enable Format Conditions Calculation	Excel 2007		Excel 2010		Excel 2013	
		Calc	Refresh	Calc	Refresh	Calc	Refresh
True	True	3.97	1.88	3.02	1.40	3.12	0.001
True	False	2.28	2.01	1.68	1.50	1.60	0.001
False	True	0.21	1.69	0.20	1.57	0.22	0.003
False	False	0.24	2.01	0.20	1.30	0.18	0.003

The conclusions of this test are:

Excel 2010 and Excel 2013 are noticeably faster than Excel 2007.

- **Turning off screen updating is the big winner**
- Switching off EnableFormatConditionsCalculation is only worthwhile if ScreenUpdating is true
- Switching off EnableFormatConditionsCalculation is much less effective than switching off ScreenUpdating
- Although Refresh looks very fast in Excel 2013 it actually just postpones the refresh to after the VBA has finished, so in fact its not faster.

I then repeated the tests, **but with the Empty sheet visible rather than the Formats sheet:**

Screen Updating	Enable Format Conditions Calculation	Excel 2007		Excel 2010		Excel 2013	
		Calc	Refresh	Calc	Refresh	Calc	Refresh
True	True	0.29	0.006	0.20	0.006	0.19	0.0009
True	False	0.25	0.006	0.20	0.006	0.18	0.0008
False	True	0.29	0.01	0.19	0.01	0.21	0.002
False	False	0.26	0.02	0.20	0.01	0.19	0.003

This completely avoids the refresh evaluation of the conditional formats and the times are comparable to the first set of tests with Screen Updating False.

I also tried repeating the tests with the **Formats sheet active but hidden behind the VBE window.**

The timings were virtually the same as with the Formats sheet visible.

So its the refresh of the conditional format cells within the activesheet window that uses the time, even if its hidden behind some other window.

I also ran TestScroll1. This times the effect of a complete scroll of the conditional formats window.

Scroll Test	Excel 2007		Excel 2010		Excel 2013	
	Up	Down	Up	Down	Up	Down
Formats Active	2.00	1.98	1.39	1.39	1.44	1.44

As you can see the scroll times are comparable to the refresh times in the first set of tests, except for Excel 2013.

But the Excel 2013 refresh timings in the first test are cheating because the refresh actually takes place after the VBA sub has ended.

Range.Calculate and Range.CalculateRowMajorOrder

If you use Range.CalculateRowMajorOrder on a single cell (or a large block of cells) it takes about 1.4 seconds – the same time as a scroll/screen refresh.

But Range.Calculate takes almost exactly twice as long – looks like it causes 2 screen refreshes not one!

Seriously Slow Conditional Formats

If you want to play with a workbook containing some seriously heavyweight conditional formats you can download [ConditionalFormatsC.xlsx](#)

This has 132K formulas =INT(RAND()*1000) in A1:V6000 and each cell has a single formatting rule to colour orange duplicated values in A1:V6000. (well of course they all turn orange).

With the Formats sheet visible pressing F9 to recalculate takes about 40 seconds.

And it looks like evaluating the conditional formats is all single-threaded: no advantage from multiple cores!

But with the Empty sheet visible F9 takes 0.03 seconds.

Conclusions

- **Heavy conditional formatting can be slow**
- **Conditional Format evaluation is single-threaded**
- **EnableFormatConditionsCalculation is not very useful**
- **Evaluation of conditional format rules takes place at screen refresh time rather than calculation time**
- **Only the conditional format rules for cells that are shown on the active window(s) get evaluated**
(large screens will be slower than small screens and zoom out slows you down!)
- **ScreenUpdating=false works well, but the final refresh time will occur when the Sub is exited.**
- **Using UDFs in conditional formats is probably not a good idea**
- **The interaction of VBA and conditional formats looks buggy**
- **Excel 2013 and 2010 are faster than 2007 for Conditional Formats**

2 other **bugs with conditional formats** have been reported, but I don't know if they have been fixed in Excel 2013:

- Opening a file created in Excel 2003 with Excel 2007 could make the conditional formats fail to refresh unless you manually set EnableFormatConditionsCalculation=true
- Repeated copy-pasting Conditional formats in Excel 2007 duplicated the conditional formatting rules so that large numbers of rules were created.