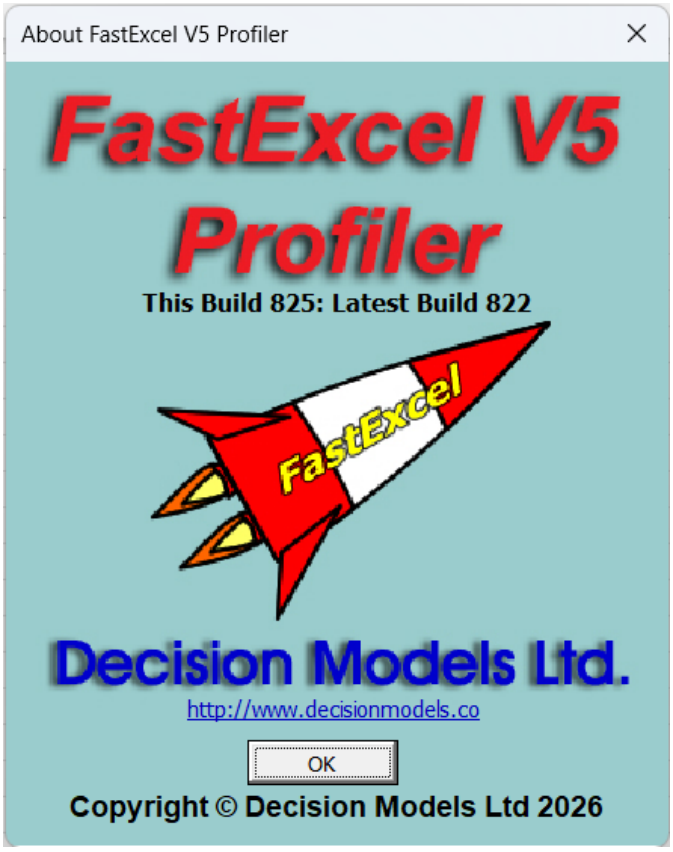
FastExcel Version 5 Profiler



User Guide

Contents

FastExcel Version 5 Profiler **1**

USER GUIDE 1

Overview **5**

Installing FastExcel V5 Profiler **6**

Automatic Installer (requires Admin level privileges) 6

Manual Installation for Windows 7

Quick Start Calculation Profiling: Drill Down **10**

Quick Start Calculation Profiling Reports 11

Quick Start VBA Profiling **12**

FastExcel VBA Profiling Report 14

FastExcel Trace Report 16

Using FastExcel Calculation Profiling **17**

How FastExcel V5 helps you run Excel faster: 18

Optimizing Excel Calculation Bottlenecks **19**

Use FastExcel V5 to Eliminate Calculation Bottlenecks 20

How fast should my spreadsheet calculate? 21

The FastExcel Profiler Ribbon **22**

FastExcel Calculation Profiler **24**

Five Calculation Profiling Commands 24

Memory and Versions 25

Check Workbook 25

The Profiling Header 26

Profile Workbook 34

Profile Formulas and Functions 43

Profile Worksheet Areas 46

Map Worksheet Cross-references. 48

FastExcel VBA Profiler **50**

ProfVBA Overview....50

ProfVBA Acknowledgements....51

Add Profiler Calls....52

Adding Drill-Down Profiler Calls using the Ribbon....54

Adding Drill-Down Profiler Calls using “Add Profiler Calls”....56

Every Nth Line....56

All Loops in Proc to Max Depth....58

Every Iteration of the Loops....59

Create a VBE Reference to FastExcel Profiler....59

Show Timings in the Immediate Window....60

All Loops in Proc Immediate Output....60

Every Iteration of Loops Immediate Output....61

Remove Profiler Calls....62

Settings, Enabling and Disabling Profiling and Tracing....63

Enable Profiling Options....63

Show Timings in the Immediate Window....64

Show ProfVBA Status....65

Show Profiling Report....66

Uncaptured Exits....67

FastExcel Execution Tracing....68

Advanced Use of ProfVBA: Inline Commands....70

Calculation Manager **71**

Calculation Manager Options and Settings....72

Excel Calculation Settings....73

Excel Current Calculation Mode....74

Excel Calculation Buttons....76

Workbook Calculation Settings....78

FastExcel Settings....80

Calculation Timing Commands 82

Getting Consistent Results from FastExcel V5 Timing....84

Calling FastExcel Methods from VBA 85

Calling FastExcel Profiling Methods....85

Calling FastExcel Calculation Methods....86

MICROTIMER function....87

MILLITIMER function....87

STRCOLID function....88

Migrating from FastExcel V4. 89

Overview

FastExcel Profiler gives you a wide variety of tools to analyze, manage, control and optimize the performance and memory usage of your workbooks. These tools fall into three main groups:

FastExcel Calculation Profiler

FastExcel Calculation Profiler finds and prioritises Excel calculation bottlenecks.
When using the Trial version only the Drill-Down Wizard is available and it is restricted to profiling a single worksheet.

- Drill-down Profiler Wizard
- Profile Workbook
- Profile Worksheet Formulas and Functions
- Profile Worksheet Areas
- Map Worksheet Cross References: This command maps the flow of calculations between worksheets.
- Memory and Versions shows memory used and available, and the versions of Excel and Windows being used.
- Check Workbook shows statistics and potential trouble spots in the active workbook.

FastExcel Calculation Manager

Calculation Manager times and controls Excel's calculation engine.

- Extended Calculation Settings
- Calculate Range
- Calculate Sheet
- Calculate Workbook
- Time a Macro

These tools can be used during development and optimization of a workbook to quickly compare the calculation speeds of different formulas, worksheets etc., and, with large slow workbooks, allow you to quickly calculate a small subset of the workbook.

FastExcel VBA Profiler

FastExcel VBA Profiler measures the time each VBA procedure takes to execute.

- Choose the projects, modules and procedures to profile
- Choose either to profile execution time or to trace execution flow
- Profiling report shows execution time from highest to lowest
- Easy-to-use methods for drilling down to profile loops and blocks of VBA
- Individual iterations of Loops can be profiled

Many thanks to **Stephen Bullen and Rob Bovey** for their help with ProfVBA: see pages 546-500 of "Professional Excel Development" ISBN 0-321-50879-3.
And many thanks also to **Don Soloway** who extended ProfVBA to handle profiling of Loops and Iterations of Loops.

Installing FastExcel V5 Profiler

FastExcel V5 Profiler requires:

- One of Excel 2021 or Office 365 Excel
- Either 32- or 64-bit Excel
- A Microsoft supported version of either 32- or 64-bit Windows
- .NET Version 4 or later

Two install methods are available:

- Automatic Installer: requires Administrative level privileges.
- Manual Installer: requires more manual steps than the automatic controller, but does NOT require Administrative level privileges.

Automatic Installer (requires Admin level privileges)

Download the Automatic Install zip file from the Decision Models website

http://www.DecisionModels.com/Downloads/InstallV5Profiler.zip

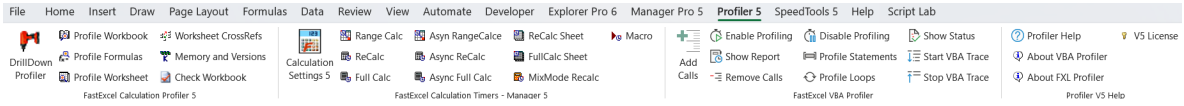
Automatic Installation requires administrative privileges.

1. Extract the installer .exe file from the downloaded zip

2. Double-click the extracted installer .exe file and follow the instructions to create a FastExcel V5 folder containing subfolders for each FastExcel V5 product. The default directory is called FastExcelV5 and is located in your Program Files (x86) directory.

Help files (.CHM) and a PDF version of the User guide will also be installed in these folders.

After successful installation FastExcel V5 will automatically be started when you start Excel, and you will find FastExcel V5 tabs for each installed product on the main ribbon.



If the ribbon does not show any FastExcel V5 tabs, or the installation was done for you by another user with administrative privileges, you may have to use the Manual Install instructions below:

Manual Installation for Windows

1. Create the FastExcelV5 folder

Extract the FastExcelV5 folder from the downloaded zip file to your chosen location

The FastExcelV5 folder contains up to 3 sub-folders, depending on which Manual Installer you chose:

- FastExcel V5 Profiler
- FastExcel V5 Manager Pro
- FastExcel V5 SpeedTools

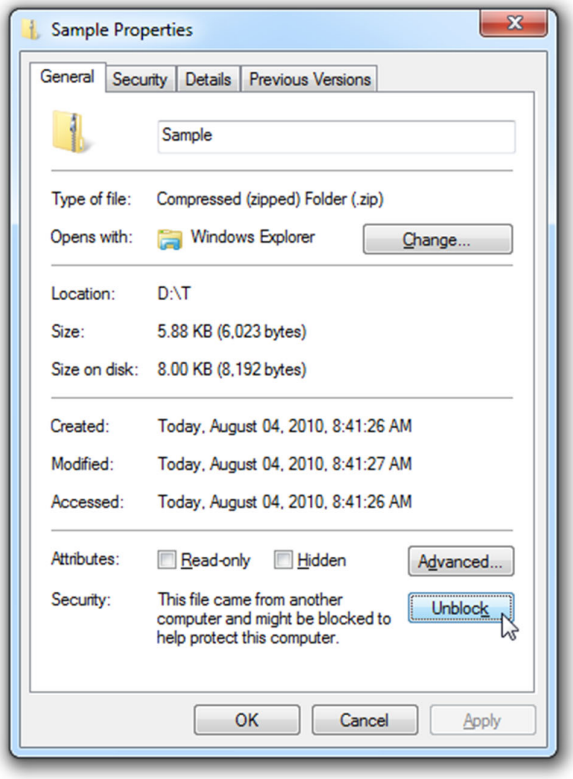
Each of these sub-folders contains an .XLAM file you need to load into Excel using Excel's Addin Manager.

2. Unblock all the XLAM, XLL and DLL files

Windows often blocks downloaded files.

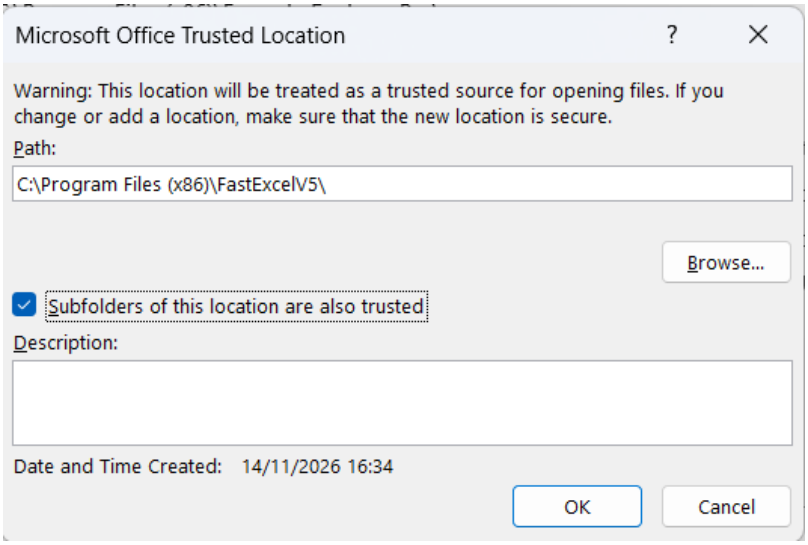
To unblock a file:

- The files you may need to unblock are:
  - FastExcelV5/FastExcel V5 Profiler/FxlV5Profiler.xlam
  - FastExcelV5/FastExcel V5 Profiler/QlmCLRHost_x64.dll
  - FastExcelV5/FastExcel V5 Profiler/QlmCLRHost_x86.dll
  - FastExcelV5/FastExcel V5 Profiler/QlmLicenseLib.dll
- Browse to each file, Select and Right-click it then select Properties
- Click the Unblock button



3. Make the FastExcelV5 folder a Trusted Location

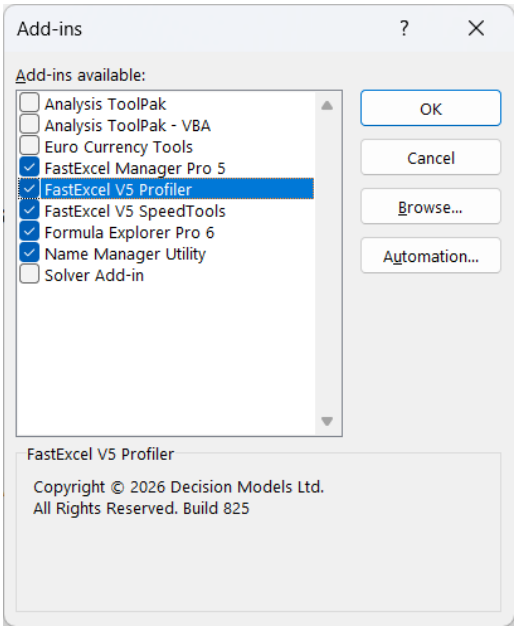
Start Excel and use File->Options->Trust Center->Trust Center Settings->Trusted Locations->Add new location->Browse to your FastExcelV5 folder->Check Subfolders of this Location are also trusted->OK



4. Use Excel's Addin Manager to install each FastExcel V5 addin

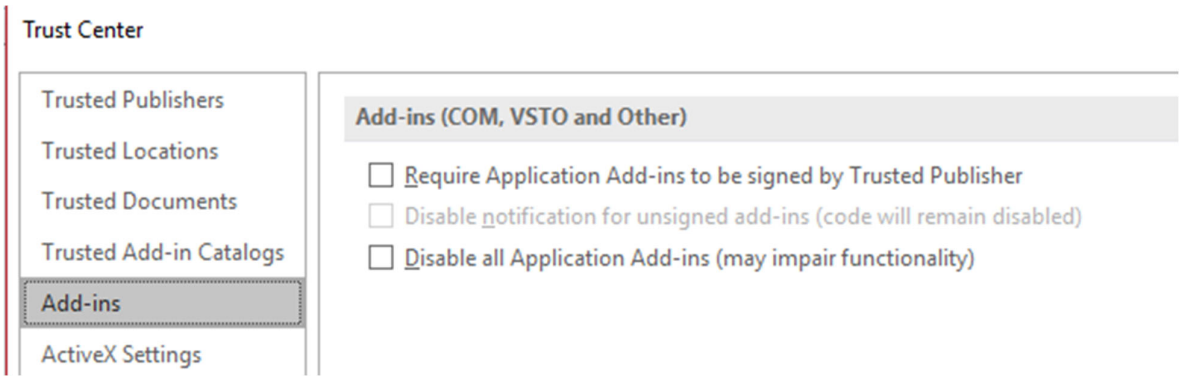
For Excel 2021 and Excel 365 Click File->Excel Options->Addins->Manage Excel Addins->Go… Press Browse and locate the FastExcelV5 folder.

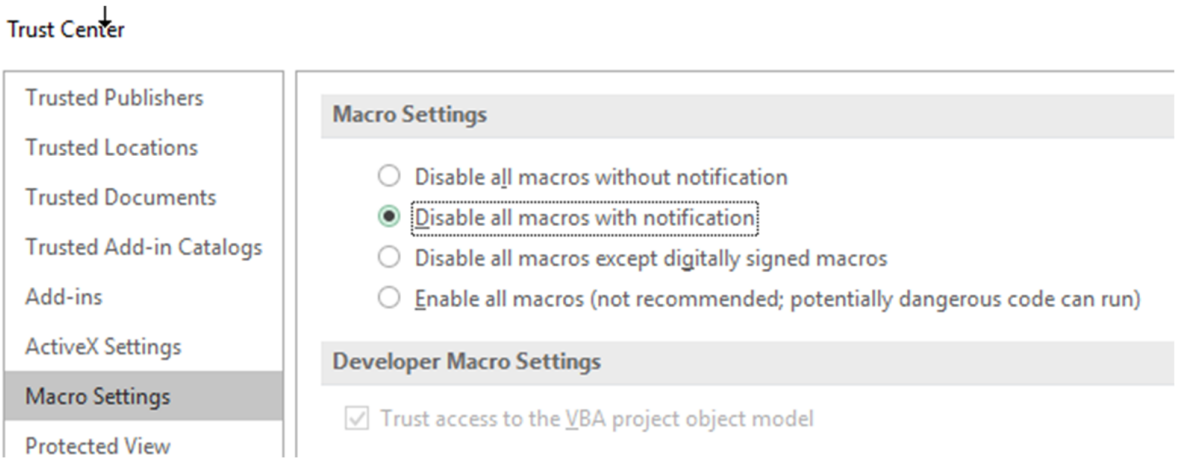
- In the FastExcel V5 Profiler sub-folder select the **FxlV5Profiler.xlam** file and click OK to return to the Addins form.
- **If asked "Do you want to copy this Addin to the Addins folder?" reply NO.**
- The Excel Addins form should now show the FastExcel V5 addins with a checkmark. Click OK to finish.



5. Recommended Trust Center Settings

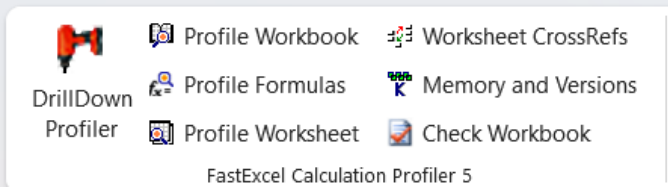
The recommended settings for addins and macros are:





Quick Start Calculation Profiling: Drill Down

If you would like to get some immediate profiling results on your workbooks without delving into lots of details you can simply use the Drill-down Profiler Wizard to find the Calculation Bottlenecks.
Once you are more familiar with some of FastExcel's features you can use the online help and FastExcel manuals to investigate additional ways of finding and eliminating bottlenecks.



Click Drill-down to Profile the active Workbook and up to 5 of the slowest worksheets: Note that the Trial Version only profiles a single worksheet

Find the problem worksheets, volatility and multi-threading inefficiencies

DrillDown When you click the Drill-down wizard button it first profiles the active workbook to find and prioritize:

- The calculation times for each worksheet.
- Workbook and worksheet Volatility and multi-threaded efficiency
- Used Range wastage by worksheet.

Find the worst formulas on the 5 worst sheets

Then the Drill-down wizard will **automatically** drill down step-by-step to profile the formulas on the worst 5 worksheets that contribute > 5% of the total calculation time. For each unique formula on each sheet you will see:

- % Contribution to total sheet calculation time
- Total and average calculation time
- Count and address of the formula
- Flags for CSE array formulas, Volatile, single-threaded formulas and formulas using UDFs.

Quick Start Calculation Profiling Reports

The Drill-down Wizard creates profile report sheets for:

- The workbook and its worksheets (sorted by calculation time).
- The formulas (sorted by calculation time) on the worst worksheets

Look for items that are highlighted in orange

The calculation profiler highlights in orange the potential problem areas you should look at.
You can see explanations of items by selecting a cell with a red triangle.

Worksheet potential problems

- Unique Formulas: Orange if any formulas contain full column references
- Conditional Formats Count: Orange if there are on average a large number of rules for each cell.
- What-If Data tables: Always orange
- Full Calc Time: Times are orange for sheets that are significantly above average
- Sheet Volatile %: Orange if calculation is significantly volatile for this sheet
- Used Range Waste: Orange if significant waste

Worksheet Profiles (Select Sheet name and click Drilldown to profile sheet formulas)

	Cell Counts			Unique	CSE		What-If	Calc Time (Millisec)			Sheet	Multi-Thread	MicroSecs	Used R
Sheetname	Used	Constants	Formulas	Formulas	Formulas	CondFmt	Tables	Single-Thread	Re Calc	Full Calc	Volatile %	Efficiency	/Formula	%Waste
Sum_Ifs	288,648		115	6	0	0	0	0.0	1.1	853.4	0.1%			25%
Count_Uniques	88,000		10,617	11	0	0	0	0.0	0.7	204.5	0.4%			28%
Lookups	4,000,080		10,004	6	0	0	0	0.0	0.2	64.0	0.4%			55%
Count_GT_100	29,820		10,005	11	0	0	0	0.0	20.0	19.4	103.1%			19%
Period_To_Date	170,017		19,995	6	0	0	0	0.0	0.7	10.1	7.2%			53%
DataTables_Ex02	130		10	2	1	0	1	0.0	0.1	3.1				62%
Tests	494		12	8	0	0	0	0.0	0.6	2.1	27.6%			0%
Examples	360		8	6	0	0	0	0.0	1.0	1.1	92.1%			71%

Worksheet Formulas Profile potential problems

- Formulas using a significant % of the worksheet calculation time are flagged orange.
- Volatile functions, single-threaded functions and UDFs are flagged orange.

Single-Threaded WorkSheet Formulas Profile for *Count_Uniques*

Formula			Formula	Microsecs	Total Millisecs		Formula %
Address	Flags	Unique Formulas	Count	/Formula	This Formula	Area/Sheet	of Sheet Time
$D$7	B-N-M	=SUM(IF(LEN(A6:A10605)>0,1/COUNTIF(A6:A10605,A6:A10605)))	1	185,538.54	185.54	389.37	48%
$D$11	B-N-M	=SUMPRODUCT((A6:A10605<>"")/COUNTIF(A6:A10605,A6:A10605&""))	1	182,684.64	182.68	389.37	47%
$D$15	O-?-S	=countu(A6:A10605)	1	11,467.74	11.47	389.37	3%
$D$19	X-N-M	=COUNTDISTINCTS(A6:A10605,2)	1	5,974.04	5.97	389.37	2%
$B$6:$B$10605	B-N-M	=IF(AND(A6<>"",A6<>A5),1,0)	10600	0.15	1.56	389.37	0%
		=COUNTA(UNIQUE(FILTER(A6:A10605,A6:A10605<>""					

When your worksheets contain formulas with volatile functions (INDIRECT, OFFSET, RAND, NOW …) calculation will be slow any time you make a change to the workbook.

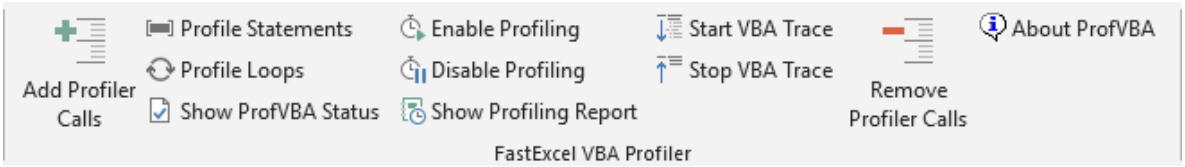
Multi-threading Efficiency

Excel's calculation engine tries to share out the calculation load across the cores of your PC.
But some functions (INDIRECT, GETPIVOTDATA, most User Defined Functions etc.) are single-threaded and so block effective use of the cores on your PC.

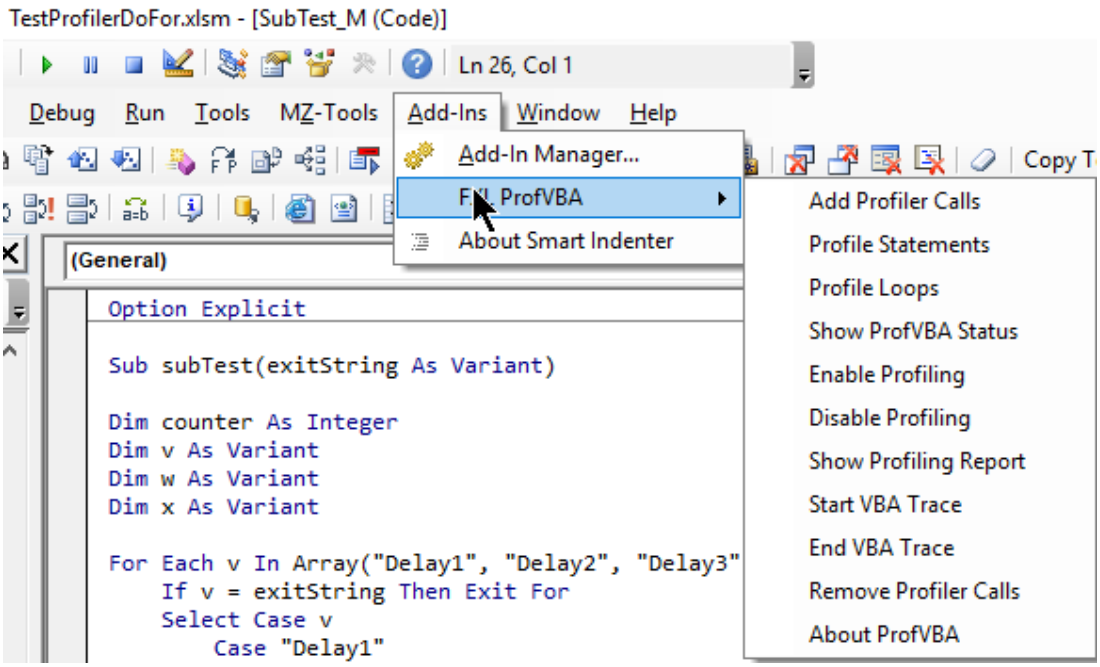
Quick Start VBA Profiling

The FastExcel VBA Profiler provides an easy-to-use way of finding out which parts of a VBA solution are using the most execution time. The profiler works by adding timing calls to your VBA. These timing calls store timing information whilst your VBA code runs. Then a profiling report can be produced showing the slowest parts of the VBA first. The VBA Profiling commands are available from:

- The Ribbon
- The ProfVBA Toolbar in the VBE
- The ProfVBA Menu in the VBE Addins Menu







The 4 Step VBA Profiling VBA Process

"Add Profiler Calls" → "Enable Profiling" → run your VBA → "Show Profiling Report"

Profiling your VBA

1. Make sure that you have enabled Access to the VBA Object Model (File->Options->Trust Centre->Settings->Macro Settings)
2. Use the VBIDE (Alt-F11) to unprotect the VBA you want to profile.
3. Use "Add Profiler Calls" to choose which projects, modules and procedures to Profile.
4. Enable Profiling
5. Run the VBA code you want to profile
6. "Show Profiling Report" to create the profiling report worksheet.

Drilling down within VBA Procedures

There are 3 different ways you can drilldown the profiling to within a specific VBA Procedure.

1. To profile a particular block of VBA statements, select them in the VBA and then use "Profile Statements".
2. To profile all the Loops in a particular VBA Procedure, select any line in the procedure that contains the Loops, then click "Profile Loops"
3. Use "Add Profiler Calls", select a Project, Module and Procedure then choose "Every Nth line of Proc"

Cleaning Up

Don't forget to "Remove Profiler Calls" to clean up your VBA!

Note:

The FastExcel timing buttons will work with VBA Profiling so that UDFs and VBA calculation event code can be profiled, but the FastExcel Profiling commands ignore VBA profiling because they trigger multiple calculation events.

FastExcel VBA Profiling Report

"Show Profiling Report" creates a report worksheet in a separate workbook.

VBA Profiling Results for fxlMgrPro470.xlsm

Total Execution Time in Milliseconds 21.35

Project.Module.Procedure	Count of Calls	Procedure Exclusive Time				% Of	Star
		Total	Average	Max	Min	Overall	
FXL4ManagerPro.fxlFormulaColorCells.SetAllExpColors	1	14.54	14.54	14.54	14.54	68.1%	
FXL4ManagerPro.ufViewEdit.GoRange	1	4.15	4.15	4.15	4.15	19.4%	
FXL4ManagerPro.ufViewEdit.FindFormRowCol	1	0.78	0.78	0.78	0.78	3.6%	
FXL4ManagerPro.fxlFormulaSelect.SelectFromTV	1	0.50	0.50	0.50	0.50	2.3%	
FXL4ManagerPro.clsTreeView.SetActiveNodeColor	6	0.27	0.04	0.06	0.02	1.2%	
FXL4ManagerPro.fxlFormulaParentChild.NodExp	51	0.16	0.00	0.01	0.00	0.7%	
FXL4ManagerPro.fxlFormulaColorCells.ResetNodeColors	1	0.15	0.15	0.15	0.15	0.7%	
FXL4ManagerPro.clsNode.mctlControl_Click	1	0.15	0.15	0.15	0.15	0.7%	
FXL4ManagerPro.fxlFormulaShow.ColorConditionalNode	25	0.14	0.01	0.01	0.00	0.6%	
FXL4ManagerPro.clsTreeView.ResetActiveNodeColor	4	0.08	0.02	0.03	0.01	0.4%	

The report timings are by default exclusive: they show the timed used by a procedure **excluding** the time taken by the calls from that procedure to other procedures. You can switch to **Inclusive timing** from the Enable Profiling command.

In this example SetAllExpColors is using 68% of the execution time.

You could then drill-down into SetAllExpColors to see which VBA statements are taking the time.

It turns out that the profile shows that 85% of the time is used turning Screen Updating back on after colouring the cells.

Call Sequence Counts

The Profiler maintains independent calling sequence counts for the first calls to a procedure and the last exit from a procedure. So, if Proc1 is the first Proc and calls Proc2 which then exits and returns to Proc1 which then exits you would get:

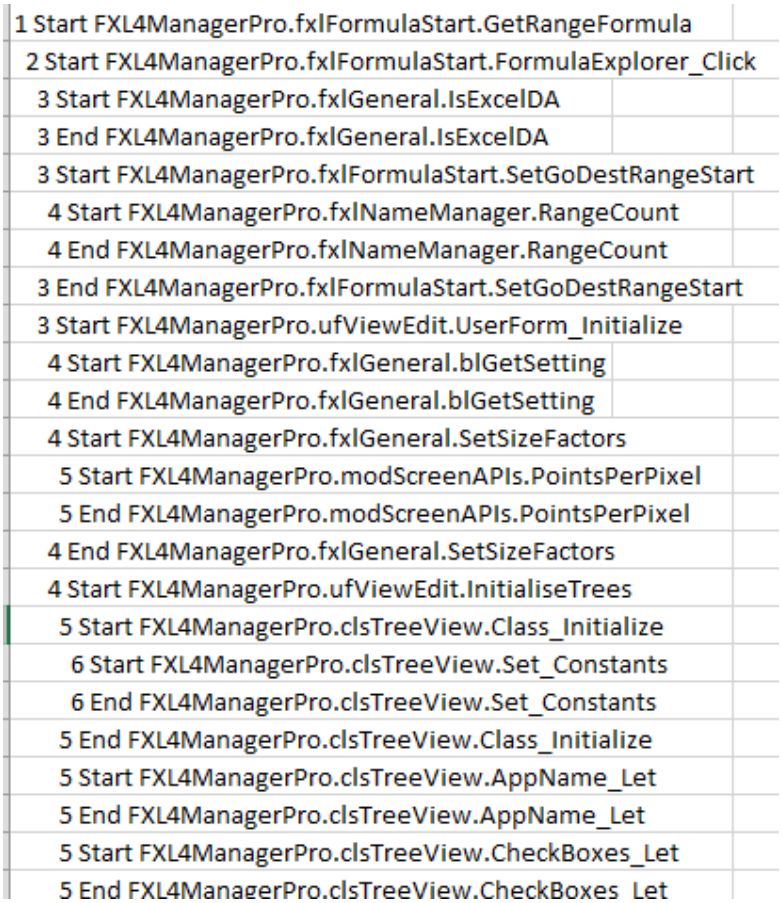
Proc Name	Start Sequence	Exit Sequence
Proc1	1	2
Proc2	2	1

Sorting by Start Sequence or Exit Sequence gives a quick way of looking at the starting or ending flow of your procedure calls.

For a more comprehensive trace see FastExcel Trace below.

FastExcel Trace Report

The Trace report is written to a .txt file in the same folder as the active workbook.



The starting number and indentation show the start and end calling level for the procedure.

Using FastExcel Calculation Profiling

To make the best of the Drill-down wizard it's a good idea to read this background material on profiling Excel calculations.

Full Calculation, Recalculation and Volatility: What and Why

To get the best out of FastExcel you need to understand the difference between an **Excel Recalculation**, where Excel works out how to recalculate the smallest number of formulas possible, and an **Excel Full Calculation**, where Excel calculates every single formula regardless of whether it has already been calculated or not.

Volatile formulas get recalculated at every calculation.

Which of these types of calculation is more important for you depends on how much of the input data changes each time you want to calculate the workbook, and how many **volatile formulas** you have. **FastExcel V5 Profiler will measure the volatility of each worksheet and can identify which built-in and XLL based functions are volatile.**

For example, if you are doing a monthly budget variance analysis each month most of the input data will change and almost all the formulas will need to be recalculated.

But if you are doing a what-if analysis on a cash-flow model then you may change only a single input number that will cause only a small number of the formulas to be recalculated.

If possible, you should decide which of these two scenarios is the most frequent for your workbook, because it can significantly affect which part of your FastExcel Profiling analysis is most significant for you, and also the methods you use to optimize your workbook calculations.

Multi-Threaded Calculation: What and Why

In Excel 2007 the Excel calculation engine was rewritten to use all the available CPUs/cores in your PC. This method of multi-threaded calculation can have a dramatic effect on your Excel calculation times: on a 4-core PC a well-designed workbook can run up to 4 times faster than with single-threaded calculation.

But some worksheet functions (all VBA user-defined functions and many add-in library functions) are single-threaded and so will seriously slow down calculation.

FastExcel V5 Profiler can measure the multithreaded efficiency both of the individual worksheets and the overall workbook, and can identify any single-threaded functions being used.

How FastExcel V5 helps you run Excel faster:

You can speed up your Spreadsheet by finding and eliminating calculation bottlenecks

Most slow-running spreadsheets contain a small number of problem areas, or Bottlenecks.
Because Excel is such a flexible spreadsheet system there are usually many different formulas that can produce the answer you want. Some of these formulas are much faster than others.

***FastExcel** V5 helps you find and prioritize bottlenecks*

In large spreadsheets it can be difficult to locate and prioritize the Bottlenecks.
You can use the Drill-down wizard and the wide variety of timing and profiling tools in FastExcel to rapidly drill down, locate and prioritize these bottlenecks.

Optimizing Excel Calculation Bottlenecks

Most Excel spreadsheets contain a number of calculation bottlenecks. Some of the most common bottlenecks are:

- **Exact Match Lookup** using MATCH, VLOOKUP, and HLOOKUP: Excel has to scan through each row of the data table until it finds a match. This can be very slow for large tables.
- **Array Formulas and SUMPRODUCT**: Using Array formulas and SUMPRODUCT can do amazing things, but forces Excel to do many calculations, which often results in slow calculations.
- **Excel calculating more than you need**: you can use FastExcel to more precisely control which parts of your workbooks should be calculated.
- **FILTER, SUM, SUBTOTAL, SUMIF, and COUNTIF**: These formulas can make Excel scan a large number of cells.
- **Single-threaded Functions**: Using single-threaded worksheet functions such as INDIRECT can slow down calculation by a large factor.
- **User-Defined Functions**: There are significant overheads involved in calling VBA UDFs and in transferring data from Excel to the UDF. With care these overheads can be minimized.
- **Volatile Functions**: Using volatile functions such as OFFSET or INDIRECT means that Excel cannot get the best out its smart recalculation engine so that each recalculation takes longer.
- **Conditional Formats**: A large number of conditional formats can significantly slow workbook calculation.
- **Large Ranges**: Using larger ranges than necessary can be expensive for calculation time.
- **Duplicated calculations**: it is very easy to build a spreadsheet where many of the calculations are being repeated many times.
- **Workbook Links**: Links to other workbooks are slow and fragile.
- **What-If Data Tables:** Data Tables use single-threaded iterative calculation methods which are slow.

Use FastExcel V5 to Eliminate Calculation Bottlenecks

Once you have identified and prioritized the calculation bottlenecks you can set about eliminating and reducing them:

Slow Lookups and Matches

- Use SpeedTools's advanced memory lookup technology to speed up recalculation of exact match lookups.
- Sort the data and use SpeedTools super-fast sorted exact match lookup.
- SpeedTools MEMMATCH, MEMLOOKUP, AVLOOKUP2 and AMATCH2 all use memory lookup and sorted exact match lookup
- Use SpeedTools AVLOOKUP2 and AMATCH2 built-in exact match error handling
- Handle multiple-condition lookups efficiently with SpeedTools AVLOOKUP2 and AMATCH2 rather than using slow array formulas or concatenation

Slow Array Formulas and SUMPRODUCT formulas

- Use SpeedTools FILTER.IFS powerful and efficient multiple condition handling to replace slow SUMPRODUCT and array formulas.
- Sort your data and use SpeedTools FILTER.IFS ability to exploit sorted data efficiently
- Minimize the effective size of the ranges you are using with FastExcel Managers Dynamic Range Wizard (part of FastExcel V5 Name Manager Pro)

Eliminate Unnecessary Calculations

- Use FastExcel V5 SpeedTools' extended calculation options to control exactly which parts of your spreadsheets should be calculated.
- Look for duplicated calculations in formulas or parts of formulas and break them out into a separate column so that they only have to be done once.

Slow VBA UDFs

- Installing FastExcel V5 SpeedTools will bypass Excel's VBE UDF refresh bug and speed up calculation when you have a large number of VBA UDFs.
- SpeedTools powerful and extensive range of functions may be able to replace some of your VBA UDFs

How fast should my spreadsheet calculate?

Studies on the effects of slow response time (see below) show that there are two 'comfort zones' of calculation times for users:

- For calculation times of less than about a tenth of a second users feel comfortable with Automatic Calculation.
- For calculation times of up to about 10 seconds in Manual Calculation mode users can maintain concentration and avoid errors.

So wherever possible you should try to use FastExcel to get your workbook calculation speed into one of these comfort zones.

The Effects of Slow Response Time

Research studies show that a user's productivity and ability to focus on the task deteriorate as response time lengthens.

- **Response time greater than 10 seconds:**

Users generally refuse to wait longer than 10 seconds

When response time is longer than 10 seconds users tend to switch to other tasks.

- **Response time greater than 1 second but less than 10 seconds:**

User errors, and annoyance level, start to increase, particularly for repetitive tasks

When response time is less than 10 seconds but longer than 1 second the user has difficulty in retaining a train of thought, but will probably not have switched to doing a different task whilst they are waiting.

- **Sub-Second Response:**

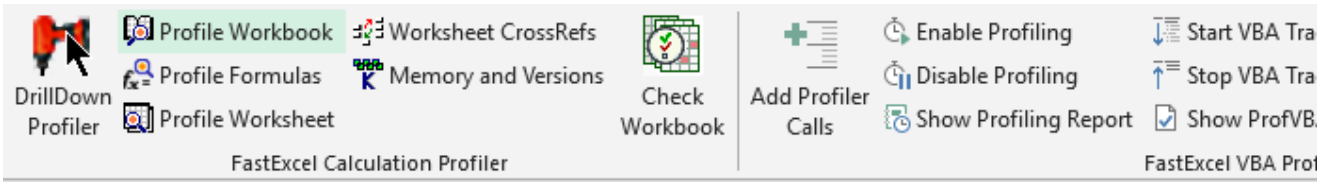
Improving calculation speed to less than a second increases productivity

For response times greater than a tenth of a second but still less than about 1 second, users can successfully keep a train of thought going, although they will notice the response time delay. IBM studies from the 1970s and 1980s showed significant productivity gains for users when response times were less than a second.

You will probably need to switch to Manual Calculation mode when entering data.

- **Instant response**

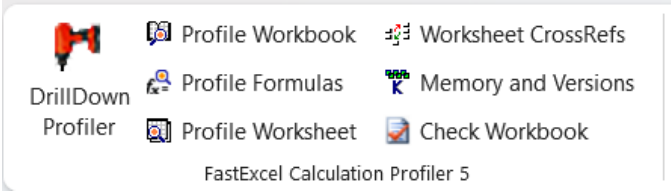
The FastExcel Profiler Ribbon



FastExcel V5 Profiler shows an additional tab on the main Ribbon.

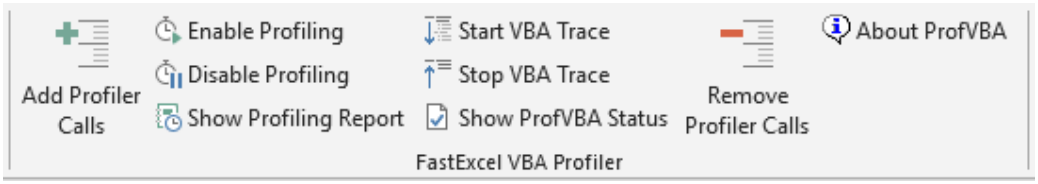
Calculation Profiling Commands:

*The **FastExcel** profiling commands allow you to find calculation bottlenecks*



Drill-Down Profiling Wizard, Profile Workbook, Profile Formulas, Profile Worksheet, Map worksheet cross-references, show Excel memory usage and Limits, produce a Check Workbook report.

VBA Profiling Commands



The VBA Profiling commands provide an easy-to-use way of profiling and tracing the flow of execution of your VBA projects.

Extended Excel Calculation Option and Calculation Timing Commands:

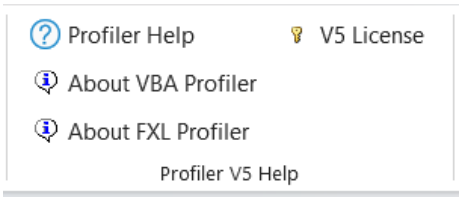


*The **FastExcel** Calculation Settings button allows you to control both Excel and FastExcel Calculation settings.*

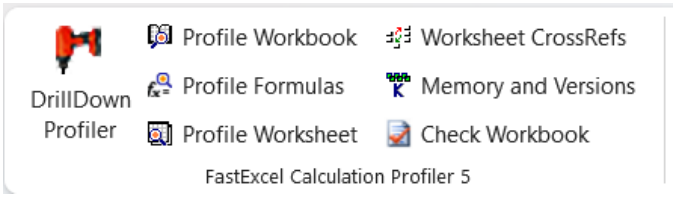
Calculation Settings for Excel, the Active Workbook and FastExcel. Includes Initial Mode, Mixed Mode and Active Book Calculation Modes. Show and change FastExcel V5 licensing options.

Calculate range, Recalculate selected sheets, Full calculate Sheet, Recalculate workbooks, Full calculate workbooks and Recalculate MixMode sheets and workbooks.

FastExcel Help buttons:



FastExcel Calculation Profiler



Five Calculation Profiling Commands

Note: when using the trial version of FastExcel only the DrillDown profiler command is available and it is restricted to a single worksheet.

Drill-down Profiler Wizard: This guides you to drill down to find the calculation bottlenecks from workbook to the individual formulas on the slowest worksheets.

Profile Workbook: Analyzes the calculation times of your workbook and worksheets.

Profile Worksheet Formulas and Functions: This analyzes the calculation times for unique formulas on a worksheet and identifies the functions used that are volatile or single-threaded.

Profile Worksheet Areas: This analyzes formula calculation times for the columns, rows and blocks of formulas in one or more of your worksheets.

Map Worksheet Cross-References: This analyzes the cross-references between worksheets and optionally re-sequences the worksheets to minimize forward cross-references.

*You can get **Contextual Help** on the output from the profiling commands either by right-clicking a selected cell on the output sheet or by selecting a cell and pressing the FastExcel Contextual Help button.*

Potential **bottlenecks** are highlighted in **orange**. The sensitivity of the orange highlighting can be controlled by setting FastExcel's Orange Tolerance (default 0.2) on the FastExcel Settings page: Cell Highlight (Orange) sensitivity.

The profiling commands help you to document calculation options, and generate timing statistics and memory usage details.

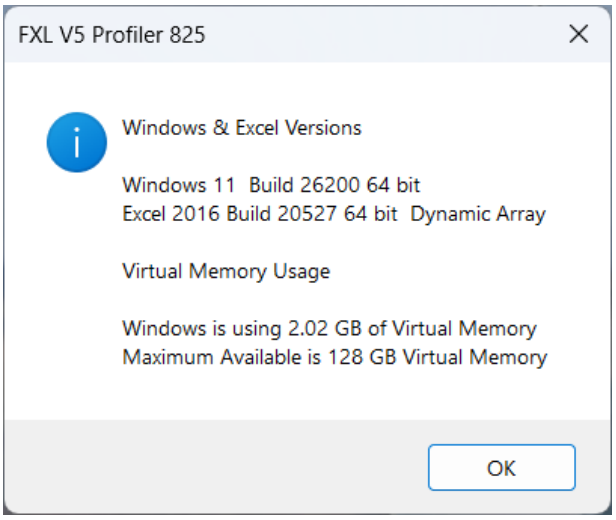
The profiling commands generate result worksheets into a separate workbook, using sheets called:

- FastXLBookn: The workbook profile results
- FastXLSheetn: the worksheet profile results
- FastXLFuncsn: the worksheet formulas and functions profile results
- FastXLPrecsn: the worksheet cross-reference map results

Multiple profiles of a given type are given an incremental number suffix.

Memory and Versions

FastExcel provides a simple way of finding how much memory your workbooks are using and what the maximum available amount of memory for your Excel configuration is.



Check Workbook

Check Workbook provides a quick way of checking a workbook for potential problems and showing statistics. The report is similar to the Profile Workbook report but replaces calculation timings with additional statistics.

FastExcel Workbook Checks

Physical Environment		
RAM(MB)	16,257	32,211
Swap File(MB)	23,094	34,259
Excel Memory(MB)	0	131,072
FileSize(K)	3,925	
Operating System	Windows 11 .26200 x64	
MHZ x Cores	3686 x 24	
Excel Version	Excel 2016.20527 x64	
XL Calc Engine	191029 Dyn Array	
Max Compat Version	2	

Profile:	14/11/2026 17:34	D:\Documents\XLEUC\XLVirtualGlobal2			
Workbook Settings					
	Shared	Decimal Places		Protection	
Shared	FALSE	Fixed	FALSE	Structure	FALSE
Changes	TRUE	Places	2	Worksheets	FALSE
Saved	FALSE	Backup	FALSE	Password	FALSE
Format	XMLWorkbookMacro	Styles	55	Views	0
Environment Counts					
Temp	192	XLB/Qat (K)	10	Com Addins	
Workbooks	2	VBE	8	Excel Addins	7

Worksheet

Sheetname	Cell Counts Used	Const	Formulas	Unique Formulas	Error Results	CSE Formulas	Data Validation	CondFmt
Examples	360	37	8	6	0	0	0	0
DataTables_Ex02	130	12	10	2	0	1	0	0
Sum_Ifs	276,621	61,951	115	6	0	0	0	0
Lookups	2,200,044	205,023	10,004	6	4	0	0	0
Formulas	98	18	9	9	1	0	1	9
BetaTest	7	5	0	0	0	0	0	0
Totals	**2,799,614**	**299,249**	**51,291**	**72**	**13**	**1**	**2**	**57**

Workbook S

Sheets Count	Cell Counts Used	Const	Formulas	Unique Formulas	Error Results	CSE Formulas	Data Validation	CondFmt
17	**2,799,614**	**299,249**	**51,291**	**72**	**13**	**1**	**2**	**57**

The Profiling Header

All five profiling commands document the profiled workbook by showing a common set of header tables:

FastExcel WorkBook Profile

Physical Environment		
RAM(MB)	15,960	32,211
Swap File(MB)	22,676	34,259
Excel Memory(MB)	0	131,072
FileSize(K)	3,925	
Operating System	Windows 11 .26200 x64	
MHZ x Cores	3686 x 24	
Excel Version	Excel 2016.20527 x64	
XL Calc Engine	191029 Dyn Array	
Max Compat Version	2	

Profile: 14/11/2026 17:18 D:\Documents\XLEUC\XLVirtualGlobal2

Workbook Settings					
Shared		Decimal Places		Protection	
Shared	FALSE	Fixed	FALSE	Structure	FALSE
Changes	TRUE	Places	2	Worksheets	FALSE
Saved	FALSE	Backup	FALSE	Password	FALSE
Format	.WorkbookMa	Styles	55	Views	0

Environment Counts					
Temp	190	XLB/Qat (K)	10	Com Addins	
Workbooks	2	VBE	8	Excel Addins	7

FastExcel V5 Profiler Build 825

2026\FastExcel_Demonstration_WkBk.xlsm

Calculation Options

Excel Calc Settings				Workbook Calculation settings			
Workbook Mode	All	Multi-threaded Calc		Mixed Mode		Other	
Current Mode	Automatic	On	TRUE	Full	TRUE	UpdateRemoteRefs	TRUE
Before Save	FALSE	Mode	Automatic	Manual	FALSE	PrecisionAsDisplay	FALSE
Initial Mode	First Workboo	Threads	24	Selected Sheets	FALSE	1904 DateSystem	FALSE
Iteration	FALSE	Force Full Calc		Restore after Open	FALSE	SaveXternalLinks	TRUE
Max Iters/change	100/0.001	Full/Recalc	Recalc	Mixedmode on Open	TRUE	Book Calc Engine	191029
Format Stale Values	FALSE			Accuracy Version	0	Compat Version	2

The Physical Environment Table

Cells with red dots contain on-sheet Notes: hover your cursor to make them pop-up.

Physical Environment		
RAM(MB)	15,960	32,211
Swap File(MB)	22,676	34,259
Excel Memory(MB)	0	131,072
FileSize(K)	3,925	
Operating System	Windows 11 .26200 x64	
MHZ x Cores	3686 x 24	
Excel Version	Excel 2016.20527 x64	
XL Calc Engine	191029 Dyn Array	
Max Compat Version	2	

The Physical Environment Table shows information about the system.

Orange if Memory Used>1MB and (Memory Used +64)>RAM (MB)

- **RAM (MB): Physical Memory Used and installed in megabytes.** You should have at least enough physical memory installed for Windows, Excel and all the open workbooks. See Bottlenecks.

*Orange if Used/Available > 3.75 * Orange Tolerance*

*Orange if Used/Available> 3.75*Orange Tolerance*

- **Swap File (MB): Swap File used and Swap file available in megabytes.** If your system is using a large amount of your existing swap file either close as many programs as possible or increase the size of your swap file.
- **Excel Memory (MB): Excel Memory Used and available.** Total Memory Used by all open Excel workbooks for data, formulas etc. (excludes Memory Used by the Excel program itself), and the maximum Excel workbook memory available for formulas etc. Megabytes.
- **File Size (K): The size on disk of the workbook in Kilobytes.**
- **Operating System: The version of OS in use when this profile was created.**
- **MHZ x Cores**: The **processor speed in MHZ, and the number of processor cores in the system.**
- **XLVersion: the Excel version number and bitness used for this profile**:
  - Excel 97 is Version 8
  - Excel 2000 is Version 9
  - Excel 2002 is Version 10
  - Excel 2003 is Version 11
  - Excel 2007 is Version 12
  - Excel 2010 is Version 14
  - Excel 2013 is Version 15
  - Excel 2016 is version 16
  - Subsequent versions also show 2016
- **XL Calculation Engine: The version number of the calculation engine that was used for this profile.**
- **Max Compat Version**: The maximum compatibility version supported by this build of Excel.
  Version 2 supports changes in the behavior of certain functions.
  version 3 supports changes in the handling of nested arrays

The Workbook Settings Table.

Workbook Settings					
Shared		**Decimal Places**		**Protection**	
Shared	FALSE	**Fixed**	FALSE	**Structure**	FALSE
Changes	TRUE	**Places**	2	**Worksheets**	FALSE
Saved	FALSE	**Backup**	FALSE	**Password**	FALSE
Format	WorkbookMa	**Styles**	55	**Views**	0

The **Workbook Settings Table** shows basic information on the profiled workbook:

- The name and path of the profiled workbook and the date and time that the profile was created. (Not shown in the above picture)

If Shared and Keep Change History are both true they are colored orange.

- **Shared**: True if the Workbook is open for Sharing. If Shared and Keep Change History are both true the Workbook size can grow considerably.
- **Changes**: Keep Change History Tracking: True if enabled, otherwise False. If Shared and Keep Change History are both true the Workbook size can grow considerably.
- **Saved**: False if the workbook has been changed in any way since the last save.

Avoid saving in Dual Format

- **Format**: the format of the saved workbook: **Dual Format (Orange)** doubles the file size.

- **Fixed**: Fixed Decimal: If True numbers will be treated as having an implied decimal point, in the position given by Places below.
- **Places**: The position of the implied decimal point, if Fixed Decimal is true.
- **Backup**: True if a backup copy of the workbook will be created when you save.
- **Styles**: The number of Styles in the workbook. If this number is large you may want to use Clean Workbook to map where the styles are being used, and to remove unused styles.
- **Protection**: The following three items show the protection status of the Workbook structure and windows, and whether the book is password protected.

Colored orange if true

- **Structure**: True if the structure of the workbook is protected. If protected with a password then opening the workbook will be slow.
- **Worksheets**: True if the worksheets in the workbook are protected.
- **Password**: True if the workbook is Password protected.
- **Views**: The number of Views in the workbook.

The Environment Counts Table

Environment Counts					
Temp	190	XLB/Qat (K)	10	Com Addins	
Workbooks	2	VBE	8	Excel Addins	7

- **XLB/Qat (K):** The size of the XLB and QAT files in K Bytes. The XLB file stores toolbar customizations. The QAT file stores ribbon customizations.
  When they get large, they may become corrupted and cause memory problems.
  The XLB file is located in %AppData%\Microsoft\Excel and the QAT file is located in %LocalAppData%\Microsoft\Office\
- **COM Addins:** The number of COM addins currently loaded
- **Temp**: Count of the temporary files in your Windows temporary directory and subfolders. A large number of Temp files may make your workbook slow to open or cause Excel to crash. The Clean Workbook command can delete all temporary files that are not currently being used.

Colored orange if >100

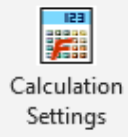
Note: **a very large number of temp files may cause FastExcel to hang with the status bar showing "Checking Temp Files".** If this occurs close Excel, reboot your PC and then delete all files in your temp folder. You can locate your temp folder using Start→Run, entering %temp% in the run box and pressing enter.

Colored orange if >15

- **VBE**: Total Number of Windows in use by the VBE IDE. Includes hidden windows. A large number of open VBE Windows consumes resources and may cause memory problems. (If blank in Excel 2002 or later check "Allow access to Visual Basic Project" in Tools, Macros, and Security). The Clean Workbook command can close all unnecessary VBE Windows.
- **Addins**: The number of Excel Add-Ins currently loaded by Excel's Addin Manager. This excludes any addins that have been loaded using File Open or the Decision Models Addin Loader.

The Calculation Settings Tables.

See FastExcel Calculation Settings for a detailed description of these options and how to set them



You can use the FastExcel Calculation Options button to change these calculation settings.

The Excel Calculation Settings Table

*By default, Excel Calculation Settings apply to **all currently open** workbooks.*

Excel Calc Settings			
Workbook Mode	All	**Multi-threaded Calc**	
Current Mode	Automatic	**On**	TRUE
Before Save	FALSE	**Mode**	Automatic
Initial Mode	First Workboo	**Threads**	24
Iteration	FALSE	**Force Full Calc**	
Max Iters/change	100/0.001	**Full/Recalc**	Recalc
Format Stale Values	FALSE		

FastExcel Active Workbook mode allows you to control Calculation for the active workbook instead of all open workbooks.
FastExcel Mixed Mode allows you to select sheets as MixMode sheets and control whether they will be calculated in Automatic, Manual or Sheet calculate.

- **Workbook Mode:** All or Active Book.
  The Excel default is to calculate all the open workbooks. With FastExcel you can restrict calculation to the active workbook only (and each workbook can have a different calculation mode, see Mixed Mode processing).

- **Current Mode**: Automatic, Manual or Semiautomatic.
  This mode applies to all the open workbooks or just to the active book. The Excel default is to apply this mode to all the worksheets in a book. With FastExcel's Mixed Mode you can control which worksheets are calculated in Automatic or Manual or Sheet Calculate mode.

- **Before Save**:
  If True Excel will recalculate before each save. Any worksheets that are NOT enabled for calculation will not be calculated.
- **Initial Mode**: Controls how Excel sets the calculation mode when started.
  First Workbook (Excel Default): Calculation mode is set from the first workbook opened.
  Automatic: Calculation mode will be set to automatic.
  Manual: Calculation mode will be set to Manual.
  Semi-Automatic: Calculation mode will be set to Automatic except Tables.
- **Iteration**: If TRUE Excel will attempt to resolve circular references by iteration.
- **Max Iters**: the maximum number of iterations used to resolve circular references.
- **MaxChange**: Iteration will stop when no cell changes by more than MaxChange

- **Format Stale Values**: if True then uncalculated formulas (stale) will be shown with strikethrough.
- **Multi-Threaded Calc:** Calculation using multiple cores and threads
- **On: True/False/ #N/A** Shows if Multi-Threaded Calculation (MTC) is on, off or not available
- **Mode:** Automatic – Excel assigns a calculation thread to each available core. Manual – the user has assigned a specific number of threads.
- **Threads:** The number of threads assigned to the calculation.
- **Force Full Calc:** Shows if this workbook will use Excel's smart recalc or always calculate every formula in the workbook at each calculation.
- **Full/Recalc:**
  Full: every formula in all open workbooks will be calculated at each calculation
  Recalc: only changed formulas and formulas dependent on changed values/volatile cells will be recalculated
  XL 64K Limit: in Excel versions prior to Excel 2007 Excel switched to force-full calculation when dependencies exceeded 64K. Colored orange if this limit has been reached

The Workbook Calculation Settings Table

These settings can be different for each open workbook

You can use Mixed Mode so that some worksheets always calculate and others calculate only when you want.

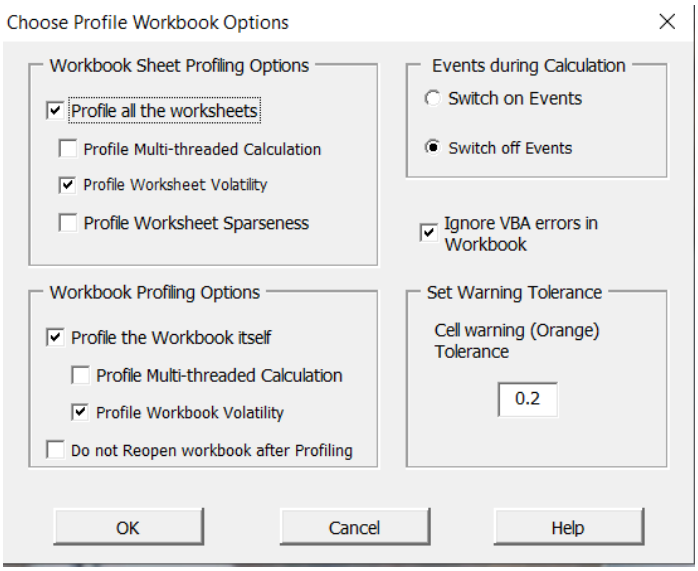
Workbook Calculation settings			
Mixed Mode		Other	
Full	TRUE	UpdateRemoteRefs	TRUE
Manual	FALSE	PrecisionAsDisplay	FALSE
Selected Sheets	FALSE	1904 DateSystem	FALSE
Restore after Open	FALSE	SaveXternalLinks	TRUE
Mixedmode on Open	TRUE	Book Calc Engine	191029
Accuracy Version	0	Compat Version	2

- **Mixed Mode** allows you to choose when any MixMode sheets will be calculated.
- **Full**: If True any MixMode worksheets will be recalculated at each full recalculation (Ctrl/Alt/F9).
- **Manual**: If True any MixMode worksheets will be recalculated at each Manual calculation (F9).
- **Selected Sheets**: If true then any MixMode sheets that are selected will be calculated when Sheet Calculate or Shift-F9 is pressed.
- **Restore After Open**: If True then FastExcel will automatically reset the calculation mode **after the workbook has been opened** to the mode that was active when the workbook was saved, or in Active Workbook Mode to the mode stored in the workbook
- **MixMode on Open**: If Excel is in Automatic mode then all the MixMode sheets in the workbook being opened will be recalculated. If Excel is in Manual mode any MixMode sheets will not be calculated when the workbook is opened unless this option is True.
- **Accuracy Version**: Controls whether certain functions use more accurate but slower algorithms.
- **Update Remote Refs**: If TRUE automatically update any remote references (DDE Links to other programs) whenever Excel recalculates.
- **Precision as Displayed**: If TRUE uses the displayed value of all cells rather than their real value as input to subsequent formulas.
  Be careful because this also permanently changes the values of numeric constants.
- **1904 Date System**: if True changes the starting date from which all dates are calculated from January 1 1900 to January 2 1904.
- **Save External Link Values**: If TRUE Excel saves values for links to external workbooks.

- **Book Calc Engine**: The version number of the calculation engine that last calculated the workbook (if not Excel 2000 or a later version then N/A). Colored orange if less than the Application calculation version
- **Compat Version**: the compatibility version currently in use for the profiled workbook.

Profile Workbook


Profile Workbook: Profile the active workbook and all its worksheets.



Because Excel calculates all open workbooks rather than just the active workbook you can get misleading calculation times if you have multiple workbooks open during Profile Workbook.

Choose Profile Workbook Options

- Because Profile Workbook can take a long time to execute these options allow you to include or omit various profiling options:
- Profile all the worksheets: if switched off the individual worksheets will not be profiled.
- Profile Multi-Threaded Calculation: If checked each worksheet will be calculated once using multi-threaded calculation and once without so that the multi-threaded calculation efficiency of each sheet can be determined.
- Profile Worksheet Volatility: If checked each worksheet will be calculated once using smart Recalc and once using Full Calculation so that the worksheet volatility can be determined.
- Profile Worksheet Sparseness: If checked the number of cells containing constants will be counted and the sparseness % of the real used range will be calculated.
- Profile the workbook itself: If switched off the workbook itself will not be profiled.
- Profile Multi-threaded Calculation: if checked the workbook will be calculated once using MTC and once without so that the multi-threaded calculation efficiency of the workbook can be determined.
- Profile Workbook Volatility: If checked the workbook will be calculated once using smart Recalc and once using Full Calculation so that the workbook volatility can be determined.
- Events during Calculation: Profile Workbook will by default attempt to prevent VBA events from occurring during the timed calculations. If your workbook contains event-handling VBA code you should consider whether to include or exclude it when running Profile Workbook.
- Ignore VBA Errors. Profiler will attempt to ignore VBA errors in the workbook being profiled.
- Set Warning Tolerance: a larger tolerance number reduces the number of orange warnings produced.

Profile Workbook produces information in several tables:

- Excel and Workbook environment information and calculation options.
- Worksheet statistics, calculation times, volatility and multi-threaded efficiency
- Workbook summary

Potential **bottlenecks** are highlighted in orange. The sensitivity of the orange highlighting can be controlled by setting FastExcel's Orange Tolerance (default 0.2).

FastExcel WorkBook Profile — **FastExcel V5 Profiler Build 825**

Physical Environment		
RAM(MB)	17,254	32,211
Swap File(MB)	24,787	34,259
Excel Memory(MB)	584	131,072
FileSize(K)	3,925	
Operating System	Windows 11 .26200 x64	
MHZ x Cores	3686 x 24	
Excel Version	Excel 2016.20527 x64	
XL Calc Engine	191029 Dyn Array	
Max Compat Version	2	

Profile: 14/11/2026 18:05 D:\Documents\XLEUC\XLVirtualGlobal2026\FastExcel_Demonstration_WkBk.xlsm

Workbook Settings

Shared		Decimal Places		Protection	
Shared	FALSE	Fixed	FALSE	Structure	FALSE
Changes	TRUE	Places	2	Worksheets	FALSE
Saved	FALSE	Backup	FALSE	Password	FALSE
Format	XMLWorkbookMacro	Styles	55	Views	0

Environment Counts

Temp	193	XLB/Qat (K)	10	Com Addins	
Workbooks	2	VBE	8	Excel Addins	7

Calculation Options

Excel Calc Settings				Workbook Calculation settings			
Workbook Mode	All	Multi-threaded Calc		Mixed Mode		Other	
Current Mode	Automatic	On	TRUE	Full	TRUE	UpdateRemoteRefs	FALSE
Before Save	FALSE	Mode	Automatic	Manual	FALSE	PrecisionAsDisplay	FALSE
Initial Mode	First Workbook	Threads	24	Selected Sheets	FALSE	1904 DateSystem	FALSE
Iteration	FALSE	Force Full Calc		Restore after Open	FALSE	SaveXternalLinks	TRUE
Max Iters/change	100/0.001	Full/Recalc	Recalc	Mixedmode on Open	TRUE	Book Calc Engine	191029
Format Stale Values	FALSE			Accuracy Version	0	Compat Version	2

Worksheet Profiles (Select Sheet name and click Drilldown to profile sheet formulas)

Sheetname	Cell Counts Used	Constants	Formulas	Unique Formulas	CSE Formulas	CondFmt	What-If Tables	Calc Time (Millisec) Single-Thread	Re Calc	Full Calc	Sheet Volatile %	Multi-Thread Efficiency	MicroSecs /Formula	Used Range %Waste	%Sparse	Page Breaks	Shapes Count	HypLinks Count	Sheet CalcMode
Sum_Ifs	276,621		115	6	0	0	0	3,840.4	1.2	1,113.6	0.1%	3.4	9,683.8	22%		FALSE	0	0	Book
Count_Uniques	88,000		10,617	11	0	0	0	457.0	1.0	218.5	0.5%	2.1	20.6	28%		FALSE	0	0	Book
Lookups	2,200,044		10,004	6	0	0	0	130.2	0.2	60.7	0.3%	2.1	6.1	18%		FALSE	0	0	Book
Count_GT_100	29,820		10,005	11	0	0	0	1,734.1	20.3	23.9	85.3%	72.7	2.4	19%		FALSE	0	0	Book
Period_To_Date	100,010		19,995	6	0	0	0	166.5	1.1	11.3	9.9%	14.8	0.6	20%		FALSE	0	0	Book
DataTables_Ex02	130		10	2	1	0	1	0.0	0.0	3.0				62%		FALSE	0	0	Book
Formulas	98		9	9	0	9	0	0.0	0.6	2.2	25.7%			0%		FALSE	0	0	Book
Tests	494		12	8	0	0	0	0.8	0.8	2.1	39.7%	0.4	66.9	0%		FALSE	0	0	Book
Log_WorkSheet1	48		10	1	0	0	0	0.0	0.0	1.5				19%		FALSE	0	0	Book
Log_WorkSheet3	27,675		503	3	0	0	0	0.3	0.0	1.4				0%		FALSE	0	0	Book
Examples	360		8	6	0	0	0	0.0	0.6	1.4	47.1%			71%		FALSE	0	0	Book
Tax_Data_Worksheet	4,032		3	3	0	48	0	0.0	0.0	1.2				23%		FALSE	1	0	Book
BetaTest	7		0	0	0	0	0	0.0	0.0	0.3				0%		FALSE	0	2	Book
Log_WorkSheet2	48		0	0	0	0	0	0.0	0.0	0.3				58%		FALSE	0	0	Book
Other								**0.4**	**0.2**	**0.4**									
Totals	**2,729,607**		**51,291**	**72**	**1**	**57**	**1**	**6,329.3**	**26.1**	**1,446.5**	**1.8%**	**4.4**	**28.2**	**18.7%**			**268**	**2**	

Workbook Summary

Sheets Count	Cell Counts Used	Constants	Formulas	Unique Formulas	CSE Formulas	CondFmt	What-If Tables	Calc Time (Millisec) Single-Thread	Re Calc	Full Calc	Book Volatile %	Multi-Thread Efficiency	MicroSecs /Formula	Used Range %Waste	%Sparse	Pivot Cache(K)	XL Links Count	Names Count	Open Time (Secs)
17	2,729,607	0	51,291	72	1	57	1	7,923.9	20.6	4,406.8	0.5%	1.8	85.9	18.7%	97.7%	0	0	87	0.63

The Worksheet Profiles Table.

The Worksheet Profiles Table shows the cell counts, calculation times and Memory Used for each worksheet. The table is only produced if the Profile All the Worksheets option is checked in the Profile Workbook Options.

The worksheets are sorted in descending sheet calculation time.

The Worksheet Profiles Table: Cell Counts Columns

Cell counts show you the key size statistics for each worksheet.

Sheetname	Cell Counts Used	Constants	Formulas	Unique Formulas	CSE Formulas	CondFmt	What-If Tables
Sum_Ifs	276,621		115	6	0	0	0
Count_Uniques	88,000		10,617	11	0	0	0
Lookups	2,200,044		10,004	6	0	0	0
Count_GT_100	29,820		10,005	11	0	0	0
Period_To_Date	100,010		19,995	6	0	0	0
DataTables_Ex02	130		10	2	1	0	1
Formulas	98		9	9	0	9	0
Tests	494		12	8	0	0	0
Log_WorkSheet1	48		10	1	0	0	0
Log_WorkSheet3	27,675		503	3	0	0	0
Examples	360		8	6	0	0	0
Tax_Data_Worksheet	4,032		3	3	0	48	0
BetaTest	7		0	0	0	0	0
Log_WorkSheet2	48		0	0	0	0	0
Other							
Totals	**2,729,607**		**51,291**	**72**	**1**	**57**	**1**

The Cell Counts columns show counts of the number of cells for:

- **Used: The Used Range**: The rectangular table of cells that Excel considers used: Excel considers cells with formatting, data or formulas used. The Used Range is not automatically reset when cells are deleted or cleared.
  You can clean up the Used Range with Clean Workbook.
- **Constants**: the number of constants on this worksheet.
- **Formulas:** the number of formulas on this worksheet.
- **Unique Formulas:** the number of different formulas on this worksheet.
- **CSE Formulas:** the number of array entered (Control-Shift-Enter) formulas on this worksheet.
- **CondFmt:** the total number of conditional formats on this worksheet. Colored orange if the average number of conditional formats per conditionally formatted cells is > 3.
- **What-If Tables**: the number of iterative what-if data tables on this worksheet.

The Worksheet Profiles Table: Calculation Times.

Calc Time (Millisec)			Sheet	Multi-Thread	MicroSecs
Single-Thread	Re Calc	Full Calc	Volatile %	Efficiency	/Formula
3,840.4	1.2	1,113.6	0.1%	3.4	9,683.8
457.0	1.0	218.5	0.5%	2.1	20.6
130.2	0.2	60.7	0.3%	2.1	6.1
1,734.1	20.3	23.9	85.3%	72.7	2.4
166.5	1.1	11.3	9.9%	14.8	0.6
0.0	0.0	3.0			
0.0	0.6	2.2	25.7%		
0.8	0.8	2.1	39.7%	0.4	66.9
0.0	0.0	1.5			
0.3	0.0	1.4			
0.0	0.6	1.4	47.1%		
0.0	0.0	1.2			
0.0	0.0	0.3			
0.0	0.0	0.3			
0.4	**0.2**	**0.4**			
6,329.3	**26.1**	**1,446.5**	**1.8%**	**4.4**	**28.2**

In the actual FastExcel output these columns are adjacent to the cell counts columns rather than as shown above.

Calc Time (Millisec); the elapsed calculation times in milliseconds.

Use this table to identify the sheets that use the most calculation time, any sheets with significant volatility or MTC inefficiency and the sheets that have the most expensive formulas.

*Colored orange if >orange tolerance * total sheet calc time and > (1+orange tolerance)*minimum sheet overhead and > orange tolerance * 500*

Single-Thread: Time in milliseconds for a Range.Calculate of the real used range in the worksheet, not using multi-threaded calculation, minus the time taken to range calculate an empty worksheet (empty sheet overhead).

Recalc: The time taken to recalculate the sheet after the workbook has been calculated, minus the time taken to recalculate an empty sheet (empty sheet overhead).
Multi-threaded calculation is used if enabled.
For most workbooks this will give a minimum worksheet calculation time because only the minimum set of formulas (mostly volatile formulas and their dependents) will be calculated.

Full Calc: The time taken to calculate all the formulae on the sheet, minus the time taken to fullcalc an empty sheet (empty sheet overhead).
Multi-threaded calculation is used if enabled.
Colored orange if > (1+ orange tolerance) *average sheet full calc time and >orange tolerance *500

Sheet Volatile %: The Ratio of Sheet Recalc time to Sheet Full Calc time.
Measures how many of the formulae on the sheet are volatile.
Colored orange if >orange tolerance * .125

*Colored orange if >(1+ orange tolerance) * average and > orange tolerance * 500*

Multi-Thread Efficiency: A measure of the efficiency of multi-threaded calculation at sheet level. The ratio of Range calculation (single-threaded) time to Full Calculation (multi-threaded) time.

Other: The three times shown in the Other row are the times taken to calculate an empty worksheet. These times are used as standard calculation overheads and are subtracted from the sheet calculation times.

Sheet % Volatile: The ratio of Recalc time to Full Calc time, excluding standard calculation overhead. When you have a high workbook volatility % you can use the sheet % volatile to determine which sheets are causing the problem.

Microseconds per Formula: calculated from the smaller of used range calc time and sheet calc time. A high value may indicate bottleneck formulas.

Worksheet Profiles Table (2)

Used Range %Waste	%Sparse	Page Breaks	Shapes Count	HypLinks Count	Sheet CalcMode
22%		FALSE	0	0	Book
28%		FALSE	0	0	Book
18%		FALSE	0	0	Book
19%		FALSE	0	0	Book
20%		FALSE	0	0	Book
62%		FALSE	0	0	Book
0%		FALSE	0	0	Book
0%		FALSE	0	0	Book
19%		FALSE	0	0	Book
0%		FALSE	0	0	Book
71%		FALSE	0	0	Book
23%		FALSE	1	0	Book
0%		FALSE	0	2	Book
58%		FALSE	0	0	Book
18.7%			**268**	**2**	

In the actual FastExcel output these columns are adjacent to the Calc Times columns rather than as shown above.

Unused Cells:
Eliminate the waste *by using FastExcel's Workbook Cleaner*

%Waste: The cells in the Used Range (formatted cells, constants, formulas, deleted/cleared cells) beyond the real Used Range (formulas and constants), as a percentage of the Used Range:
Colored orange if > orange tolerance * 10000 cells. Significant waste can use additional memory and calculation time.

%Sparse: Cells not containing formulas or constants, as a percentage of the used range.
Colored Orange if > orange tolerance * 10000 cells.

On sheet notes are counted as shapes

Shapes Count: Count of the number of shapes in the Drawing layer for this sheet. Includes hidden and zero-height shapes. Use FastExcel's Clean Workbook command to remove zero-height shapes.

Colored **orange** if any of the shapes are linked by formula (slows calculation and editing).

HyperLinks Count: Count of the number of Hyperlinks on this sheet. The hyperlinks can be internal or external to the workbook. Large numbers of hyperlinks can be slow.

Sheet Calc Mode: Shows the MixMode calculation mode for the worksheet.

- **Book:** The Sheet will be calculated whenever the Workbook calculates
- **F:** The Sheet will be calculated at every Full workbook calculation.
- **M:** The Sheet will be calculated at every Manual workbook re-calculation.
- **S:** Shift-F9/worksheet Calculate will calculate the worksheet when it is selected

Either Book or any combination of F M and S is allowed.

The Workbook Summary Table.

Volatility% and MTC are key measures of calculation efficiency

The ***Workbook Summary*** table shows the cell counts, workbook calculation times, workbook volatility, workbook overhead, bytes per cell, and names count, for the profiled workbook. The workbook Summary table is only produced if the Analyze the Workbook option is checked in the FastExcel Settings.

The Workbook Summary Table Cell Counts.

Sheets	Cell Counts			Unique	CSE		What-If
Count	Used	Constants	Formulas	Formulas	Formulas	CondFmt	Tables
17	2,729,607	0	51,291	72	1	57	1

This shows totals of the counts and what-if tables for all the worksheets.

Sheets Count: the total number of sheets in the workbook including Chart, Dialog, XLM Macro sheets as well as normal worksheets.

The Workbook Summary Table Calculation Times.

Workbook Summary

Calc Time (Millisec)			Book	Multi-Thread	MicroSecs
Single-Thread	Re Calc	Full Calc	Volatile %	Efficiency	/Formula
7,923.9	20.6	4,406.8	0.5%	1.8	85.9

- **Single-Thread:** Total Time in milliseconds for Range.Calculate for the worksheets. The calculation is single-threaded.
- **Recalc**: milliseconds to recalculate (F9) the workbook immediately after a full calculation.
- **FullCalc**: milliseconds to fully calculate the workbook (Ctrl/Alt/F9).

Keeping Workbook Volatility low is often an excellent way to improve response time.
Colored orange if > orange tolerance * 0.125

Book Volatile % is Recalculate Time as a percentage of full calculation time: a high value means one or more of:

- High proportions of the formulas are Volatile
- Evaluating the dependency trees takes a long time.
- The workbook is too complex for Excel to build a dependency tree (status bar **always** shows Calculate and there are no circular references).
- ForceFull calculation is true

Multi-Thread Efficiency: A measure of the efficiency of multi-threaded calculation at book level. The ratio of the sum of range calculation (single-threaded) times for each sheet to Full Calculation (multi-threaded) time for the book.

MicroSecs per Formula: Microseconds per formula based on full calculation time.

The Workbook Summary Table Used Range etc.

Used Range		Pivot	XL Links	Names	Open
%Waste	%Sparse	Cache(K)	Count	Count	Time (Secs)
18.7%	97.7%	0	0	87	0.63

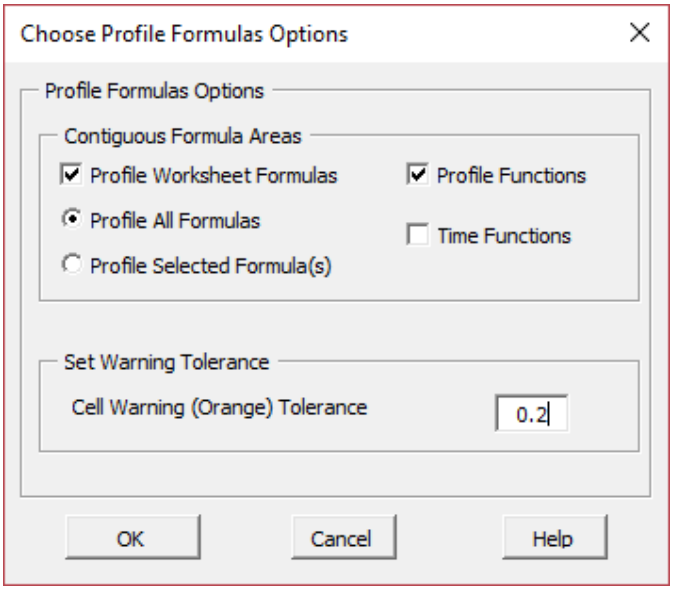
The actual FastExcel workbook summary output includes cell count column and calculation times. These columns are not shown in the above figure.

- **%Waste:** Cells in the UsedRange Outside the last Row and Column containing data or formulas, as a percentage of the cells in the usedrange.
- **%Sparse:** Sparseness of the UsedRange: Cells inside the used range, but not containing formulas or constants, as a percentage of the used range.
- **Pivot Cache (K):** Memory Used by Pivot Table Caches in this workbook in K bytes.
- **XLLinks Count:** The Number of Excel workbooks that are linked to by the profiled workbook.
- **Names Count**: count of the defined Names in the workbook.
- **Open Time (Secs)**: The time taken to reopen the workbook.

Profile Formulas and Functions

This command allows you to profile and prioritize each unique formula on a worksheet and to analyse the functions used. Use this to drill down on a problem worksheet or worksheet area to the formulas you need to change.

You can use this analysis to determine which formulas and functions are volatile and which formulas are single-threaded.



You can choose to profile one or both of:

- Worksheet formulas: either all formulas or the formulas in a selected area.
- Functions used: either in all formulas or the formulas in a selected area.

You can also request to time the individual functions, and set the warning tolerance level.

Worksheet Formulas Profile Table

This table contains one row for each unique formula, sorted in descending calculation time order.

Single-Threaded WorkSheet Formulas Profile for							*Count_Uniques*
Formula			Formula	Microsecs	Total Millisecs		Formula %
Address	Flags	Unique Formulas	Count	/Formula	This Formula	Area/Sheet	of Sheet Time
$D$7	B-N-M	=SUM(IF(LEN(A6:A10605)>0,1/COUNTIF(A6:A10605,A6:A10605)))	1	185,538.54	185.54	389.37	48%
$D$11	B-N-M	=SUMPRODUCT((A6:A10605<>"")/COUNTIF(A6:A10605,A6:A10605&""))	1	182,684.64	182.68	389.37	47%
$D$15	O-?-S	=countu(A6:A10605)	1	11,467.74	11.47	389.37	3%
$D$19	X-N-M	=COUNTDISTINCTS(A6:A10605,2)	1	5,974.04	5.97	389.37	2%
$B$6:$B$10605	B-N-M	=IF(AND(A6<>"",A6<>A5),1,0)	10600	0.15	1.56	389.37	0%
$D$27	B-N-M	=COUNTA(UNIQUE(FILTER(A6:A10605,A6:A10605<>"")))	1	630.74	0.63	389.37	0%
$D$6,$D$10,$D$14,$D$18,$D$22,$D$26,D	B-N-M	=FORMULATEXT(D7)	8	60.67	0.49	389.37	0%
$D$31	B-N-M	=SUM(IF(UNIQUE(A6:A10605)<>"",1,0))	1	370.64	0.37	389.37	0%
$D$29	B-N-M	=LET(U,UNIQUE(A6:A10605),COUNTA(FILTER(U,U<>0)))	1	365.74	0.37	389.37	0%
$A$4	B-N-M	=FORMULATEXT(B7)	1	147.74	0.15	389.37	0%
$D$23	B-N-M	=SUM(B6:B10605)	1	144.04	0.14	389.37	0%

Formula Address: Gives the addresses of the first few locations where this formula is used. **If the formula is used in many different locations not all locations will be shown.**

Flags: One or more of the following formula flags:

- D – Spilling Dynamic Array Formula
- L – Formula contains a Lambda
- A - This is a CSE Array-entered Formula
- B - This formula contains a Built-in, native Excel function
- X - This formula contains a registered XLL function
- O – Other: This formula contains a JS, VBA, Automation, or XLM function
- N - This formula is not Volatile
- V - This formula is Volatile
- ? - Cannot tell if this formula is Volatile or not
- M - This formula is Multi-threaded
- S - This formula is probably Single-threaded

Most of the formulas in this example contain functions that are Built-in to Excel (B), are Non-volatile (N) and are Multi-threaded (M): so the flag is B-N-M.
VBA, Automation and XLM functions are always single-threaded but it is not currently possible to determine within the analysis if they are volatile or not, so they are flagged as? – flag is O-?-S
COUNTDISTINCTS is a registered, non-volatile, multithreaded XLL function so is flagged as X-N-M

The analysis does not currently detect Defined Names containing volatile functions or single-threaded functions.

Unique Formulas: Formulas that are copied and contain relative or absolute addresses are treated as being the same (they have the same R1C1 formula).

Formula Count: The total number of occurrences of this formula within the area analysed.

Microsecs per Formula: The average time to calculate (single-threaded) a single occurrence of this formula

Total Millisecs for this formula: Time in Milliseconds to calculate all the occurrences of this formula in this Sheet/Area using a single thread. =Time/Formula * Count
Sorted largest first.

Total Millisecs for this Area/Sheet: Total Millisecs for a single-threaded calculate of all the Formulas in this Sheet/Area.

Formula % of Sheet Time: Total single-threaded calculation time for this formula as a percentage of the time for a single-threaded calculation of all formulas.

Function Profile Table

This table shows an analysis of the functions used, with optionally some indicative calculation timing.

Local Function Name	Function Flags	English Function Name	Function Count
FORMULATEXT	B-N-M	FORMULATEXT	9
SUM	B-N-M	SUM	3
IF	B-N-M	IF	10602
LEN	B-N-M	LEN	1
COUNTIF	B-N-M	COUNTIF	2
SUMPRODUCT	B-N-M	SUMPRODUCT	1
countu	O-?-S	countu	1
COUNTDISTINCTS	X-N-M	COUNTDISTINCTS	1
COUNTA	B-N-M	COUNTA	2
UNIQUE	B-N-M	UNIQUE	3
FILTER	B-N-M	FILTER	2
LET	B-N-M	LET	1
AND	B-N-M	AND	10600

Local Function Name: The name of the function in the current User Interface Language
Function Flags: Flags that describe the attributes of the function.

- L- This function is a Lambda function
- B-This function is a native, Built-in Excel function
- X- This function is registered as an XLL function
- O- Other: This function is a VBA or Automation function
- V- This function is Volatile
- ?- Cannot tell if this function is volatile or not
- N- This function is Not volatile
- S- This function is probably single-threaded
- M- This function is multi-threaded
- Y - This function is an Asynchronous function

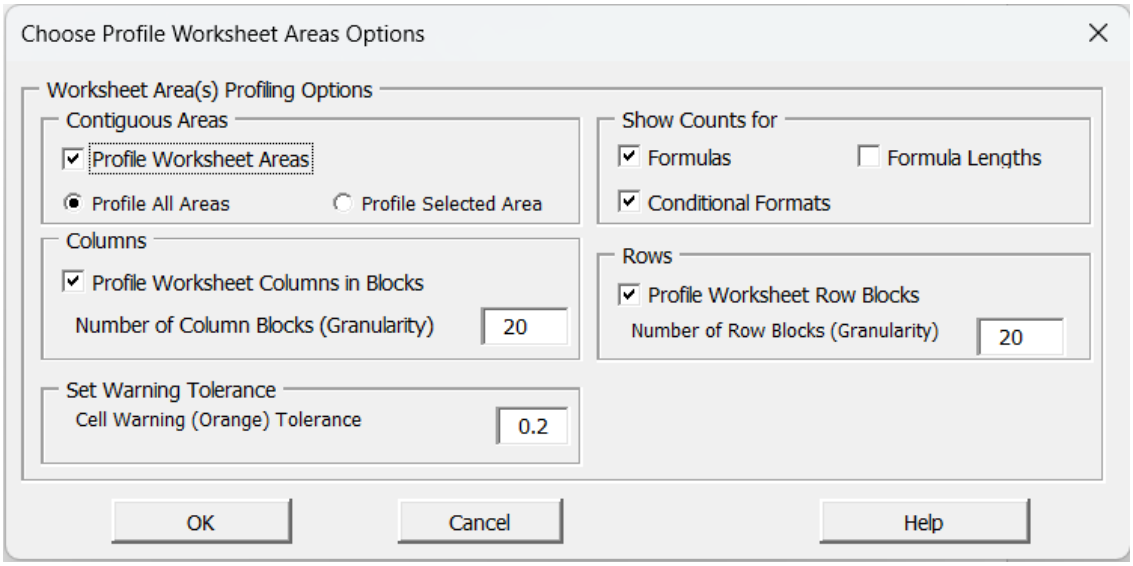
Function Count: The total number of occurrences of this function.
Average Millisecs per function: The Average Time to calculate this function In Milliseconds:
(Note that there may be other function calls etc. embedded inside this function, and the function may also be used in several different formulas)
Total Millisecs This Function: Time in Milliseconds to calculate all the occurrences of this function in this Sheet/Area using a single thread. =Time/Function * Count
Note: Timings for functions are indicative only and may not be accurate. Function timing is most accurate when the function returns only a single value and when each occurrence of the function is used in an identical formula.

Profile Worksheet Areas


Profile Worksheet Areas: Shows the Profile Header block, and profiles each rectangular block of formulas and, optionally, each column and each block of rows on each of the selected worksheets.

The Profile Worksheet Areas command shows you a form to allow you to choose between options.

Choose Profile Worksheet Areas Options



You can choose to profile one or more of:

- The contiguous areas on the worksheet(s): either all or the selected areas.
- The Worksheet Columns within the areas: choose the number of groups of columns you want – larger number gives finer granularity but takes longer.
- The Worksheet Rows within the areas: choose the number of groups of rows you want – larger number gives finer granularity but takes longer.

You can also choose to show counts of formulas and formula lengths – but this uses more time.

The warning tolerance (Orange tolerance) controls the sensitivity of flagging the results as warnings. A larger number means more warnings.

Profile Selected Area

After reviewing the output from Profile Worksheet Areas you can select one or more of the problem areas on the worksheet that was profiled and use profile Selected Area to drill down into that area.

Worksheet Areas Profile Table

Use the tables to identify Bottleneck areas on the worksheet.

There will be three **Worksheet Areas Profile** Tables for each of the selected Worksheets.

All calculation times in these tables are single-threaded calculation times.

- Cell counts of formulas, array cells and conditional formats.
- Calculation times for the area, column and the sheet.
- Total, average and maximum formula lengths.

Each table is sorted in descending calculation time sequence.

The first table **profiles** each separate contiguous **rectangular area of formulas** on the worksheet:

WorkSheet Areas Profile (Use DrillDown on this Sheet to Profile a Formula Block) for Worksheet								
Address	Count of		Cond Format Rules		Calc Millisecs		MicroSecs	Area/Col %
Area	Formula	ArrayCells	Count	Average	Area/Col	Sheet	/Formula	of Sheet
$P$5:$P$26	22	0	0		3,373.44	3,453.00	153,338.36	97.7%
$Q$7:$Q$28	22	0	0		36.08	3,453.00	1,639.85	1.0%
$R$9:$R$30	22	0	0		12.25	3,453.00	556.67	0.4%
$T$13:$T$34	22	0	0		10.03	3,453.00	456.01	0.3%
$S$11:$S$32	22	0	0		8.34	3,453.00	378.88	0.2%
$Q$4	1	0	0		3.17	3,453.00	3,172.10	0.1%
$U$12	1	0	0		0.27	3,453.00	265.00	0.0%
$T$10	1	0	0		0.24	3,453.00	242.50	0.0%
$R$6	1	0	0		0.24	3,453.00	236.40	0.0%
$S$8	1	0	0		0.22	3,453.00	223.90	0.0%

The second table profiles the columns: the number of column groups should be no more than the number specified in the options. The column groups are sorted in descending calculation time sequence.

Column							
$P$5:$P$26	22	0	0	3,434.10	3,453.00	156,095.41	99.5%
$Q$4,$Q$7:$Q$28	23	0	0	39.71	3,453.00	1,726.45	1.1%
$R$6,$R$9:$R$30	23	0	0	11.22	3,453.00	487.69	0.3%
$T$10,$T$13:$T$34	23	0	0	10.15	3,453.00	441.32	0.3%
$S$8,$S$11:$S$32	23	0	0	8.27	3,453.00	359.42	0.2%
$F$1:$F$10300	0	0	0	0.65	3,453.00		0.0%
$B$1:$B$10300	0	0	0	0.61	3,453.00		0.0%
$D$1:$D$10300	0	0	0	0.56	3,453.00		0.0%
$A$1:$A$10300	0	0	0	0.49	3,453.00		0.0%

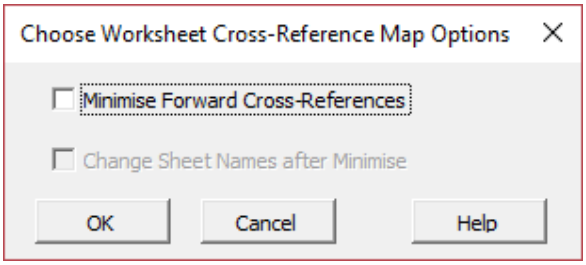
The third table profiles the rows in the same way as the columns.

RowBlock							
$Q$4,$R$6,$S$8,$T$10,$U$12,$P$5:$P$26,$Q$7:$Q$28,$R$9:$R$30,$S$11:$S$32,$T$13:$T$34	115	0	0	3,442.99	3,453.00	29,939.03	99.7%
$A$516:$U$1030	0	0	0	3.36	3,453.00		0.1%
$A$1031:$U$1545	0	0	0	0.64	3,453.00		0.0%
$A$5666:$U$6180	0	0	0	0.35	3,453.00		0.0%
$A$5151:$U$5665	0	0	0	0.33	3,453.00		0.0%
$A$6696:$U$7210	0	0	0	0.32	3,453.00		0.0%

Map Worksheet Cross-references.


Map Worksheet Cross-references produces a tabular map showing how the references between worksheets.

Worksheet Cross-references occur when a formula on one worksheet refers to another worksheet.



Optionally you can also request that the worksheets are re-sequenced to minimize the number of worksheet cross-references: **but note that with current Excel versions this will generally have little or no impact on calculation times.**

Worksheet Calculation Sequence Forward Cross-reference Tables.

The output is one or two tables of worksheet cross-reference counts, with forward cross-references highlighted in orange:

The first table shows forward cross-reference counts using the **Current Worksheet Calculation Sequence**.

Worksheet Cross-reference Map: Forward CrossReferences=3628							
Sheet Refers To-->	1DASHBOARD	2CtryCalc	3CityCalc	4LaneCalc	5Coverage	6CityCtryContVols	7Summary
1DASHBOARD		147	2,315				
2CtryCalc	420			5	106		
3CityCalc	29,001	1,155			530	525	
4LaneCalc	154,505		454,250				
5Coverage			210	2,730			
6CityCtryContVols	1,995		1,050	1,785	3,570		
7Summary	640	440	220		110	346	

The second table shows **the Optimized Worksheet Calculation Sequence**, and its effect on the forward cross-reference counts.

Optimised WorkSheet Calculation Sequence: Forward CrossReferences=72						
Sheet Refers To-->	A0_RespondentData	A1_SegmentNames	A2_UniVerse	A3_Sectors	A4_Respondents	A5_Exp
A0_RespondentData						
A1_SegmentNames						
A2_UniVerse	72				72	
A3_Sectors	180		36			
A4_Respondents	2,505	501	2,004	1,336		
A5_Explanation						
A6_ProductNames						
A7_ProductData	533					
A8_Counts	1	2	1			
A9_Products	20,787	1,599	5,330		6,396	
B0_Answers					4,950	

Worksheet Forward Cross-reference Tables: Sheet Refers-To and Referenced-By:

Rows show Refers-To counts — Reading a row ***across*** the table shows you the ***refers-To*** counts.

Columns show Referenced-By counts

Worksheet Forward Cross-reference Tables: Color-Coding:

Orange cells show problem areas — **Orange** highlighted cells show forward worksheet references. These are the most common cause of excessive workbook overhead.

Bright green cells on the diagonal are empty because **FastExcel** does not count references within a sheet.

Light green columns show worksheets that are not r***eferenced-by*** any other sheet. These are typically report sheets: for example, the Summary sheet.

Light green rows show worksheets that do not r***efers-to*** any other sheet. These are typically sheets containing the input data.

Using the optimized Sheet calculation sequence:

Changing the worksheet names changes the worksheet calculation sequence — If FastExcel has significantly reduced the forward references you will be asked if you want to automatically change the sheet names.

NOTE: In Excel97 and Excel2000 Excel's worksheet calculation sequence is the alphabetically sorted worksheet name sequence rather than the sequence of the worksheets in the workbook.
In Excel 2002 and later versions the worksheet calculation sequence is generally not important to calculation time.

Circular Worksheet Cross-reference Paths table

If **FastExcel** can't find a worksheet calculation sequence with zero forward cross-references it's probably because the workbook contains circular cross-reference paths.

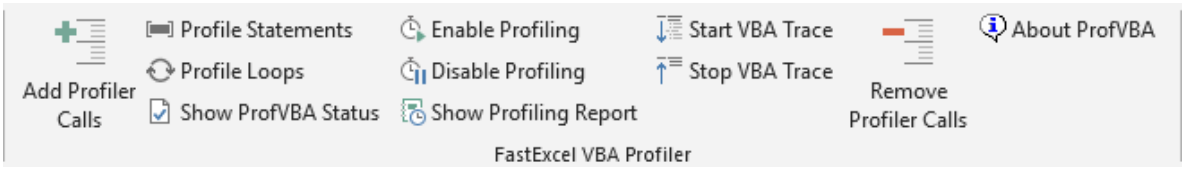
Circular WorkSheet CrossReference Paths

Path Length	Forward Refs	Total Refs	First	Second	Third	Fourth	Fifth
3	72	2076	UniVerse	Respondents	UniVerse		
4	72	1444	UniVerse	Respondents	Sectors	UniVerse	

The Path Length column shows the number of worksheets involved in the circular path.
In complex workbooks there may be thousands of unique circular paths. **FastExcel** is designed show you the shorter paths first, and will not attempt to find paths containing more than 6 sheets.

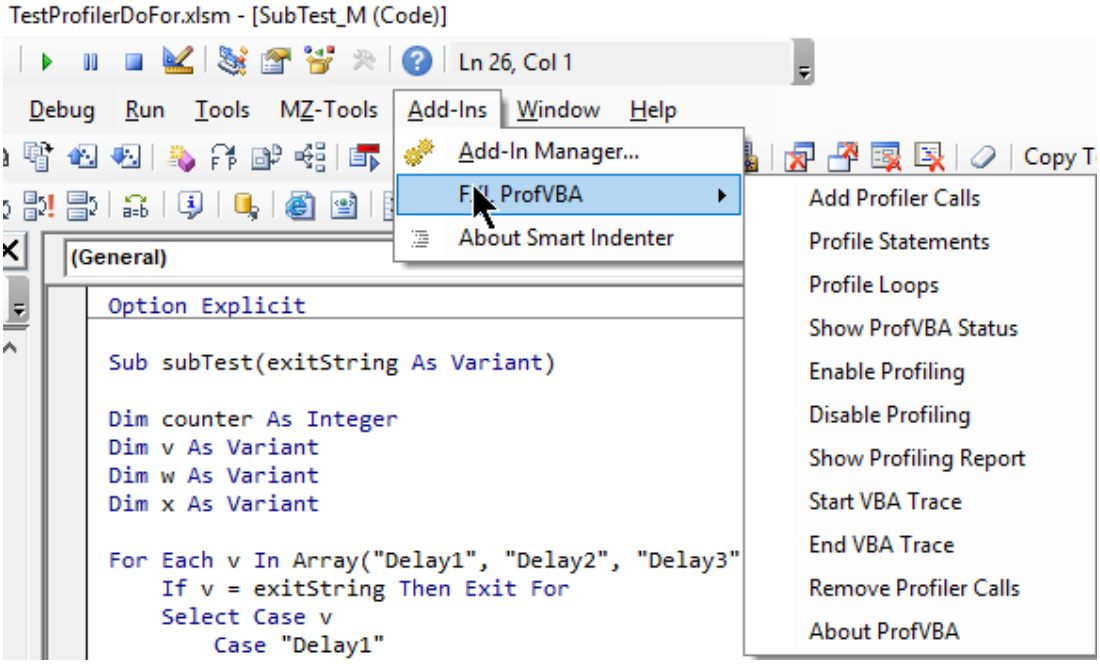
FastExcel VBA Profiler

ProfVBA Overview



FastExcel VBA Profiler works by adding timing and tracing calls to the projects and procedures you want to profile, so that profiling statistics are accumulated when you run the VBA.

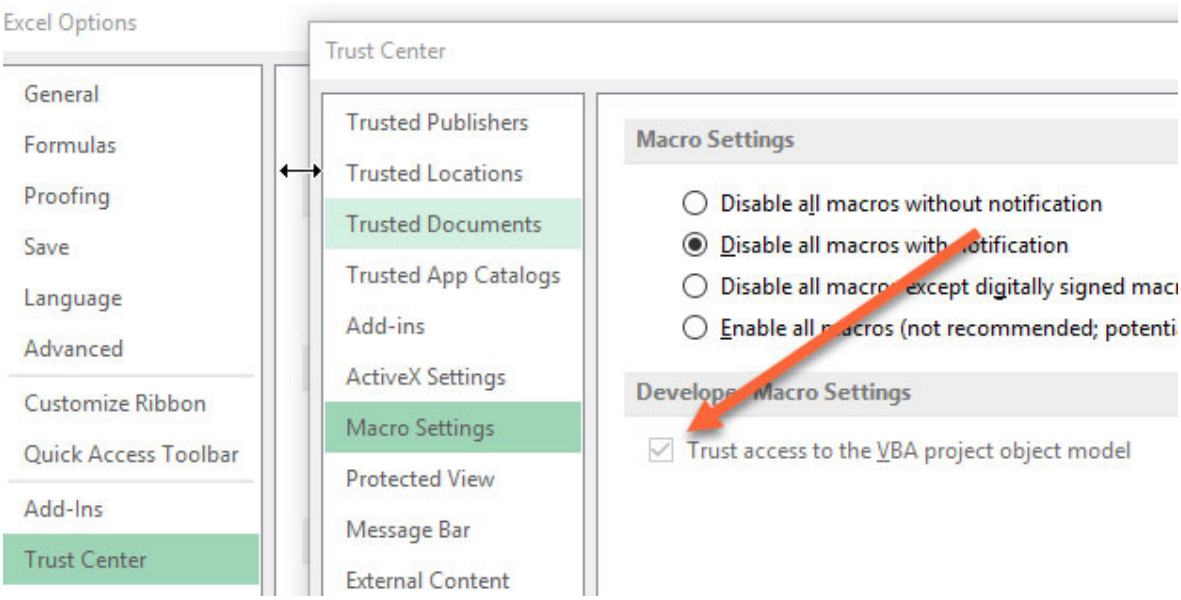
These ProfVBA commands are also available in the VBE IDE from the ProfVBA Toolbar and the Add-Ins menu item.



Use the ProfVBA toolbar when working with particular procedures open in the VBE to add profiling calls around selected blocks of VBA statements or around all the loops in particular procedures.



Note: Adding Profiling Calls requires "Trust Access to the VBA Project Object Model" to be checked.



ProfVBA Acknowledgements

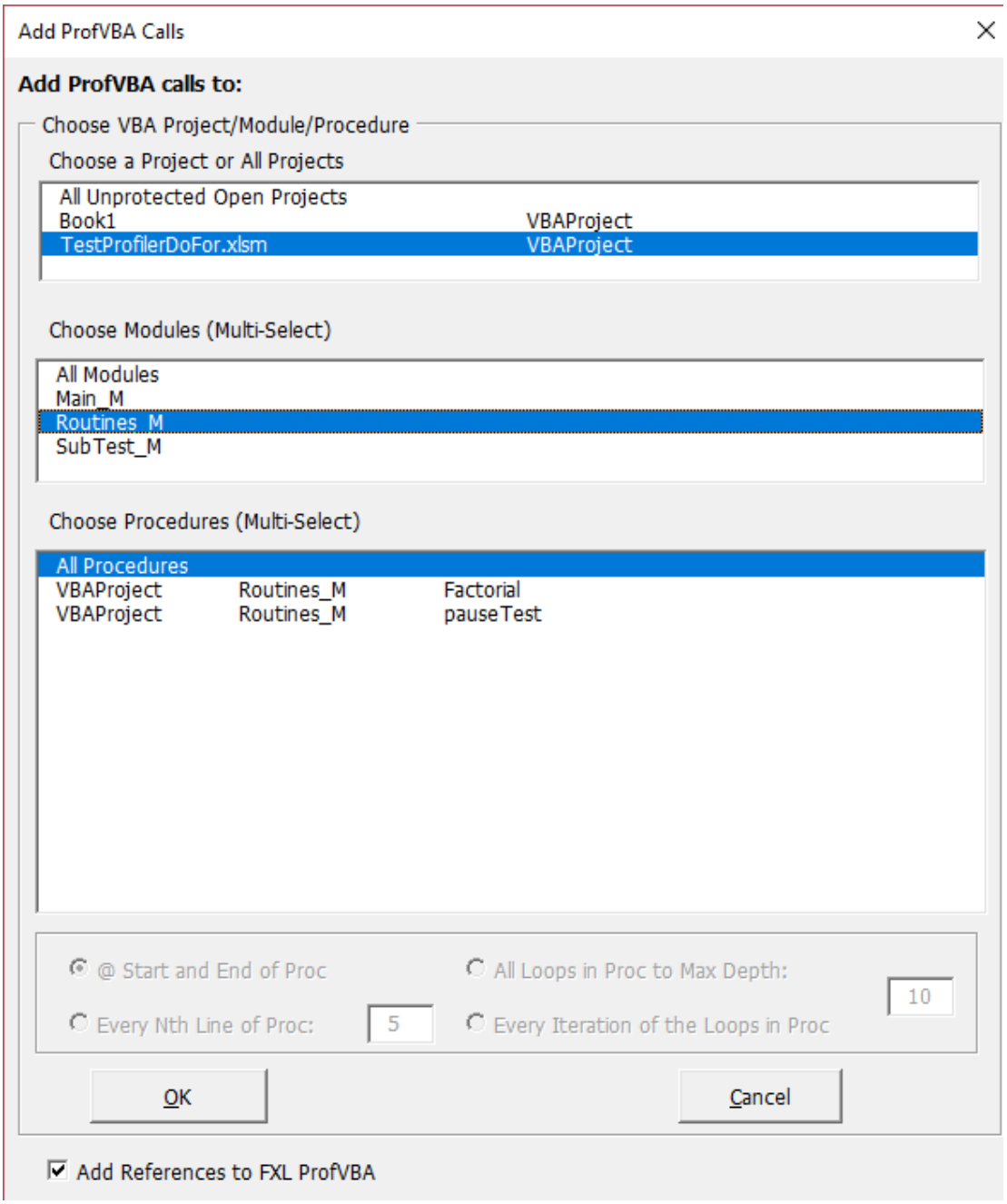
Many thanks to **Stephen Bullen and Rob Bovey** for their help with ProfVBA: see pages 546-500 of "Professional Excel Development" ISBN 0-321-50879-3.

And many thanks also to **Don Soloway** who extended ProfVBA to handle profiling of Loops and Iterations of Loops and added the Inline Command options.

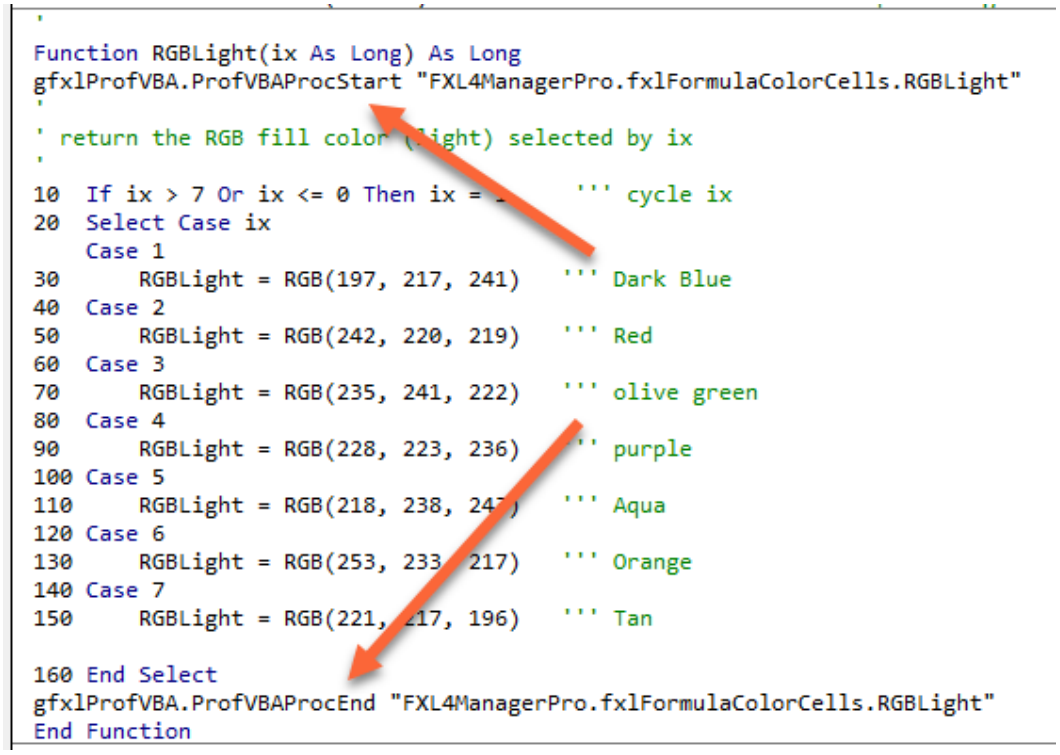
Add Profiler Calls

Add Profiler Calls allows you to control which projects, modules and procedures to profile.

Selecting a Project will show all the modules in that project.
Selecting one or more Modules will show all the Procedures in those Modules.



Note: Profiler calls can only be added to unprotected projects
Clicking OK adds calls to the FXL VBA Profiler at the start and end of each procedure.



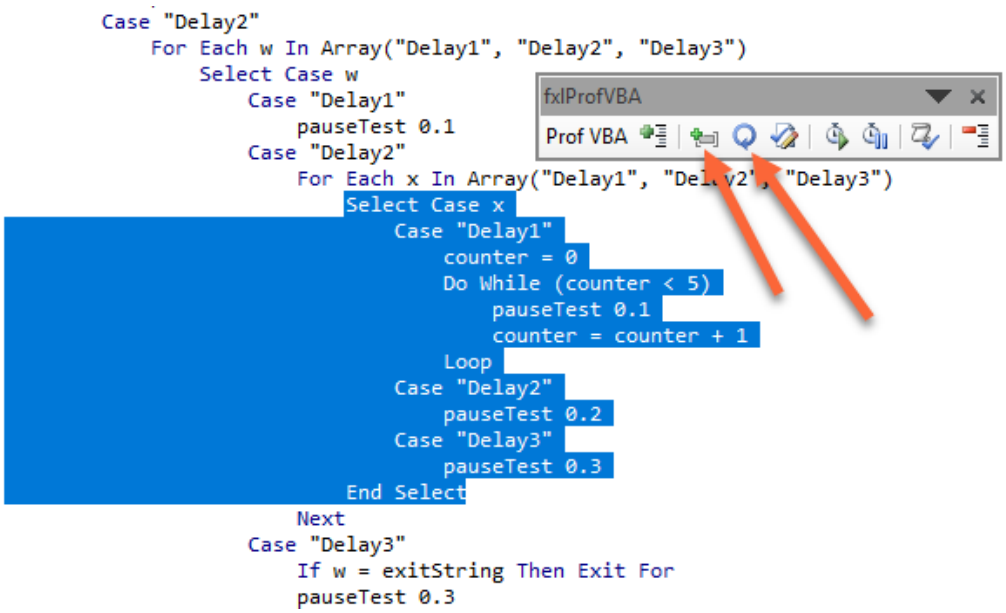
```
'
Function RGBLight(ix As Long) As Long
gfxlProfVBA.ProfVBAProcStart "FXL4ManagerPro.fxlFormulaColorCells.RGBLight"
'
' return the RGB fill color (light) selected by ix
'
10  If ix > 7 Or ix <= 0 Then ix =      ''' cycle ix
20  Select Case ix
    Case 1
30      RGBLight = RGB(197, 217, 241)   ''' Dark Blue
40  Case 2
50      RGBLight = RGB(242, 220, 219)   ''' Red
60  Case 3
70      RGBLight = RGB(235, 241, 222)   ''' olive green
80  Case 4
90      RGBLight = RGB(228, 223, 236)   ''' purple
100 Case 5
110     RGBLight = RGB(218, 238, 243)   ''' Aqua
120 Case 6
130     RGBLight = RGB(253, 233, 217)   ''' Orange
140 Case 7
150     RGBLight = RGB(221, 217, 196)   ''' Tan

160 End Select
gfxlProfVBA.ProfVBAProcEnd "FXL4ManagerPro.fxlFormulaColorCells.RGBLight"
End Function
```

Adding Drill-Down Profiler Calls using the Ribbon

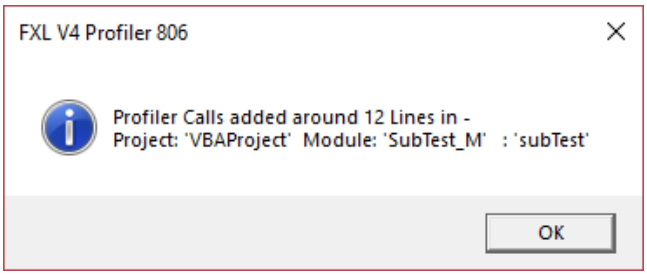
You can add drill-down profiler calls to locate slow VBA blocks of code or slow VBA loops.

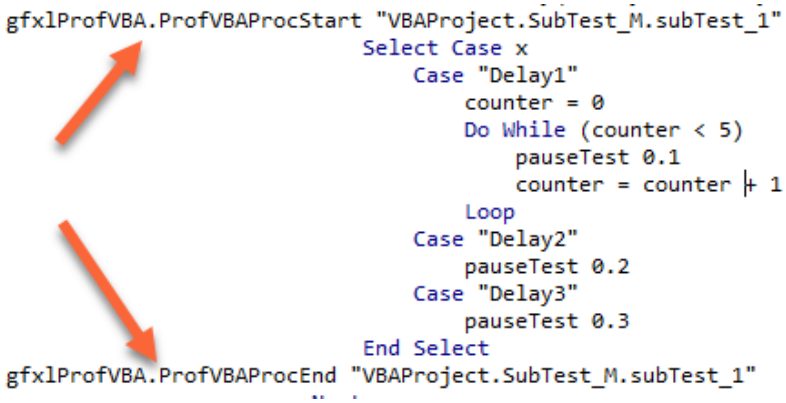
The recommended way of doing this is to:

- Select statements or a procedure (Sub or Function) in the VBE
- Then click the “Profile Statements” or “Profile Loops” buttons on the ProfVBA Toolbar in the VBE

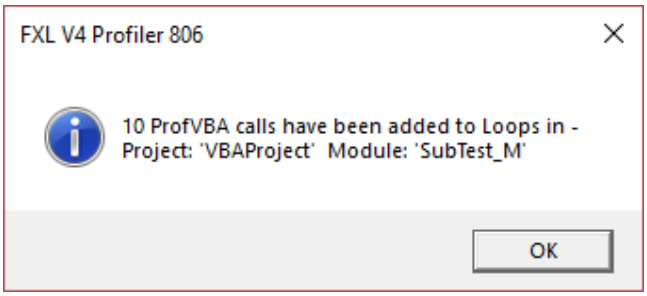


Profile Statements will wrap the selected VBA statements in profiler Start and End calls.





Profile Loops will wrap each loop within the selected procedure in Profiler Start and End calls, and will handle nested loops (up to a maximum (default 10) set in Add Profiler Calls).




```
gfxlProfVBA.ProfVBAProcStart "VBAProject.SubTest_M.subTest.ForEach_3.x"
                    For Each x In Array("Delay1", "Delay2", "Delay3")
                        Select Case x
                            Case "Delay1"
                                counter = 0
gfxlProfVBA.ProfVBAProcStart "VBAProject.SubTest_M.subTest.DoLoop_4"
                                Do While (counter < 5)
                                    pauseTest 0.1
                                    counter = counter + 1
                                Loop
gfxlProfVBA.ProfVBAProcEnd "VBAProject.SubTest_M.subTest.DoLoop_4", True
                            Case "Delay2"
                                pauseTest 0.2
                            Case "Delay3"
                                pauseTest 0.3
                        End Select
                    Next
gfxlProfVBA.ProfVBAProcEnd "VBAProject.SubTest_M.subTest.ForEach_3.x", True
```

You can repeat these steps to add profiler calls to many different blocks of VBA.

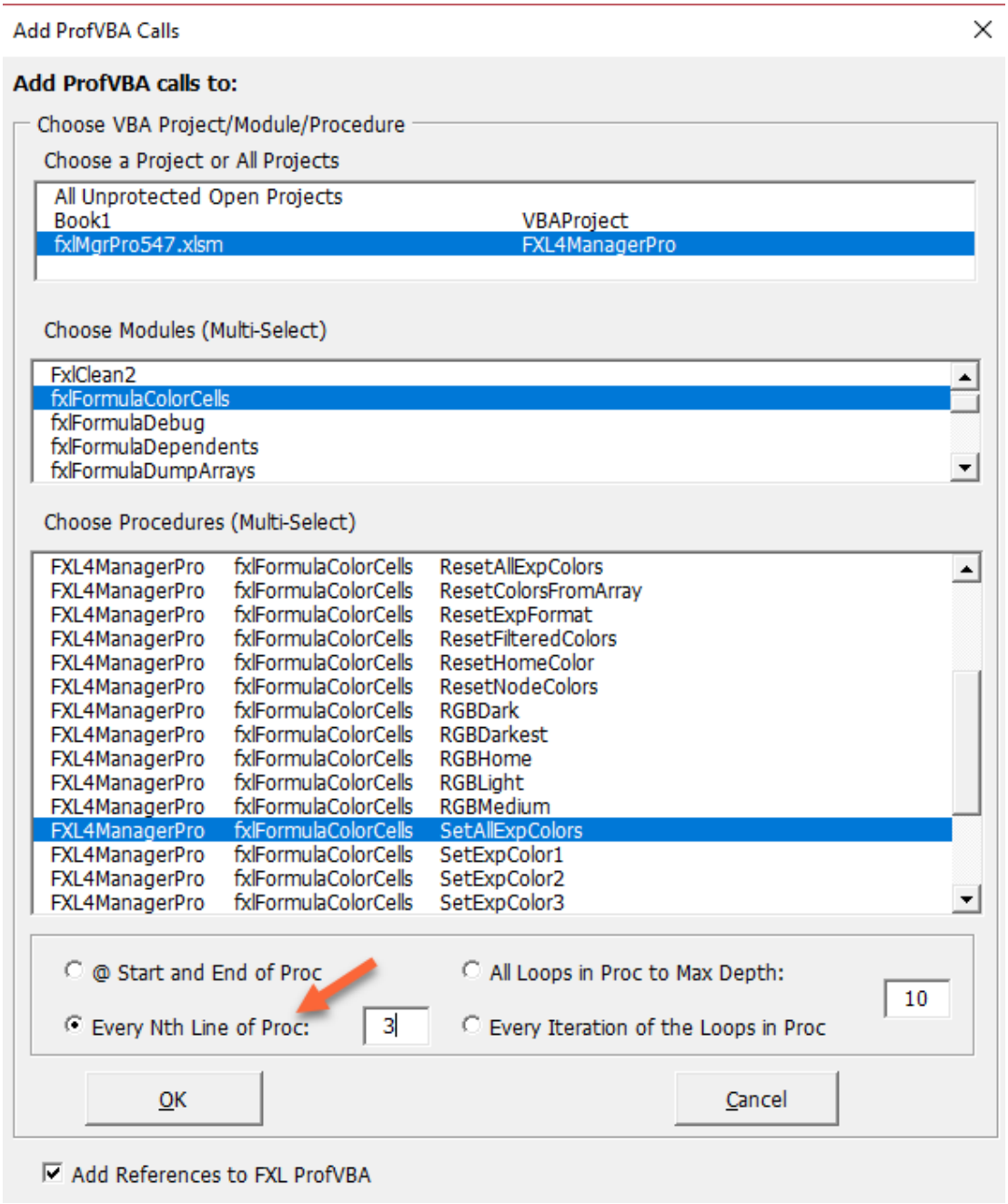
Adding Drill-Down Profiler Calls using "Add Profiler Calls"

You can also add drill-down calls from the "Add Profiler Calls" form.

Every Nth Line

Selecting one or more Procedures enables the "Every Nth line of Proc", "All Loops in Proc" and "Every Iteration of Loops" options.

Use one of these options when you want to drill-down into specific procedures to see which lines of code in the procedures are consuming the time.



The **Nth Line option** adds End and Start pairs of calls after every Nth line (3 in this example)

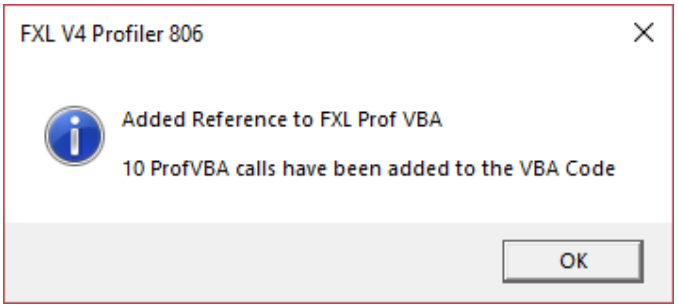
```
gfxlProfVBA.ProfVBAProcStart "FXL4ManagerPro.fxlFormulaColorCells.SetAllExpColors_2"
70  Else
80      GoTo Fail:
90  End If

gfxlProfVBA.ProfVBAProcEnd "FXL4ManagerPro.fxlFormulaColorCells.SetAllExpColors_2"
gfxlProfVBA.ProfVBAProcStart "FXL4ManagerPro.fxlFormulaColorCells.SetAllExpColors_3"
100 ResetNodeColors cTree, jExps(), strExps(), jFuncs()
110 If gblClosing Then
120     gblClosing = False
130     gfxlProfVBA.ProfVBAProcEnd "FXL4ManagerPro.fxlFormulaColorCells.SetAllExpColors_3": Exit
gfxlProfVBA.ProfVBAProcEnd "FXL4ManagerPro.fxlFormulaColorCells.SetAllExpColors_3"
gfxlProfVBA.ProfVBAProcStart "FXL4ManagerPro.fxlFormulaColorCells.SetAllExpColors_4"
```

All Loops in Proc to Max Depth

This option wraps every loop in the selected procedure on profiler calls.

Max Depth refers to the maximum level of loops nested inside other loops that is allowed.

The Profiler recognises FOR, DO and WHILE loops.




```
gfxlProfVBA.ProfVBAProcStart "VBAProject.SubTest_M.subTest.ForEach_3.x"
                    For Each x In Array("Delay1", "Delay2", "Delay3")
                        Select Case x
                            Case "Delay1"
                                counter = 0
gfxlProfVBA.ProfVBAProcStart "VBAProject.SubTest_M.subTest.DoLoop_4"
                                Do While (counter < 5)
                                    pauseTest 0.1
                                    counter = counter + 1
                                Loop
gfxlProfVBA.ProfVBAProcEnd "VBAProject.SubTest_M.subTest.DoLoop_4", True
                            Case "Delay2"
                                pauseTest 0.2
                            Case "Delay3"
                                pauseTest 0.3
                        End Select
                    Next
gfxlProfVBA.ProfVBAProcEnd "VBAProject.SubTest_M.subTest.ForEach_3.x", True
```

The Profiling report shows the results for each Loop.

VBA Profiling Results for TestProfilerDoFor.xlsm

Total Execution Time in Milliseconds 1325.17

Project.Module.Procedure	Count of Calls	Exclusive Procedure Time				% Of	Call Sequence Counts	
		Total	Average	Max	Min	Overall	Start Seq Counter	Exit Seq Counter
VBAProject.SubTest_M.subTest.DoLoop_4	1	507.67	507.67	507.67	507.67	38.3%	9	5
VBAProject.SubTest_M.subTest.ForEach_3.x	1	504.65	504.65	504.65	504.65	38.1%	8	6
VBAProject.SubTest_M.subTest.ForEach_1.v	3	205.55	68.52	103.84	0.01	15.5%	2	8
VBAProject.SubTest_M.subTest.ForEach_2.w	1	102.02	102.02	102.02	102.02	7.7%	7	7
VBAProject.SubTest_M.subTest	3	5.28	1.76	2.78	1.09	0.4%	1	9

Every Iteration of the Loops

This option inserts Profiler calls **inside each loop**, so that the profiler records each individual pass through the loop.

```
                For Each x In Array("Delay1", "Delay2", "Delay3")
gfxlProfVBA.ProfVBAProcStart "VBAProject.SubTest_M.subTest.ForEach_3.x:" & gfxlProfVBA.ProfVBALoopIndex("Push")
                    Select Case x
                        Case "Delay1"
                            counter = 0
                            Do While (counter < 5)
gfxlProfVBA.ProfVBAProcStart "VBAProject.SubTest_M.subTest.DoLoop_4:" & gfxlProfVBA.ProfVBALoopIndex("Push")
                                pauseTest 0.1
                                counter = counter + 1
gfxlProfVBA.ProfVBAProcEnd "VBAProject.SubTest_M.subTest.DoLoop_4:" & gfxlProfVBA.ProfVBALoopIndex("Pop"), True
                            Loop
                        Case "Delay2"
                            pauseTest 0.2
                        Case "Delay3"
                            pauseTest 0.3
                    End Select
gfxlProfVBA.ProfVBAProcEnd "VBAProject.SubTest_M.subTest.ForEach_3.x:" & gfxlProfVBA.ProfVBALoopIndex("Pop"), True
                Next
```

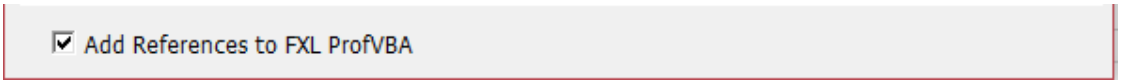
The Profiling Report shows results for each iteration. I sorted the results on the first column to show them in execution sequence.

n Milliseconds 1325.65

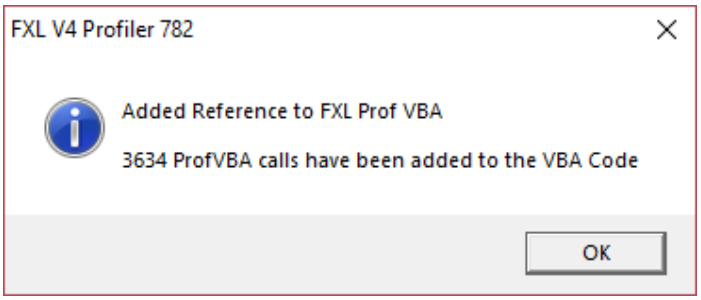
Sorted Results

Project.Module.Procedure	Count of Calls	sive Procedure Time Total	Average	Max	Min	% Of Overall	Call Sequence Counts Start Seq Counter	Exit Seq Counter
VBAProject.SubTest_M.subTest	3	6.65	2.22	2.91	1.34	0.5%	1	20
VBAProject.SubTest_M.subTest.DoLoop_4:4.10	1	100.63	100.63	100.63	100.63	7.6%	16	12
VBAProject.SubTest_M.subTest.DoLoop_4:4.6	1	100.23	100.23	100.23	100.23	7.6%	12	8
VBAProject.SubTest_M.subTest.DoLoop_4:4.7	1	98.87	98.87	98.87	98.87	7.5%	13	9
VBAProject.SubTest_M.subTest.DoLoop_4:4.8	1	101.61	101.61	101.61	101.61	7.7%	14	10
VBAProject.SubTest_M.subTest.DoLoop_4:4.9	1	99.64	99.64	99.64	99.64	7.5%	15	11
VBAProject.SubTest_M.subTest.ForEach_1.v:1.1	3	203.43	67.81	101.73	0.00	15.3%	2	6
VBAProject.SubTest_M.subTest.ForEach_1.v:1.14	1	0.00	0.00	0.00	0.00	0.0%	20	19
VBAProject.SubTest_M.subTest.ForEach_1.v:1.2	2	3.24	1.62	3.23	0.00	0.2%	5	18
VBAProject.SubTest_M.subTest.ForEach_2.w:2.13	1	0.00	0.00	0.00	0.00	0.0%	19	17
VBAProject.SubTest_M.subTest.ForEach_2.w:2.3	1	100.50	100.50	100.50	100.50	7.6%	9	7
VBAProject.SubTest_M.subTest.ForEach_2.w:2.4	1	2.97	2.97	2.97	2.97	0.2%	10	16
VBAProject.SubTest_M.subTest.ForEach_3.x:3.11	1	201.45	201.45	201.45	201.45	15.2%	17	14
VBAProject.SubTest_M.subTest.ForEach_3.x:3.12	1	300.21	300.21	300.21	300.21	22.6%	18	15
VBAProject.SubTest_M.subTest.ForEach_3.x:3.5	1	6.21	6.21	6.21	6.21	0.5%	11	13

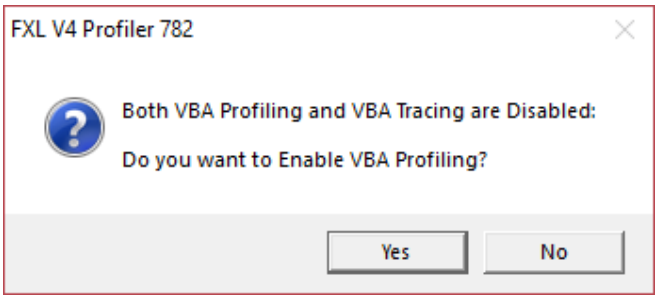
Create a VBE Reference to FastExcel Profiler



The profiler calls that are added to your VBA code call into FastExcel VBA Profiler, and so require a reference to be created.
Clicking OK will add the references and provide a status message.



And prompt you to Enable Profiling or Tracing.



Show Timings in the Immediate Window

Choosing this option will show timing results in the immediate window. This enables rapid speed testing of changes to the VBA.

All Loops in Proc Immediate Output

```
subTest.ForEach_1.v         0 Milliseconds
subTest       1.36 Milliseconds
subTest.ForEach_1.v         101.73 Milliseconds
subTest       1.09 Milliseconds
subTest.DoLoop_4            507.63 Milliseconds
subTest.ForEach_3.x         503.67 Milliseconds
subTest.ForEach_2.w         102.78 Milliseconds
subTest.ForEach_1.v         102.64 Milliseconds
subTest       0.9 Milliseconds
```

Note: The results are output to the immediate window at the conclusion of each Profile_Start and Profile_End block and are not aggregated. This in contrast to the Profiling Report where the results of each pass through a block are aggregated:

Total Execution Time in Milliseconds 1321.81								
		Exclusive Procedure Time				% Of	Call Sequence Counts	
Project.Module.Procedure	Count of Calls	Total	Average	Max	Min	Overall	Start Seq Counter	Exit Seq Counter
VBAProject.SubTest_M.subTest.DoLoop_4	1	507.63	507.63	507.63	507.63	38.4%	9	5
VBAProject.SubTest_M.subTest.ForEach_3.x	1	503.67	503.67	503.67	503.67	38.1%	8	6
VBAProject.SubTest_M.subTest.ForEach_1.v	3	204.37	68.12	102.64	0.00	15.5%	2	8
VBAProject.SubTest_M.subTest.ForEach_2.w	1	102.78	102.78	102.78	102.78	7.8%	7	7
VBAProject.SubTest_M.subTest	3	3.36	1.12	1.36	0.90	0.3%	1	9

Every Iteration of Loops Immediate Output

```
subTest.ForEach_1.v:1.1     0.01 Milliseconds
subTest       1.49 Milliseconds
subTest.ForEach_1.v:1.1     100.69 Milliseconds
subTest.ForEach_1.v:1.2     0.01 Milliseconds
subTest       105.76 Milliseconds
subTest.ForEach_1.v:1.1     203.38 Milliseconds
subTest.ForEach_2.w:2.3     99.54 Milliseconds
subTest.DoLoop_4:4.6        100.57 Milliseconds
subTest.DoLoop_4:4.7        99.56 Milliseconds
subTest.DoLoop_4:4.8        101.61 Milliseconds
subTest.DoLoop_4:4.9        100.53 Milliseconds
subTest.DoLoop_4:4.10       99.66 Milliseconds
subTest.ForEach_3.x:3.5     507.54 Milliseconds
subTest.ForEach_3.x:3.11    201.44 Milliseconds
subTest.ForEach_3.x:3.12    300.2 Milliseconds
subTest.ForEach_2.w:2.4     1012.15 Milliseconds
subTest.ForEach_2.w:2.13    0 Milliseconds
subTest.ForEach_1.v:1.2     1114.59 Milliseconds
subTest.ForEach_1.v:1.14    0 Milliseconds
subTest       1325.94 Milliseconds
```

Total Execution Time in Milliseconds 1632.4								
		Inclusive Procedure Time				% Of	Call Sequence Counts	
Project.Module.Procedure	Count of Calls	Total	Average	Max	Min	Overall	Start Seq Counter	Exit Seq Counter
VBAProject.SubTest_M.subTest	3	1,325.94	441.98	1,220.18	1.49	81.2%	1	20
VBAProject.SubTest_M.subTest.DoLoop_4:4.10	1	99.66	99.66	99.66	99.66	6.1%	16	12
VBAProject.SubTest_M.subTest.DoLoop_4:4.6	1	100.57	100.57	100.57	100.57	6.2%	12	8
VBAProject.SubTest_M.subTest.DoLoop_4:4.7	1	99.56	99.56	99.56	99.56	6.1%	13	9
VBAProject.SubTest_M.subTest.DoLoop_4:4.8	1	101.61	101.61	101.61	101.61	6.2%	14	10
VBAProject.SubTest_M.subTest.DoLoop_4:4.9	1	100.53	100.53	100.53	100.53	6.2%	15	11
VBAProject.SubTest_M.subTest.ForEach_1.v:1.1	3	203.38	67.79	102.69	0.01	12.5%	2	6
VBAProject.SubTest_M.subTest.ForEach_1.v:1.14	1	0.00	0.00	0.00	0.00	0.0%	20	19
VBAProject.SubTest_M.subTest.ForEach_1.v:1.2	2	1,114.59	557.29	1,114.58	0.01	68.3%	5	18
VBAProject.SubTest_M.subTest.ForEach_2.w:2.13	1	0.00	0.00	0.00	0.00	0.0%	19	17
VBAProject.SubTest_M.subTest.ForEach_2.w:2.3	1	99.54	99.54	99.54	99.54	6.1%	9	7
VBAProject.SubTest_M.subTest.ForEach_2.w:2.4	1	1,012.15	1,012.15	1,012.15	1,012.15	62.0%	10	16
VBAProject.SubTest_M.subTest.ForEach_3.x:3.11	1	201.44	201.44	201.44	201.44	12.3%	17	14
VBAProject.SubTest_M.subTest.ForEach_3.x:3.12	1	300.20	300.20	300.20	300.20	18.4%	18	15
VBAProject.SubTest_M.subTest.ForEach_3.x:3.5	1	507.54	507.54	507.54	507.54	31.1%	11	13

The Loop iterations are labelled slightly different for the 3 different types of loops (For To, For Each and Do).

The general format is

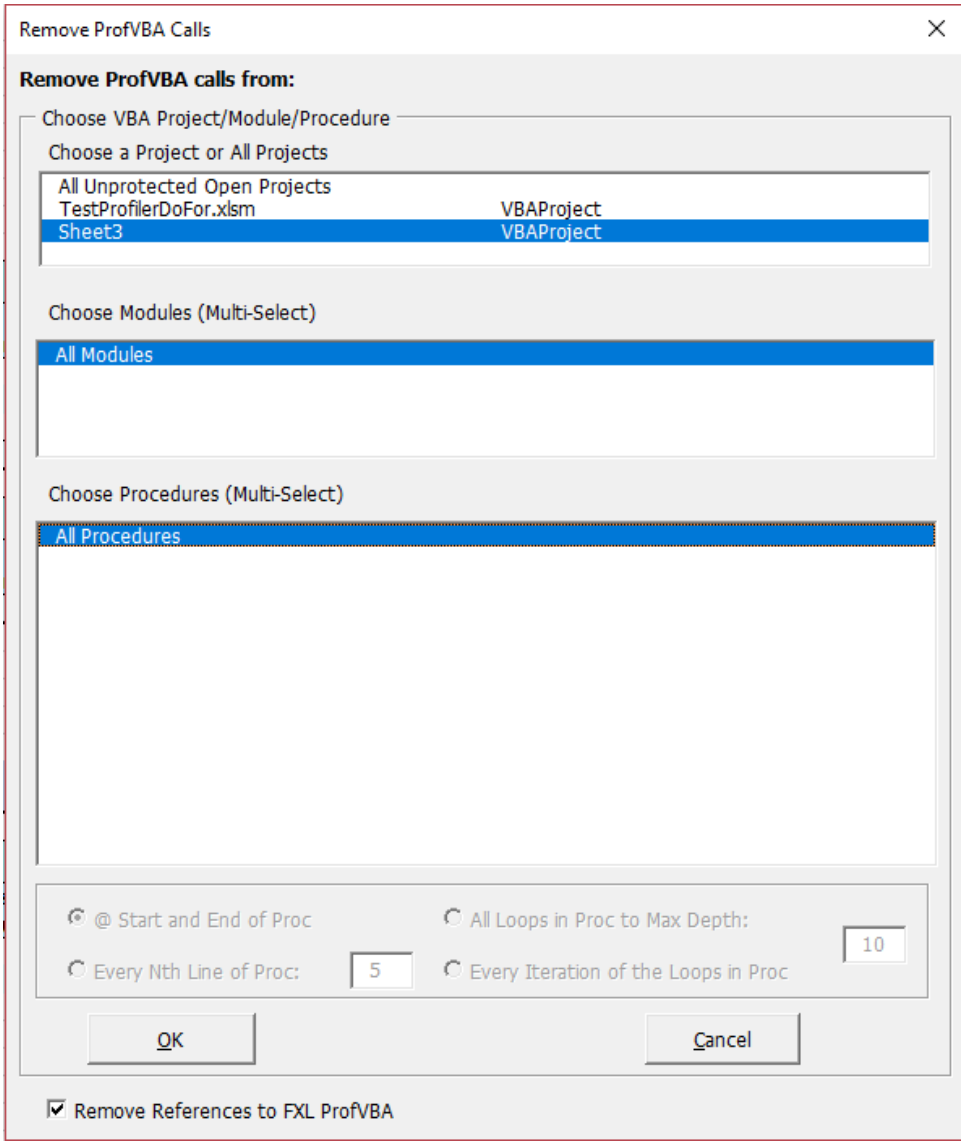
procedureName.loopType_UID1.variableName:depth.UID2

Where

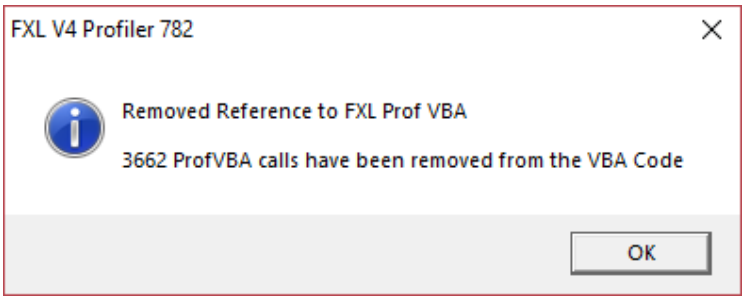
- **loopType** is one of the following, For To, For Each, or Do Loop
- **UID1** is a loop unique identifier that can be seen in the user code. This is to support in finding the loop which this timer measured,
- **variableName** is the loop variable for the For To and For Each loops. The Do loop does not have a loop variable,
- **depth** is the nested depth value. Subroutine and functions will be counted as a depth down,
- **UID2** is a unique identifier that is generated at runtime. This is required to identify the path a For Each or a Do loop is taking. For the For Each and Do loops, the UID2 is based on a counter that labels the loops sequentially when that loop is first called. For the For To loop, the UID2 is the value of the increment variable.

Remove Profiler Calls

This command tidies up your VBA code by removing ProfVBA calls and References in a similar way to Add Profiler Calls. If you have manually added profiler calls note that Remove is case-sensitive.



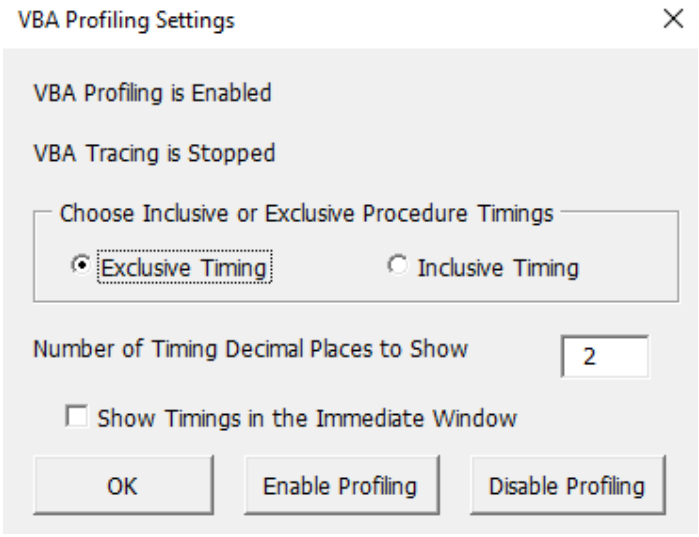
And gives a status message.



Settings, Enabling and Disabling Profiling and Tracing

After you have added Profiler Calls to your VBA code, profiling and tracing does not start until you enable it **and then run your VBA code**.

Note: Profiling and Tracing are mutually exclusive: starting one stops the other.



Enable Profiling Options

Exclusive timing

Shows only the actual time taken by a procedure and excludes time taken by child procedures called by the procedure.

Inclusive timing

Shows the total time taken by a procedure including the time taken by all its child procedures.

Decimal Places

Controls the number of decimal places of milliseconds to show in the result. Minimum 0, Maximum 4, Default 2.

Show Timings in the Immediate Window

When this is checked results will be shown in the VBE Immediate window as well as being available to the Profiling report.

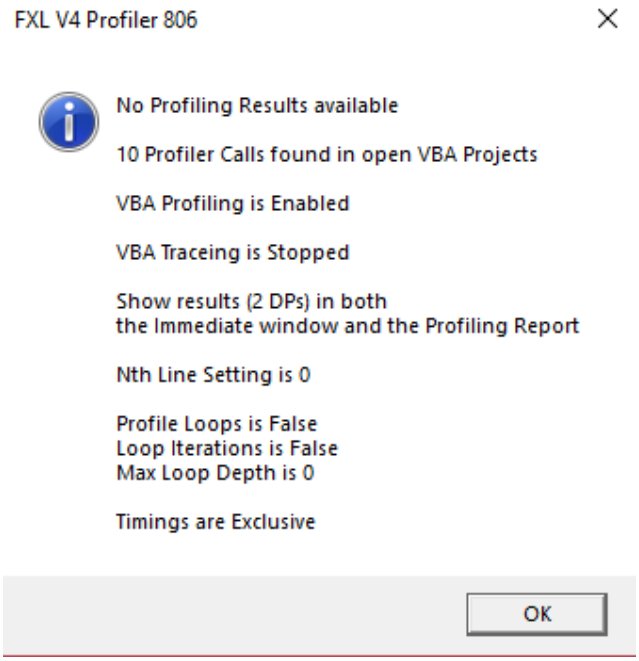
```
subTest.ForEach_1.v         0 Milliseconds
subTest       1.36 Milliseconds
subTest.ForEach_1.v         101.73 Milliseconds
subTest       1.09 Milliseconds
subTest.DoLoop_4            507.63 Milliseconds
subTest.ForEach_3.x         503.67 Milliseconds
subTest.ForEach_2.w         102.78 Milliseconds
subTest.ForEach_1.v         102.64 Milliseconds
subTest       0.9 Milliseconds
```

As noted above the Immediate Window results are output at the conclusion of each Profile_Start and Profile_End block and are not aggregated. This in contrast to the Profiling Report where the results of each pass through a block are aggregated.

Total Execution Time in Milliseconds 1321.81								
		Exclusive Procedure Time				% Of	Call Sequence Counts	
Project.Module.Procedure	Count of Calls	Total	Average	Max	Min	Overall	Start Seq Counter	Exit Seq Counter
VBAProject.SubTest_M.subTest.DoLoop_4	1	507.63	507.63	507.63	507.63	38.4%	9	5
VBAProject.SubTest_M.subTest.ForEach_3.x	1	503.67	503.67	503.67	503.67	38.1%	8	6
VBAProject.SubTest_M.subTest.ForEach_1.v	3	204.37	68.12	102.64	0.00	15.5%	2	8
VBAProject.SubTest_M.subTest.ForEach_2.w	1	102.78	102.78	102.78	102.78	7.8%	7	7
VBAProject.SubTest_M.subTest	3	3.36	1.12	1.36	0.90	0.3%	1	9

The Immediate Window results can be Exclusive or Inclusive.

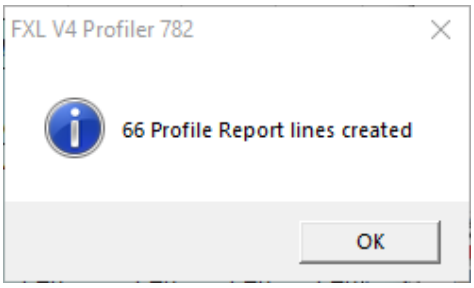
Show ProfVBA Status



ProfVBA Status shows you the current profiling status and settings.

Note: there are Profiling Results available to report.

Show Profiling Report



The profiling report is created as a worksheet called FastXLProfVBAn in a separate workbook.

	A	B	C	D	E	F
1	*VBA Profiling Results for fxlMgrPro473.xlsm*					
2						
3	*Total Execution Time in Milliseconds 68*					
4			Procedure Exclusive Time			
5	Project.Module.Procedure	Count of Calls	Total	Average	Max	Min
6	FXL4ManagerPro.fxlFormulaColorCells.StorePrecFormat	1	42.27	42.27	42.27	42.27
7	FXL4ManagerPro.fxlFormulaColorCells.SetAllExpColors	1	14.76	14.76	14.76	14.76
8	FXL4ManagerPro.ufViewEdit.GoRange	1	4.43	4.43	4.43	4.43
9	FXL4ManagerPro.fxlFormulaColorCells.SetExpColor3	1	1.46	1.46	1.46	1.46
10	FXL4ManagerPro.fxlFormulaShow.MoveEvalLeft	1	1.09	1.09	1.09	1.09

The report timings shown are exclusive: they show the timed used by a procedure **excluding** the time taken by the calls from that procedure to other procedures. When using the Inclusive setting the time taken by the calls for a procedure will include both the time taken by the procedure itself and the time taken by all the procedures it has called.

In this example SetAllExpColors is using 22% of the execution time. To drill down into that procedure, “Remove Profiler Calls” then “Add Profiler Calls” for SetAllExpColors using either the Nth line option or the Profile Selected Statements option or one of the Profile Loops options.

For ease of use Profile Statements and profile Loops can be used from within the VBE and act on the VBE selected statements or procedure.

It turns out that in this case the profile shows that 85% of the time used in SetAllExpColors is turning Screen Updating back on after colouring the cells.

Call Sequence Counts

The Profiler maintains independent calling sequence counts for the first entry call to a procedure and the last exit from a procedure. So, if Proc1 is the first Proc and calls Proc2 which then exits and returns to Proc1 which then exits you would get:

Proc Name	Start Sequence	Exit Sequence
Proc1	1	2
Proc2	2	1

Sorting by Start Sequence or Exit Sequence gives a quick way of looking at the starting or ending flow of your procedure calls.

For a more comprehensive trace see FastExcel Trace below.

Uncaptured Exits

If a VBA procedure exits without reaching a ProfVBAProcEnd statement it will not show in the Profiling Report.
The time taken by an uncaptured exit procedure is accumulated into the procedure that it exits to.

The most frequent cause of uncaptured exits is a higher-level error handler:

Sub1 contains an On Error statement to capture VBA errors, and calls Sub2.

Sub2 calls Sub3

Sub3 calls Sub4 and then **causes a divide by zero error after the Sub4 call.**
This error passes control directly to the error handler in Sub1, bypassing the execution of the ProfVBAProcEnd calls in both sub 3 and sub2.

The Profiling Report therefore will not show the time taken by Sub2 and Sub3 and it will be accumulated into Sub1.

VBA Profiling Results for fxlV4Profiler798.xlsm

Total Execution Time in Milliseconds 3590.18

		Exclusive Procedure Time				% Of	Call Sequence Counts	
Project.Module.Procedure	Count of Calls	Total	Average	Max	Min	Overall	Start Seq Counter	Exit Seq Counter
VBAProject.Module1.sub1	1	2,699.07	2,699.07	2,699.07	2,699.07	75.2%	1	2
VBAProject.Module1.sub4	1	891.11	891.11	891.11	891.11	24.8%	4	1

Uncaptured Procedure Exit detected: Uncaptured Procedure time will be accumulated into its calling Procs

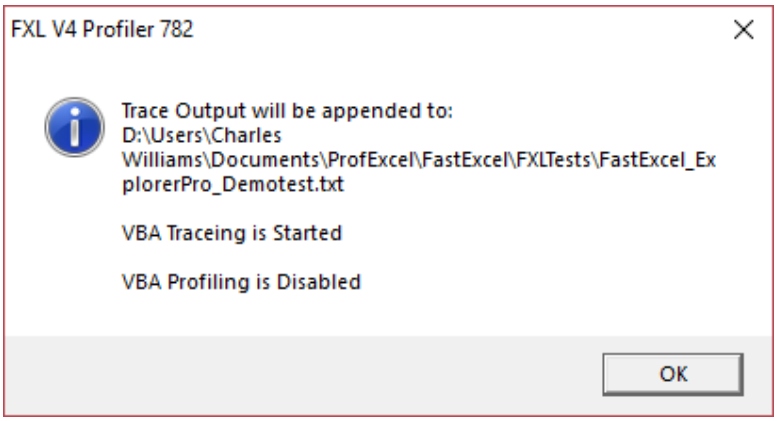
FastExcel Execution Tracing

Because VBA is an event-driven language it can be difficult to work out the path execution takes through your code.

If you need to trace the flow of execution through a sequence of procedure and event calls you can use FastExcel Trace Profiling.

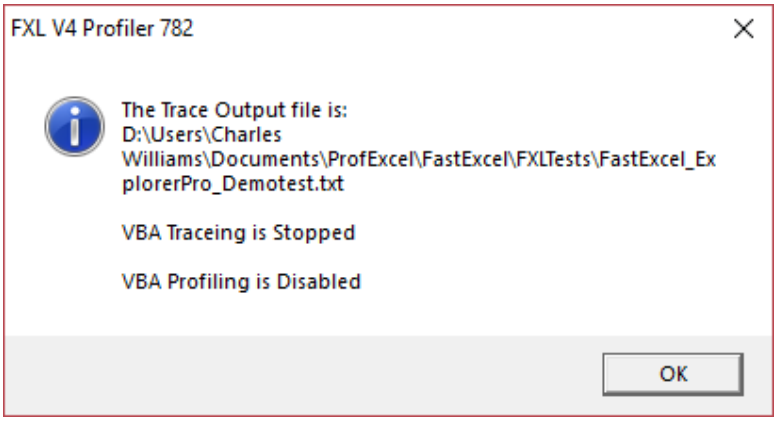
First you need to Add Profiler Calls to the VBA code you want to trace (usually start by adding to all modules and procedures), in exactly the same the same way as for Profiling.

Start VBA Trace



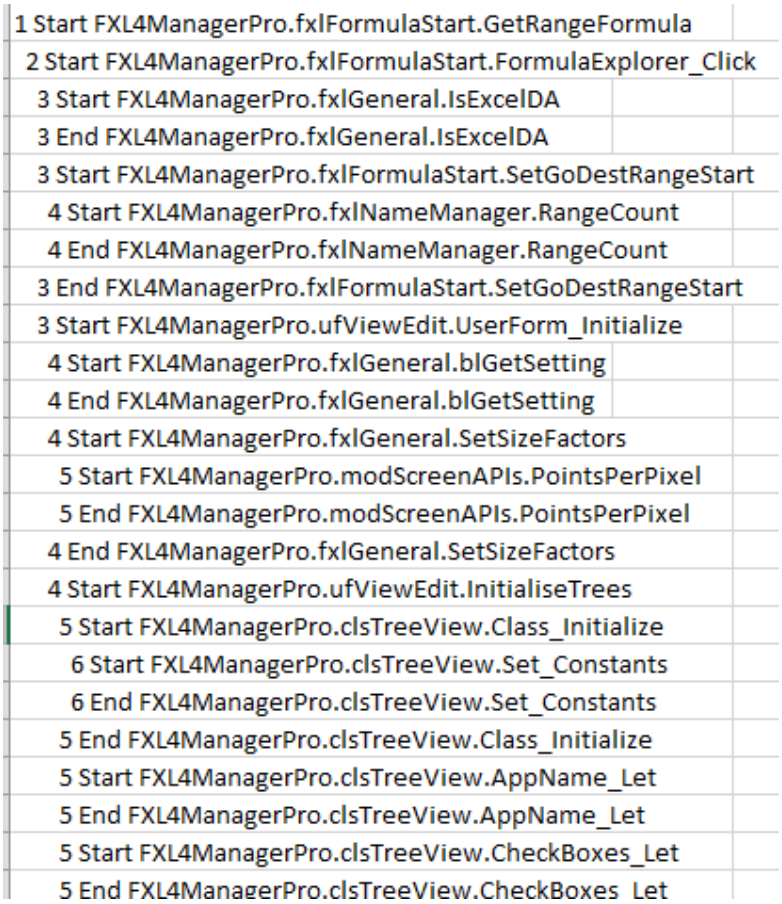
The Trace report is appended to a .txt file in the same folder as the active workbook, rather than as a worksheet report. This ensures that as much of the trace as possible survives a crash.

Stop VBA Trace



Use Stop VBA Trace to ensure that all trace results are written to the file.

The Trace Report .txt File



The starting number and indentation show the start and end calling level for the procedure.

Uncaptured Exits in the Trace Report

If uncaptured exits have occurred the last trace line shown will be indented and its calling level will not be 1.

See Show Profiling Report – Uncaptured Exits for a more detailed explanation.

Advanced Use of ProfVBA: Inline Commands

Don Soloway extended ProfVBA to be able to use commands embedded as comments in the VBA code to drive the addition of Profiler calls for Loops.

This allows you to retain profiling instrumentation in interesting areas of your code without impacting execution.

Then you can call Add Profiler Calls to easily regenerate all the profiling start and end calls. Calling Remove Profiler calls will remove the Start and End calls but leave the Inline Commands intact.

The format of the inline commands is as follows:

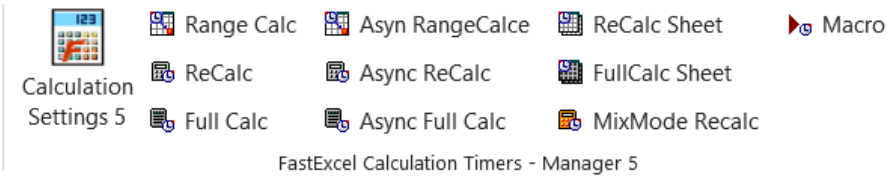
'TagString Start profilerName [variableName] [stackPush]
'TagString End profilerName [variableName] [stackPop] [displayToImmediate]

Where

- **TagString** is an identifying string to indicate that an inline command follows. The TagString currently equals gfxlProfVBA.
- **profilerName** is a user specified name for that timer
- **variableName** is the increment variable name for the For To loops
- **stackPush** is to save the UID2 for nested loops
- **stackPop** is to retrieve the UID2 for nested loops
- **displayToImmediate** is to indicate that output is sent to the immediate window. This takes the value of True or False.

Calculation Manager

Note: Calculation Manager is not available with Windows FastExcel Manager Pro.
FastExcel Extended Calculation Modes are only available with FastExcel V5 SpeedTools and FastExcel Manager Pro for Mac

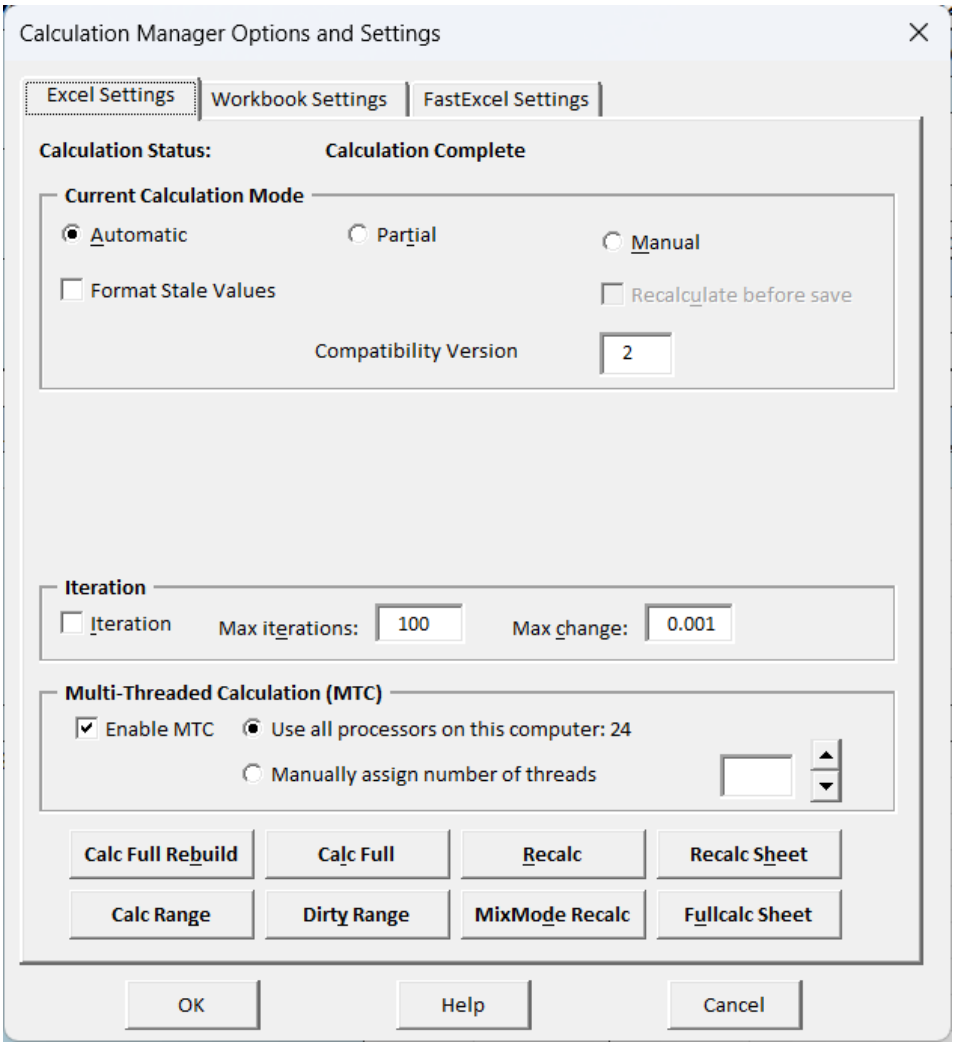


Calculation Manager gives you:

- Full Control of all Excel's calculation settings:
- Call the Calculation methods from buttons or keystrokes
    - Asynchronous Calculation Timing: Range, ReCalc, Fullcalc
      Asynchronous function timing includes the wait for the async function to return.
    - Calculate Full Rebuild Dependencies
    - Calculate Full all open workbooks
    - Recalculate all open workbooks
    - Recalculate Worksheet
    - Full Calculate Worksheet
    - Calculate Range
    - Mixed Mode Recalculation
- Accurate timing for each calculation method
    - On Windows timing accuracy is at the Microsecond level, although in practice other factors such as Windows multi-tasking prevent timing tasks reaching this level of accuracy.
    - On Mac timing is accurate at best to around 4 milliseconds.

Calculation Manager Options and Settings

The settings are in three groups: Excel Settings, Workbook Settings and FastExcel Settings.



Excel Calculation Settings

Excel's default methods for Determining Calculation Mode.

By default, Excel uses the same calculation mode for all open workbooks.
The Calculation mode is initially set from the first workbook opened and is not changed when other workbooks are opened.

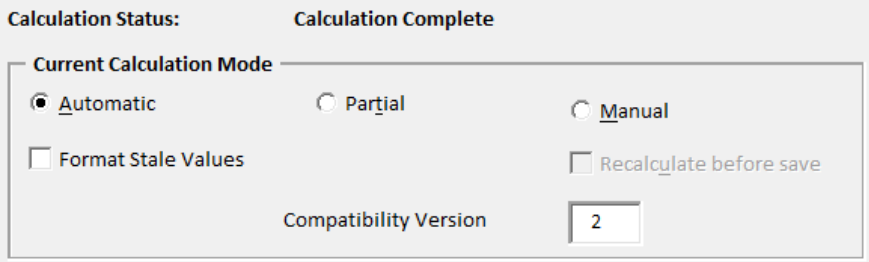
Excel uses the same calculation mode for all open workbooks.

Excel sets the initial calculation mode from the first (previously saved) workbook opened. This initial calculation mode will not change when additional workbooks are opened, and will only change when either the user or a VBA program changes it. If you start Excel with a brand-new empty workbook and make changes to that workbook then the calculation mode is Automatic.

If a manual mode workbook is opened in automatic mode the workbook will be calculated.

Excel Current Calculation Mode

You can use this form to view and change Excel's current calculation status and settings.



By default, the current Excel calculation mode applies to all the open workbooks.
Also, all the open workbooks are recalculated, not just one

Excel's default calculation mode is at the Excel session level.
This means that Excel will use the same calculation mode for all open workbooks, regardless of the calculation mode that was saved with an individual workbook.

Whenever Excel calculates the default behavior is to calculate all the open workbooks, rather than just the active workbook

Automatic

Formulas are recalculated automatically whenever anything changes so that the workbook(s) are always calculated.

Partial

Similar to automatic except that Excel Iterative Tables and Python for Excel cells are not automatically calculated. This can be useful with large workbooks because Excel Tables and cause multiple calculations of the workbook.

Manual

The status bar will also show "Calculate" if there are circular references or ForceFullCalculation has been turned on.

Formulas are only recalculated when the user requests it by pressing F9 or the FastExcel recalculate button. When the workbook is not completely calculated the status bar shows "calculate".

Format Stale Values Uncalculated (Stale) Cells are shown with Strikethrough

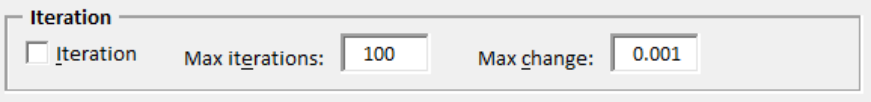
Recalculate before save.

If in Manual mode checking this option will cause Excel to recalculate uncalculated formulas in the workbook each time it is saved.

Compatibility Version Controls the compatibility version calculation mode of the current workbook.

Excel Iteration

These settings control how Excel handles circular references.



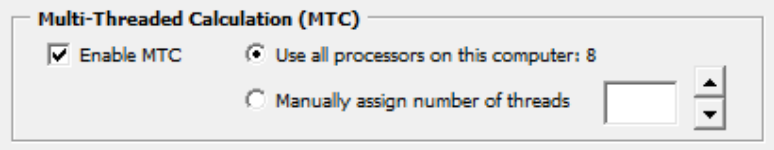
Iteration

When checked Excel will attempt to resolve formulas that are linked by circular references by repeatedly calculating them.

Maximum iterations and Maximum change

Excel will stop the repeated calculation of the circular references as soon as either the maximum number of iterations has been reached or the maximum change in the values of all of the formulas is less that the specified maximum change.

Multi-threaded calculation Settings



Enable MTC

When Checked Excel (Windows Excel 2007 and Mac Excel version 16 and later) will split the calculation process over multiple threads. This will improve calculation speed on many problems, particularly with large workbooks and multiple recalculations.

Use all Processors/Manually Assign threads

You can control the maximum number of threads to be used by Multi-Threaded calculation

Excel Calculation Buttons



The calculation timer for these buttons is controlled by your FastExcel settings

The Excel Calculation Settings Tab also contains buttons to request various types of calculation. **FastExcel** will show the time taken by the calculation if the calculation timer for buttons is switched on.

Using FastExcel defaults Excel will calculate all enabled worksheets in all open workbooks, and will calculate all MixMode sheets with a Full calculate, and selected MixMode sheets with a Sheet calculate.

Calc Full Rebuild (Shift-Ctrl-Alt-F9)

By default, MixMode sheets are calculated if Mixed Mode Full Calculate is on

Calculate Full Rebuild rebuilds the dependency trees and then fully calculates all formulas in all open workbooks. This option is only available from Excel 2002 onwards.

Calc Full (Ctrl-Alt-F9)

By default, MixMode sheets are calculated if Mixed Mode Full Calculate is on

Calculate Full fully calculates all formulas in all open workbooks.

Recalc (F9)

Recalculation is the default Excel calculation process. Only changed cells and formulas dependent on changed cells are calculated

Recalculate recalculates all formulas in all open workbooks that have been changed or are volatile, or are dependent on a cell that has been changed or recalculated.
By default MixMode sheets are calculated if calculation mode is Manual and Mixed Mode Manual is on.

Recalc Sheet (Shift-F9)

Sheet Calculate re calculates the selected Sheets assuming that all precedent worksheets and linked workbooks have been correctly calculated.

Recalculate Sheet recalculates all formulas on each selected sheet that have been changed or are volatile or are dependent on a cell on the selected sheet that has been changed or recalculated. By default, MixMode sheets are calculated if Mixed Mode Selected Sheets is on.

Fullcalc Sheet (Alt-Shift-F9)

Full Calculate Sheet calculates every formula on the selected Sheets assuming that all precedent worksheets and linked workbooks have been correctly calculated.

Full Calculate Sheet calculates all formulas on each selected sheet. By default MixMode sheets are calculated if Mixed Mode Selected Sheets is on.

Calc Range (Alt-F9)

Use FastExcel Calculation Control->FastExcel Settings to choose between the 2 available Range Calculate methods (with or without dependencies internal to the selected range) and the number of Range Calculate trials (default 3). Range Calculate then shows the median time of all the trials

Range Calculate calculates the currently selected range of cells.
If more than one Area is selected; each Area is calculated left-to-right and top-to-bottom, in the sequence that the areas were selected.
If more than one worksheet is selected then only the selected range on the active worksheet is calculated.
If one or more of the selected cells is part of a multi-cell array formula then all the cells in the array formula will be included in the calculation.
If Calculation is set to Automatic both the time taken to calculate the range in Manual mode and the workbook recalculation time are shown.

Dirty Range

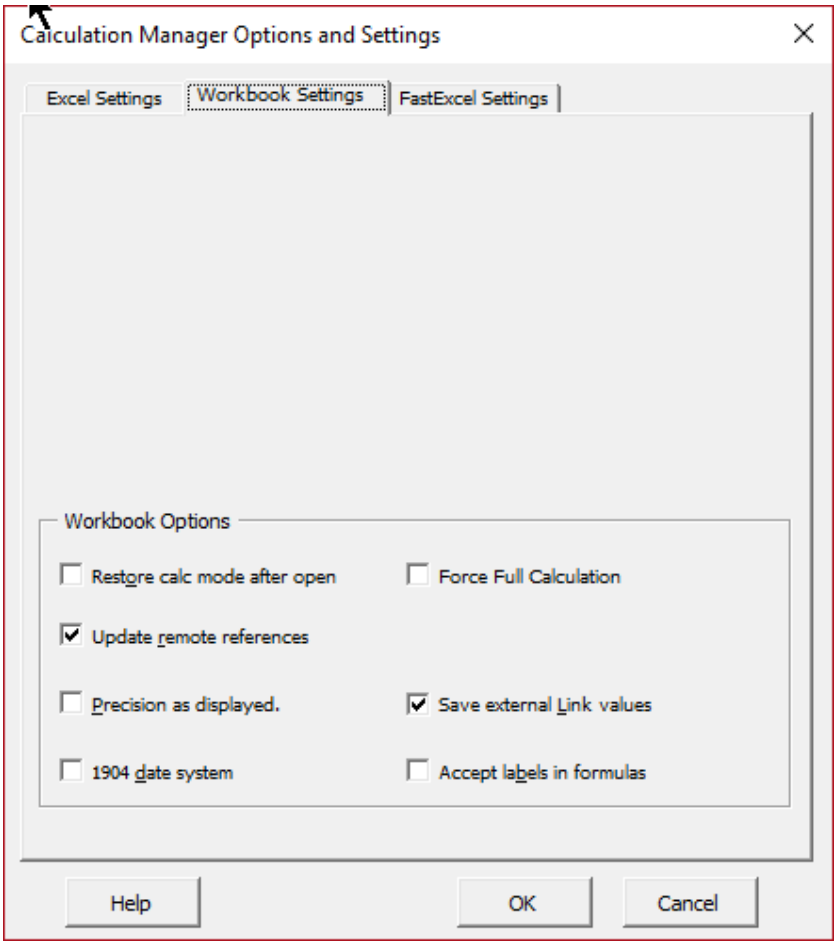
Flags the selected cells as uncalculated. This option is only available in Excel 2002 or later. In automatic mode this will trigger a recalculation.

MixMode Recalc (Ctrl-Shift-F9)

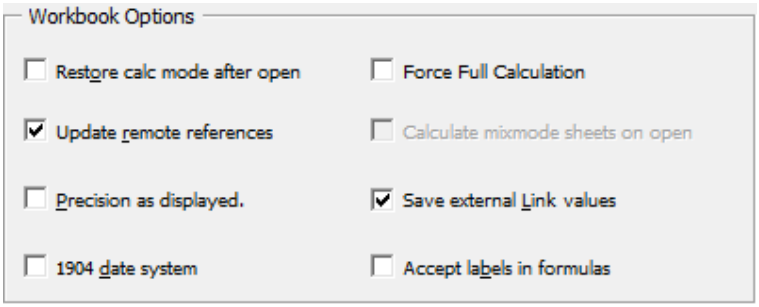
Recalculates both MixMode and ordinary sheets.

MixMode Recalc is similar to Recalc (F9), except that the recalculation also includes all Mixed Mode sheets.

Workbook Calculation Settings



Workbook Calculation Settings: Workbook Options



Restore Workbook Calculation Mode after open

This option is only available with FastExcel V5 SpeedTools and FastExcel Manager Pro for Mac

Force Full Calculation

This option is only available in Excel 2007 and Mac Excel Version 16 or later.

If checked every calculation on this workbook will be a full calculation rather than a recalculation, and the time taken to rebuild the dependency trees will not be needed at workbook open time or when the workbook is edited.

Switch on Force Full Calculation if adding a row at the top of a worksheet is very slow and Full Calculate time is close to Recalculate time.

Update Remote References

If TRUE automatically update any remote references (DDE Links to other programs) whenever Excel recalculates.

Save External Link Values

If TRUE Excel will save values for the links to external workbooks. I recommend keeping this option as TRUE

Precision as Displayed

Checking this box will force Excel to calculate to the number of decimal places that appear as a result of your formatting, ***and will permanently change any numbers stored in cells***.
You need to be sure you have thought through the full implications of this before using it.
Precision as Displayed slows down calculation.

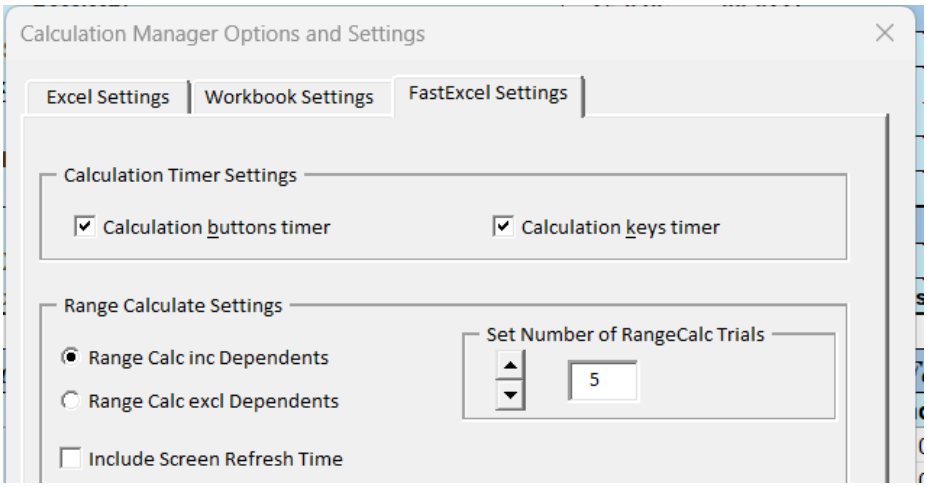
1904 Date System

If TRUE changes the starting date from which all dates are calculated from January 1 1900 to January 2 1904.

Accept Labels in Formulas

This allows Excel to use natural language labels in formulas. Because there are circumstances when this will give you unexpected or ambiguous results, ***I recommend you do NOT use this feature.***

FastExcel Settings



Calculation Timers for buttons and keys

You can control the FastExcel calculation time display separately for the calculate buttons and the calculate keys.

When checked you will see a message box after the calculation showing the time taken by the calculation.

Range Calculate Settings

Range Calc including or excluding Dependents

In Excel 2007 and later there are 2 different Range calculate methods. One includes within-range dependency calculations and the other ignores within-range dependencies and calculates row by row strictly left-to-right and top-to-bottom. Range Calculate always ignores dependencies outside the selected range, although the selected range will be expanded to include all the cells in multi-cell array formulas.

Set Number of RangeCalc Trials

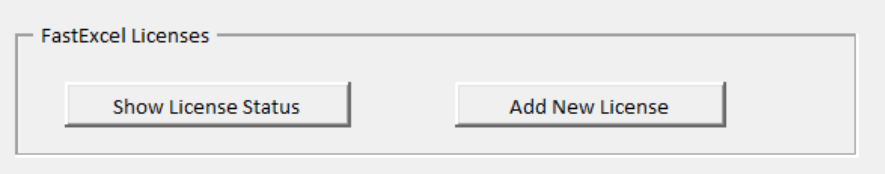
Because RangeCalc timing can be sensitive to Windows multi-tasking it is better to allow a number of trials. The Median (middle-most) calculation time will automatically be used, thus discarding incorrectly high and low values.

You can set the number of trials: the default number is 3.

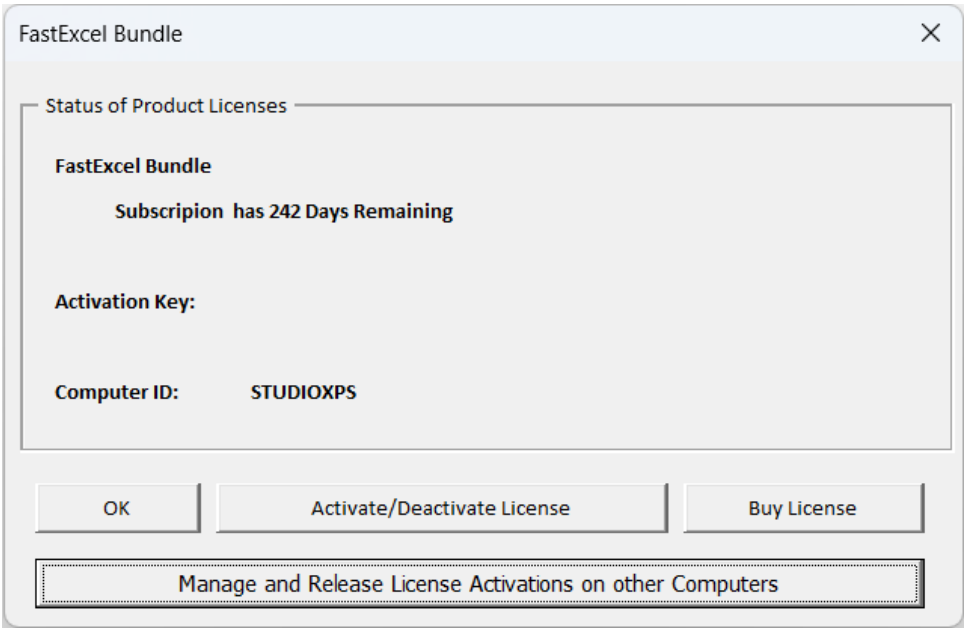
Include Screen Refresh Time

If checked timings for Range Calculate will include the time used by Excel to refresh the visible window.

FastExcel Licenses



Show License Status



Shows the current status of the FastExcel license for this FastExcel product. A Bundle license covers the individual profiler, Manager Pro and SpeedTools products.

Activate/Deactivate License launches the QLM license Wizard.

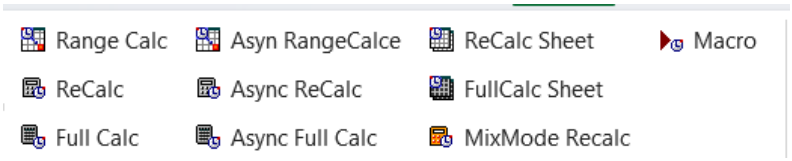
Buy License takes you to the Decision Models FastExcelV5 page.

Manager and Release License Activations takes you to the QLM Decision Models Licensing Portal.

Calculation Timing Commands

There are ten FastExcel calculation timing commands.

These buttons calculate the selected items and optionally show you the elapsed time taken by the calculation.



Use FastExcel Calculation Control->FastExcel Settings to choose between the 2 available Range Calculate methods (with or without dependencies internal to the selected range) and the number of Range Calculate trials (default 3). Range Calculate then shows the median time of all the trials.


The currently selected cells, using **Range Calculate**. Alt-F9
If more than one Area is selected; each Area is calculated left-to-right and top-to-bottom, in the sequence that the areas were selected.
If more than one worksheet is selected then only the selected range on the active worksheet is calculated.
If one or more of the selected cells is part of a multi-cell array formula then all the cells in the array formula will be included in the calculation.
If Calculation is set to Automatic both the time taken to calculate the range in Manual mode and the workbook recalculation time are shown.

Sheet Calculate re calculates the selected Sheets assuming that all precedent worksheets and linked workbooks have been correctly calculated.


The currently selected sheet(s), using **Sheet Recalculate**. Shift-F9.
If more than one worksheet is selected all the selected worksheets are recalculated. If Calculation is set to Automatic both the time taken to recalculate the sheet(s) in Manual mode and the subsequent workbook recalculation time are shown. If MixMode sheets are selected by default they will be calculated

Full Calculate Sheet calculates every formula on the selected Sheets assuming that all precedent worksheets and linked workbooks have been correctly calculated.


The currently selected sheet(s), using **Sheet Full Calculate**. Alt-Shift-F9.
If more than one worksheet is selected all the selected worksheets are recalculated. If Calculation is set to Automatic both the time taken to recalculate the sheet(s) in Manual mode and the subsequent workbook recalculation time are shown. If MixMode sheets are selected by default they will be calculated

Recalculation is the default Excel calculation process. Only changed cells and formulas dependent on changed cells are calculated


All enabled sheets in open workbooks. **Recalculate**. F9.

Full calculation calculates all formulas in all open workbooks

Recalc MixMode Sheets recalculates all sheets including MixMode sheets.

MixMode sheets are not calculated unless Manual Calculation has been checked in MixMode options.

Calculate all open workbooks. **Full Calculation**. Ctrl-Alt-F9

**Recalculate MixMode Sheets**: All enabled sheets and any MixMode sheets in all open workbooks are recalculated. Ctrl-Shift-F9.

Select, run and time a macro.

The Async timing commands time both the Excel calculation time and the wait time for the Async operation to complete.

Getting Consistent Results from FastExcel V5 Timing

Sometimes you will find that your **FastExcel** timing results are not repeatable, even though **FastExcel** measures elapsed time using a high-resolution timer with microsecond accuracy.

Power Saving and Multi- Core processors:

Power saving features, particularly on dual-core processors, may dynamically vary the speed of your processor. This can create problems for consistent timings using the high-resolution timer. If timing consistency is important try switching off the power-saving features of your processor.

Excel minimizes the number of calculations:

When Excel calculates it remembers:

- Which cells have been calculated.
- The optimum calculation path used for the final answer.

The second calculate is often much faster than the first

So, the next time Excel calculates it can save time by:

- Only calculating the cells that have changed (or depend on cells that have changed)
- Re-using the last calculation path

Why FastExcel Timing results may vary from run to run:

Both Excel and Windows cache information to speed up subsequent operations

Both Excel and Windows try to optimize performance by storing recent information in memory. This includes the calculation path used by the most recent calculation.

So, Excel's second calculate is often significantly faster than the first.

Windows is a multitasking system so Excel does not get the whole system to itself

Excel also saves some of this information, including the most recent calculation path, with the workbook.

Excel's background calculation process can easily be interrupted

Windows is a multitasking, paged virtual memory operating system so other system tasks are always running at the same time as Excel. Try and make sure there are no other user tasks active whilst you are using **FastExcel.**

Excel's background calculation process can be interrupted if you press any key or click the mouse whilst **FastExcel** is active. This may give you inaccurate timings.

FastExcel V5 can help you quickly find out how much memory Excel is using.

Calling FastExcel Methods from VBA

Calling FastExcel Profiling Methods

FastExcel V5 Profiler has a VBA API that enables you to call the FastExcel Profiling commands from VBA.

You first need to add a reference to your VBA project that references FastExcel V5 Profiler:
Alt-F11 to go to the VBE, then Tools → References and check the box beside fxl4Profiler.

Now in your VBA code you can add statements like

Fxl4Profiler.vbaProfDrillAPI

This calls the Drilldown profiler on whatever is currently the active workbook.
It is not generally a good idea to make this call when your VBA code is within the active workbook, because by default the Profiler closes and then reopens the workbook, thus interrupting your VBA.

The available Profiling APIs are:

fxl4Profiler.gblSilentProfile – a boolean switch that, when True, prevents the Profiling commands from showing messages and user-forms.

fxl4Profiler.gblNotReopenBook – a boolean switch that, when set to True, prevents the Profiling commands from finding worksheet overheads and closing and re-opening the profiled workbook.

Use this switch if you are using the ProfDrillAPI from within the workbook to be profiled.

fxl4Profiler.vbaProfDrillAPI – this runs the FastExcel Drilldown Profiler on the active workbook

fxl4Profiler.vbaProfBookAPI – this runs FastExcel Profile Workbook on the active workbook

fxl4Profiler.vbaProfSheetAPI – this runs FastExcel Profile Worksheet on the active worksheet

fxl4Profiler.vbaProfFormulasAPI – this runs FastExcel Profile Formulas on the active worksheet

fxl4Profiler.vbaProfCheck – this runs FastExcel Check workbook on the active workbook

NOTES:

- **Calling one of these profiling commands will generally change the active workbook.**
- **If you want to call two commands in succession from your VBA, make sure the VBA resets the active workbook and active worksheet appropriately before each command.**
- **When using gblNotReopenBook it is NOT recommended to save the profiled workbook after ProfDrill or ProfBook. Instead close and reopen the workbook manually after Profiling.**

Calling FastExcel Calculation Methods

If you are using any of FastExcel's additional calculation modes and options and your VBA program is requesting Excel calculations you should use FastExcel's enhanced calculation methods.

If you use Excel's standard calculation methods either FastExcel's additional calculation modes and options will not operate or you may get unexpected results.

You can access FastExcel's calculation methods from VBA either by using Application.Run or by setting a reference in your VBA project to SpeedToolsV5: Alt-F11 to go to the VBE, then Tools→References and check the box beside SpeedToolsV5.

The available calculation methods are:

FxlRangeCalc

FxlRangeCalc calculates the selected range on the active sheet in the same way as the FastExcel Calculate Range button.

FxlSheetCalc

FxlSheetCalc recalculates the selected worksheet(s) in the same way as the FastExcel Recalculate Sheets button.

FxlFullSheetCalc

FxlFullSheetCalc full calculates the selected worksheet(s) in the same way as the FastExcel Full Calculate Sheets button.

FxlBookCalc

FxlBookCalc recalculates workbooks in the same way as the FastExcel Calculate Book button.

FxlFullbookCalc

FxlFullBookTime does a full workbook calculation in the same way as the FastExcel Full Calculate button.

FxlDisabledSheetsCalc

FxlDisabledSheetsCalc recalculates workbooks including Mixed mode worksheets.

FxlFullDependCalc

The workbooks will have their dependency trees rebuilt and a full calculation will be done.

MICROTIMER function

The MICROTIMER function is a wrapper function for the Windows high-resolution timer API.

The function returns a double containing seconds. The resolution of the timer is dependent on the clock speed of your PC, but is about 1 microsecond on a 1200MHZ AMD. The function itself takes about 5 microseconds to execute when called from VBA on the same machine.

The function is not volatile, and is primarily designed to be called from VBA Subs/Functions rather than from a worksheet.

MICROTIMER Syntax

MICROTIMER()

MICROTIMER Remarks

To call the function from VBA you must first make a reference from the VBE to FastExcelV5:

- Open the VBE (Alt F11)
- Open a module within your VBA project
- Tools->References and check FastExcelV5.

It is also possible to evaluate the function from VBA as a worksheet function using Evaluate, but this incurs significant extra overhead.

MILLITIMER function

The MILLITIMER function is a wrapper function for the Windows medium-resolution timer API.

The function returns a long containing milliseconds. The resolution of the timer is about 5 milliseconds. The function itself takes about 1 millisecond to execute when called from VBA on the same machine.

The function is not volatile, and is primarily designed to be called from VBA Subs/Functions rather than from a worksheet.

MILLITIMER Syntax

MILLITIMER()

MILLITIMER Remarks

To call the function from VBA you must first make a reference from the VBE to FastExcelV5:

- Open the VBE
- Open a module within your VBA project
- Tools->References and check FastExcelV5.

It is also possible to evaluate the function from VBA as a worksheet function using Evaluate, but this incurs significant extra overhead.

STRCOLID function

The STRCOLID function returns an alphabetic column name (A to XFD) given a column number.

STRCOLID Syntax

STRCOLID(jColNo)

JColNo

jColNo is the number of the column you want to convert to a column name.

STRCOLID Remarks

If the column number is greater than 16384 STRCOLID returns #N/A

STRCOLID can be used either as a worksheet function or from VBA.

The main use of STRCOLID as a worksheet function is to calculate a cell address as a string for use with INDIRECT.

Migrating from FastExcel V4.

Both FastExcel V4 and FastExcel V5 can be installed on the same system whilst testing and migrating workbooks, but it would be better to uninstall FastExcel V4 once this period is over.