TM1 Tools Help File

Table of Contents

TM1 Tools Help File

TM1 Tools Introduction

This help file explains the features of the TM1 Tools add-in for Microsoft Excel (tm1tools.xla). This is a small collection of tools which help when you're using the TM1 OLAP product from IBM / Cognos / Applix, (collectively if informally referred to as "Iboglix") depending on where you are in the timeline.

The add-in was created by Martin Ryan, Alan Kirk and Steve Rowe, with a particular nod to Mike Cowie of Quebit in relation to some of the TM1 API code, and Steve Vincent in relation to changes in the version 1.0 release of the Hierarchy Documenter. A general nod is also offered to the other members of the TM1 community who have contributed feedback and suggestions that have improved the add-in over time.

The authors can be found at www.olapforums.com.

See Also:

- An Outline of the Add-In's Functionality
- The TM1 Tools Options
- The Trace Formulas Tool
- Clearing Formulas
- The Bulk Paste Tool
- The Hierarchy Documenter
- The File Lister Tool
- The File Searcher Tool
- The TM1 API
- Worksheet Functions

Licence Conditions

This tool is provided on the following conditions:

- All rights remain with the original authors. You may use and modify the code for your own use. The code may not be distributed or sold, or incorporated into commercial products, without the express written agreement of the authors.
- The code authors provide this tool for use as a service to the TM1 community. None of the authors or any administrators or members of, or persons or entities associated with www.olapforums.com (whether under that or any future name) or related sites will be liable for any loss or problems stemming from the use of the tool.

If you do not agree to these conditions, please remove the add-in from your computer and discontinue use of it.

Submissions of additional functionality are welcomed from the TM1 user base, though we'd ask that:

- All such code be clearly documented; and
- If any modifications have been made to existing code, that it's clearly noted. As the tool becomes more expansive, there's a risk of breaking something if you modify existing code to work with your own addition.

Outline Of The Add-In's Functionality

As of Version 1.0 (February 2011) the add-in offers:

- A DBRW/DBSW Trace Formulas Tool which allows you to identify the elements that such a worksheet function is pointing to;
- Three ways to convert formulas to values; two of which operate on TM1-specific functions only (such as DBRW, SubNm etc), and one of which will remove all formulas regardless of whether they are TM1-specific. See the topic **Clearing TM1 Formulas**.
- An option for pasting copied data from the Windows Clipboard into a range of TM1 DBR/DBRW cells. This is called the Bulk Paste Tool.
- A "one-click" option for toggling between automatic and manual calculation modes, including having a toolbar icon which gives you a visual cue about the mode that's currently in use. See below.
- An option to change the current path to the path of the file that you have open.
- An option to open Windows Explorer to the path of the file that you have open.
- An option to create a graphical chart of one or more of the consolidations in a dimension and, optionally, to list the elements in a single column for filtering and to return an attribute of the elements. See the topic The Hierarchy Documenter.
- The ability to time the calculation of a single sheet or a whole workbook. (See the topic The Timers.)
- The ability to list the files in a folder or tree of folders (plus related information like age and size) into an Excel workbook. See the topic The File Lister Tool.
- The ability to load text files (including .pro and .rux files) into an excel workbook and then search them for a specified text expression. See the topic File Searcher.
- Wrapper functions which call some of the TM1 API functionality, and which can be called from your own code if you set a reference to the TM1 Tools project. (Discussed under the heading TM1 API .) There are three at the moment:
    - A function which you can call to execute a TurboIntegrator process. See the topic RunTIProcess.
    - A function which will return a list of the names of the available servers as a string array. See the heading GetServerList
    - A function which will return a collection of custom TM1Client objects which have information about clients and the security groups that they are in. This function will only work for users with Admin access to the server. See the heading GetClientsCollection.
- Two additional worksheet functions:

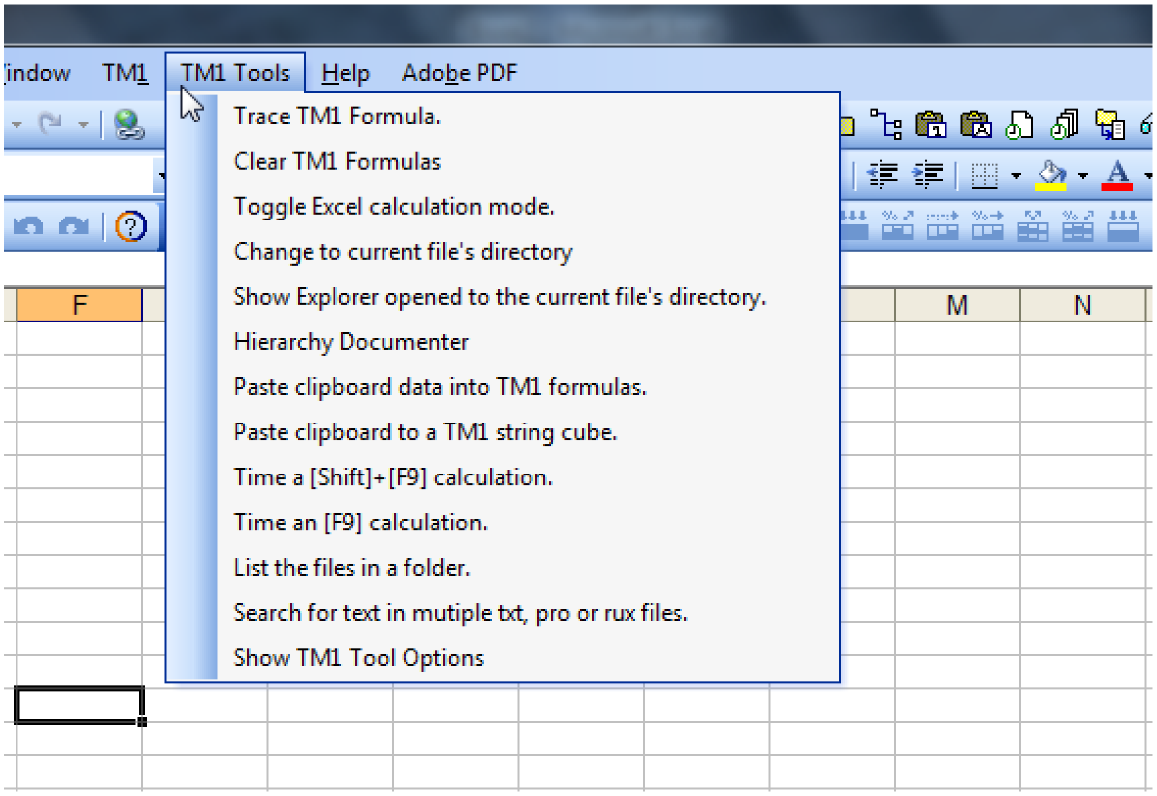
- o ElIsAnc since IBM seems unwilling to get off its backside and provide that itself; and
- o IsFolderValid which returns True if a folder name exists on the local computer or False if it doesn't.

This list may expand over time if there is enough interest.

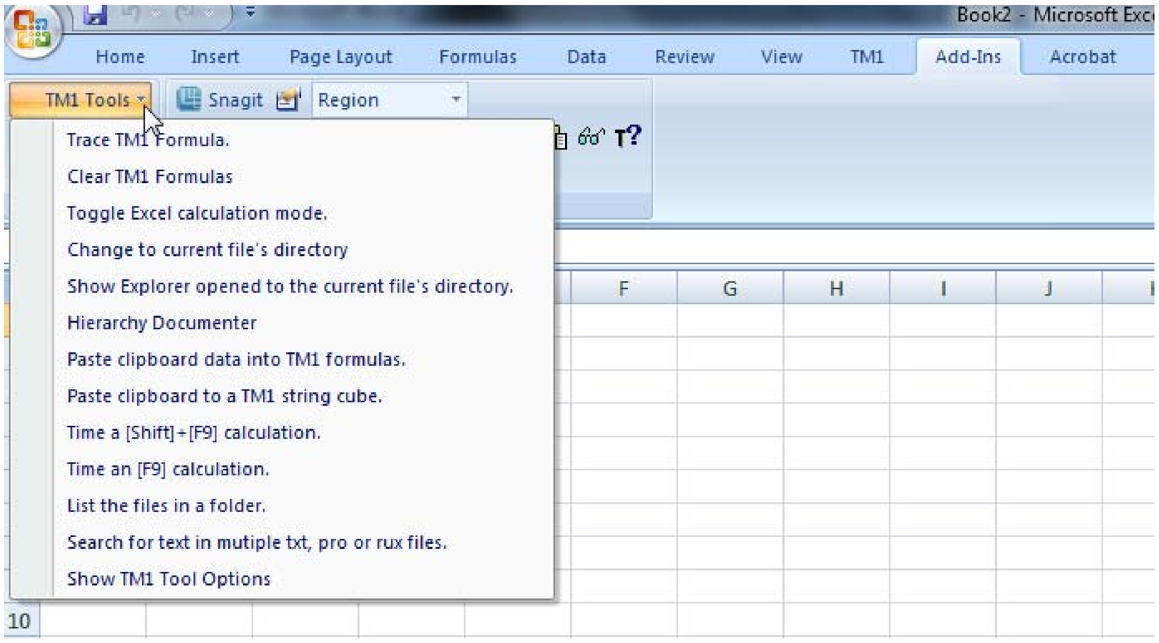
The Tool Bar And Menu

The tool bar currently consists of about a dozen different icons representing different tool functions. By default only a subset of them will be visible when you first use the tool. You can determine which icons are visible via the TM1 Tools Options form, which can be accessed through the last button on the toolbar. (That button can't be switched off, for obvious reasons.)

In version 1.0 of the tool a menu was also added to the standard worksheet menu bar by default, though it can be turned off via the Options. That menu shows all of the available tools, regardless of whether you have the corresponding toolbar button on or off.


indow
TM1
TM1 Tools
Help
Adobe PDF
Trace TM1 Formula.
Clear TM1 Formulas
Toggle Excel calculation mode.
Change to current file's directory
Show Explorer opened to the current file's directory.
Hierarchy Documenter
Paste clipboard data into TM1 formulas.
Paste clipboard to a TM1 string cube.
Time a [Shift]+[F9] calculation.
Time an [F9] calculation.
List the files in a folder.
Search for text in mutiple txt, pro or rux files.
Show TM1 Tool Options
F
M
N

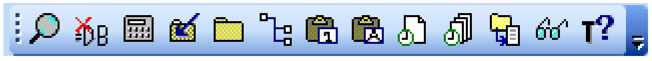
In Excel 2007, that menu will appear in the Add-Ins tab on the ribbon.


Book2 - Microsoft Exce
Home
Insert
Page Layout
Formulas
Data
Review
View
TM1
Add-Ins
Acrobat
TM1 Tools
Snagit
Region
Trace TM1 Formula.
Clear TM1 Formulas
Toggle Excel calculation mode.
Change to current file's directory
Show Explorer opened to the current file's directory.
Hierarchy Documenter
Paste clipboard data into TM1 formulas.
Paste clipboard to a TM1 string cube.
Time a [Shift]+[F9] calculation.
Time an [F9] calculation.
List the files in a folder.
Search for text in mutiple txt, pro or rux files.
Show TM1 Tool Options

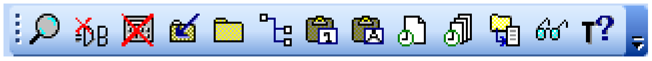
In pre-release versions of TM1 Tools the button icons varied depending on your version of Windows and Excel. The icons are now incorporated into the tool so they should be consistent across any platform.

The following shot shows all of the icons on the version 1.0 toolbar when you're in manual calculation mode. More may be added in future versions but since (a) I'm sick of having to do screenshots each release, and (b) the exact appearance of the toolbar will vary according to which tools you have on, take it as read that the full list may vary in later versions:

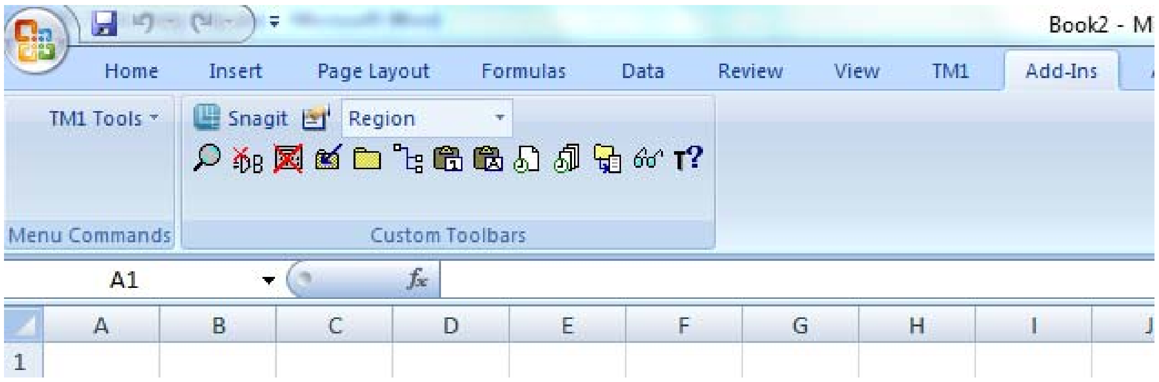
Excel up to 2003, Manual Calculation:



Automatic Calculation:



In Excel 2007 the icons will appear on the Add-Ins tab along with the Menu (if enabled):



Trace

This icon summons the Trace Formula tool, which allows you to tell what arguments make up a TM1 DBRW (for example) formula.

Clear Formulas

This icon allows you to replace TM1-specific formulas (or, depending on the option selected, all formulas) from the active workbook (or, again depending on the option, a single sheet). See the section titled Clearing TM1 Formulas.

Toggle Calc Mode

This button will have one of two faces.

If you're currently in manual calculation mode the icon will look like a calculator, showing you that you can switch to automatic calculation mode.

If you're in automatic or semi-automatic mode, the calculator is crossed out telling you that you can switch to manual calculation mode.

CAUTION! The button can't track whether you manually change the calculation mode via the Options menu in Excel, or whether other VBA code does it. Its icon is only refreshed when you run any TM1 Tools function. However it will always toggle the calculation mode to the alternative state from its current one (manual to automatic, or automatic to manual) regardless of the icon displayed. The icon is therefore more of a guideline than something that you should rely on.

Change To Current File's Directory

Also known as Directory Quickchange. This is useful if you've opened a file from a shortcut, recently used files list or Outlook Journal entry. In some cases you may want to switch to that file's folder so that you can open other related files without needing to drill up and down the trees in Windows Explorer. This button will change your current file directory to that of the active file (if any).

Show Explorer

This icon will launch Windows Explorer to the folder that contains the currently active file. If there are no files opened or the file hasn't been saved, it will launch to the standard Explorer location.

The Hierarchy Documenter

This tool will allow you to generate a graphical representation of one or more consolidations in your dimension into an Excel workbook. See the topic titled The Hierarchy Documenter.

The Bulk Paste Tool

This tool will take the data that you have copied to the clipboard, and paste it into a range of DBR or DBRW formulas, thus sending it to the cube. The icon on the left works with numeric data or string (text) data that is not in numeric format. The one on the right forces the values to be sent as strings, regardless of the format of the data. See the topic titled Bulk Paste Tool.

The Timers

The one on the left does a [Shift] + [F9] calculation. The one on the right does a full [F9] calculation (hence the larger icon). Both use the TM1 calculation macro, not the native Excel methods which means that you can't use them if the TM1 add-in isn't loaded. (But then, that's true of most of the tool's functionality.) When the calculation has been done a message box will display showing the time taken to do it. For advanced users who want to run and compare multiple tests, you'll find the information also written to the Immediate window in the Visual Basic Editor.

The File Lister Tool

This icon launches the File Lister tool, which allows you to generate a list of files (or a subset of files) from a folder into an Excel worksheet.

The Multi File Search Tool

This allows you to load multiple .pro, .rux or .txt files into a single workbook. You can then search that workbook and have the search terms highlighted.

See the topic File Searcher.

Options

This button brings up the TM1 Tools Options dialog.

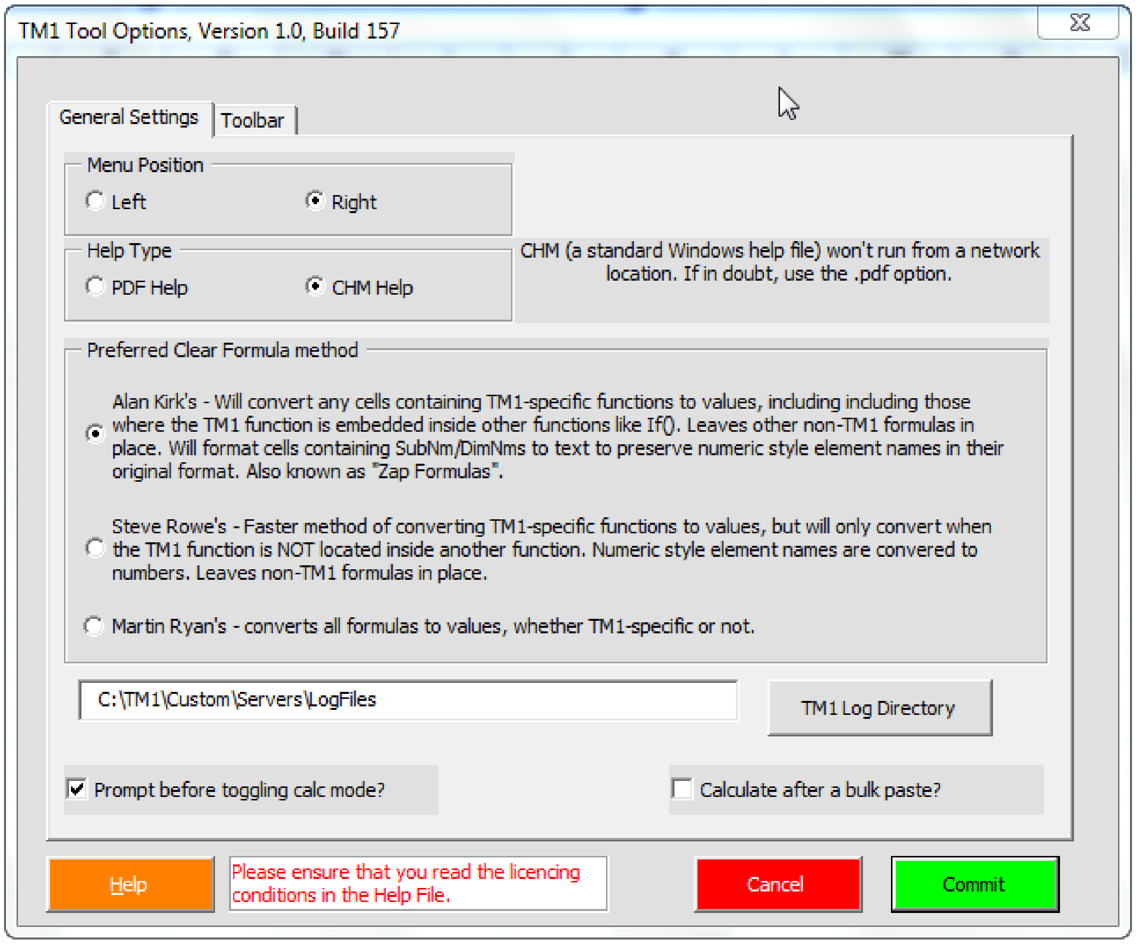
Setting User Options

TM1 Tools Options

The options for TM1 Tools can be set via the following icon:

T?

That launches the Options form shown below. The form now consists of two tabs.



TM1 Tool Options, Version 1.0, Build 157

Menu Position: Left [✓ Right]

Help Type: PDF Help [✓ CHM Help]

CHM (a standard Windows help file) won't run from a network location. If in doubt, use the .pdf option.

Preferred Clear Formula method

- [✓] Alan Kirk's - Will convert any cells containing TM1-specific functions to values, including including those where the TM1 function is embedded inside other functions like If(). Leaves other non-TM1 formulas in place. Will format cells containing SubNm/DimNms to text to preserve numeric style element names in their original format. Also known as "Zap Formulas".
- Steve Rowe's - Faster method of converting TM1-specific functions to values, but will only convert when the TM1 function is NOT located inside another function. Numeric style element names are convered to numbers. Leaves non-TM1 formulas in place.
- Martin Ryan's - converts all formulas to values, whether TM1-specific or not.

C:\TM1\Custom\Servers\LogFiles

- [✓] Prompt before toggling calc mode?
- [ ] Calculate after a bulk paste?

Please ensure that you read the licencing conditions in the Help File.

Most of the options are fairly self explanatory.

Title Bar

The title bar shows the version and build number. (The build number is merely the number of times that the file has been saved (which, obviously, is "frequently"), so it won't increment by 1 with each release.)

Menu Position

The TM1 Tools toolbar will try to position itself relative to the Standard toolbar in Excel 97 to 2003. (In Excel 2007 it's stuck in the Add-Ins tab on the non-customisable Office Ribbon, and there's nothing we can do about that.)

If the Standard toolbar is visible you have the choice of setting the TM1 Tools toolbar on the right or left of the Standard one. Your choice will be remembered the next time you start Excel.

(The toolbar is automatically generated when the add-in loads, so setting the position via the Excel GUI won't work.)

If the Standard toolbar is not visible, then the TM1 Tools toolbar will be a floating one. The Menu Position radio buttons will be disabled on the Options form.

Help Type

This selection determines how you'll be seeing this help file.

In their wisdom Microsoft have crippled most help systems to the point of being unusable, at least from a network location. If you open a compiled help file (.chm) from a network location, the content won't be visible unless you've modified some entries in your computer's registry to allow it.

HTML help operates under many similar restrictions.

If you have the add-in installed locally, you can use the .chm Windows help file.

If you're picking it up from your network, you can still get at least a form of help via an Adobe .pdf document.

In both cases the file must be located in the same folder as the add-in itself.

Note that if you're using Excel's "Insert Function" dialog to insert the ElIsAnc function, the "Help On This Function" link will only work if you are able to use the .chm file. The setting that you make in the Options doesn't affect that; the link will try to use the .chm even if you have .pdf selected here.

Preferred Clear Formula Method

The clear formula methods are discussed under the heading Clearing TM1 Formulas. By default the Alan Kirk method is used, but you can change to your preferred method via this form.

TM1 Log Directory

This is used in conjunction with the RunTIProcess wrapper function which goes around the TM1 API, and allows you to execute TurboIntegrator processes from your own Visual Basic code. If you have access to the directory that TM1 writes its log files to, you can specify that directory here, and will then be able to have any error logs presented to you upon completion of the process.

This is discussed further under the topic RunTIProcess.

Prompt Before Toggling Calc Mode

There's a button on the TM1 Tools toolbar which will toggle your Excel calculation mode between Automatic and Manual. (This is described under the heading Toggle Calc Mode.)

By default the button will ask you to confirm that you want to swap modes before it does that. You can turn the prompt off here if you don't want it to bother you again.

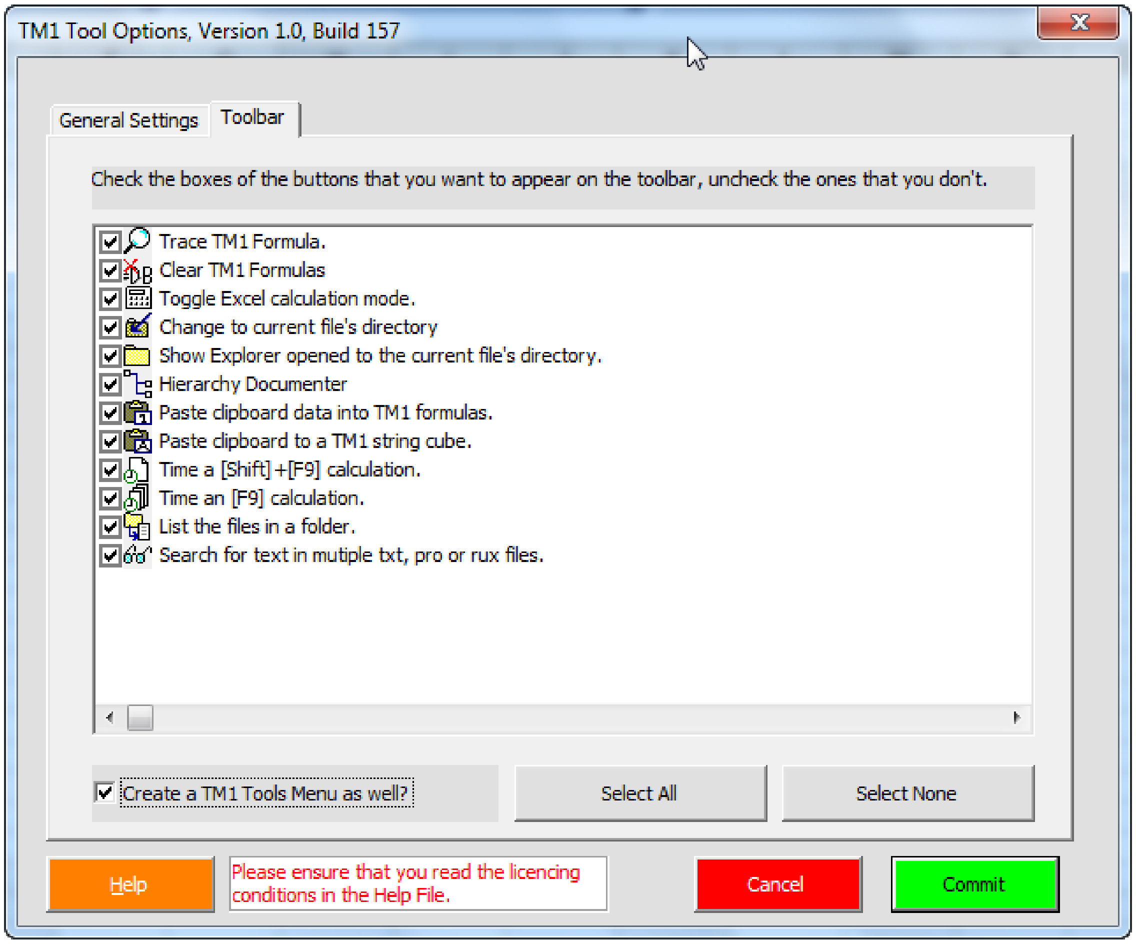
Calculate After A Bulk Paste

Used in conjunction with the Bulk Paste tool. See that heading for more details. In short, when you paste a range of values into TM1 formulas using the Bulk Paste tool you can either have the worksheet recalculated after the paste has been done, or not.

Help

The [Help] button launches this file.

The Toolbar Tab



Check the boxes of the buttons that you want to appear on the toolbar, uncheck the ones that you don't.

- [✓] Trace TM1 Formula.
- [✓] Clear TM1 Formulas
- [✓] Toggle Excel calculation mode.
- [✓] Change to current file's directory
- [✓] Show Explorer opened to the current file's directory.
- [✓] Hierarchy Documenter
- [✓] Paste clipboard data into TM1 formulas.
- [✓] Paste clipboard to a TM1 string cube.
- [✓] Time a [Shift]+[F9] calculation.
- [✓] Time an [F9] calculation.
- [✓] List the files in a folder.
- [✓] Search for text in mutiple txt, pro or rux files.

[✓] Create a TM1 Tools Menu as well?

Please ensure that you read the licencing conditions in the Help File.

You can select any or all of the buttons on the toolbar to be shown, with the exception of the TM1 Options button (which is always visible). The buttons can be quickly checked or cleared using the [Select All] and [Select None] buttons at the bottom.

The ***Create a TM1 Tools Menu as well?*** checkbox determines whether a menu will be added to the standard menu bar in Excel versions up to 2003, or to the Add-Ins tab in Excel 2007. This menu is not affected by the toolbar buttons that you select; it will always show all of the commands available in TM1 Tools, and the menu items will be in text rather than icon format.

Tracing The Arguments of a TM1 Formula

Trace Formula Tool

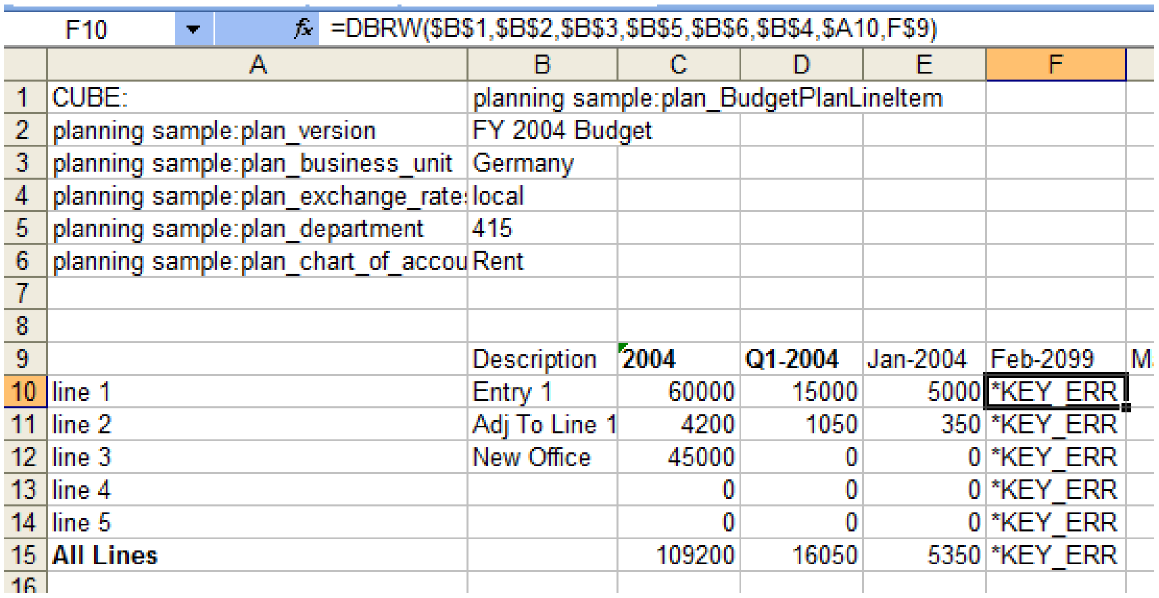
Most TM1 reports begin life as a slice. They're later modified to meet the user's needs; cells are shifted around, formatted and so on.

Unfortunately in the process of doing that the TM1 DBRW formulas can get corrupted, and may end up pointing to the wrong place.

Slices use cell references rather than range names, which can make tracing errors rather difficult since you have to look at each argument individually.

To help overcome this, the Trace Tool will show you all of the arguments that make up a DBR/DBRW formula. If any are invalid, they'll be highlighted.

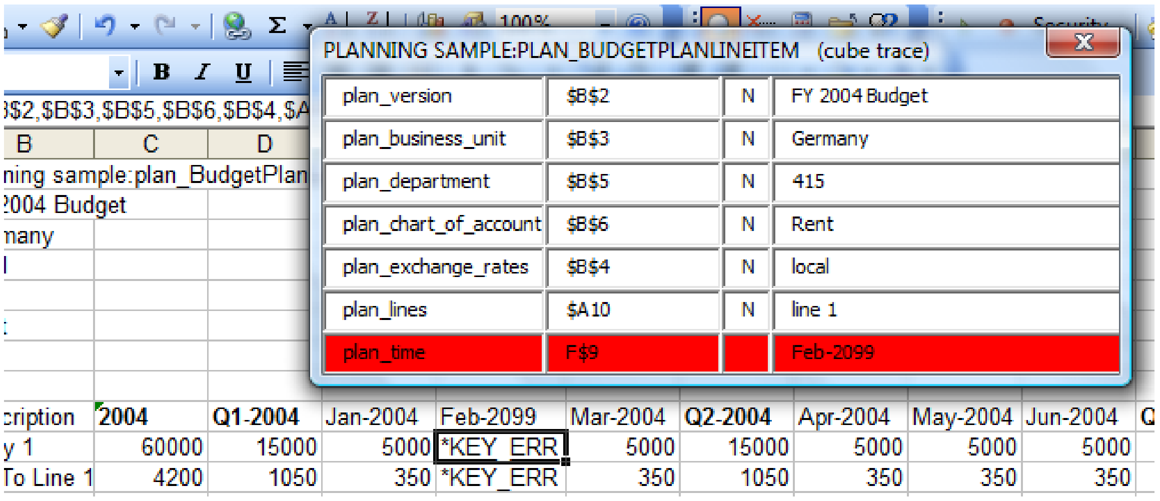
For example, some of the formulas in the sheet below have key errors. The formula bar shows the arguments in the first of these. To find which one(s) are wrong you'd have to check each cell individually, then check them against the elements in each dimension.



F10 =DBRW(B1,B2,B3,B5,B6,B4,$A10,F$9)

	A	B	C	D	E	F	
1	CUBE:	planning sample:plan_BudgetPlanLineItem					
2	planning sample:plan_version	FY 2004 Budget					
3	planning sample:plan_business_unit	Germany					
4	planning sample:plan_exchange_rate:	local					
5	planning sample:plan_department	415					
6	planning sample:plan_chart_of_accou	Rent					
7							
8							
9		Description	**2004**	**Q1-2004**	Jan-2004	Feb-2099	M
10	line 1	Entry 1	60000	15000	5000	*KEY_ERR	
11	line 2	Adj To Line 1	4200	1050	350	*KEY_ERR	
12	line 3	New Office	45000	0	0	*KEY_ERR	
13	line 4		0	0	0	*KEY_ERR	
14	line 5		0	0	0	*KEY_ERR	
15	**All Lines**		109200	16050	5350	*KEY_ERR	
16							

Alternatively, you can use the Trace Tool which will show you a grid like this:



PLANNING SAMPLE:PLAN_BUDGETPLANLINEITEM (cube trace)

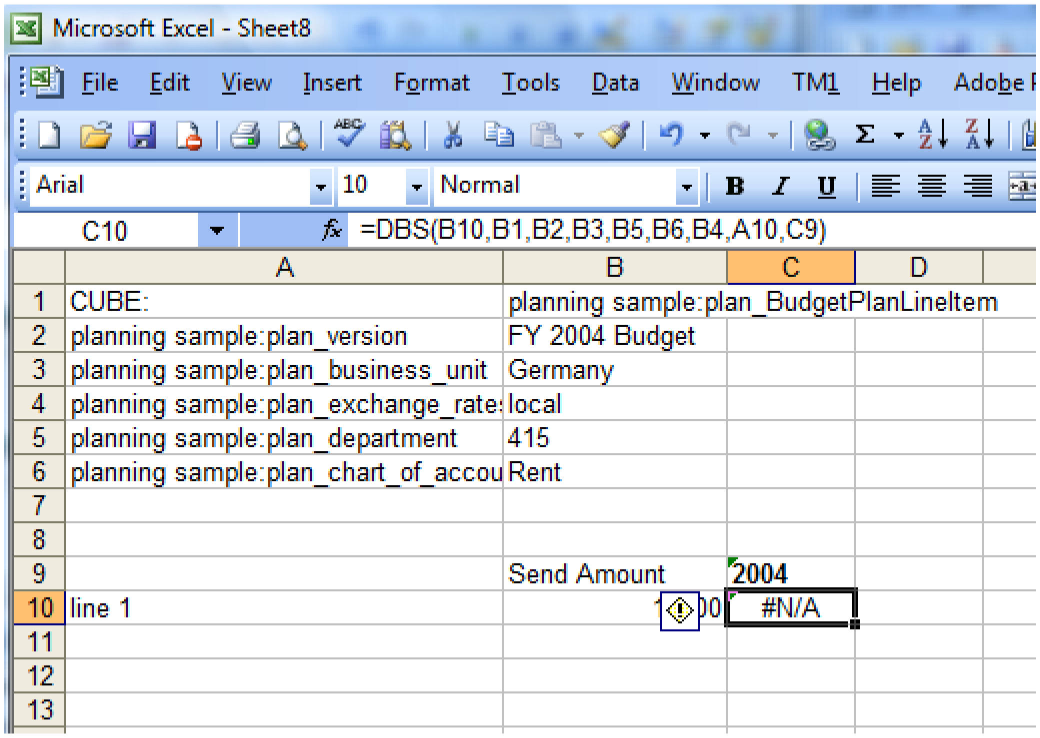
plan_version	B2	N	FY 2004 Budget
plan_business_unit	B3	N	Germany
plan_department	B5	N	415
plan_chart_of_account	B6	N	Rent
plan_exchange_rates	B4	N	local
plan_lines	$A10	N	line 1
plan_time	F$9		Feb-2099

cription	2004	Q1-2004	Jan-2004	Feb-2099	Mar-2004	Q2-2004	Apr-2004	May-2004	Jun-2004
y 1	60000	15000	5000	*KEY_ERR	5000	15000	5000	5000	5000
To Line 1	4200	1050	350	*KEY_ERR	350	1050	350	350	350

This shows that it's the plan_time element which is the problem; the element specified (Feb-2099) does not exist in the dimension.

The Trace Tool also works on DBS/DBSW formulas. If the element is invalid, it will be highlighted in the same way as a DBR/DBRW one. However not all problems are highlighted as obviously.

Take the following example:

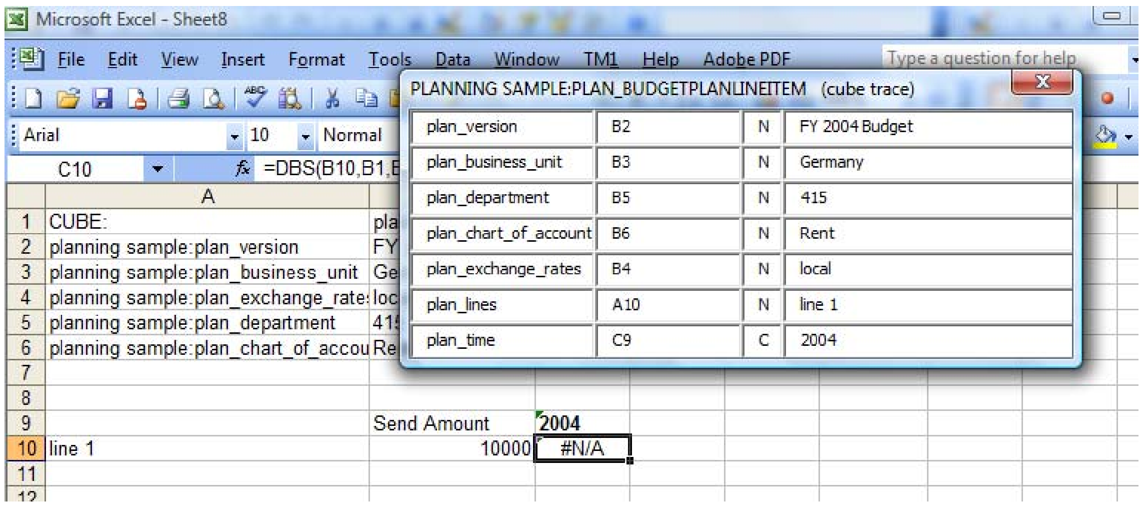


Microsoft Excel - Sheet8

C10 =DBS(B10,B1,B2,B3,B5,B6,B4,A10,C9)

	A	B	C	D
1	CUBE:	planning sample:plan_BudgetPlanLineItem		
2	planning sample:plan_version	FY 2004 Budget		
3	planning sample:plan_business_unit	Germany		
4	planning sample:plan_exchange_rate:	local		
5	planning sample:plan_department	415		
6	planning sample:plan_chart_of_accou	Rent		
7				
8				
9		Send Amount	2004	
10	line 1	1 00	#N/A	
11				
12				
13				

The DBS formula is returning an #N/A error. The Trace Tool won't highlight any of the elements as having an error and yet if you look closely it does tell you what the problem is:



	A	B	C
1	CUBE:	pla	
2	planning sample:plan_version	FY	
3	planning sample:plan_business_unit	Ge	
4	planning sample:plan_exchange_rate:	loc	
5	planning sample:plan_department	41	
6	planning sample:plan_chart_of_accou	Re	
7			
8			
9		Send Amount	2004
10	line 1	10000	#N/A
11			

PLANNING SAMPLE:PLAN_BUDGETPLANLINEITEM (cube trace)

plan_version	B2	N	FY 2004 Budget
plan_business_unit	B3	N	Germany
plan_department	B5	N	415
plan_chart_of_account	B6	N	Rent
plan_exchange_rates	B4	N	local
plan_lines	A10	N	line 1
plan_time	C9	C	2004

The third column tells you what type each element is. In this case the last one is a C type… and you can't write to a consolidation.

Converting Formulas To Values

Clearing TM1 Formulas

TM1 uses a number of special Excel functions which derive from the tm1p.xla and tm1.xla add-ins. As long as the user has those add-ins loaded in their Excel session and are logged into a TM1 server, the functions provide and/or allow dynamic updates of the data in TM1 cubes (DBR, DBRWs), will send values to the cubes (DBS/DBSW), return elements from the dimensions (DimNm, SubNm) and so on.

The problem is that if you send a workbook to a user who does ***not*** have TM1 installed, then all of the formulas which use those functions are going to evaluate as #NAME error values because without the TM1 add-ins Excel will have no idea what (for example) a DBRW function is.

Consequently TM1 Tools allows you to replace formulas with hard coded values, so that the workbook can be sent to non-TM1 users.

There are three methods available for you to do this. Your preferred one can be set from the **Options form**.

The methods are:

- The Alan Kirk Method (a.k.a. Zap Formulas); replaces formulas which contain TM1-specific functions with values, including those where the TM1 functions embedded inside other functions.
- The Steve Rowe Method; very fast method of removing TM1-specific functions provided that they aren't inside or following another function.
- The Martin Ryan Method; Removes ***all*** formulas, regardless of whether they contain TM1-specific functions.

Method 1: The Alan Kirk Method, a.k.a. Zap Formulas

Advantages

- It will clear only formulas which contain TM1 functions. If you have any other formulas (such as Sum, SubTotal, etc), those will be left in place. (This may be a disadvantage if you want a completely frozen snapshot of the workbook.)
- For any SubNm / DimNm element names which are numeric in format, the cell will be formatted as text before the replacement is done. That means that cost centre 0004 won't suddenly become 4.
- It allows you to select whether you want the formulas to be zapped in only a single sheet or the entire workbook, and gives you several other options as well. (See *How Do You Use It?* below.)
- It's the only method which will remove ***all*** TM1 functions (regardless of where they are in the cell formula) yet leave non-TM1 functions and formulas in tact.

Disadvantages

- Because it has to filter out which cells contain TM1 functions somewhere (anywhere) in the cell, it's the slowest of the three methods. (Though this is relative; in tests it cleared a workbook of 1,644 rows by 22 columns with 24,479 formula cells in 11 seconds. This compared with 1 minute 2 seconds for a comparable commercial product.)

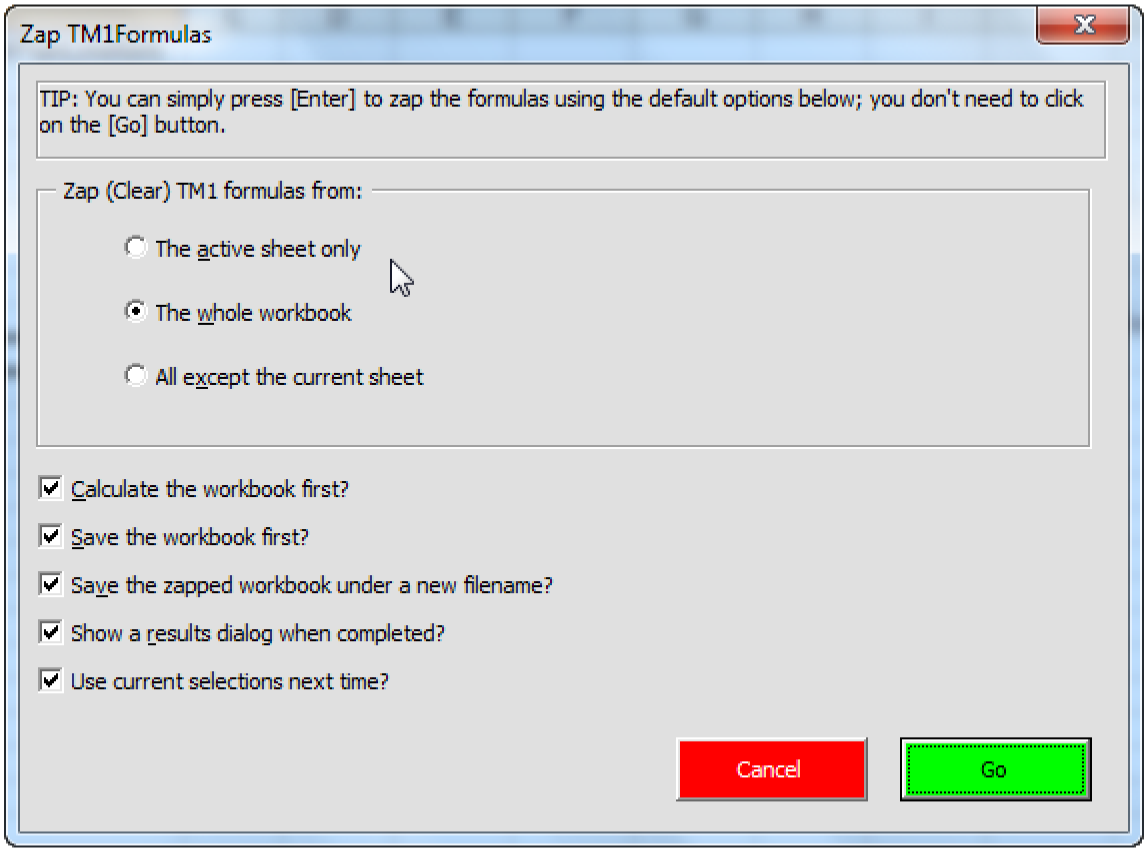
How Do You Use It?

This is the default mode in the current version of the software and will be selected automatically if you haven't previously made an alternative selection.

If you have, click on the Options form toolbar button and select this method on the Options form.

That will automatically assign the method to the "Convert TM1 Formulas to Values" toolbar button.

When you click on that button it will launch an options form as shown below.


Zap TM1Formulas

TIP: You can simply press [Enter] to zap the formulas using the default options below; you don't need to click on the [Go] button.

Zap (Clear) TM1 formulas from:

- ○ The active sheet only
- ● The whole workbook
- ○ All except the current sheet

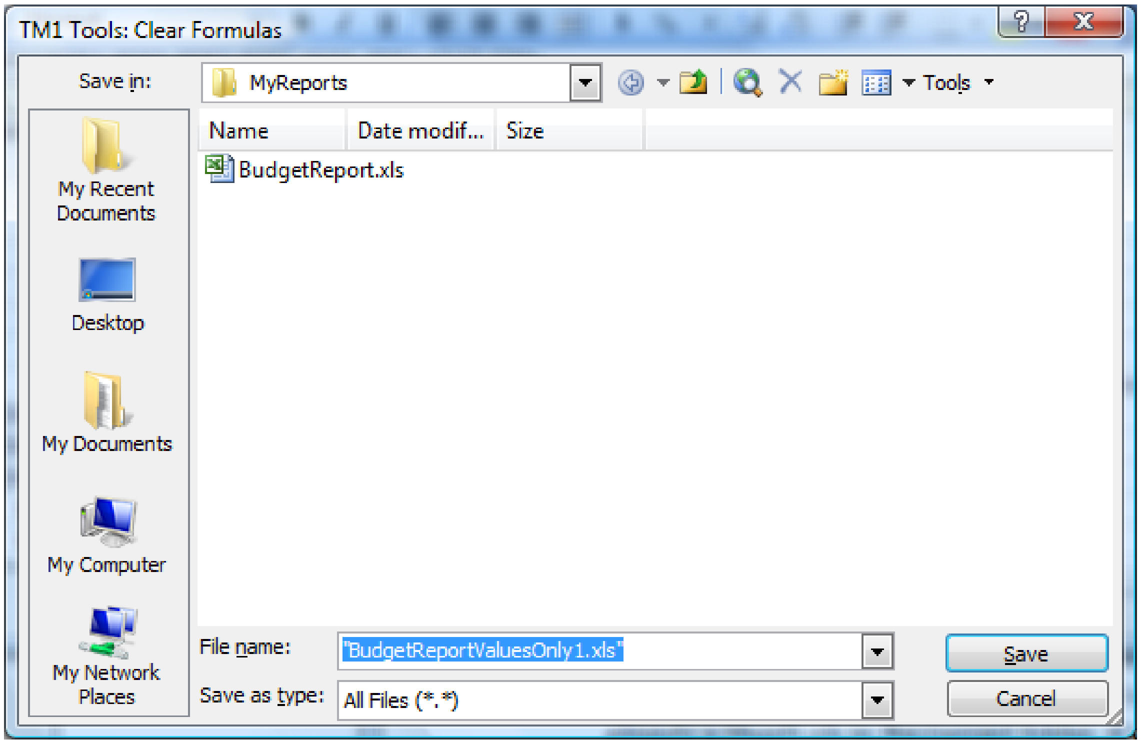
- ☑ Calculate the workbook first?
- ☑ Save the workbook first?
- ☑ Save the zapped workbook under a new filename?
- ☑ Show a results dialog when completed?
- ☑ Use current selections next time?

These options are largely self-explanatory:

- ***Zap TM1 Formulas From***: You can choose to clear the formulas from the active sheet alone, the entire workbook, or (new in version 1.0) all of the sheets except the current one. (The last option being useful for when you're keeping a reconciliation workbook, for example; you may need to keep the current sheet live but snapshot the historical ones.) Consideration was given to allowing users to zap only a specified range, but I can't think of an instance where that would ever be used in practice.
- ***Calculate the workbook first?*** You have the option of recalculating the workbook first to ensure that the values are up to date before the formulas are removed. This will in fact do a universal calculation (the same as pressing [F9]) so make sure that you don't have any large workbooks hanging around in the background. Also if you elect to have this on, make sure that you're connected to the TM1 server.
- ***Save the workbook first?*** There is no option to undo the zapping of TM1 formulas. Instead this checkbox allows you to do a final save of the workbook before the formulas are removed. If you change your mind and have checked this box, you can just close the zapped workbook and reopen the one containing the formulas.

  - *What if it's a new slice?* If it's a fresh slice which hasn't been saved and has a name like "Sheet1", the program will save it as Sheet1.xls. If there's already a Sheet1.xls in the current folder, it will ask you to confirm that you want to overwrite it. If you select [No] or [Cancel], then the process will error out and a dialog will ask you to save the sheet manually.

- *Save the zapped workbook under a new filename?* If this is selected then the program will make a copy of the workbook under a new name. It will suggest a default name (as shown in the dialog below) but you're free to change it. By default it will be the original name followed by `ValuesOnly1`. If there's already a file of that name it will suggest `ValuesOnly2`, then `ValuesOnly3` and so on. (NB: The other two methods do not use the same system for assigning the default file name.)

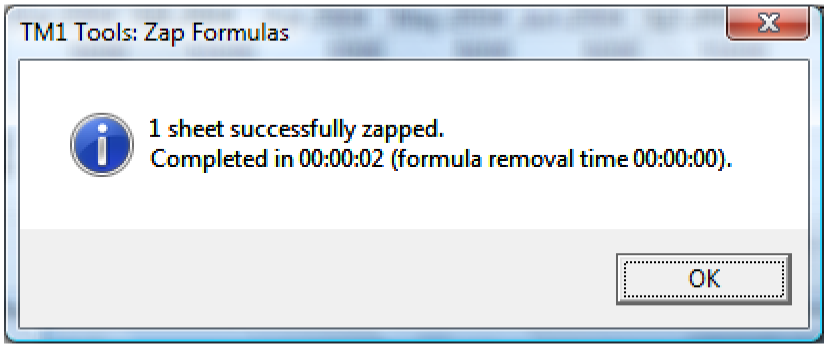


BudgetReport.xls

File name: "BudgetReportValuesOnly1.xls"

Save as type: All Files (*.*)

- *Show a results dialog when completed?* If checked, the program will present a dialog telling you how many sheets were zapped and how long it took.



1 sheet successfully zapped.
Completed in 00:00:02 (formula removal time 00:00:00).

- ***Use current selections next time?*** I normally zap only the current worksheet and got sick of having to remember to reselect that after I'd zapped a whole workbook. From version 1.0 onwards you can now choose whether or not to keep your current selections the next time you run the tool. If you don't, it will default to the last saved settings next time you run it.

Accelerator Keys

Note that each of the controls on the form have an accelerator key denoted by an underscore character somewhere in the name. You use that character with the `[Alt]` key to make your selections from the keyboard instead of with the mouse. For example, `[Alt]` + `[S]` will toggle the Save Workbook First? option on and off.

Method 2: The Steve Rowe Method

Advantages

- It will clear only TM1 functions. If you have any other formulas (such as Sum, SubTotal, etc), those will be left in place. (This may be a disadvantage if you want a completely frozen snapshot of the workbook.)
- It's extremely fast.

Disadvantages

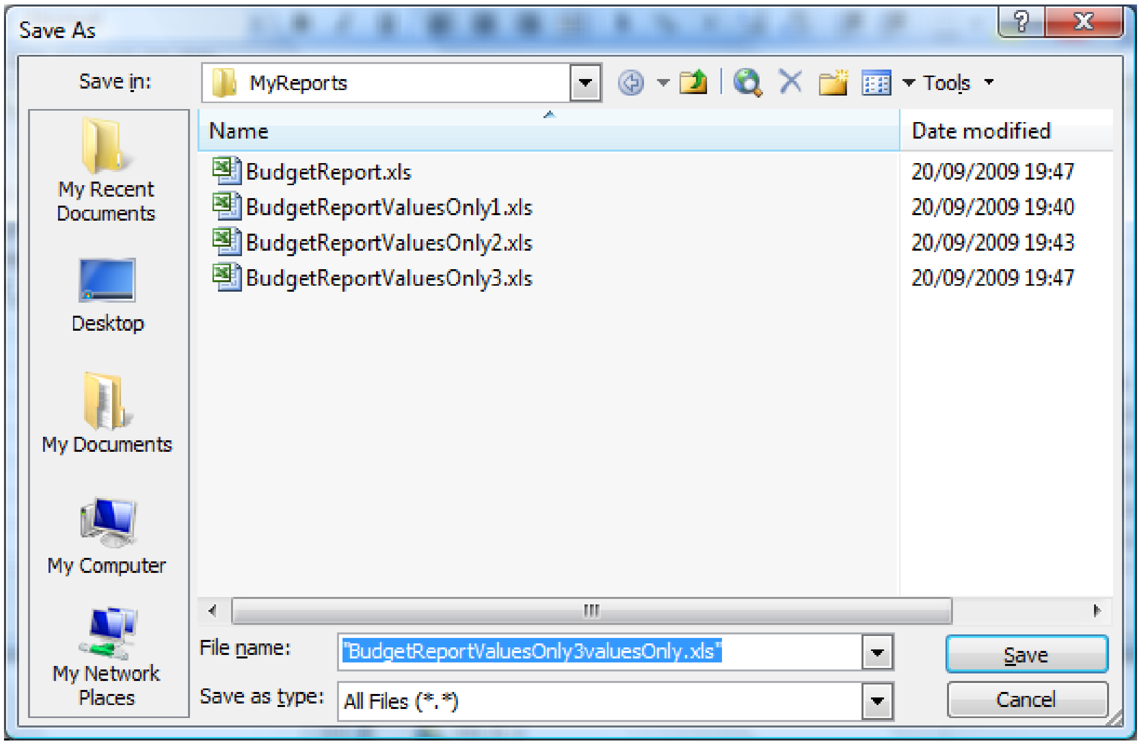
- It will remove TM1 formulas from the whole workbook; you can't specify that you want it for a single sheet only. This may be an advantage if you always want to do whole books.
- For any SubNm / DimNm element names which are numeric in format, the cell will ***not*** be formatted as text before the replacement is done. That means that a cost centre like 0004 will appear as 4.
- If you have TM1 functions inside (for example) an If() function or following another function, those will not be removed.

How Do You Use It?

Click on the Options form toolbar button and select this method from the Options form.

That will automatically assign the method to the "Convert TM1 Formulas to Values" toolbar button.

When you click on that button it will ask you for a filename to save the workbook with the removed formulas under:



Name	Date modified
BudgetReport.xls	20/09/2009 19:47
BudgetReportValuesOnly1.xls	20/09/2009 19:40
BudgetReportValuesOnly2.xls	20/09/2009 19:43
BudgetReportValuesOnly3.xls	20/09/2009 19:47

File name: "BudgetReportValuesOnly3valuesOnly.xls"

Save as type: All Files (*.*)

This is not quite the same as the Alan Kirk method, which adds an incrementing number at the end of the file name. Instead it adds `ValuesOnly` to the end of the filename. Note that it does this after each save. If you have a workbook that you're constantly adding new slices to then clearing the formulas from, you could potentially end up with a very log filename. ***However*** you don't have to accept the name suggested, and can substitute a different one.

After that the TM1 formulas are cleared. There is no dialog telling you when it's done and in most cases you don't need one.

Method 3: The Martin Ryan Method

Advantages

- It will clear ***all*** functions. Whether this is an advantage or disadvantage depends on whether you need other functions to continue working.
- It's extremely fast.

Disadvantages

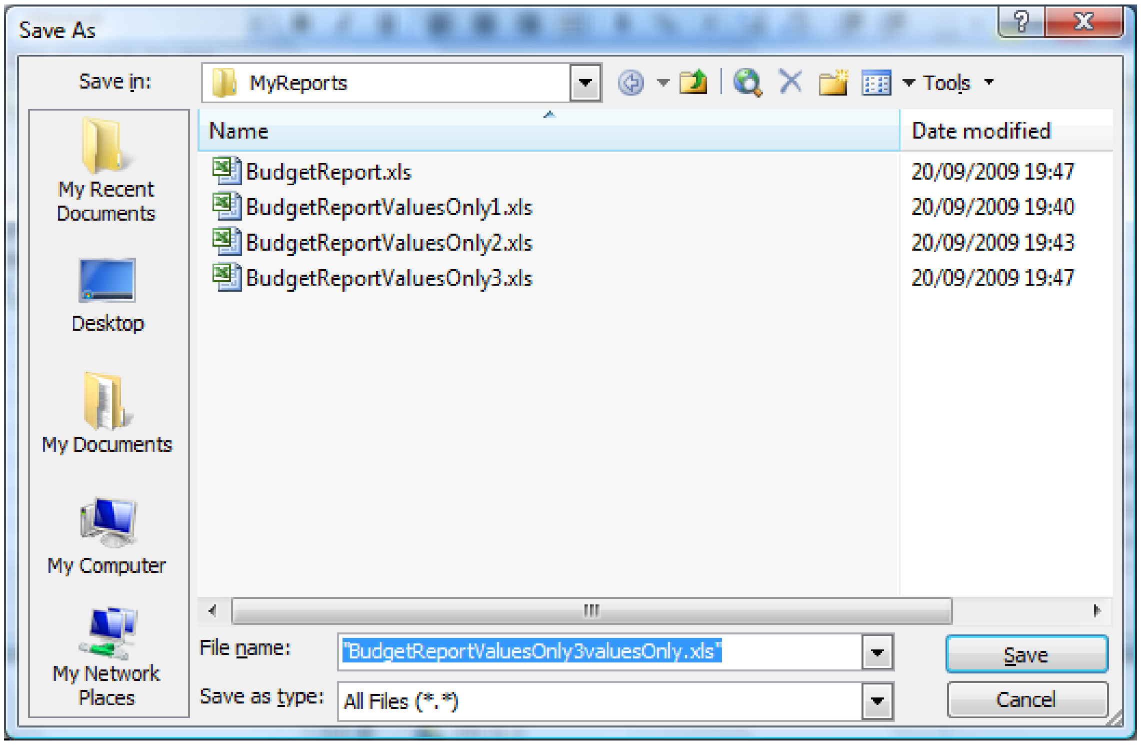
- It will clear ***all*** functions. Whether this is an advantage or disadvantage depends on whether you need other functions to continue working.
- It will remove TM1 formulas from the whole workbook; you can't specify that you want it for a single sheet only. This may be an advantage if you always want to do whole books.
- For any element names which are numeric in format, the cell will ***not*** be formatted as text before the replacement is done. That means that a cost centre like 0004 will appear as 4.

How Do You Use It?

Click on the Options form toolbar button and select this method from the Options form.

That will automatically assign the method to the "Convert TM1 Formulas to Values" toolbar button.

When you click on that button it will ask you for a filename to save the workbook with the removed formulas under:



Name	Date modified
BudgetReport.xls	20/09/2009 19:47
BudgetReportValuesOnly1.xls	20/09/2009 19:40
BudgetReportValuesOnly2.xls	20/09/2009 19:43
BudgetReportValuesOnly3.xls	20/09/2009 19:47

File name: "BudgetReportValuesOnly3valuesOnly.xls"

Save as type: All Files (*.*)

This is not quite the same as the Alan Kirk method, which adds an incrementing number at the end of the file name. Instead it adds `ValuesOnly` to the end of the filename. Note that it does this after each save. If you have a workbook that you're constantly adding new slices to then clearing the formulas from, you could potentially end up with a very log filename. ***However*** you don't have to accept the name suggested, and can substitute a different one.

After that the TM1 formulas are cleared. There is no dialog telling you when it's done and in most cases you don't need one.

Pasting Values Into TM1 DBR Type Functions

Bulk Paste Tool

Introduction

This tool allows you to:

- Copy a range of data into the Windows clipboard. This need not be from another Excel range, but can be from a Word document, or a Web page for example. The only requirement is that the range that you copy must be an exact rectangle. You can't copy from a table which has two columns on the first row and four on the second.
- Paste that data into a block of TM1 DBR, DBRW or DBRA functions (which we'll collectively call DBR-type functions for brevity), and have it sent to the cube(s) that the functions refer to.

Note that the DBR type function must be at the ***start*** of the cell. You can't have them nested inside If() or other functions. This is by design; it would be too risky to try sending values to expressions which may not in fact be intended for data entry. If there are other expressions beyond the DBR (for example `=DBRW($A$1, $A3, B$2) / 1000`), the extra expressions won't affect the send operation. The only requirement is that the =DBRW (or =DBR or =DBRA) be at the start of the cell.

Caution: Web page data doesn't always render as a perfect table when it's copied. You may want to test it first. PDF tables are sometimes also a bit dodgy. Excel ranges and Word tables should be safe though, as should "plain text" delimited ranges of values.

Using The Tool

Simply copy the data that you want to paste, then position your cursor in a cell at the start of the range that you want to paste into. With one exception you do ***not*** have to select the whole range that you'll be pasting to; Bulk Paste will decide the size of that range based on the clipboard content. It will ignore any cells which don't contain DBR type functions.

The one exception is where you want to paste a single value over a range of cells. This will most commonly be used in zeroing out a range, but it could also be used to set an initial value. (Data spreading could also do this, but in only one direction (rows or columns, not both) at a time.) To do that, copy the value that you want to paste, select the range that you want to paste it into, then select the relevant bulk paste icon.

Which Icon To Use

In most cases you will click in the icon on the left:

The icon on the right is for a special set of circumstances; it's for when you are:

- Writing data to a string cube, ***but***
- That data includes values which are in numeric format.

This is discussed more fully in *Limitations* below. In short, it forces numeric values to be sent as strings (text). If you're writing strings which are ***not*** in numeric format, you can use either button.

Limitations

Please be aware of the following limitations:

- The tool will have no effect if you use it on a DBR type function which points to a consolidation (it doesn't do spreading), or to a cell that you don't have write access to.
- To write the values to the cube, the tool will temporarily convert the functions from DBR/DBRW/DBRA to DBS, DBSS or DBSA types, then back. Although there are no known instances in which this process should fail, it's still recommended that you save the workbook which contains your TM1 formulas before you use Bulk Paste to be on the safe side. You can therefore simply reload the file if there are any problems.
- For DBR or DBRW functions, Bulk Paste doesn't know what type of element it will be writing to. (That is, whether it's an N (numeric) type, or an S ("string" or text type).) And because the data source can be anything and not just an Excel range, it can't check the formatting of the source content. It therefore assumes that anything that's numeric is going to an N element, and anything that isn't is going to an S element. It therefore will not write numeric values into a string element unless you use the second of the two Bulk Paste buttons as described above. Most people won't find this to be a problem.

Errors

You will get an error message in any of the following circumstances:

- You have not copied anything and your clipboard is empty when you attempt the paste operation;
- You have copied something into the clipboard which cannot be represented as an array of numeric or string data, like a picture;
- You have copied a non-rectangular range of data;
- Your worksheet is protected;
- You have copied more rows or columns than there are in your worksheet as measured from the active cell when you paste;
- There are no TM1 DBR or DBRW formulas in range in your worksheet.

You will ***not*** get an error message if you attempt to paste into consolidations or to cells to which you have no write access, but the values won't change either.

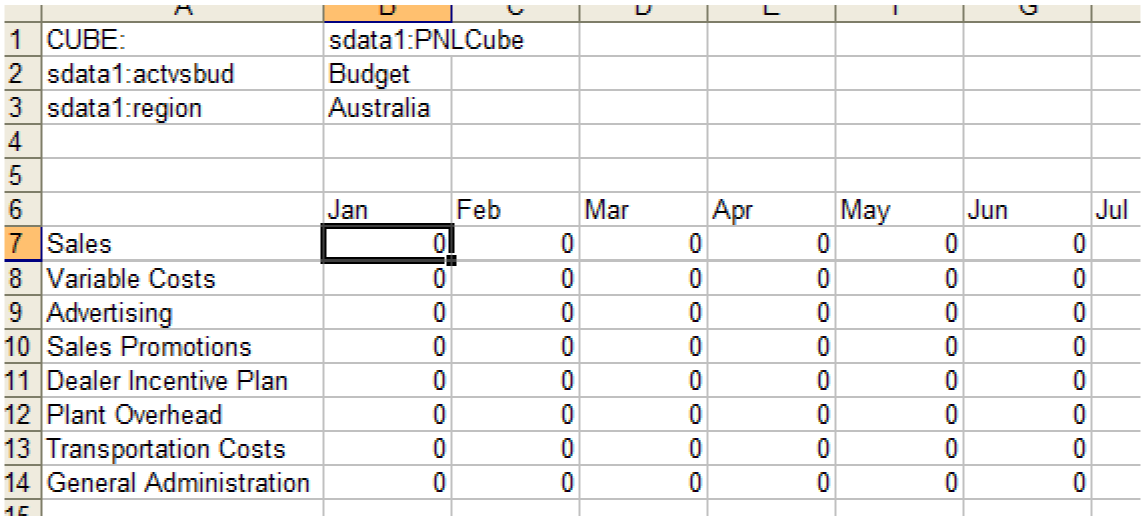
About Blank or Missing Rows

Important: This behaviour changed slightly in version 0.9.

Suppose that your source is the following table (note the third row of data, which consists of empty cells), and you copy the four rows below the heading one:

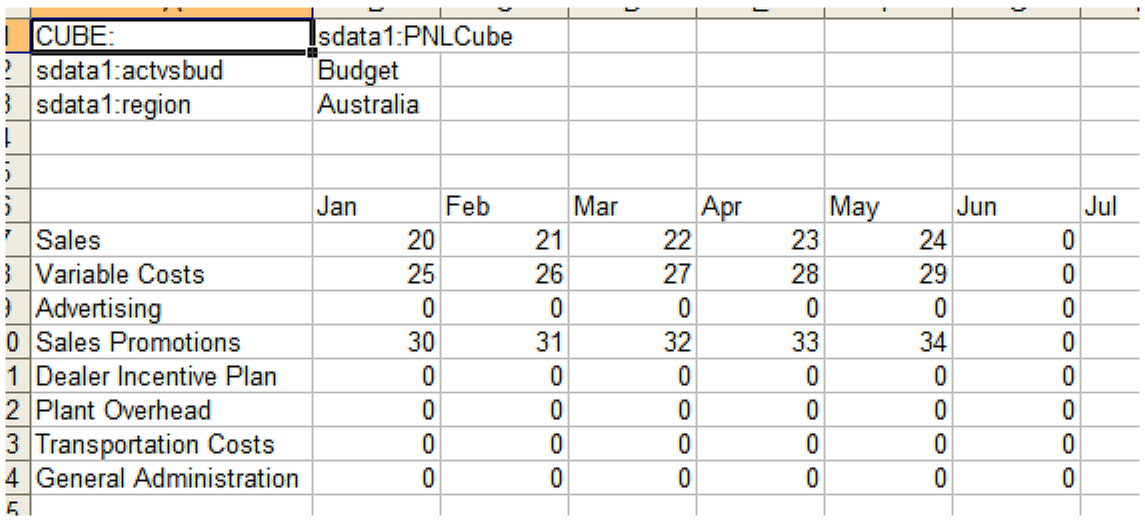
Column 1	Column 2	Column 3	Column 4	Column 5
20	21	22	23	24
25	26	27	28	29
30	31	32	33	34

And your receiving worksheet looks like this:



1	CUBE:	sdata1:PNLCube						
2	sdata1:actvsbud	Budget						
3	sdata1:region	Australia						
4								
5								
6		Jan	Feb	Mar	Apr	May	Jun	Jul
7	Sales	0	0	0	0	0	0	
8	Variable Costs	0	0	0	0	0	0	
9	Advertising	0	0	0	0	0	0	
10	Sales Promotions	0	0	0	0	0	0	
11	Dealer Incentive Plan	0	0	0	0	0	0	
12	Plant Overhead	0	0	0	0	0	0	
13	Transportation Costs	0	0	0	0	0	0	
14	General Administration	0	0	0	0	0	0	

You'll get the following result:



CUBE:	sdata1:PNLCube						
sdata1:actvsbud	Budget						
sdata1:region	Australia						
	Jan	Feb	Mar	Apr	May	Jun	Jul
Sales	20	21	22	23	24	0	
Variable Costs	25	26	27	28	29	0	
Advertising	0	0	0	0	0	0	
Sales Promotions	30	31	32	33	34	0	
Dealer Incentive Plan	0	0	0	0	0	0	
Plant Overhead	0	0	0	0	0	0	
Transportation Costs	0	0	0	0	0	0	
General Administration	0	0	0	0	0	0	

However, if you already had values on the Advertising line then those values would ***not*** be touched.

Why?

Because the clipboard would show the empty row of cells as containing empty strings. Since the paste functionality is intended to support both numeric and string values, the program has no way of knowing that you intend to send zeroes to the empty cells. For all it knows, you could have a row of formulas which point to string cells there. Consequently it will try to send zero length strings to the formulas, fail, and leave the original values untouched.

You get a different result if the entire row was ***missing*** from the table, however.

Column 1	Column 2	Column 3	Column 4	Column 5
20	21	22	23	24
25	26	27	28	29
30	31	32	33	34

In this case it will still see four rows of data, not three. Because there are no actual cells on the third row, though (and therefore, unlike the previous example, the clipboard is giving no information about what is there), the program will "interpret" the missing cells. If you're doing a standard paste it will treat the cells as containing all zeroes. If you're doing a forced string paste, it will treat them as containing empty strings.

Confused? The easy way around it is "Don't use a data source which has blank rows or columns".

Information Tools

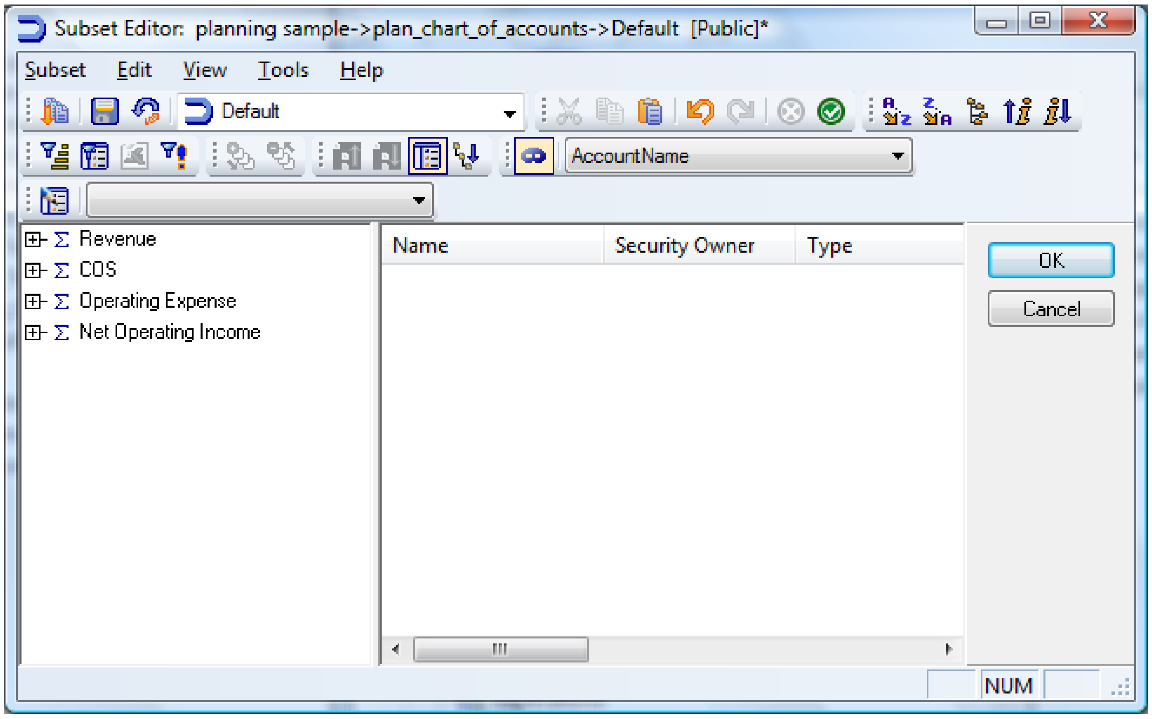
The Hierarchy Documenter

A hierarchy is the structure of the elements that sit under a consolidated element in a TM1 dimension. In a chart of accounts consolidation called `Profit or Loss` you might have two consolidated elements called `Revenue` and `Expenses`. `Revenue` may be broken down into further consolidations and N level elements, as might `Expenses`.

With large dimensions which have complex consolidation hierarchies, it can often be useful to get a graphical printout showing how all of the elements relate to each other. For a small dimension you can see that in the Subset Editor, but for most larger trees (like charts of accounts) it can be very difficult to get a feel for the overall structure of a consolidation.

This tool will generate such a tree in an Excel file for you.

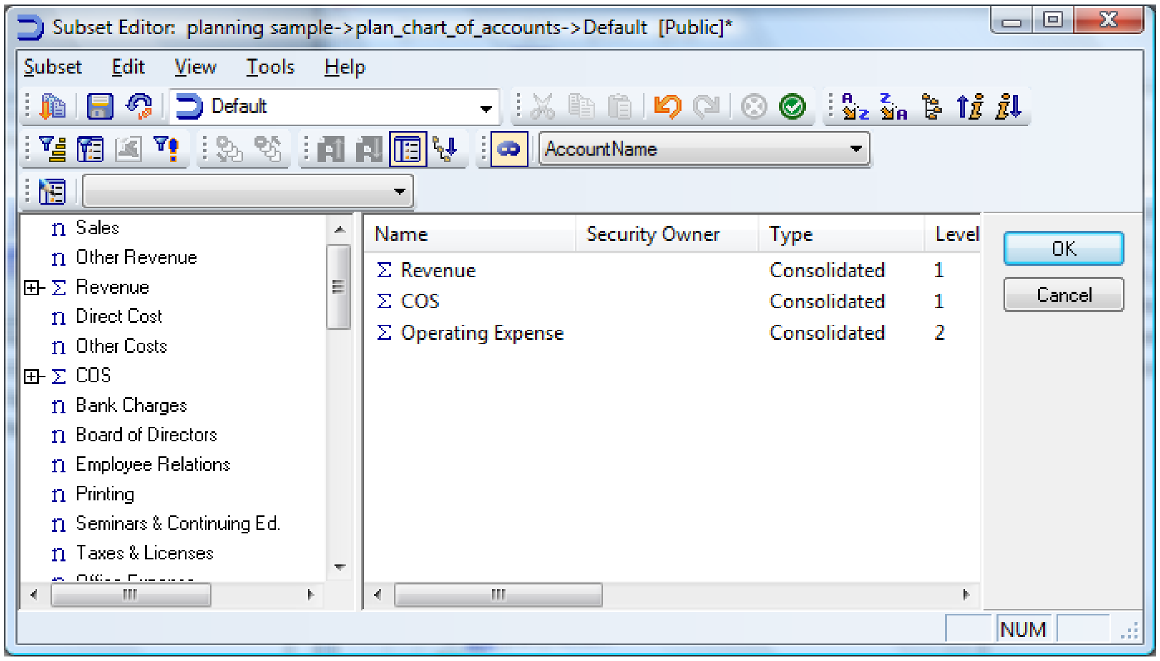
Take, as an example, the plan_chart_of_accounts dimension in the Planning Sample database that comes with TM1. The default subset isn't very intuitive given that you can't see how the elements relate to each other:



Subset Editor: planning sample->plan_chart_of_accounts->Default [Public]*

⊞ Σ Revenue
⊞ Σ COS
⊞ Σ Operating Expense
⊞ Σ Net Operating Income

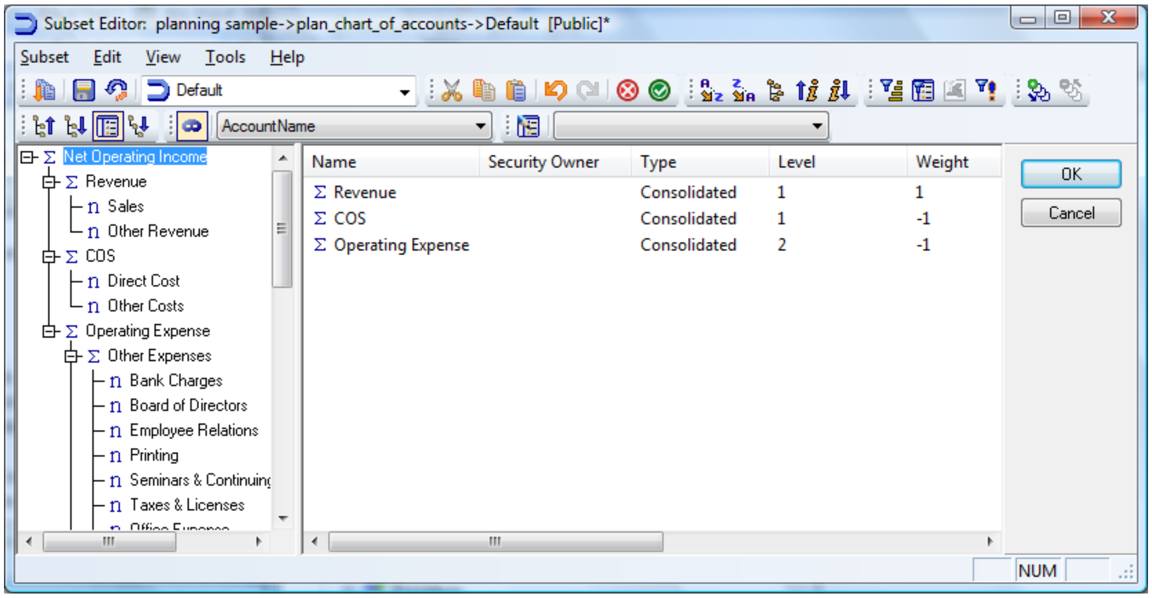
If you select the show All elements option, it gets even worse:


Subset Editor: planning sample->plan_chart_of_accounts->Default [Public]*
Subset Edit View Tools Help
Default
AccountName
OK
Cancel
NUM

Sales
Other Revenue
Revenue
Direct Cost
Other Costs
COS
Bank Charges
Board of Directors
Employee Relations
Printing
Seminars & Continuing Ed.
Taxes & Licenses

Name	Security Owner	Type	Level
Σ Revenue		Consolidated	1
Σ COS		Consolidated	1
Σ Operating Expense		Consolidated	2

If you wanted to see how Net Operating Income was put together you would need to select that element alone then Expand All Elements, like so:


Subset Editor: planning sample->plan_chart_of_accounts->Default [Public]*
Subset Edit View Tools Help
Default
AccountName
OK
Cancel
NUM

Net Operating Income
Revenue
Sales
Other Revenue
COS
Direct Cost
Other Costs
Operating Expense
Other Expenses
Bank Charges
Board of Directors
Employee Relations
Printing
Seminars & Continuing
Taxes & Licenses

Name	Security Owner	Type	Level	Weight
Σ Revenue		Consolidated	1	1
Σ COS		Consolidated	1	-1
Σ Operating Expense		Consolidated	2	-1

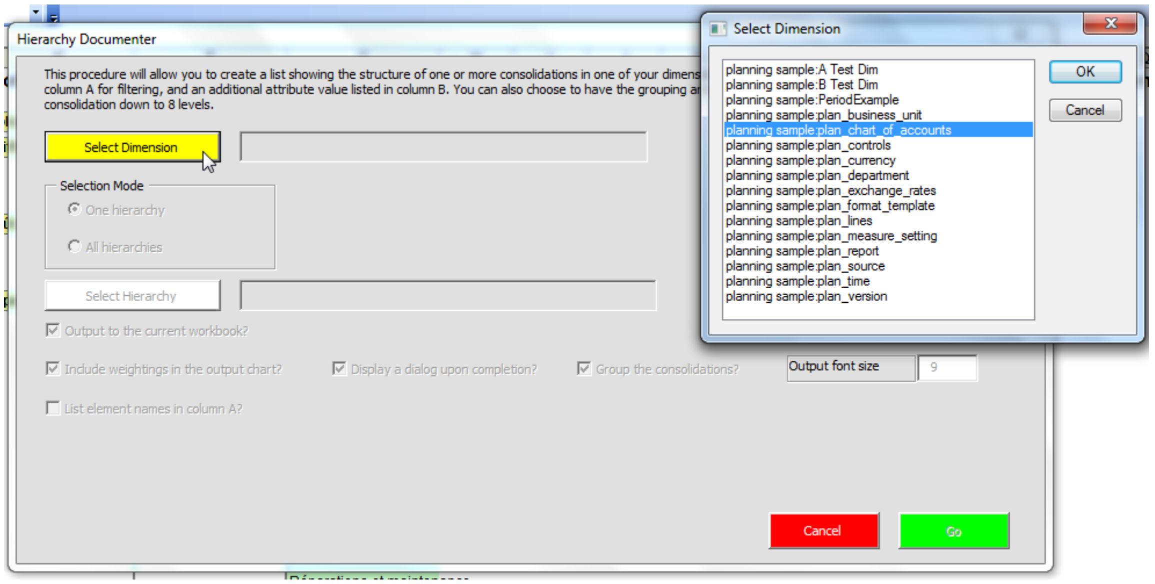
That would give you a pretty good overview except that:

- Most charts of accounts in real life aren't as conveniently brief as this example one is; and
- Notice how `COS` and `Operating Expense` have a weighting of -1? (Meaning that they're multiplied by -1 before being added to `Revenue` to give the `Net Operating Income` figure.) You can see the weightings of Net Operating Income's ***immediate*** children, but suppose that there were consolidations further down which had children which are weighted as something other than 1. You'd have to go hunting for them through the entire tree.

The Documenter can show you the whole tree on a single sheet (if it'll fit) and will also show any elements which have a weighting of something other than 1 (if you choose).

The Documenter is launched from this icon on the toolbar:

That will reveal the following form. In this illustration, I've already clicked the [Select Dimension] button. Note that from version 1.0 on, this is initially highlighted in yellow to guide you to the step that you need to take. Once you've selected a dimension it'll revert to grey.



Hierarchy Documenter

This procedure will allow you to create a list showing the structure of one or more consolidations in one of your dimens
column A for filtering, and an additional attribute value listed in column B. You can also choose to have the grouping ar
consolidation down to 8 levels.

Selection Mode: [✓ One hierarchy] All hierarchies

☑ Output to the current workbook?

☑ Include weightings in the output chart? ☑ Display a dialog upon completion? ☑ Group the consolidations? Output font size 9

☐ List element names in column A?

Select Dimension

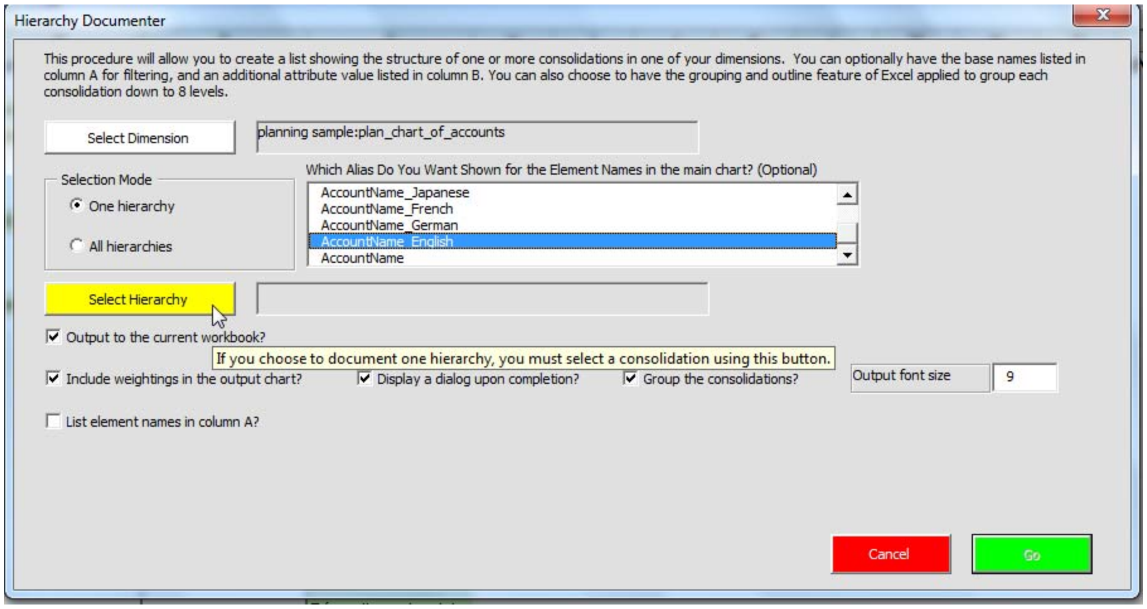
planning sample:A Test Dim
planning sample:B Test Dim
planning sample:PeriodExample
planning sample:plan_business_unit
planning sample:plan_chart_of_accounts
planning sample:plan_controls
planning sample:plan_currency
planning sample:plan_department
planning sample:plan_exchange_rates
planning sample:plan_format_template
planning sample:plan_lines
planning sample:plan_measure_setting
planning sample:plan_report
planning sample:plan_source
planning sample:plan_time
planning sample:plan_version

The options are fairly straightforward:

- `[Select Dimension]` will launch a dialog which allows you to select the dimension that contains the hierarchy that you want to document, as shown above. All of the other controls (aside from the [Cancel] button) are disabled until you've made a selection.
- The `[Selection Mode]` buttons allow you to determine whether you want to document just one hierarchy, or whether you want to have the program scan through the whole dimension looking for any consolidations which don't have parents of their own, and document each one that it finds. (The ones which do have parents will, obviously, form part of the hierarchy of their ancestor elements.) The program will remember the last selection that you made, and that will be the default next time.
    - o "How come I can't choose to just select ***some*** but not ***all*** consolidations?", I hear you ask. That's a very good question. I suggest that you ask it of your Iboglix representative. I've been asking for years now for a version of their element selection dialog which lets you select multiple elements and hey, guess what... they've never provided one. Accordingly you can select ***one*** element, or you can have the program loop through all of them. While there would be ways of getting multiple selections, the

amount of work involved wouldn't be worth the money that we're getting paid for this tool. Because any percent of nuthin' is still nuthin'.

Once you've selected a dimension, more controls become visible. In particular:


Hierarchy Documenter

This procedure will allow you to create a list showing the structure of one or more consolidations in one of your dimensions. You can optionally have the base names listed in column A for filtering, and an additional attribute value listed in column B. You can also choose to have the grouping and outline feature of Excel applied to group each consolidation down to 8 levels.

Select Dimension: planning sample:plan_chart_of_accounts

Selection Mode: (●) One hierarchy ( ) All hierarchies

Which Alias Do You Want Shown for the Element Names in the main chart? (Optional)

- AccountName_Japanese
- AccountName_French
- AccountName_German
- AccountName_English (selected)
- AccountName

Select Hierarchy

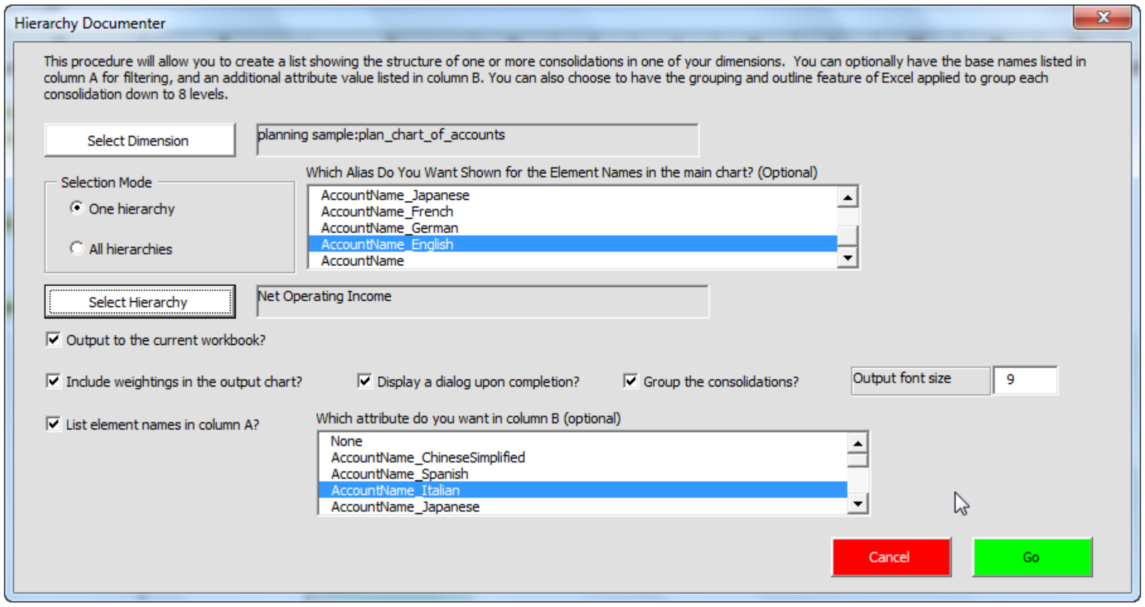
[✓] Output to the current workbook?

If you choose to document one hierarchy, you must select a consolidation using this button.

[✓] Include weightings in the output chart? [✓] Display a dialog upon completion? [✓] Group the consolidations? Output font size: 9

[ ] List element names in column A?

- `Which Alias Do You Want Shown...?` If the dimension that you select has any aliases, they'll be listed in the selection box, like the example below. You can choose to have the output show the selected alias names of the elements instead of the element base name, which I expect most people probably will. (Selecting `None` (the default) means that the base element name will be used.) The program does ***not*** store your previous attribute selections. You must select the appropriate one each time you run the program, since the attributes may change and the dimension selection may as well.
- In the illustration above the selection mode has been set to **One Hierarchy**. In that case you'd now have to tell the Documenter ***which*** hierarchy you want to document. The `[Select Hierarchy]` button is highlighted to indicate this. The `[Go]` button, on the other hand, is still disabled. If you select a non-consolidated element, you will get a warning and have to reselect.
    - **Note**: Had you selected **All Hierarchies**, the button would have been grey and disabled. The `[Go]` button would have been enabled as the Documenter would have enough information to go on with.
- You can now select any options that you want, which are in the lower part of the form:



This procedure will allow you to create a list showing the structure of one or more consolidations in one of your dimensions. You can optionally have the base names listed in column A for filtering, and an additional attribute value listed in column B. You can also choose to have the grouping and outline feature of Excel applied to group each consolidation down to 8 levels.

Select Dimension: planning sample:plan_chart_of_accounts

Selection Mode: [✓ One hierarchy] All hierarchies

Which Alias Do You Want Shown for the Element Names in the main chart? (Optional)
AccountName_Japanese
AccountName_French
AccountName_German
[✓ AccountName_English]
AccountName

Select Hierarchy: Net Operating Income

[✓] Output to the current workbook?

[✓] Include weightings in the output chart?
[✓] Display a dialog upon completion?
[✓] Group the consolidations?
Output font size: 9

[✓] List element names in column A?

Which attribute do you want in column B (optional)
None
AccountName_ChineseSimplified
AccountName_Spanish
[✓ AccountName_Italian]
AccountName_Japanese

- `Output to the current workbook?` This is only visible if the current workbook was previously generated by the documenter tool. It's useful if you're generating a number of different hierarchies for your dimensions and want them to all be in the one workbook for reference or distribution; it saves having to manually move them into the one workbook. Because the documenter uses particular colours and styles, you can't output to a workbook which wasn't created by the tool. The option also won't be available if you've locked the workbook.

- `Include Weightings In Output?` If you select this box (which is the default, though the Documenter will remember your last selection and use that the next time) you'll see the weighting given to an element in calculating its parent's value *if* that weighting is something other than 1. For example, both `COS (Cost Of Sales)` and `Operating Expense` have a weighting of -1 in the `Net Operating Income` consolidation:

- `Display dialog upon completion?` This is just what it says; if you have it selected a message box will pop up at the end showing you how many hierarchies were done, and how long it took to run.
  - Incidentally, there's no easy way to estimate that, but the procedure is reasonably fast. I took the worst case scenario I could come up with; a dimension which has literally thousands of hierarchies (don't ask), selecting the `All hierarchies` option. The output ran to 115,037 rows, and it took 5 minutes and 46 seconds to run. On the other hand an `All hierarchies` run on our chart of accounts (25 hierarchies, 25,383 rows of output) took 1 minute and 4 seconds. The main P&L hierarchy by itself took only 9 seconds.

- `Group The Consolidations?` By default the consolidations (down to the maximum level anyway) will be "grouped" using Excel's Group And Outline functionality. This allows you to click on the `[+]` and `[-]` controls to expand or condense individual consolidations or groups of them. An example of this can be seen in the illustration above. Some people didn't care for this so you now have the option of turning it off. As with most of the settings, your choice will be remembered for the next time you use the Documenter.
- The `Output font size`. This determines the size of font that will be used in the output sheet. (The output is always in Arial.) You can specify any value between 6 and 24 points. If you specify a value outside that range, or specify an invalid entry (like text) the default of 8 will be used. If you specify a decimal value, the nearest valid value will be used. (Some decimal values are valid, but there are usually minimum increments between valid values.) The Documenter will remember the last (valid) font size that you specified, and that will become your default for next time.
- `List Element Names In Column A`? This is a feature (introduced in version 1.0) which was requested by Steve Vincent. Because the element names are not in a single column it's not possible to filter them. You therefore have the option to place the base element names in column A for this purpose. Whether you do or not, the output now starts in column D, but if you have this option switched off then columns A and B will be hidden.
- `Which Attribute Do You Want In Column B?` This is an extension of the request above. You can elect to have any one of the attributes of the dimension's elements entered into column B. This can be an alias or not; up to you. This list box will only be visible if you have checked the `List Element Names In Column A` box.

The Output

An example of the output has already been shown in this topic.

- Consolidations are shown in bold text on a yellow background, and "straddle" the branch that links to their child elements. (The two consolidation name cells aren't merged, because working with merged cells in Excel can be painful.)
- N level elements are shown in green;
- S elements are shown in blue.

If the hierarchy exceeds 65,536 rows then if you're using Excel 2007, it keeps on writing. If you're not, it will create a second (or third, or however many are needed) sheet, and continue the output on that.

The consolidations may be grouped, depending on your option selection. Any consolidations which are deeper than the maximum grouping level will remain ungrouped.

The element names may extend beyond their cells. It was decided not to automatically resize the columns to make them fit because it may make the output worksheet ridiculously wide. Similarly wrapping the text may make individual rows too bloated. This design decision is up for grabs; if anyone believes that it should be changed, drop us a line at OlapForums.

Directory Quickchange

This is launched from the following icon:

This is useful if youopened a file from a shortcut, recently used files list or Outlook Journal entry. In some cases you may want to switch to that file's folder so that you can open other related files without needing to drill up and down the trees in Windows Explorer. This button will change your current file directory to that of the active file (if any).

It should work with either mapped drive paths (ones which look like D:\TM1\) or with UNC ones (eg \\TM1\TM1Files\).

Show Explorer

This is launched from the following icon:

The icon launches Windows Explorer, but does so to the folder of the workbook file that you have opened at the time. If you have no workbook open, or the workbook has not been saved, it will open to the default Windows Explorer folder.

The Timers

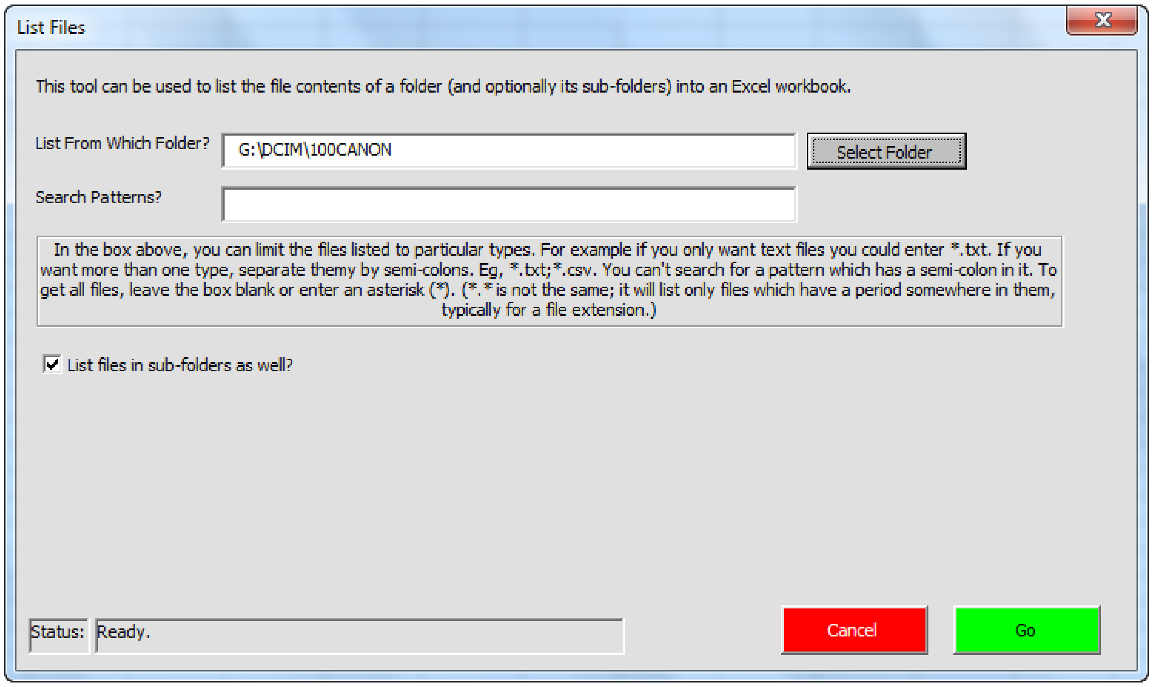
The two timers are useful if you want to check the performance of one of your TM1 (or indeed any Excel) reports. (***Note***: You still need to have the TM1 add-in loaded since these tools use the TM1 recalculation macros, not the native Excel methods.)

The one on the left does a [Shift] + [F9] calculation. The one on the right does a full [F9] calculation (hence the larger icon). Both use the TM1 calculation macro, not the native Excel methods. When the calculation has been done a message box will display showing the time taken to do it. For advanced users who want to run and compare multiple tests, you'll find the information also written to the Immediate window in the Visual Basic Editor. ([Alt] + [F11] to get to the Visual Basic Editor, then [Ctrl]+[G] to show the Immediate Window.)

File Lister Tool

This tool enables you to get a list of all of the files in a folder, or a subset of those. For example, you can limit the output to .txt or .log files.

This button launches the main dialog, which is shown below:



List Files

This tool can be used to list the file contents of a folder (and optionally its sub-folders) into an Excel workbook.

List From Which Folder? G:\DCIM\100CANON

Search Patterns?

In the box above, you can limit the files listed to particular types. For example if you only want text files you could enter *.txt. If you want more than one type, separate themy by semi-colons. Eg, *.txt;*.csv. You can't search for a pattern which has a semi-colon in it. To get all files, leave the box blank or enter an asterisk (*). (*.* is not the same; it will list only files which have a period somewhere in them, typically for a file extension.)

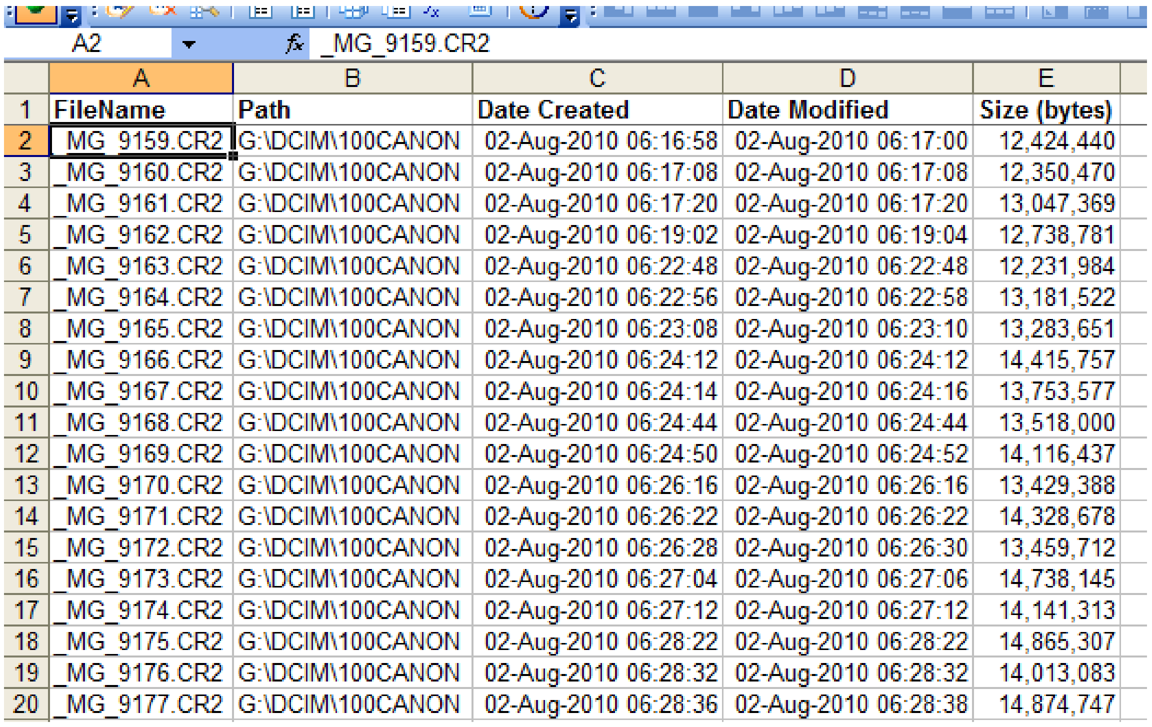
☑ List files in sub-folders as well?

Status: Ready.

The options are all self-explanatory. This isn't only for use with TM1 files, but can be used for any kind.

Because the output includes the date of modification you can, for example, use it to locate older files for archiving or culling. The tool's ability to list files in subfolders as well as the specified folder makes it more useful than Windows Explorer for that purpose, and is considerably faster than most Windows search options.

An example of the output is shown below:



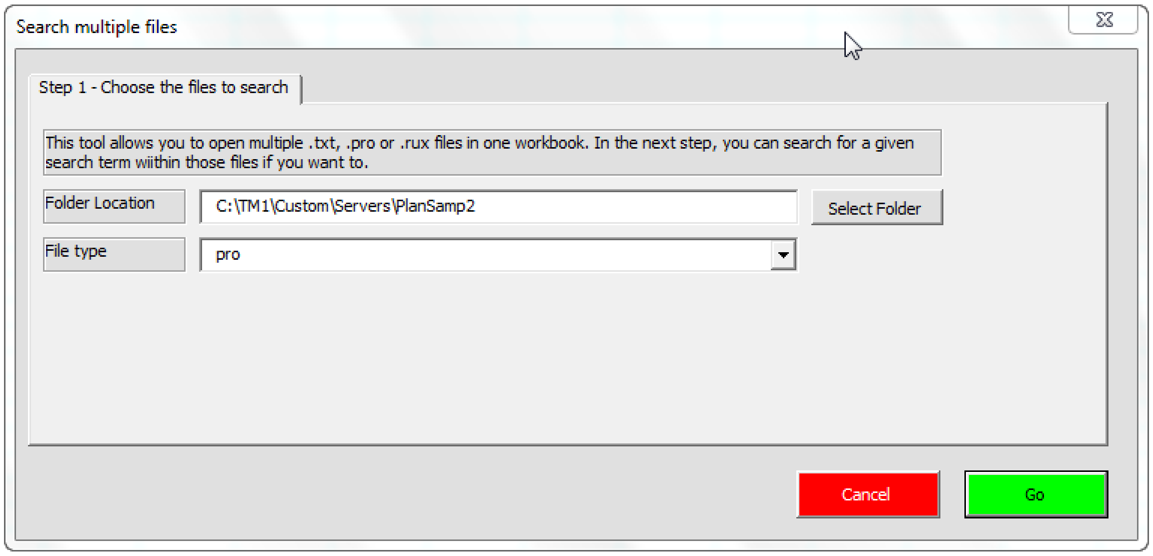
A2 _MG_9159.CR2

	A	B	C	D	E
1	FileName	Path	Date Created	Date Modified	Size (bytes)
2	_MG_9159.CR2	G:\DCIM\100CANON	02-Aug-2010 06:16:58	02-Aug-2010 06:17:00	12,424,440
3	_MG_9160.CR2	G:\DCIM\100CANON	02-Aug-2010 06:17:08	02-Aug-2010 06:17:08	12,350,470
4	_MG_9161.CR2	G:\DCIM\100CANON	02-Aug-2010 06:17:20	02-Aug-2010 06:17:20	13,047,369
5	_MG_9162.CR2	G:\DCIM\100CANON	02-Aug-2010 06:19:02	02-Aug-2010 06:19:04	12,738,781
6	_MG_9163.CR2	G:\DCIM\100CANON	02-Aug-2010 06:22:48	02-Aug-2010 06:22:48	12,231,984
7	_MG_9164.CR2	G:\DCIM\100CANON	02-Aug-2010 06:22:56	02-Aug-2010 06:22:58	13,181,522
8	_MG_9165.CR2	G:\DCIM\100CANON	02-Aug-2010 06:23:08	02-Aug-2010 06:23:10	13,283,651
9	_MG_9166.CR2	G:\DCIM\100CANON	02-Aug-2010 06:24:12	02-Aug-2010 06:24:12	14,415,757
10	_MG_9167.CR2	G:\DCIM\100CANON	02-Aug-2010 06:24:14	02-Aug-2010 06:24:16	13,753,577
11	_MG_9168.CR2	G:\DCIM\100CANON	02-Aug-2010 06:24:44	02-Aug-2010 06:24:44	13,518,000
12	_MG_9169.CR2	G:\DCIM\100CANON	02-Aug-2010 06:24:50	02-Aug-2010 06:24:52	14,116,437
13	_MG_9170.CR2	G:\DCIM\100CANON	02-Aug-2010 06:26:16	02-Aug-2010 06:26:16	13,429,388
14	_MG_9171.CR2	G:\DCIM\100CANON	02-Aug-2010 06:26:22	02-Aug-2010 06:26:22	14,328,678
15	_MG_9172.CR2	G:\DCIM\100CANON	02-Aug-2010 06:26:28	02-Aug-2010 06:26:30	13,459,712
16	_MG_9173.CR2	G:\DCIM\100CANON	02-Aug-2010 06:27:04	02-Aug-2010 06:27:06	14,738,145
17	_MG_9174.CR2	G:\DCIM\100CANON	02-Aug-2010 06:27:12	02-Aug-2010 06:27:12	14,141,313
18	_MG_9175.CR2	G:\DCIM\100CANON	02-Aug-2010 06:28:22	02-Aug-2010 06:28:22	14,865,307
19	_MG_9176.CR2	G:\DCIM\100CANON	02-Aug-2010 06:28:32	02-Aug-2010 06:28:32	14,013,083
20	_MG_9177.CR2	G:\DCIM\100CANON	02-Aug-2010 06:28:36	02-Aug-2010 06:28:38	14,874,747

File Searcher

This tool enables you to search multiple text files for a specific term. It does this by loading them into an Excel workbook and allowing you to search in various ways. For example, if you know that a specific cube is referenced in some of your TI Process files but aren't sure which ones, you can load all of the .pro files into a new workbook, then search for the cube's name. The tool is launched from the following icon on the toolbar:

Normally you will see step 1 of a two step wizard displayed:


Search multiple files

Step 1 - Choose the files to search

This tool allows you to open multiple .txt, .pro or .rux files in one workbook. In the next step, you can search for a given search term wiithin those files if you want to.

Folder Location: C:\TM1\Custom\Servers\PlanSamp2

File type: pro

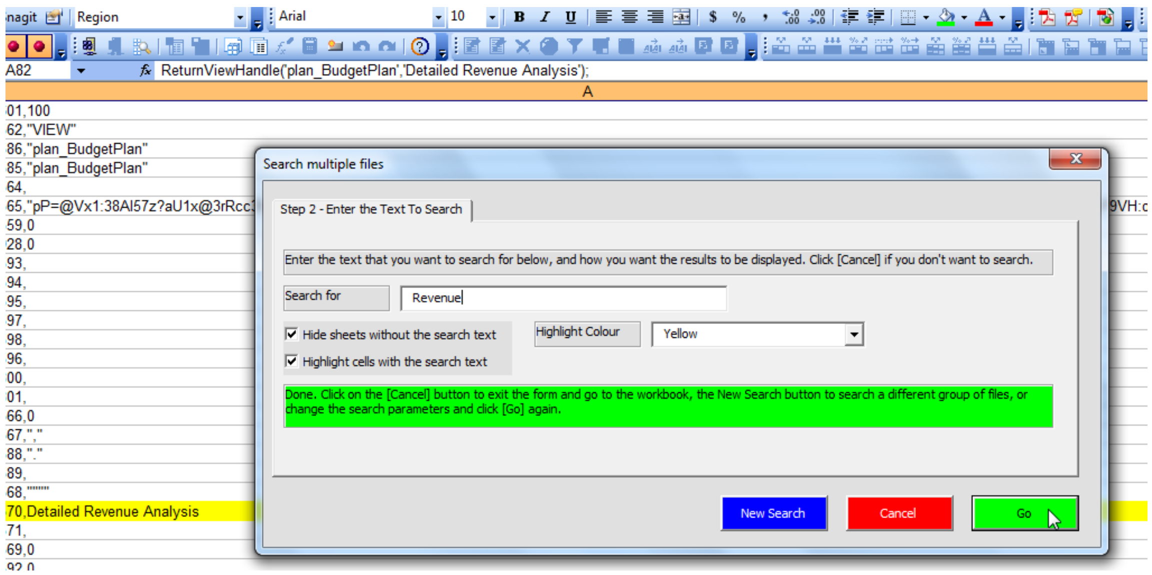
To use the tool, you need to tell it where to look for the files, and which file type(s) you're looking for:

- First, enter the folder which contains the files that you want to search. If you've used this tool before, the last folder that you did a search in will be shown. Otherwise, either enter the directory path straight into the `Folder Location` box, or click on the `[Select Folder]` button to select the path from the GUI.
- Note that the `[Go]` button will be disabled until you have made a folder selection.
- Next, select one of the three file types from the `File Type` drop down. You can select either `.pro` to search TurboIntegrator process files, `.rux` to search rules files, or `.txt` to select text files. You *cannot* search any other type of file.

When the selections are made, select the `[Go]` button. You will get an error message if:

- The folder is invalid; or
- There are no matching files in it.

Otherwise, you move to Step 2 of the dialog, shown below.



```
A82    ReturnViewHandle('plan_BudgetPlan','Detailed Revenue Analysis');
01,100
62,"VIEW"
86,"plan_BudgetPlan"
85,"plan_BudgetPlan"
64,
65,"pP=@Vx1:38Al57z?aU1x@3rRcc
59,0
28,0
93,
94,
95,
97,
98,
96,
00,
01,
66,0
67,","
88,"."
89,
68,""""
70,Detailed Revenue Analysis
71,
69,0
```

Search multiple files

Step 2 - Enter the Text To Search

Enter the text that you want to search for below, and how you want the results to be displayed. Click [Cancel] if you don't want to search.

Search for: Revenue

☑ Hide sheets without the search text

☑ Highlight cells with the search text

Highlight Colour: Yellow

Done. Click on the [Cancel] button to exit the form and go to the workbook, the New Search button to search a different group of files, or change the search parameters and click [Go] again.

Search For

The illustration above shows the state of the dialog after a search has been done. Initially the Search For box will be empty; the previous search is not stored, and you'll usually need to enter something into it.

Hide Sheets

You can choose to hide the sheets (files) which don't contain the expression that you're searching for. If you uncheck that box, ***all*** of the sheets will be unhidden.

The workbook will always have at least one sheet visible, namely `Sheet1`. (Even if you've deleted that sheet previously, it will be added back in since normal workbooks must have at least one sheet visible.)

Highlight

You can also choose whether to highlight any cells which contain the text that you're looking for.

If you elect to highlight the text, you can choose to do so with a background colour of either red, green, blue or yellow. If you have a non-standard palette in your workbooks, the nearest approximate colour will be used.

Go

Once you've made those selections, click on the `[Go]` button.

In the illustration above I've already done that, which is why the "Done" message is displayed in the dialog, and one of the rows of the worksheet in the background is highlighted in yellow.

You can now either:

- Load a different set of files for searching by clicking on the blue `[New Search]` button; or
- Click on `[Cancel]` to exit the search tool and go to the workbook.

Note: Clicking on `[Cancel]` simply cancels any further search actions. It does ***not*** close the workbook. In fact it's the only way that you can get to the workbook.

To Clear The Search Results

Leave the `Search For` dialog blank and click `[Go]`. That will remove all of the highlighting (if any) and unhide any hidden sheets.

If You Have Already Loaded The Files

If you want to do a different search within the files that you've already loaded, just click on the spectacles icon again. The tool will load straight to step 2 of the dialog, and you can modify the search that you've made.

Note that that will only work with workbooks which were created via this tool, which are tagged with a special property. You can't use the tool to search normal workbooks, not least because most people wouldn't want their hard-applied formatting destroyed accidentally.

TM1API

TM1 API

The TM1 Application Programming Interface (API) provides a way of writing your own code which interacts with TM1 objects like servers, cubes, dimensions and so on. The code can be written in Visual Basic, Visual Basic for Applications (VBA) and C.

Note: There is a different API for VB.Net, which we aren't discussing here and which plays no part in TM1 Tools. A third API is available for Java.

In the case of VB and VBA, a module called tm1api (supplied as part of your TM1 installation) is incorporated into the project. That module contains a raft (actually more like an ocean liner) of function declarations which reference functions in the various TM1 libraries which are placed on your computer when you install the application.

You interact with TM1 by calling those functions and passing values to them.

Even for experienced programmers the TM1 API is quite an esoteric place, and programming with it should be your ***last*** option. As an alternative, the tm1p.xla add-in exposes a large number of functions which can be called via the Application.Run method, and you're strongly advised to take advantage of them.

However there are some things which you can't do through automation via the Excel interface, and it's sometimes necessary to use the API in such cases.

To ease the pain of working with the API, we've started to include "wrapper" functions which sit around the API code, and can be called as easily as any "normal" VBA function. All you need to do is set a reference to the TM1 Tools project, as shown below.

In version 1.0 there are only a handful of these:

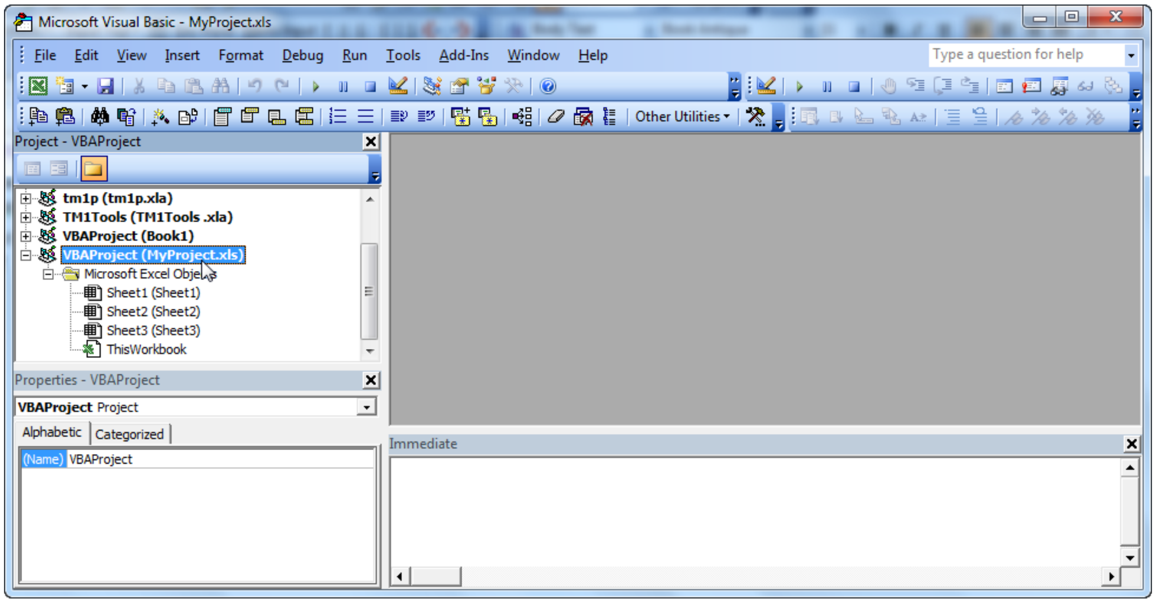
- One to run a TurboIntegrator process, optionally with parameters;
- One to return a list of available TM1 servers as a string array;
- One which returns a collection of custom TM1Client objects which you can use to query information about the clients.

Before you can call the TM1 Tools API functions you need to set a reference to it.

Referencing TM1 Tools

Begin by creating your own project by creating and saving a new workbook. (Or modifying an existing one.) Go to the Visual Basic Editor (VBE) by pressing [Alt]+[F11].

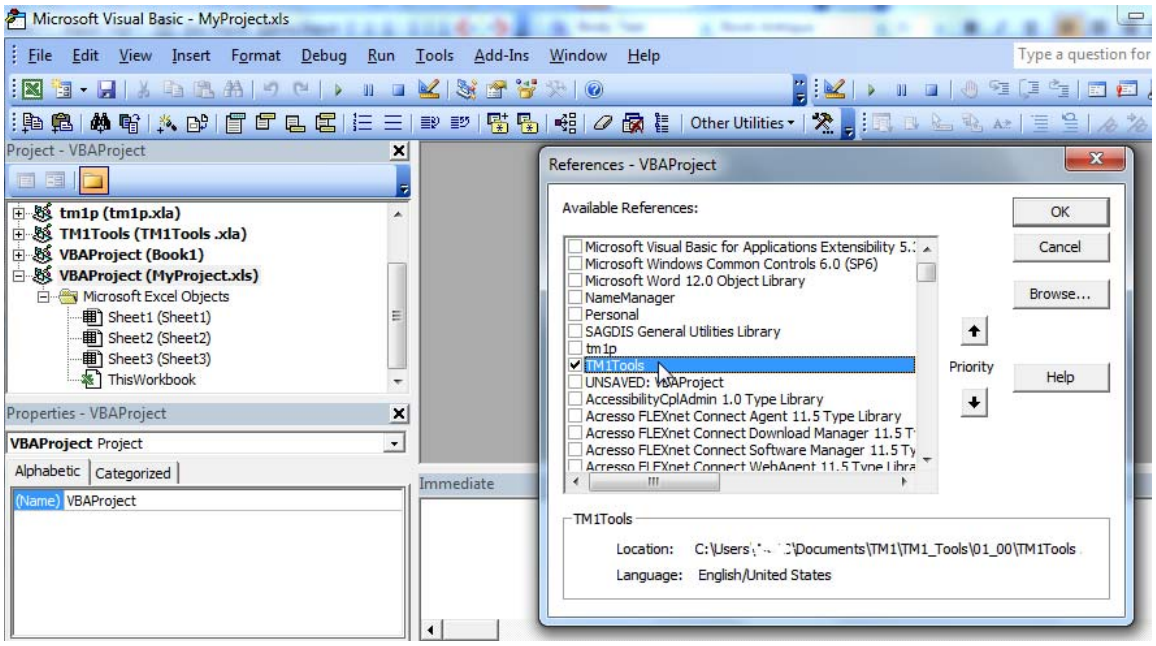
If the Project Explorer isn't visible, activate it either from the View menu or by [Ctrl]+[R]. In this case it's kept in the top left hand corner.



Ensure that you have your project selected as shown above.

Now go to `Tools` -> `References...`

Locate the TM1Tools project as shown below, and check it.



- [ ] Microsoft Visual Basic for Applications Extensibility 5.:
- [ ] Microsoft Windows Common Controls 6.0 (SP6)
- [ ] Microsoft Word 12.0 Object Library
- [ ] NameManager
- [ ] Personal
- [ ] SAGDIS General Utilities Library
- [ ] tm1p
- [✓] TM1Tools
- [ ] UNSAVED: VBAProject
- [ ] AccessibilityCplAdmin 1.0 Type Library
- [ ] Acresso FLEXnet Connect Agent 11.5 Type Library
- [ ] Acresso FLEXnet Connect Download Manager 11.5 T
- [ ] Acresso FLEXnet Connect Software Manager 11.5 Ty
- [ ] Acresso FLEXnet Connect WebAgent 11.5 Type Libra

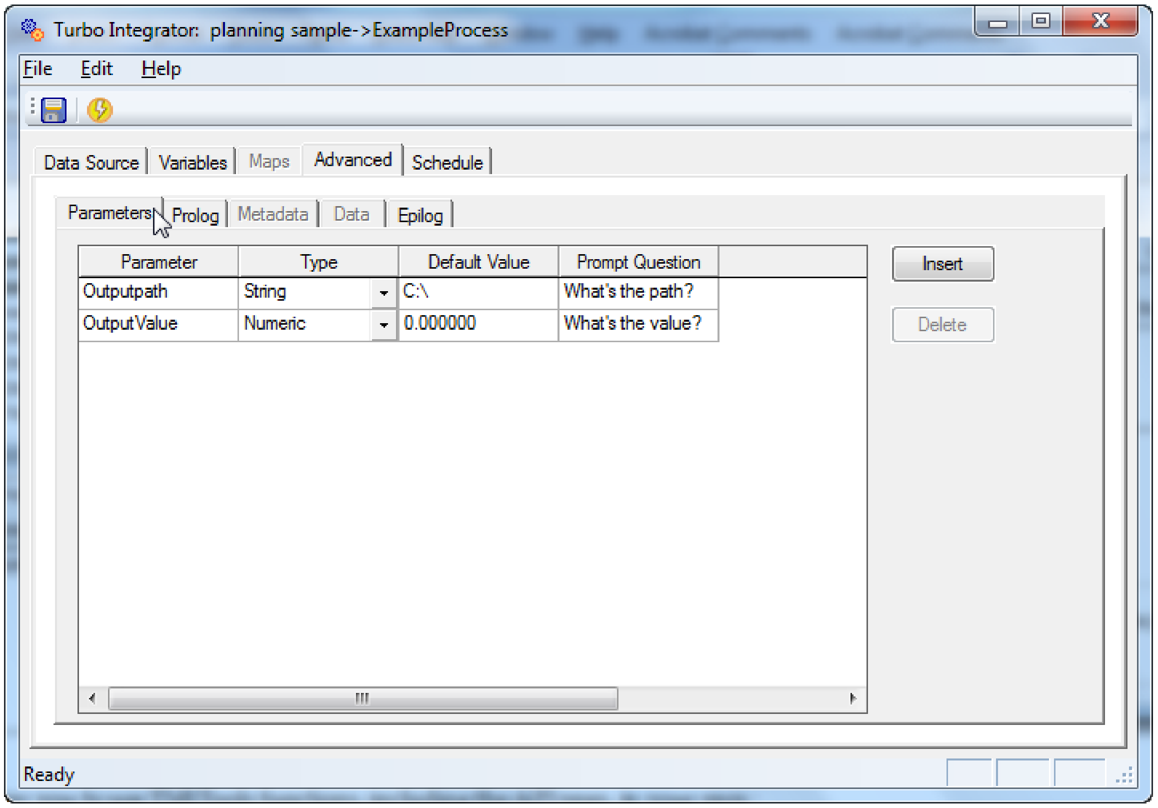
TM1Tools
Location: C:\Users\[illegible]\Documents\TM1\TM1_Tools\01_00\TM1Tools
Language: English/United States

This will allow you to use TM1Tools functions, including the API ones, in your own project.

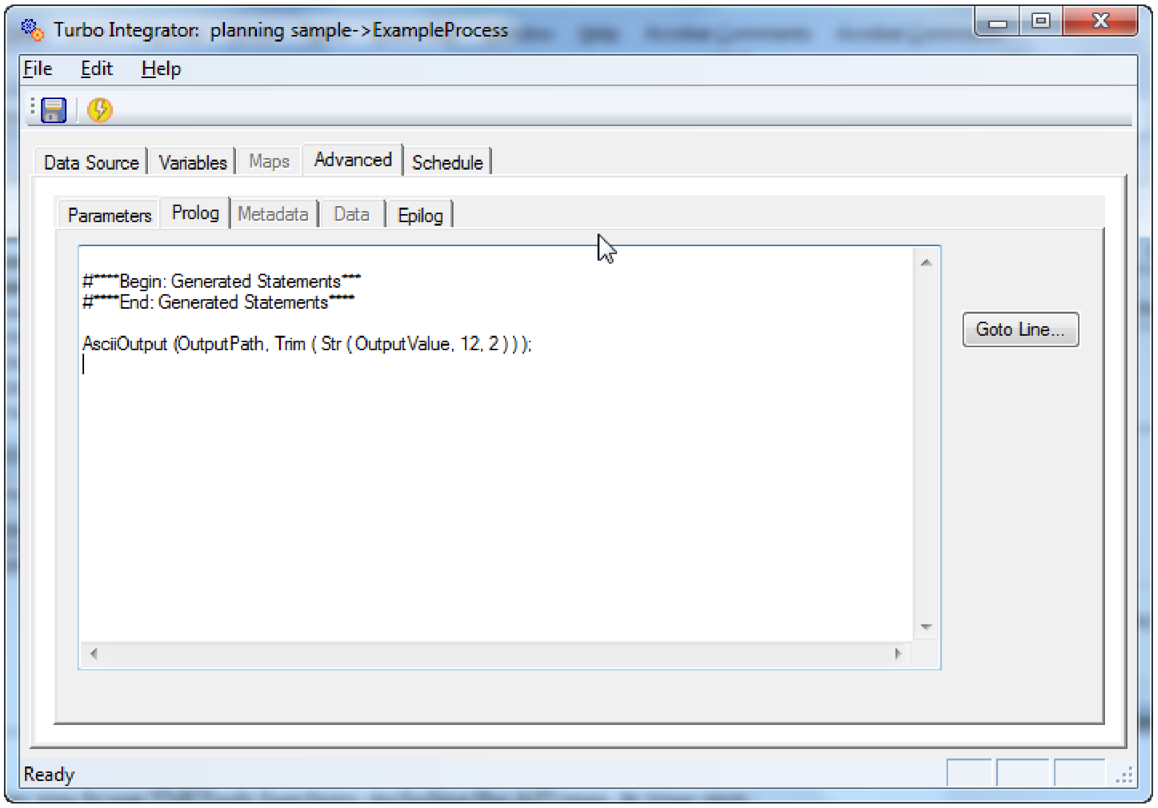
Now add a module to your project by right clicking on it and selecting `Insert` -> `Module`

After that, write some code to call one of the API functions. We'll use the one which runs a TI process as an example.

For the purposes of this demonstration a simple TI process was created in the Planning Sample server. It has two parameters, one string and one numeric.



Parameter	Type	Default Value	Prompt Question
Outputpath	String	C:\	What's the path?
OutputValue	Numeric	0.000000	What's the value?


Turbo Integrator: planning sample->ExampleProcess
File Edit Help
Data Source | Variables | Maps | Advanced | Schedule
Parameters | Prolog | Metadata | Data | Epilog
Goto Line...
Ready

```
#****Begin: Generated Statements***
#****End: Generated Statements****

AsciiOutput (OutputPath, Trim ( Str ( OutputValue, 12, 2 ) ) );
```

To run this process, it's possible to call the RunTIProcess function in TM1 Tools as if it was built into your own project:

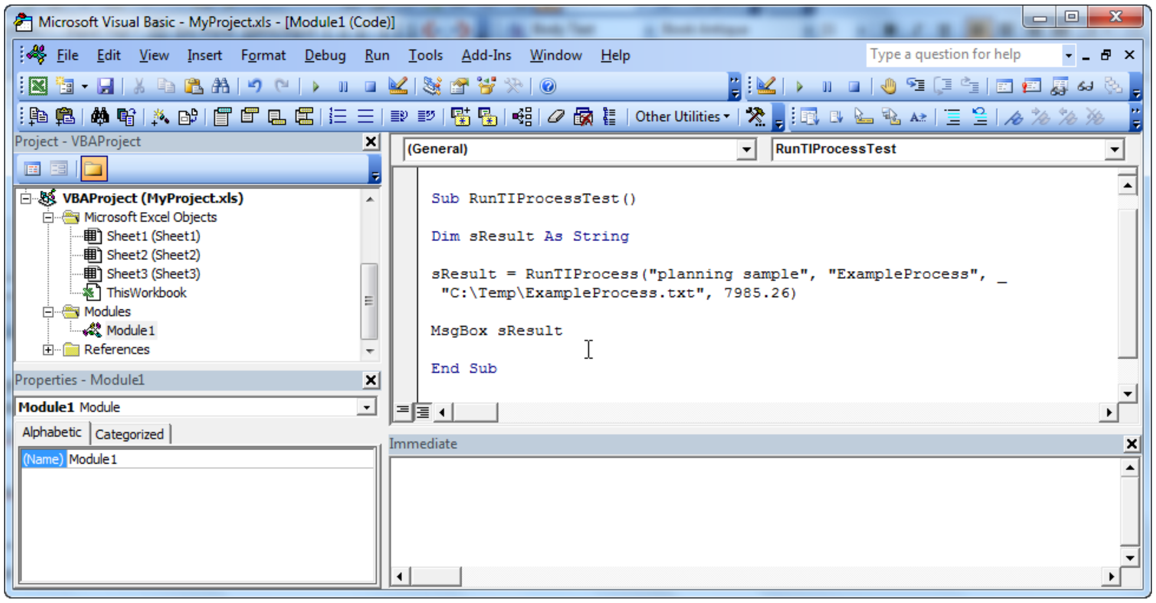
```
Sub RunTIProcessTest()

Dim sResult As String

sResult = RunTIProcess("planning sample", "ExampleProcess", _
 "C:\Temp\ExampleProcess.txt", 7985.26)

MsgBox sResult

End Sub
```



```
Sub RunTIProcessTest()

Dim sResult As String

sResult = RunTIProcess("planning sample", "ExampleProcess", _
 "C:\Temp\ExampleProcess.txt", 7985.26)

MsgBox sResult

End Sub
```

The same principle applies to any of the other TM1 API functions which are or will be incorporated into later versions of the tool.

Calling Other Functions In TM1 Tools From VBA

It's actually possible for you to call ***any*** public functions which are in TM1 Tools, not just the API ones, using this method as well. For example, Excel offers the GetOpenFileName function if you need to get a specific file, but doesn't have an equivalent for getting a folder rather than a file. With a reference to TM1Tools set you can reference our GetOpenFolder function without needing to regenerate the code yourself, or copying it into your own project. (Rather fortunate, as it relies on Windows API callback functions which are somewhat mind-stretching.)

```
Sub GetAFolder()

Dim s_Folder As String

s_Folder = GetOpenFolder("C:\", "Which folder do you need?", _
 "TM1 Tools References")

If s_Folder = "" Then
    MsgBox "No folder was selected"
Else
    MsgBox s_Folder & " was selected"
End If

End Sub
```

Caution: TM1 Tools is not a commercial product, and it put together by the authors in their spare time. That means that we don't spend many, many (unpaid!) hours sitting down and planning out the TM1 Tools API in fine detail. And that in turn means that we may over time change the functions that exist in the application as we find improved ways of doing things, though now that we've designated it as version 1.0 we'll try to minimise that as much as possible. If public function argument lists change from now on we'll try to incorporate that information into the release notes, but we can't guarantee that it won't happen.

Functions

RunTIProcess

Purpose: Executes a Turbo Integrator process from Excel.

Syntax

```
sResult=RunTIProcess(servername, processname, p1, p2, … ,
pN)
```

Arguments

servername (string)	A valid TM1 instance, to which you are already connected.
processname (string)	The name of the TI process to be run.
pN (optional, string or numeric)	The parameters to be passed to the process.

Returns

A string. Either "Process executed successfully", or information about the failure. If an error log is generated then the function will attempt to show it to the user within the body of the function, and then also return a string to the calling process.

Notes

- In order to view any error log generated, the user must have read privileges to the logging directory.

- That directory needs to be set through the TM1 Tools Options form. (See that topic for details.)

- To use this function your VBA project must reference the TM1 Tools add-in as discussed under the topic headed TM1 API.

GetServerList

Purpose: Return a list of the names of the currently available TM1 Servers via by ref string array. The function itself returns the number of servers found. You do not need to be logged into any of the servers for this function to work.

Syntax

```
NoOfServers = GetServerList(ListOfServers)
```

Arguments

- ByRef `ListOfServers` (string ***array***)   A dynamically defined string array.

Returns

Directly: The number of servers found.

Indirectly: A string array containing the names of the servers via the ListOfServers argument. That variable is populated by the function.

Notes

- The array returned is zero based. If there are no servers, the array will be undimensioned.

- For that reason do not rely on the VBA UBound function to get the size of the array; it returns an error on undimensioned arrays. (It should really return -1, but tell that to Microsoft.) Instead, use the return value of the function as shown in the usage example.

- To use this function your VBA project must reference the TM1 Tools add-in as discussed under the topic headed TM1 API.

Usage Example

```
Sub DemonstrateServerList()

Dim i_NoOfServers As Integer
Dim i_Server As Integer
Dim sa_ServerNames() As String

'You MUST use error handling in calls to
'the API procedures.
On Error GoTo ErrorHandler

i_NoOfServers = GetServerList(sa_ServerNames)

Debug.Print i_NoOfServers & " server(s) found"

If i_NoOfServers > 0 Then

    For i_Server = 0 To i_NoOfServers - 1

        Debug.Print sa_ServerNames(i_Server)
    Next
End If
Exit Sub
ErrorHandler:
MsgBox "Error " & Err.Description
End Sub
```

GetClientsCollection

Purpose: Return a collection containing custom TM1Client objects (defined in the TM1Tools workbook), each of which has information about TM1 clients embedded in them. Each TM1Client object will in turn have a collection of TM1Group objects.

In an ideal world (where we were getting paid for this application) we'd use custom collection objects rather than a generic Collection object, which can contain any kind of data. In practice they're a pain to do in VBA, and the generic object will do just as well for the purposes of the exercise.

This procedure can only be used by clients who have Admin privileges to the server; non-Admins will get a runtime error.

Syntax

```
NoOfClients = GetClientsCollection(ServerName, _
 CollectionOfClients, NumberOfActiveUsers, _
 ZeroCountIsError)
```

Arguments

- `ServerName` (String). The name of the server that you want to get the list of clients for. Do not append a trailing colon to this; it would be something like `sData`, not `sData:`.
- ByRef `CollectionOfClients` (Collection). A variable defined as a Collection object. Any existing content in this variable will be deleted. The variable will be populated with a collection of TM1Client objects (see that heading for details), each of which contains information about a client on the server.
- Optional ByRef `NumberOfActiveUsers` (Long). You can optionally pass a long integer to this argument; it will be returned with the number of users who are currently active on the server. This saves you from having to iterate through the collection to determine how many users are on.
- Optional `ZeroCountIsError` (Boolean, default True). If the server appears to have no clients, a runtime error will be returned to your code. This does not apply if you aren't logged in to a server; in that case the function will simply return zero and the collection will be empty.

Returns

Directly: The number of clients found.

Indirectly: A collection of TM1Client objects via the CollectionOfClients argument, and optionally the number of active clients via the numberofActiveUsers argument.

Notes

- The method can only be used by clients who have Admin rights to the server.

- To use this function your VBA project must reference the TM1 Tools add-in as discussed under the topic headed TM1 API.

Usage Example

```
Sub DemonstrateClientCollection()

Dim l_NoOfClients As Long
Dim l_NoOfActiveClients As Long

Dim col_Clients As Collection
Dim obj_Client As TM1Client
Dim obj_Group As TM1Group

'You MUST use error handling in calls to
'the API procedures.
On Error GoTo ErrorHandler

l_NoOfClients = GetClientsCollection("Planning Sample", _
 col_Clients, l_NoOfActiveClients, True)

Debug.Print l_NoOfClients & " client(s) found, " _
 & l_NoOfActiveClients & " active."

For Each obj_Client In col_Clients

    Debug.Print obj_Client.ClientName & ", " _
     & IIf(obj_Client.ClientHasPassword, "has password", "no password")

    Debug.Print "Belongs to " & obj_Client.Groups.Count & " group(s)"

    For Each obj_Group In obj_Client.Groups
        Debug.Print vbTab & obj_Group.GroupName
    Next

Next

Exit Sub

ErrorHandler:

MsgBox "Error " & Err.Description

End Sub
```

Objects

TM1Client Object

This is an object which has been defined in TM1Tools to store information about a client on a TM1 server. A collection of these can be obtained by calling the GetClientsCollection procedure after setting a reference the TM1 Tools add-in as discussed under the topic headed TM1 API.

You can declare a variable to be a TM1Client type as long as you have the reference to TM1Tools set.

The objects can only be returned by users who have Admin rights to the server.

Methods

The object has no methods.

Events

The object raises no events.

Properties

ClientName (String)

The name of the client on the server.

ClientActive (Boolean)

Whether the client was active on the server at the time that the object was created. This value is ***not*** dynamic; it's merely a snapshot at the time of object creation.

ClientHasPassword (Boolean)

Whether or not the client had a password defined at the time that the object was created.

ClientPasswordExpiryDays (Long)

If the client's password is set to expire after a specified number of days, this tells you how many. Remember that if this value is set the client won't simply be forced to change their password; it will simply stop working.

ClientPortsMax (Long)

The number of ports that the client can have active on the server.

ClientPasswordUpdate (Date)

The date and time that the client changed their password. This would be UTC time not local time.

Groups (Collection)

This is a collection of TM1Group objects providing the names of the TM1 security groups that the client belongs to on the server.

TM1Group Object

This is an object which has been defined in TM1Tools to store information about a security group on a TM1 server. A collection of these will be attached to each TM1Client object returned from the GetClientsCollection procedure. That procedure requires a reference the TM1 Tools add-in as discussed under the topic headed TM1 API.

You can declare a variable to be a TM1Group type as long as you have the reference to TM1Tools set.

A procedure which returns all of the groups for a server will be included in a future release of TM1Tools.

The objects can only be returned by users who have Admin rights to the server.

Methods

The object has no methods.

Events

The object raises no events.

Properties

GroupName (String)

Name of the security group on the server.

Clients (Collection)

For future expansion only. The procedure which will return all security groups (not included in the current release) will incorporate a collection of TM1Client objects in this property. That collection will provide a list of all of the clients who are in that group.

However when the TM1Group object is returned as part of the Groups property of a TM1Client object, this collection is unpopulated. (If it were otherwise you'd have an infinite recursive pattern all the way downuntil all of the computer's memory was used.)

Additional Worksheet Functions

Worksheet Functions

This section deals with any worksheet functions that have been added to TM1 Tools. At the time of writing there were only two.

See Also:

- TM1ElIsAnc
- IsFolderValid

TM1ElIsAnc

Purpose: To determine whether an element is an ancestor of another element in a dimension. It performs similarly to the ElIsAnc function in Rules and TurboIntegrator.

Reason: Because Applix then Cognos and now IBM canbe bothered providing this obviously missing functionality.

Syntax

TM1ElIsAnc(Dimension, Ancestor, Descendant)

Arguments

- Dimension (String)   A valid dimension name, including the server name if required.
- Ancestor (String)   A consolidation element in the dimension. For the function to return True, this needs to be somewhere above the Descendant element in one or more consolidation hierarchies of the dimension.
- Descendant (String)   An element (either N or C type) which is being tested to determine whether it falls below the Ancestor element in one or more dimension hierarchies.

Returns

- If Descendant falls below Ancestor in any consolidation hierarchy: Boolean True
- If it doesn, or any of the three arguments are invalid: Boolean False.
- If an error occurs (typically, the TM1 add-in is not loaded): An Excel #Value error. (This will ***not*** be the case if you merely pass a non-existent dimension name or element name(s) to the function; that will return False instead of an error.)

The Rules / TI functions return 1 and 0 rather than True or False, but the return value used is consistent with the Worksheet functions that Iboglix ***has*** bothered to provide like ElIsPar.

Notes

- It doesnmatter how many levels there are between Ancestor and Descendant. If Ancestor is an immediate parent, itstill an ancestor even though italso a parent.
- The function does not take into account whether the Descendant element is weighted. Even if Descendant is effectively weighted as zero relative to Ancestor, if it exists below Ancestor in ***any*** dimension hierarchy the function will still return True.

Usage Advisories

This function isn't going to be as fast as a native TM1 function would be. Use it sparingly if possible.

Example

```
= TM1ElIsAnc("Server:Region", "Australia", "New South
Wales")
```

In this hypothetical dimension Region, the element `Australia` is an ancestor of the element `New South Wales`. The example returns True.

IsFolderValid

Purpose: To determine whether a drive or folder exists on the current computer. This can be either a mounted drive (like T:\TM1Clients\) or a UNC type (like \\TM1Server\TM1Share\TM1Clients). The only regular use might be if you're having a user specify the output path for saved copies of a report generated by VBA, especially if you want to check that a network drive is still available.

Reason: It's unlikely that this will often be needed as a worksheet function, but it's used to validate folder paths in TM1 Tools itself and there's no reason not to provide it as a function as well. It enhances the standard VBA Dir() function since that function will generate an error if a UNC root path is specified without a trailing backslash. (`\\TM1Server\TM1Share` rather than `\\TM1Server\TM1Share\`)

Syntax

IsFolderValid(FolderToCheck As String) As Boolean

Arguments

- FolderToCheck (String)   A path to the folder that you're checking for. As noted above this can be either a mounted drive letter, or a UNC reference.

Returns

- If the path exists, Boolean True,
- If it doesn, or the argument is invalid: Boolean False.

Notes

- None

Usage Advisories

- None.

Example

```
= FolderToCheck ("T:\TM1Clients")
```

Index