Auditing Protocol 8.1

January 20, 2006

1. Preliminaries

1. Run *XLAnalyst* by selecting XLAnalyst ► Analyse This Workbook. *XLAnalyst* generates a separate workbook containing test results in a worksheet titled XLAnalystReport. (Note: if you see a warning related to unsaved changes, click on *Yes*.)

2. Run *Spreadsheet Professional* by selecting Test ► Spreadsheet Test Reports tool from the toolbar. Select the following options for *Spreadsheet Professional*:

a. Under *Sheets*, check All Sheets.
b. Under *Include:*, check Summary, Range Names, Maps, General Tests, and Calculation Tests.
c. Click on Options, go to the Test Options tab, and make sure *all* the boxes are checked. Go to the Translation Options tab. Below *Copied calculations are defined as those that are the same as*, check the radio button for *in the column to*

the left or the row above. Click OK to return to the main screen. Click OK to run the report. (Note: if you see a warning related to merged cells, click on *Yes*). *Spreadsheet Professional* will run the Test Report tool and record the results in a

Yes.) *Spreadsheet Professional* will run the Test Report tool and record the results in a new workbook. Note that *Spreadsheet Professional* analyzes each sheet in the workbook you are auditing and for each sheet creates related sheets in the same workbook called Sheet Name – Map and Sheet Name – Errors.

3. Create a single, integrated auditing workbook by merging the spreadsheet you are auditing, the Audit Report Template, the *XLAnalyst* sheet, and the *Spreadsheet Professional* workbook. Add "Audited" and your initials to the original name of the workbook you are auditing (for example, ROI.Audited.KTS.xls) and save it (be sure not to change the original version of the workbook you are auditing).

The workbook should consist of the following sheets in the order given below:

(Note: Right mouse click on the respective *Spreadsheet Professional* sheet name and select Move/Copy. Then select the audit workbook and move the sheet into its proper place.)

- a. Audit Report Template
- b. XL Analyst report
- c. Worksheet 1 of the workbook to be audited
- d. SS Pro Map 1 (for Worksheet 1)
- e. Worksheet 2
- f. SS Pro Map 2
- g. Worksheet 3

h. SS Pro Map 3
.
i. Error Report 1
j. Error Report 2
k. Error Report 3
.
1. SS Pro Summary sheet

m. SS Pro range names sheet n. SS Pro General Tests sheet

2. Record numerical data

(Note: all cell addresses refer to locations in the Audit Report Template worksheet unless otherwise specified.)

1. In cell C2 enter the name of the workbook.

2. In row 2, starting in cell D2, enter the name of each sheet. The sheet names will be copied to row 17.

3. Go to cell H6 in the XLAnalystReport and copy the Overall Risk Rating into cell C7 in the Audit Report Template. Use Edit – Paste Special – Values in order to maintain the cell formatting throughout the Audit Report Template.

4. Copy the eight numbers from L33:L40 of the XLAnalystReport to the Audit Report Template starting in cell C8 and ending in cell C15. Do not change the cell formats.

5. Go to the *Spreadsheet Professional* Errors report for each sheet. Copy the numbers between B4 and B10 into rows 18-24 in the appropriate column in the Audit Report Template. Then delete row 22, which should be blank. Check that the row summaries are calculated correctly in cells C19-C23.

7. Return to the *Spreadsheet Professional* Errors report for each sheet. Starting in row 12, these sheets report the number and cell locations of up to 25 types of potential errors. (All 25 potential errors are listed in the Audit Report Template in rows 55-79 and 81-105.) The names of the potential errors are listed starting in cell A14, and the frequencies (number of offending cells) in cell B14. *Spreadsheet Professional* lists only potential errors identified on the current sheet, not all 25 potential error types. Thus the lengths of these lists will vary from sheet to sheet. *Spreadsheet Professional* also reports the cell addresses of all cells that violate a certain error type. This information can be found farther down in column A. The name of a given error type appears first in bold type and

centered. Then a description of the error follows, which occupies two or three rows. Finally, the cell addresses of the offending cells are listed.

First, record the *frequency* of each of the potential errors that appear under **Potential** errors summary starting in row 12 in the Errors sheet. You will have to copy these numbers one at a time into the appropriate cell in the Audit Report Template in rows 55 to 79. First find the column that corresponds to the relevant sheet. Then find the row that corresponds to the appropriate type of potential error, and enter the frequency.

Second, record the *cell address(es)* for each of the potential errors as given in the Errors sheet. You will have to copy these cell addresses one at a time into the appropriate cell in the Audit Report Template in rows 81 to 105. First find the column that corresponds to the relevant sheet. Then find the row that corresponds to the appropriate type of potential error, and enter the cell address(es).

8. Determine which functions (if any) are used in each sheet. For each sheet, use Display All Formulas (Control+`) or Edit – Find – "=" – Find All to display the formulas. Record the names of the functions you find in rows 26-34 under these categories:

- Financial (e.g., NPV, IRR, PMT)
- Date & Time (e.g., MONTH, TODAY, NOW)
- Math (e.g., SUM, EXP, INT, MAX, ROUND)
- Statistical (e.g., AVERAGE, STDEV)
- Lookup & Reference (e.g., VLOOKUP, OFFSET, INDIRECT)
- Database (e.g., DCOUNT, DVAR)
- Text (e.g., CONCATENATE, LEFT)
- Logical (e.g., IF, AND, COUNTIF)
- Information (e.g., ISERROR, ISTEXT)

If you are not sure which of these categories a given function falls under, use the Function Wizard in Excel to list functions by category.

3. Purpose

1. Take a few minutes to review the workbook and then write a brief description of its purpose in cell C3.

2. In row 3, starting in cell D3, enter a brief description of the purpose of each sheet.

3. In row 4, starting in cell D4, enter the sequence number(s) of the sheet(s) that are referenced in each sheet. That is, record the sequence number(s) of the sheets the provide data *directly* to the given sheet.

4. XLAnalyst error analysis

Return to the XLAnalystReport. Column K lists 17 possible error conditions (excluding Measures and System messages) in rows 10-30. These same possible errors are listed in the Audit Report Template in B37:B53. Wherever you see "Found" in column K of the

XLAnalyst Report, record the cell address (without the sheet name) from column M under the appropriate sheet name in the Audit Report Template, starting in column D, row 37.

Check each set of potential error cell addresses in rows 37-53 in the Audit Template. If any are errors record them under Errors starting in row 131. For information on error types and XLAnalyst Potential Errors see Appendices I and II, respectively.

Follow these steps to document an error:

- a. Record the cell address (or addresses in the case of an error that is repeated) under Errors starting in B131 (be sure to include the *sheet name* in all cell addresses)
- b. Record the total number of cells involved in cell C131
- c. Check the appropriate entry in cell D131 if in your judgment *XLAnalyst* correctly identified the error
- d. Select No credit in cell E131
- e. Select the type of error in F131 (a description of the error types with examples is given in Appendix I)
- f. Select how it was discovered in G131 (*XLAnalyst: numerical tests*)
- g. Enter any comments in H131.

5. SS Pro Map error analysis

Examine the Map sheet for each sheet in turn. Look for patterns that suggest potential errors. Then examine the corresponding cell in the test worksheet to determine if it is in fact an error. For example, you might see an F (for Function) followed to the right by several "<" signs (indicating the function has been copied to the right). If another F appears in the same row, that might indicate an error. For a guide to the symbols used in the SS Pro Map sheets see Appendix III.

Follow these steps to document an error:

- a. Record the cell address (or addresses in the case of an error that is repeated) under Errors starting in B131 (be sure to include the *sheet name* in all cell addresses)
- b. Record the total number of cells involved in cell C131
- c. Select No credit in cell D131
- d. Select No credit in cell E131
- e. Select the type of error in F131 (a description of the error types with examples is given in Appendix I)
- f. Select how it was discovered in G131 (SS Pro: Map analysis)
- g. Enter any comments in H131.

6. SS Pro error analysis

Working with one sheet at a time, spot-check each set of potential error cell addresses in rows 88-105 in the Audit Report Template. For information on SS Pro Potential Errors see Appendix IV.

Follow these steps to document an error:

- a. Record the cell address (or addresses in the case of an error that is repeated) under Errors starting in B131 (be sure to include the *sheet name* in all cell addresses)
- b. Record the total number of cells involved in cell C131
- c. Select No credit in cell D131
- d. Check the appropriate entry in cell E131 if in your judgment *SS Pro* correctly identified the error
- e. Select the type of error in F131 (a description of the error types with examples is given in Appendix I)
- f. Select how it was discovered in G131 (SS Pro: numerical analysis)
- g. Enter any comments in H131.

7. Code inspection

Identify any remaining errors in the workbook. Use the Trace Precedents tool in the Formula Auditing toolbar, the Display All Formulas (Control+`) option, and/or the Edit – Find – "=" – Find All option in your search for errors. Ultimately, you must examine *every formula* in *every sheet*, at least quickly, in order to be confident that you have found all the errors in the workbook.

List any cells that contain errors starting in cell B131.

Follow these steps to document an error:

- a. Record the cell address (or addresses in the case of an error that is repeated) under Errors starting in B131 (be sure to include the *sheet name* in all cell addresses)
- b. Record the total number of cells involved in cell C131
- c. Select No credit in cell D131
- d. Select No credit in cell E131
- e. Select the type of error in F131 (a description of the error types with examples is given in the Appendix I)
- f. Select how it was discovered in G131 (*Code inspection*)
- g. Enter any comments in H131.

8. Performance Testing

Test the performance of the spreadsheet by:

• testing extreme inputs

- try large or small values for individual inputs, including negative values; try unlikely combinations of values
- using Data Sensitivity
 - perform one-way or two-way sensitivity analysis using the Data Sensitivity tool from the Sensitivity Toolkit (<u>http://mba.tuck.dartmouth.edu/toolkit/</u>); look for unexpected behaviors
- using Tornado Chart from the Sensitivity Toolkit
 - vary each input one-at-a time and create a Tornado Chart (also available in the Sensitivity Toolkit); look for inputs that have no effect on the output or an unexpectedly large or small impact.

List any cells that contain errors starting in cell B131.

Follow these steps to document an error:

- a. Record the cell address (or addresses in the case of an error that is repeated) under Errors starting in B131 (be sure to include the *sheet name* in all cell addresses)
- b. Record the total number of cells involved in cell C131
- c. Select No credit in cell D131
- d. Select No credit in cell E131
- e. Select the type of error in F131 (a description of the error types with examples is given in Appendix I)
- f. Select how it was discovered in G131 (Performance testing)
- g. Enter any comments in H131.

9. Qualitative assessments

1. Rate the workbook as a whole on the following eight criteria. Use a scale of 1 to 5, where 1 means "least positive" and 5 means "most positive." Start in cell C108.

- Overall ease of understanding
- Use of modules
- Use of parameterization
- Use of range names for parameters
- Use of range names for formulas
- Ease of use
- Ease of communication
- Technical quality (relative degree of good spreadsheet practices)

2. Note whether these features are observable in the workbook. Use Yes/Some/No as possible responses. Start in cell C117.

- Model assumptions
- Sources for inputs
- Guide to sheets

- Cell comments
- Pseudocode for formulas
- Notes in cells

3. Note whether these features are observable in the workbook. Use Yes/Some/No as possible responses. Start in cell C125.

- Protected cells or sheets (Note: Select Tools – Protection and check each of the four options listed for any type of protection.)
- Hidden cells (Note: You can find this in the XLAnalystReport, cell L20)
- Data validation (Note: You can use Edit – Go To – Special – Data Validation - All to highlights the cells in a worksheet that use Data Validation.)

10. Last steps

Two additional pieces of information should be recorded after completing the audit. One is the total time you have expended on the audit (in minutes) in cell B146 (or below if the Errors section expands beyond row 145). The other is any observations that you wish to make on either the workbook you have audited or on the auditing protocol. Enter these comments in cell B147 (or below if necessary). *Be sure to delete any unused rows in the Error report section*.

Appendix I – Error Type Glossary

An individual cell in a spreadsheet is clearly an error if the value in it is wrong. But there are a number of other conditions which we will record as errors. For example, a cell containing a formula in which an input parameter is duplicated is so dangerous that it warrants being called an error. This appendix lists the seven categories of errors we will track and gives examples of situations in which each error occurs

Logic errors

A formula is used incorrectly, leading to an incorrect result.

Examples:

1. An NPV function that incorrectly discounts the first payment.

2. A VLOOKUP function that refers to non-sorted data inputs.

3. An IF function one or more of whose arguments is calculated incorrectly.

Notes:

1. If a formula is correct except that it refers to incorrect cell addresses, enter that error as a Reference error.

2. This includes conceptual errors, for example, calculating taxes while ignoring depreciation.

Reference errors

A formula contains one or more incorrect references to other cells.

Examples:

1. A formula refers to a parameter in C2 but the parameter is located in B2.

Notes:

1. If the reference error can clearly be traced to incorrect use of relative and absolute addresses, this error should be classified as a Copy/Paste error.

Hard-coding numbers in formulas

One or more numbers appear in formulas and the practice is sufficiently dangerous.

Examples:

1. The discount rate for an NPV calculation appears as a parameter but it also appears as a number within the NPV function itself.

Notes:

1. A number appearing in a formula is not automatically an error.

Data input

An incorrect data input is used.

Examples:

Notes:

Copy/Paste

A formula is wrong due to inaccurate use of copy/paste.

Examples:

1. A correct formula is entered in cell A1 and then copied to B1:F1. The formulas in B1:F1 are incorrect, due to incorrect use of relative and absolute addresses in A1.

Notes:

Omission

A formula is wrong because one or more of its input cell is blank.

Examples:

Notes:

Syntax

A formula is typed incorrectly.

Examples:

Notes:

Circular References

Reason for inclusion

These are normally accidental, if deliberate (using iteration) they still may hide other accidental circular references.

Experience

Circular references are common in full financial models as net profit depends on interest earned which depends on profit made.

Fix Suggestions

It is plausible to leave a circular reference in a spreadsheet and use Excels iteration settings to converge on a result. BUT this is only valid if the reference will actually converge. AND it knocks out a useful check for later users of the model. To fix, trace the dependency tree and if it is an error, correct the wrong reference(s). If it is deliberate or unavoidable, replace some formulas with values at a sensible point.

Modelling Advice

A neat, tidy, well written and well documented VBA routine that replaces the formulas with their results, at one point in the chain, will remove the circular references in a controlled fashion.

Test Limitations

As Circular References are circular (!) they have no start or end, so the test can only highlight one of the cells in the circular chain. Note that the Excel-wide Iteration setting (Tools>Options>Calculation>Iteration) can hide all circular references (see Other below).

Cells Displaying A Number But Storing Text

Reason for inclusion

The contents look like a number, but actually evaluates to zero in many formulas. It is considered non-numeric by Excel.

Experience

One UK School had a GBP 30,000 error in its budget submission because one key cell was showing a number but was treated as zero by the formulas.

Fix Suggestions

Visit each cell found (re-run report until there are no more) and reset its format to a numerical one. Then click in the cell to edit it and press enter to accept. There is no simple fix because each cell must be edited to reset its type to a numerical one.

Modelling Advice

Do not use this format (Text), input text works fine under the default general format. In the very unlikely event that you need to show a number, but for it not to be treated as numeric, prepend an ' (apostrophe) or some other text (Eg frozen>). At least an edit find/replace will work on these.

Test Limitations

The test checks that the problem cell is used by another cell in the same worksheet. This prevents false positives, but does not address the potential for one of these problem cells to be used later. The pro version checks the whole workbook for dependents.

Mixed Formulas And Values

Reason for inclusion

Formulas containing a mix of cell references and numbers can be difficult to maintain. They are often quick fixes and may no longer be required.

Experience

One company had a GBP 1,000,000 error in its budget one year because a late change the previous year had been left in one of the formulas.

Fix Suggestions

If its wrong, go the cell and remove the number part.

Modelling Advice

In general it is better to hold such values in a cell on their own. You can then document them and reference the cell.

Test Limitations

Occasionally this test may bring up a cell containing a formula with an embedded text string that does not refer to another cell. It also ignores 0 and 1 as these are common and usually acceptable (eg IF(formulaResult > 0, ...). It does however return numeric arguments in formulas such as VLOOKUPs, INDEX etc. The pro version is more configurable.

Formulas Evaluating To An Error

Reason for inclusion

A cell containing an error result suggests at least one of the input values to the formula is outside the range expected by the developer.

Experience

A very common cause of error results in finance models is the profit margin calculation (Profit/Revenue). If there is no revenue then this formula returns a '#DIV/0' error.

Fix Suggestions

Check the source cells contain data that is in the acceptable range. If not the error might be a good warning. Often it will be better to extend the formula to cope with the values that caused the errors (zero often).

Modelling Advice

Plan models carefully, try and anticipate all possible data ranges. To be safe, wrap possible error producing formulas with IF(ISERROR(theFormula), yourErrorMessageOrValue, theFormula).

Test Limitations

The tests uses the VBA IsError function, and picks up compatible errors, such as #REF, #DIV/0, #N/A, #NAME etc. There may be others that are missed.

Vlookups Expecting An Ordered List

Reason for inclusion

The 3 argument version of Vlookup expects the table to be sorted. It gives an approximate match if an exact one is not found, rather than an error.

Experience

One company's account code remapping exercise went badly wrong when new codes where added at the bottom rather than inserted in the correct order.

Fix Suggestions

Either add a FALSE as a fourth argument, or make sure the list is sorted. Be sure you know what the formula is doing, and there are no other formulas using the table before making any changes. Approximate matches can be very useful, but the table MUST be sorted.

Modelling Advice

Either version is fine, but if the look up table should be sorted then that must be made clear so a later user doesn't inadvertently create errors.

Test Limitations

The test only looks for the lookups, it does not try to test the table to see if it is sorted.

Hlookups Expecting An Ordered List Reason for inclusion

The 3 argument version of Hlookup expects the table to be sorted. It gives an approximate match if an exact one is not found, rather than an error.

Experience

One company's sales reporting system went badly wrong when a sales representative was replaced with another with a different surname.

Fix Suggestions

Either add a FALSE as a fourth argument, or make sure the list is sorted. Be sure you know what the formula is doing, and there are no other formulas using the table before making any changes. Approximate matches can be very useful, but the table MUST be sorted.

Modelling Advice

Either version is fine, but if the look up table should be sorted then that must be made clear so a later user doesn't inadvertently create errors.

Test Limitations

The test only looks for the lookups, it does not try to test the table to see if it is sorted.

Links To External Workbooks

Reason for inclusion

Changes in the other workbook may not be reflected in this one, or may lead to invalid data being referenced.

Experience

One company linked to subsidiary companies' management accounts, in order to consolidate. One month one of the subsids 'tidied up' their reports. Luckily this caused huge errors and so was spotted, but still cost a lot of time to fix.

Fix Suggestions

Try to prevent users from changing sheet layout - education is better than protection (protection is easy to circumvent). Try using Vlookups or VBA to pull in the data by row description rather than cell position.

Modelling Advice

Links always carry an element of risk, often they are a sign that this piece of analysis would be better in a database. If links are required, have a map of which books link to which. A VBA import routine can be more robust than links.

Test Limitations

The test returns the first external link in the workbook links collection

Presence Of Very Hidden Sheets

Reason for inclusion

Very hidden sheets do not appear in the Format>Sheets>unhide menu, and may contain untested formulas and/or invalid data.

Experience

One consultant had provided a company with a model they could not reconcile. There was an error on a very hidden sheet that they did not know existed. A lot of senior executive time was wasted in the process.

Fix Suggestions

Use the VBA window to check for sheets that are very hidden, look at the visible property. If found, reset to xlSheetVisible, flick back to Excel to see what is on the sheet.

Modelling Advice

Very hidden sheets can be useful in reducing unauthorised changes. Be sure their contents are thoroughly tested, and will not need manual maintenance. Remember that references to very hidden sheets are going to be difficult to audit.

Test Limitations

The test returns the name of the first very hidden sheet in the worksheets collection.

Hidden Rows Or Columns

Reason for inclusion

Data in hidden rows and columns can be inadvertently corrupted when copying and pasting whole rows or columns.

Experience

One spreadsheet stopped working one day, just after some titles had been updated. A key column that a VBA procedure used was overwritten, and the code failed.

Fix Suggestions

Click between A and 1 to select the whole sheet, unhide rows and unhide columns. Do this before making any significant changes.

Modelling Advice

If you need to change the 'view' of the sheet frequently, consider using views, grouping and outlining or a VBA maintenance procedure.

Test Limitations

The test returns the address of the first hidden row or column that it finds.

"=+" Construct

Reason for inclusion

The worksheet could be an old Lotus file, and could be out of date, or the developer used lotus techniques.

Experience

One spreadsheet would not open one day. The back up was recovered and investigated, all formulas had this construct. The '+'s were removed, the file size decreased by about 10% and it never got corrupted again.

Fix Suggestions

Do an edit replace all on '=+' replace with '='. Back up your work first!

Modelling Advice

Don't do it, its scruffy, wasteful, inefficient and unstable. It is also often an indication of poor planning and design.

Test Limitations

The test returns the first '=+' it finds in a formula. This could possibly be in an embedded text string.

Conditional Formatting

Reason for inclusion

Conditional formatting can make it difficult to apply and enforce standard styles to assist user understanding.

Experience

One spreadsheet had a corporate colour scheme to assist users to enter the right data type in the right places. After conditional formatting was added the rules became extremely complex. Each cell's format conditions then had to be edited individually.

Fix Suggestions

To be sure you understand what is happening in a worksheet you may need to temporarily delete conditional formats. Back up first!

Modelling Advice

Control the application and removal of conditional formats with VBA as a development/maintenance routine. You can then switch them on and off simply to test the with and without versions.

Test Limitations

The test checks to see if any cells have conditional formatting set. The cell result does not have to trigger the formatting, so the cell may appear normal. Currently blank cells are not tested, only those with a formula or constant.

Use Of Pivot Tables

Reason for inclusion

Pivot tables may be displaying out of date data.

Experience

Pivot tables are very powerful, but it is easy to forget to refresh them to take account of data changes. One company reported the same figures for several months on certain lines until someone realised the pivot tables supplying that data needed refreshing.

Fix Suggestions

If the sheet is not linked to, you can use the sheet activate event to trigger a refresh. You can also set the pivot table to refresh on opening the workbook. To manually refresh a pivot table, right click anywhere in the table and click refresh.

Modelling Advice

Be sure pivot tables carry a warning about refresh state. Pick the most appropriate event and force a refresh in that event handler. Be aware people may link to the pivot and they should be looking at up to date data too.

Test Limitations

The test finds the first pivot table in the workbook. It does no further checking.

Array Formulas

Reason for inclusion

Array formulas are difficult to break into simple understandable parts. They can also cause problems when copying and pasting

Experience

One company could not manipulate regional submissions into a useable layout as many formulas were array formulas spanning 12 columns.

Fix Suggestions

Where possible break out into multiple cells, use a whole row or column if required.

Modelling Advice

Often you can use SUMPRODUCT instead of certain types of array formula. Its faster and easier to follow. If not, consider an intermediate row or column to enable you to break the formula into multiple simple ones.

Test Limitations

The test finds the first cell whose HasArray property is true.

Nested If Statements

Reason for inclusion

With nested lfs, many return values are possible. Testing all possible results therefore becomes difficult.

Experience

One nested IF had to be broken out into 4 separate cells to understand why it didn't seem to work. It turned out one set of choices was the wrong way round, but the error only appeared very rarely, as this choice was unusual.

Fix Suggestions

Break out into multiple IF formulas in multiple cells.

Modelling Advice

You will need to step though this formula one argument at a time to test it. Make it easy and spread it over several cells, hide the intermediate ones if necessary. Consider using a lookup (approximate if necessary).

Test Limitations

The test looks for any formulas that have IF in them twice - this could possibly occur in an embedded text string.

Use Of Sumif

Reason for inclusion

SUMIFs make testing and auditing difficult, as it is not always obvious from the context what is included and what isn't. It is not possible to know which actual cell values make up the total - this makes it difficult to confirm the value is correct.

Experience

One spreadsheet made heavy use of SUMIFs to group products by product codes. The values from one product were missed for months because its code had been mistyped. Just checking the direct precedents gave a false sense of security that everything was included.

Fix Suggestions

To test SUMIFs, use another row or column and put an IF statement to include the value if it matches the criteria and a zero if not. Then use a standard SUM.

Modelling Advice

If possible, lay out the information so you can use simple SUMs, or use a 'flag' column or row. If not use SUMIFs, but test them as suggested.

Test Limitations

The test looks for any formulas that have SUMIF in them - this could possibly occur in an embedded text string.

Use Of Database Functions (Dsum Etc) Reason for inclusion

Dfunctions make testing and auditing difficult as it is not always obvious from the context what is included and what isn't. It is not possible to know which actual cell values make up the total - this makes it difficult to confirm the value is correct.

Experience

One spreadsheet used DSUMs to build reports. A new business unit was added, but its results would not report correctly. To fix this error, advanced filtering and pivot tables were used, as it is not possible to audit to the actual cells making up the result. It turned out to be an ambiguity in the criteria.

Fix Suggestions

To test, use another row or column and put an IF statement to include the value if it matches the criteria and a zero if not. Then use a standard SUM.

Modelling Advice

If using these functions you should test using pivot tables or advanced filtering and extract results to a separate sheet for analysis. It may be more appropriate to use a fully featured database, perhaps using a pivot table to provide flexible Excel reporting.

Test Limitations

The test looks for any formulas that have any of the main D functions (DSUM, DMAX, DMIN, DCOUNT, DCOUNTA, DAVERAGE) in them - this could possibly occur in an embedded text string. It does not look for any of the other D functions.

Use Of Indirect Reason for inclusion

Use of INDIRECT breaks cell dependency methods of testing and auditing. This makes it complex to trace.

Experience

One consultant provided a company with a model that made significant use of INDIRECT (100's of them). Opening this model with calculation set to automatic crashed Excel. This model was replaced with a simple stable VBA solution (100 lines of code).

Fix Suggestions

Replace with INDEX or MATCH or a V or H LOOKUP.

Modelling Advice

There are better, easier to audit functions that provide similar functionality, use those instead. Consider one of the fix suggestions or maybe use OFFSET.

Test Limitations

The test looks for any formulas that have INDIRECT in them - this could possibly occur in an embedded text string.

Longest Formula

Reason for inclusion

Longer formulas are generally harder to understand - and some may be so long as to be un-editable. **Experience**

Some formulas in one model could not be edited as they exceeded Excels formula length limit (1024 characters in Excel XP). They seemed to work, but it wasn't possible to be truly sure. The model had to be re-written.

Fix Suggestions

Split long formulas into multiple cells if possible

Modelling Advice

Keep names (eg sheet names, range names) meaningful but short (perhaps 5-12 characters) so they are easy to follow in formulas. Use camel case (mixed upper and lower) rather than spaces or extended characters for the same reason (eg: use 'PLReport' rather than 'P & L Report' as a sheet name).

Test Limitations

The test counts all the characters within a formula and returns the longest it finds.

Most Complex Formula Reason for inclusion

Formulas with lots of variables and lots of operators are difficult to test and to trace errors through.

Experience

One client's spreadsheet had formulas that contained six nested IFs and five ANDs. This had to be broken out into six separate cells to be understandable.

Fix Suggestions

Split complex formulas into multiple cells if possible.

Modelling Advice

Split into multiple cells so the intermediate results can be seen, testing is then much easier.

Test Limitations

The test counts the number of arithmetical operators it finds in each formula and returns the largest number. Some of these operators may occur in embedded text strings.

Total Number Of Formulas

Reason for inclusion

As the number of formulas increases, models become harder to develop, test and audit.

Experience

With more than 20,000 formulas all depending on each other, it is essential that the right ones go in the right places.

Fix Suggestions

Always remove any unused parts of a model as soon as they are retired. Keep an archive model if necessary, but try to keep the live version as light as possible.

Modelling Advice

In general more simple formulas is better than less very complicated ones, so there is a balance. Never combine formulas just to reduce the total number.

Test Limitations

The test counts every cell that Excel considers a formula. Merged cells are counted once for each cell within the merged area.

Total Number Of Unique Formulas

Reason for inclusion

The total number of unique formulas is a better gauge of the model complexity than just the total number of formulas.

Experience

Any model that has over 1,000 different formulas is doing a wide range of work and will be challenging to test and maintain.

Fix Suggestions

It may be possible to split the model into multiple models that can be more easily understood separately. If so do it.

Modelling Advice

Overall model complexity is related to the number of distinct functions it performs. Several models, each of which does one thing well, is better than a single huge model that is barely understandable but seems to do a few things ok.

Test Limitations

The test counts every unique formula. It considers the relative address of any precedents to exclude similar formulas.

Workbook Size

Reason for inclusion

Larger workbooks are harder to develop, test and audit.

Experience

32Mb files can be difficult to open on some systems. 16Mb file sometimes open, but never feel stable. 2-3 Mb or less is a more comfortable size.

Fix Suggestions

Large workbooks should be split into smaller ones if possible and sensible, or maybe rebuilt in a database.

Modelling Advice

A model should be as simple as it can be and still do the job well. All elements should justify their place in the model. Data that is common for many models (eg tax rates, or absorption rates) should be kept separately if possible. This reduces duplication, which reduces the chances for inconsistencies and makes maintenance easier.

Test Limitations

The test checks the size of the workbook on disc. It does not take account of changes since it was last saved.

No Of Worksheets

Reason for inclusion

Often a sheet represents an analysis unit, such as business unit, or time, or report type. High numbers of sheets often means a complex workbook.

Experience

One client had a workbook containing 48 monthly worksheets. Keeping track whilst paging up and down through these was difficult. Figures were not always in the same place on each sheet which added to the challenge.

Fix Suggestions

A model that has many worksheets may be better in a database. If each sheet is fairly small, it may be better to change the modelling and use something else as the sheets (eg instead of month, use year, or business unit, and put month as a column or row on the sheets).

Modelling Advice

Data that is spread across many sheets is difficult to manipulate and consolidate. If the sheet structure is similar, consider a database and pivot table. If each sheet is different then consider multiple models.

Test Limitations

The test counts worksheets only, not macro sheets or chart sheets.

Total All Lines of VBA Code

Reason for inclusion

VBA can be the source of very bizarre behaviour and hard to track errors.

Experience

9,000 Lines of code becomes a full model in its own right. Well written, well commented code is fine, but rare in Excel VBA development. Even 50 lines of bad, uncommented spaghetti code can take several days to unravel.

Fix Suggestions

Be sure you understand what the VBA is meant to be doing, or get someone in who does. Open workbooks with macros disabled to try and assess if VBA could be causing problems. You can sometimes trap the sheet change event and see which piece of code is doing the changing.

Modelling Advice

Write neat, tight, well designed, well documented code or get someone in to do it. Remember the most important reader of your code is the maintenance programmer. They will greatly appreciate your efforts to make their job achievable. The macro recorder can help with syntax, but is just not robust enough alone for live code.

Test Limitations

This is a very quick very approximate test. It just counts the total number of lines in each VBA component, including blanks and comments. VBA editor settings can impact this, even for workbooks with no 'active' VBA code. Use XLAnalyst Pro VBA for a more detailed analysis.

Largest Formula Result

Reason for inclusion

The largest number gives an indication of the sort of numbers being analysed. Big numbers bring their own problems of rounding and presentation

Experience

Rounding errors at one company were GBP Billions. Custom formats and explicit rounding were needed to manage them.

Fix Suggestions

Make sure this number is reasonable in the context of your model. Use ROUND to explicitly control precision, use similar formats throughout the model.

Modelling Advice

Test Limitations

The test finds the largest numerical value returned by a formula in the workbook. Values in the 30,000-36,000 range may be recent dates that are the result of a formula.

Codematic Risk Management Components

Reason for inclusion

Risk management components increase accountability and indicate spreadsheet risks are taken seriously.

Experience

When people have to personally, visibly take responsibility for their spreadsheets, they check them more carefully.

Fix Suggestions

If they are found, be sure they are being used properly, if they are not found consider some method of connecting the person responsible for a spreadsheet and the consequences of material errors. Visit www.codematic.net for more information.

Modelling Advice

Any important spreadsheet should always have a sheet that explains a bit about the model, the author and the potential risks. This should not be so onerous as to discourage its use.

Test Limitations

The test only looks for the components of the Codematic Risk Management program, other in-house features may provide the same quality improvements.

Protected Worksheets

Reason for inclusion

Several XLAnalyst tests fail if the sheet is protected, in particular you cannot return the set of cells containing formulas from a protected sheet. Those tests that are affected are hatched out in red, this is nearly all tests!

Experience

In all versions of Excel up to and including 2003, sheet protection breaks all auditing functionality. This makes understanding a model very hard. It also suggests that it may not have been tested effectively recently (since it was protected).

Fix Suggestions

To work properly with the sheet, it needs to be unprotected. The proversion of this tool breaks sheet protection as part of its tests. Password protection isn't something that needs a 'fix' as such.

Modelling Advice

If you are protecting sheets, use VBA to enable multiple sheets to be protected and unprotected easily. Don't rely on password protected sheets remaining stable.

Test Limitations

This version cannot test protected sheets, the pro version removes worksheet protection so the tests can be run.

Protected Workbook Structure

Reason for inclusion

Workbook structure protection stops the sheets visible property from being changed. This causes some XLAnalyst tests to fail.

Experience

Workbook structure protection is rarely used, but is likely break any VBA routines that work with sheets.

Fix Suggestions

Workbook structure protection stops people hiding and unhiding, deleting and adding sheets and changing the sheet order. It doesn't have a 'fix' as such.

Modelling Advice

Use it if you need it, but better to rely on VBA to control the user experience of your model. Workbook structure protection is fairly straightforward to break.

Test Limitations

None of the tests on this particular research version will fail due to workbook structure protection. But earlier versions (prior to version R.1.3.64) were affected and later ones might be. Any failures caused by protection would be reported on this line.

Other

Reason for inclusion

This area is reserved for system messages that do not fall into the other categories. This version only considers the Iteration setting. Iteration is an Excel wide setting. Iteration can be used with a circular reference to iterate to a solution. But it may mask a problem elsewhere in the workbook, or indeed any open one, until the iteration setting is changed or Excel shut down.

Experience

During testing it was discovered this tool did not always find known circular references. One test workbook had the iteration set to on, all subsequent tests failed to find circular references, even after that workbook was closed.

Fix Suggestions

Routinely check the calculation settings. Turn iteration off to check for unexpected circular references. Watch out for a 'calculate' in the status bar that will not go away - this may mean something somewhere has set Iteration to on.

Modelling Advice

Don't use circular references and don't set the Iteration setting (Tools>Options>Calculation). Use a VBA routine instead, it may not be as slick, and may be slower, but it is so much more robust and controllable.

Test Limitations

The test checks the Iteration status at the end of the other tests, this means any circular references will have been missed. Turn iteration off and re-run the test if you need to be sure. The PRO version operates differently and does not miss them.

Appendix III – Spreadsheet Professional Map Analysis Key

Map Analysis Key:

- L Label =
- Numeric Input Ν =
- F =
- Unique Formula Formula copied from left = <
- \wedge =
- Formula copied from right Formula copied from both left and above +=

Appendix IV – *Spreadsheet Professional* Descriptions of tests

Each test is described below together with the reasons as to why this might create an error.

Tests on Specific Cells

Unused input value

A number has been entered which is not used in any calculation.

1. A calculation contains an incorrect reference.

2. The writer of the spreadsheet may have forgotten that this input value exists and hard coded the value directly into subsequent calculations.

Unused calculation

The results from this calculation are not used elsewhere on this worksheet 1. The results should be used but there is an incorrect reference in a subsequent calculation.

No precedents

This formula does not depend on any other cells. Usually this implies that an input has been entered as a combination of values.

1. Unless the individual values that make up the input are documented then it will be impossible to subsequently understand how the results were derived.

Blank cells referenced

The following calculations reference a blank cell.

1. An input value has not been entered.

2. The calculation contains an incorrect reference.

3. There maybe no error at present but users may subsequently enter values or formulas into the blank cells causing inconsistent results.

Errors referenced

The following calculations reference one or more cells containing an error.

1. Excel is indicating that one or more calculations result in errors. This will lead to unpredictable results.

Non numeric cells referenced

Although the following calculations have numeric answers they refer to non-numeric cells.

1. A value has been entered as a non numeric.

2. The calculation contains an incorrect reference.

Forward row reference

The following calculations refer to a row after the row in which they are situated.

1. Well written spreadsheets should be read from top to bottom like a book. Forward references often indicate a late additional piece of code which has been inadequately checked.

2. The calculation contains an incorrect reference.

Forward column reference

The calculation refers to a column to the right of the column in which it is situated. 1. Well written spreadsheets should be read from left to right like a book. Forward references often indicate a late additional piece of code which has been inadequately checked.

2. The calculation contains an incorrect reference.

Hidden cell referenced

The following calculations reference to hidden cells.

1. It is impossible to check these calculations by examining the spreadsheet. In practice they are often found to contain errors.

NPV function

The following calculations use the NPV function. This is used incorrectly in over 30% of all cases examined.

 The Excel NPV function assumes that all cashflows occur at the end of each period. If you are assuming mid period or opening period cashflows substantial errors will occur.
 The cashflows must also be equally spaced in time.

VLOOKUP function

The following calculations use the VLOOKUP function.

1. The range being searched must be in ascending order or errors may occur.

HLOOKUP function

The following calculations use the HLOOKUP function. 1. The range being searched must be in ascending order or errors may occur.

LOOKUP function

The following calculations use the LOOKUP function.

1. The range being searched must be in ascending order or errors may occur.

IF function

The following calculations contain an IF statement.

1. The calculation used is dependent on the input values to the spreadsheet therefore these cells must be checked particularly carefully.

Double IF function

The following calculations contain two or more IF statements.

1. These are complex calculations which have been found to be incorrect in more than 25% of cases.

2. Ensure that all possible values are allowed for.

Numeric rule

The following calculations contain a number. This is the most common cause of errors within a spreadsheet.

1. A number has been added to the calculation as a 'quick fix' and not been subsequently removed.

2. A number has been used within the calculation even though it is also input elsewhere on the spreadsheet. Changing the input then has no effect.

3. The number is being used to convert from one set of units to another (000s to millions etc). This is often performed incorrectly.

Dependents rule

Although this calculation is a copy of the one to its left, it is referenced by a different number of subsequent calculations.

1. A calculation contains an incorrect reference.

Complex calculation

This calculation is particularly complex and therefore likely to contain errors. 1. Errors can be of all types.

Unprotected calculation.

This calculation is not protected and can be overwritten by the users of the spreadsheet. 1. Subsequent amendment (either accidently or deliberately) may result in incorrect results being derived.

External reference

This calculation references a value on a separate spreadsheet. 1. These cells often contain out of date values which have not been updated.

Date reference

This calculation references a date. 1. This calculation may not work over the year 2000. Check formula.

Two digit integer reference

This calculation references a two digit integer. It maybe a date.

1. If this two digit integer is a date, there is a high probability that the formula may not work past the year 2000.

References the value 99

This calculation references the value 99. It maybe the year 99.

1. If this value represents 1999, check the formula in each column to ensure that it will work past the year 2000.

General tests

These tests examine potential errors that are not specific to one particular cell.

Relative range names

This test examines for relative range names. These often cause errors, because the cells they refer to depend on where the relative range name is referenced.

Lotus eval rules

If a spreadsheet is imported from Lotus 123 it inherits the Lotus evaluation order and rules. This is slightly different to the evaluation order within Excel and therefore may create an error.

External links

This test checks for external links. These must be updated before the final results are produced. – They often aren't.

Calculation manual

If worksheet protection is set to off, this may create an error if an input is updated, and the spreadsheet is not recalculated.

Duplicate range names

This test looks for duplicate range names which maybe an indication that a particular range has not been correctly named.