

TK Solver Case Study: Converting Excel Formulas

Many customers are interested in converting their existing spreadsheet files to TK Solver files. One reason is that TK files can be run with RuleMaster, effectively distributing those models worldwide from a single server. Application developers can maintain their code more easily in TK Solver and they don't have to send new files to their end users every time they make a change. Users also benefit from TK's list solving, backsolving, and unit conversion capabilities – features not available in Excel. And they can combine their formulas with those from the existing TK-based applications such as Roark's Formulas, Heat Transfer, Queueing Theory, and Financial Management.

The conversion process is simple. Highlight and copy the spreadsheet cells to be converted. Then switch to TK Solver and paste into the Rule Sheet. For example, if cell C4 contains the formula $=\text{SQRT}(A4^2+B4^2)$, and that cell is copied to TK, the rule will appear as $C4 = \text{SQRT}(A4^2+B4^2)$. The process can be improved by first using the MathLook for Excel utility to provide more meaningful names to the spreadsheet cells. The cell names are automatically transferred to TK with the formulas.

Many spreadsheets contain tables of calculations based on copied formulas. For example, if cell B4 in the above example is copied and pasted to the range B5..B20, the same formula will simply be repeated sixteen times with the row number changing in each cell. If that column were then copied and pasted into TK, you would have sixteen rules. The first five are shown below.

Rule
$C5 = \text{SQRT}(A5^2+B5^2)$
$C6 = \text{SQRT}(A6^2+B6^2)$
$C7 = \text{SQRT}(A7^2+B7^2)$
$C8 = \text{SQRT}(A8^2+B8^2)$
$C9 = \text{SQRT}(A9^2+B9^2)$

It is unlikely that this is the approach you would want to take. You would suddenly have 45 new variables in your TK model. There are two better options. You can list the rule once and use TK's List Solver to solve it repeatedly over a range of inputs. You can create a procedure function which loops through the range of inputs, solving for the new value of C at each step. The best choice depends on the requirements of the end user. Procedure functions are used when more automation is desirable. List Solving gives the end user more control. For RuleMaster applications, procedure functions are the best choice.

Another conversion issue deals with incompatibility of built-in functions. For example, if cell G10 contains the formula $=\text{ROUND}(F10,2)$ and F10 has the value 1.23456, Excel reports the value 1.23. If that formula is copied to TK, the result will be 2. This is because the ROUND function is defined differently. The correct TK translation would be $G10 = \text{ROUND}(F10,.01)$. Here is a listing of incompatible functions: CEILING, COUNT, DATE, FLOOR, ROUND, TIME, VALUE.

Excel includes many functions which are not available as TK built-in functions. For example, the Excel function SQRTPI returns the square root of the product of a number and the constant pi. In such cases, an equivalent TK function must be made available on the Function Sheet. A collection of functions has already been created and is available through the UTS web site. The functions are stored in a single TK file which can be merged into your TK applications as required.

The Excel function IF is a special case. Excel handles conditions through the use of the IF function. For example, the Excel formula $=\text{IF}(B3<B4,100)$ will return 100 if B3 is less than B4, otherwise it will return the word FALSE. There is a second usage which returns a second value if the condition is false. The formula $=\text{IF}(B3<B4,100,80)$ will return 80 if B3 is not less than B4. TK provides an IF THEN ELSE construct which is not compatible with Excel's function. To make the translation possible, two TK functions are provided in the collection on our website. The TK Function IF assumes three arguments, and performs the equivalent of TK's IF THEN ELSE construct. Function IF1 assumes the simpler case with

only two arguments. When you copy formulas from Excel, check the resulting TK rules. If any have an IF function with only two arguments, change the IF to IF1 and it will work just as it did in Excel.

More advanced spreadsheets with multiple worksheets and VBA code are probably not good candidates for direct conversion to TK. In these cases, it has been my experience that starting from the initial flowchart and building the TK model “from scratch” is most efficient.

If you discover any other shortcuts in converting spreadsheets to TK, I’d appreciate hearing from you.

- Todd Piefer, UTS Project Mgr.