Command Reference: WriteTableToExcel()

Write a table to a Microsoft Excel workbook file

Version 11.03.02, 2015-06-10

The WriteTableToExcel() command writes a table to a worksheet in a Microsoft Excel workbook file. A contiguous block of cells (rectangle) must be specified in one of the following ways to receive the table:

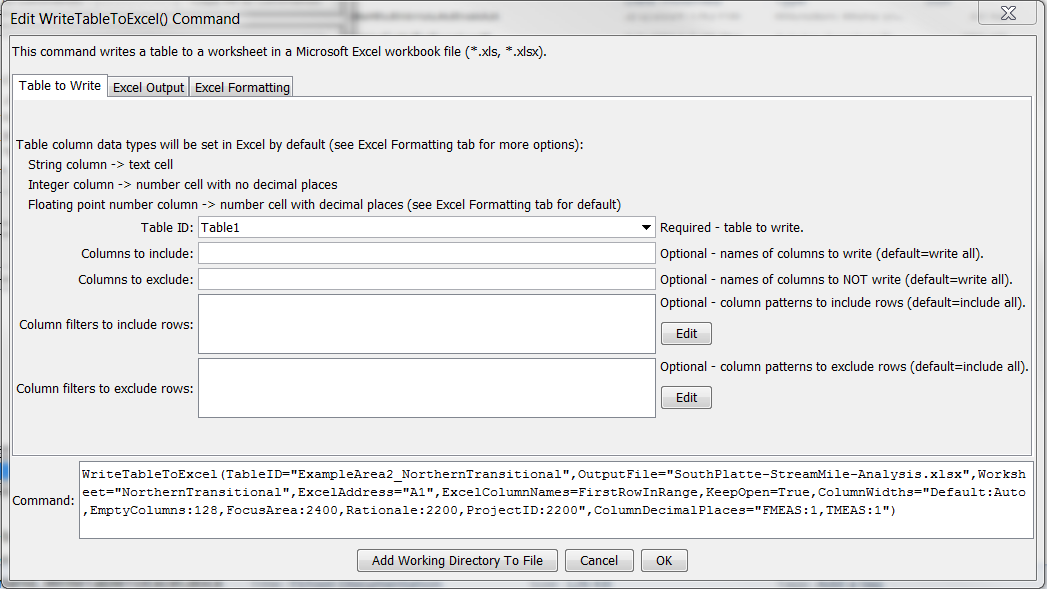
* Specify the upper-left cell in a range of cells using Excel address notation (e.g., A1)
* Specify a range of cells using Excel address notation (e.g., A1:D10)
* Specify the name of an Excel named range.
* Specify a table name (essentially a named range).

TSTool uses the Apache POI software (http://poi.apache.org) to read/write the Excel file and consequently functionality is constrained by the features of that software package.

The following are limitations of this command:

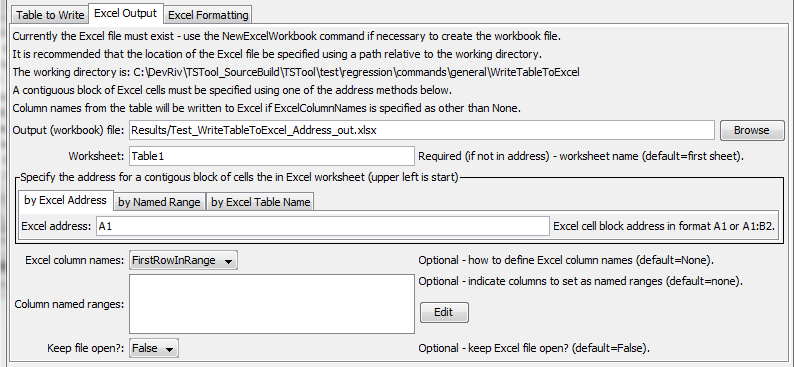
* Currently only the upper-left cell is utilized but in the future the range may be used to limit output.
* Cell data types are determined from the table columns being written. In the future a parameter may be provided to allow option of using original Excel formatting.

The following figures illustrate the dialog used to edit the command and the syntax for the command.



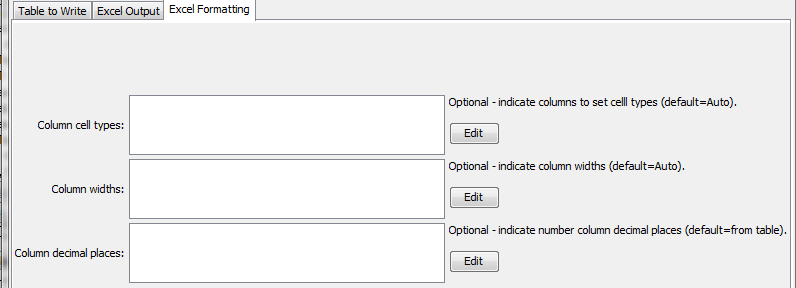
WriteTableToExcel

WriteTableToExcel() Command Editor



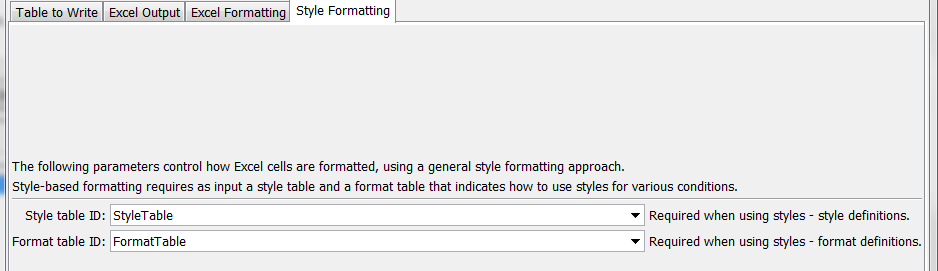
WriteTableToExcel\_Excel

WriteTableToExcel() Command Editor for Excel Output Parameters



WriteTableToExcel\_Formatl

WriteTableToExcel() Command Editor for Excel Formatting Parameters



WriteTableToExcel\_Style

WriteTableToExcel() Command Editor for Style Formatting Parameters

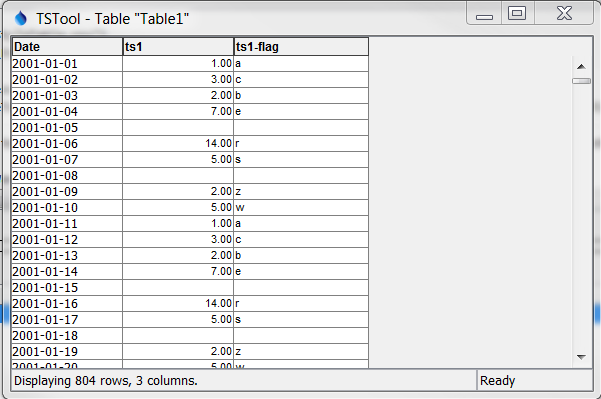
The command syntax is as follows:

WriteTableToExcel(Parameter=Value,…)

Command Parameters

| Parameter | Description | Default |
| --- | --- | --- |
| TableID | Identifier for table to write. Can specify using processor ${Property}. | None – must be specified. |
| IncludeColumns | Names of columns in table to write. | Write all columns. |
| ExcludeColumns | Names of columns in table to exclude from writing. | Write all columns. |
| ColumnInclude  Filters | Indicate table column names and pattern to use to include rows. For example, include rows with blanks in columns. The format of the parameter is:  ColumnName1:Pattern1,  ColumnName2:Pattern2,…  where patterns can contain \* to match a substring. | Include all rows. |
| ColumnExclude  Filters | Indicate table column names and pattern to use to exclude rows. For example, exclude rows with blanks in columns. The format of the parameter is:  ColumnName1:Pattern1,  ColumnName2:Pattern2,…  where patterns can contain \* to match a substring. | Include all rows. |
| OutputFile | The name of the Excel workbook file (\*.xls or \*.xlsx) to write, as an absolute path or relative to the command file location. If the Excel file does not exist it will be created. Can specify using processor ${Property}. | None – must be specified. |
| Worksheet | The name of the worksheet in the workbook to write. If the worksheet does not exist it will be created. Can specify using processor ${Property}. | Write to the first worksheet. |
| ExcelAddress | Indicates the block of cells to write, using Excel address notation (e.g., A1:D10). | Must specify address using one of available address parameters. |
| Excel  NamedRange | Indicates the block of cells to write, using an Excel named range. | Must specify address using one of available address parameters. |
| ExcelTableName | Indicates the block of cells to write, using an Excel named range. | Must specify address using one of available address parameters. |
| ExcelColumn  Names | Indicate how to determine the column names for the Excel table (in order to not overwrite with data rows), one of:   * FirstRowInRange – column names are written to the first row in the Excel address range * None – column names are not written * RowBeforeRange – column names are written to the row before the Excel address range | None |
| ColumnNamed  Ranges | The map of column names to named ranges, useful when the column of values is used as choices in Excel data validation. | No named ranges will be defined. |
| KeepOpen | Indicate whether to keep the Excel file open (True) or close after creating (False). Keeping the file open will increase performance because later commands will not need to reread the workbook. Make sure to close the file in the last Excel command. | False |
| Column  CellTypes | Column names and corresponding cell types using notation: ColumnName1:CellType1,ColumnName2:CellType2. Column name can be Default to set the default for all output columns. Supported cell types are:   * Auto – determine cell type from table column * Text – Excel text cell | Auto |
| ColumnWidths | Column names and corresponding widths using notation: ColumnName1:Width1,ColumnName2:Width2. Column name can be Default to set the default for all output columns and EmptyColumns to set the width for columns with no data values. Supported width values are:   * Auto – determine width from table contents * N – number of 1/256 of character widths (maximum is 256\*256) | Default column with determined by Excel. |
| ColumnDecimal  Places | Column names and corresponding number of decimal places, for floating point numbers, using notation: ColumnName1:Num1,ColumnName2:Num2. | Determine from table column precision, or 6 if unable to determine from table. |
| StyleTableID | Identifier for style table (see below). Can be specified using processor ${Property}. | Style table not used. |
| FormatTableID | Identifier for style table (see below). Can be specified using processor ${Property}. | Style table not used. |

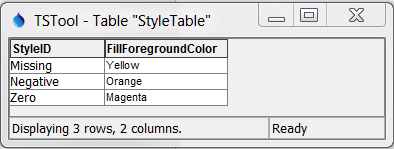
Excel cell formatting consists of number formatting, cell colors, cell width, etc. The Excel Formatting tab allows several formatting parameters to be specified. However, a more general formatting capability based on styles is being phased in and is configured using the Style Formatting tab. Consider the following data table, where the goal is to write the TSTool table to Excel and format cells to indicate specific conditions of interest. Note that this approach is being implemented in a similar way for the WriteTimeSeriesToExcel() command.



WriteTableToExcel\_DataTable

Data Table used with WriteTableToExcel() Command Style Formatting

To configure style-based formatting, a style table is defined listing properties for formatting a cell. This table can be defined as a CSV file, Excel worksheet or other format and read into TSTool using a suitable command. The following figure illustrates a basic style table, which can be shared among commands.



WriteTableToExcel\_StyleTable

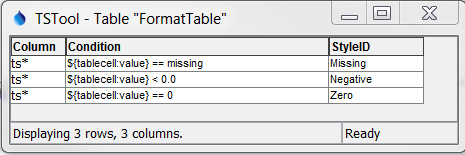
Style Table used with WriteTableToExcel() Command Style Formatting

The following style table columns are recognized. The default values for cell style properties not listed in the table are those provided by Excel.

Recognized Style Table Columns

| Column Name | Description | Default |
| --- | --- | --- |
| StyleID | An identifier for the style, which is used in the format table below. | None – must be specified. |
| FillForegroundColor | The foreground fill color as a named color (e.g., “Red”), RGB triplet (255,255,255), or hex color 0xFFFFFF. The following named colors are recognized: black, blue, cyan, darkgray, gray, green, lightgray, magenta, none, orange, pink, red, white, yellow. | No fill color. |
| FillPattern | Fill pattern for cells using FillForegroundColor and FillBackgroundColor. | Currently always defaults to solid. |

The format table indicates how the styles are used for columns and cell values, as shown in the following example, which indicates that any columns with names starting with “ts” should be processed to evaluate for missing and negative values.



WriteTableToExcel\_FormatTable

Format Table used with WriteTableToExcel() Command Style Formatting

For the above style and format tables, the column names must be specified as shown. The Condition column recognizes the following specifiers (note that the WriteTimeSeriesToExcel() command uses ${tsdata:value}, ${tsdata:flag}, and ${ts:property}):

* ${tablecell:value} – the cell value (raw value before any formatting)
* ${tablecell:comment} – the cell comment (currently not implemented in TSTool tables but is reserved for future implementation and Excel cell comments)

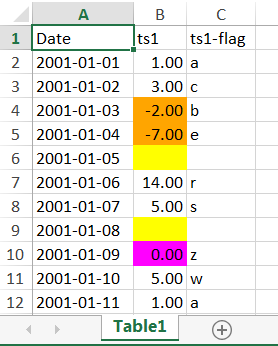
Values on the left and right of the operator must be separated with spaces to facilitate parsing the condition. The Condition column recognizes the following operators:

Format Table Condition Operators

| Operator | Description |
| --- | --- |
| < | Less than. |
| <= | Less than or equal to. |
| == | Equal to. Specify the right-side value as missing to check for missing. |
| != | Not equal to. Specify the right-side value as missing to check for missing. |
| > | Greater than. |
| >= | Greater than or equal to. |
| contains | Specify for string values to check for substring (case-independent). |

Currently only one condition per format table row can be specified but more complex conditions will be supported in the future.

The following figure illustrates the output from the above example.



WriteTableToExcel\_Output

WriteTableToExcel() Command Example Output

This page is intentionally blank.