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:

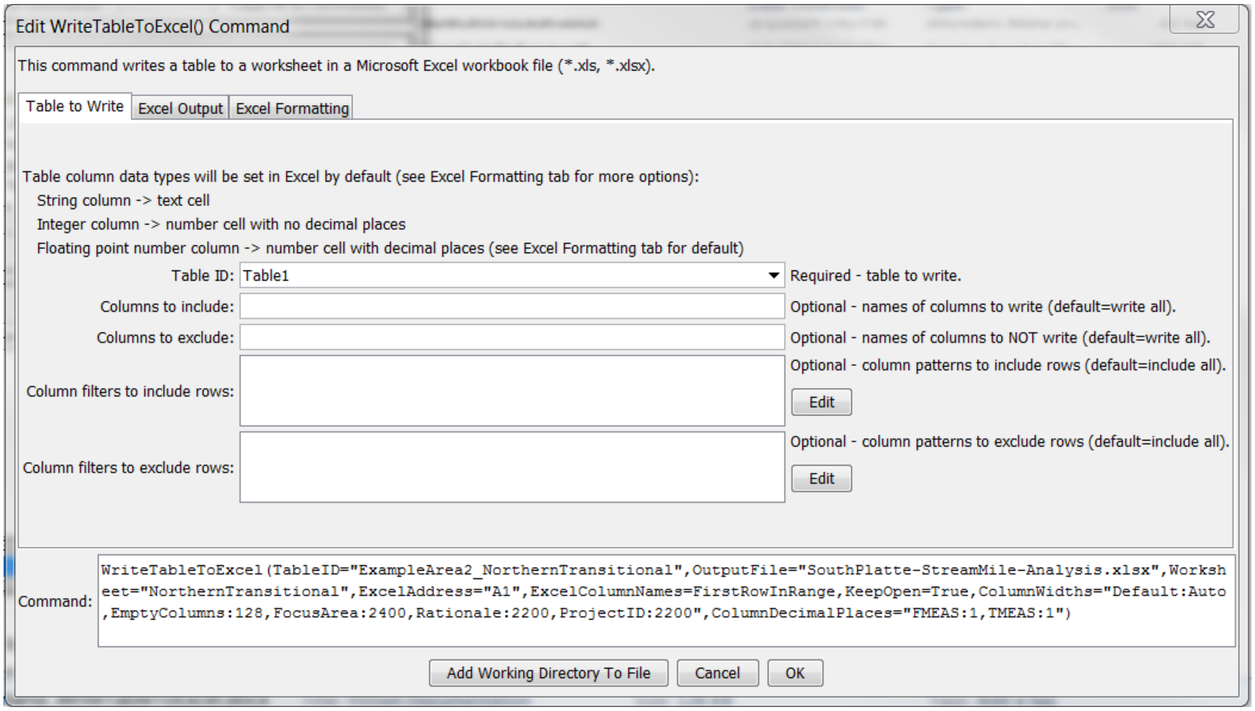
- Specify the upper-left cell in a range of cells using Excel address notation (e.g., `A1`)
- Specify a range of cells using Excel address notation (e.g., `A1:D10`)
- Specify the name of an Excel named range.
- Specify a table name (essentially a named range).

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.



This command writes a table to a worksheet in a Microsoft Excel workbook file (*.xls, *.xlsx).

Table column data types will be set in Excel by default (see Excel Formatting tab for more options):
String column -> text cell
Integer column -> number cell with no decimal places
Floating point number column -> number cell with decimal places (see Excel Formatting tab for default)

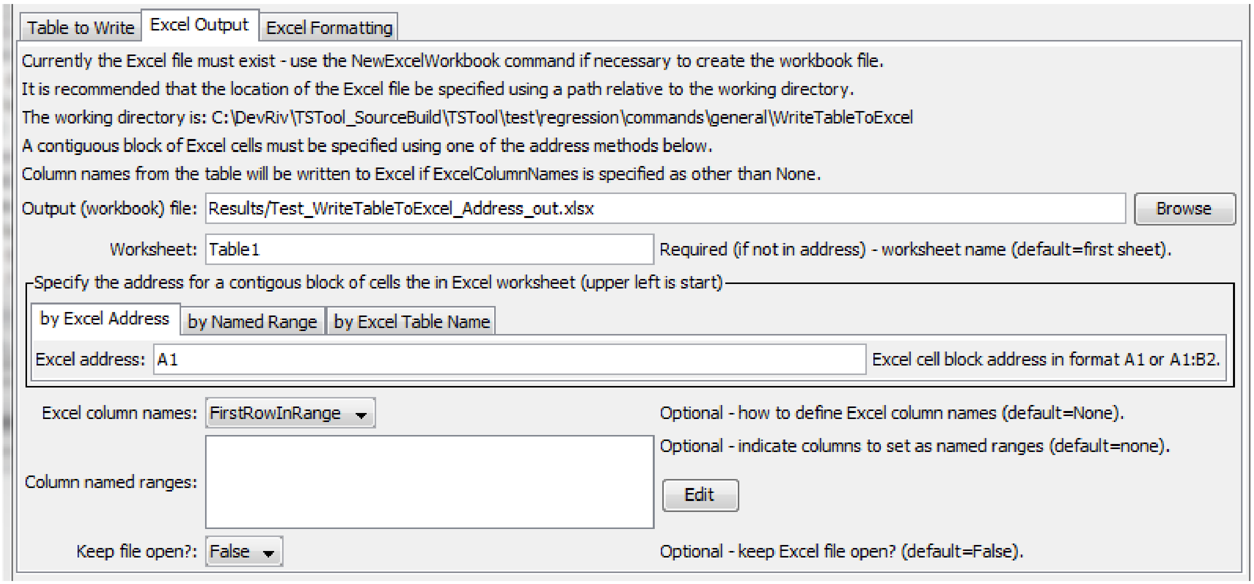
Table ID: Table1 Required - table to write.
Columns to include: Optional - names of columns to write (default=write all).
Columns to exclude: Optional - names of columns to NOT write (default=write all).
Column filters to include rows: Optional - column patterns to include rows (default=include all).
Column filters to exclude rows: Optional - column patterns to exclude rows (default=include all).

Command:

```
WriteTableToExcel(TableID="ExampleArea2_NorthernTransitional",OutputFile="SouthPlatte-StreamMile-Analysis.xlsx",Worksh
eet="NorthernTransitional",ExcelAddress="A1",ExcelColumnNames=FirstRowInRange,KeepOpen=True,ColumnWidths="Default:Auto
,EmptyColumns:128,FocusArea:2400,Rationale:2200,ProjectID:2200",ColumnDecimalPlaces="FMEAS:1,TMEAS:1")
```

WriteTableToExcel

WriteTableToExcel() Command Editor



Currently the Excel file must exist - use the NewExcelWorkbook command if necessary to create the workbook file.
It is recommended that the location of the Excel file be specified using a path relative to the working directory.
The working directory is: C:\DevRiv\TSTool_SourceBuild\TSTool\test\regression\commands\general\WriteTableToExcel
A contiguous block of Excel cells must be specified using one of the address methods below.
Column names from the table will be written to Excel if ExcelColumnNames is specified as other than None.

Output (workbook) file: Results/Test_WriteTableToExcel_Address_out.xlsx

Worksheet: Table1 Required (if not in address) - worksheet name (default=first sheet).

Specify the address for a contigous block of cells the in Excel worksheet (upper left is start)

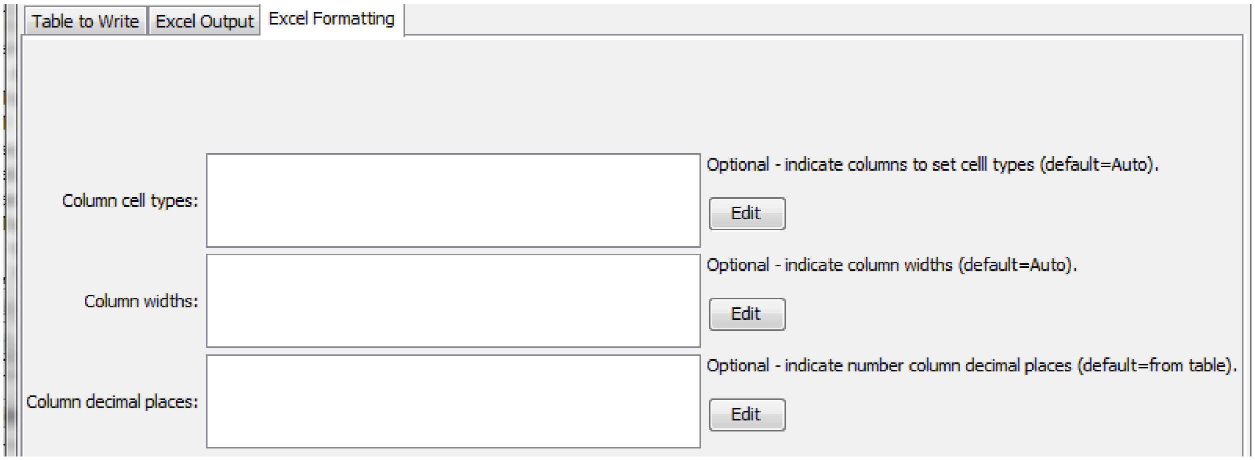
Excel address: A1 Excel cell block address in format A1 or A1:B2.

Excel column names: FirstRowInRange Optional - how to define Excel column names (default=None).
Column named ranges: Optional - indicate columns to set as named ranges (default=none).

Keep file open?: False Optional - keep Excel file open? (default=False).

WriteTableToExcel_Excel

WriteTableToExcel() Command Editor for Excel Output Parameters



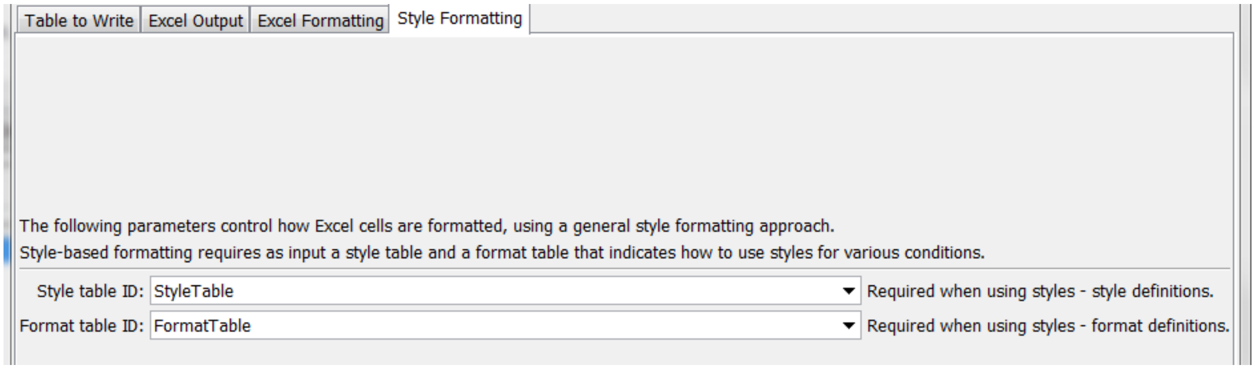
Column cell types: — Optional - indicate columns to set celll types (default=Auto).

Column widths: — Optional - indicate column widths (default=Auto).

Column decimal places: — Optional - indicate number column decimal places (default=from table).

WriteTableToExcel_Formatl

WriteTableToExcel() Command Editor for Excel Formatting Parameters



The following parameters control how Excel cells are formatted, using a general style formatting approach.
Style-based formatting requires as input a style table and a format table that indicates how to use styles for various conditions.

Style table ID: StyleTable — Required when using styles - style definitions.

Format table ID: FormatTable — Required when using styles - format definitions.

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.

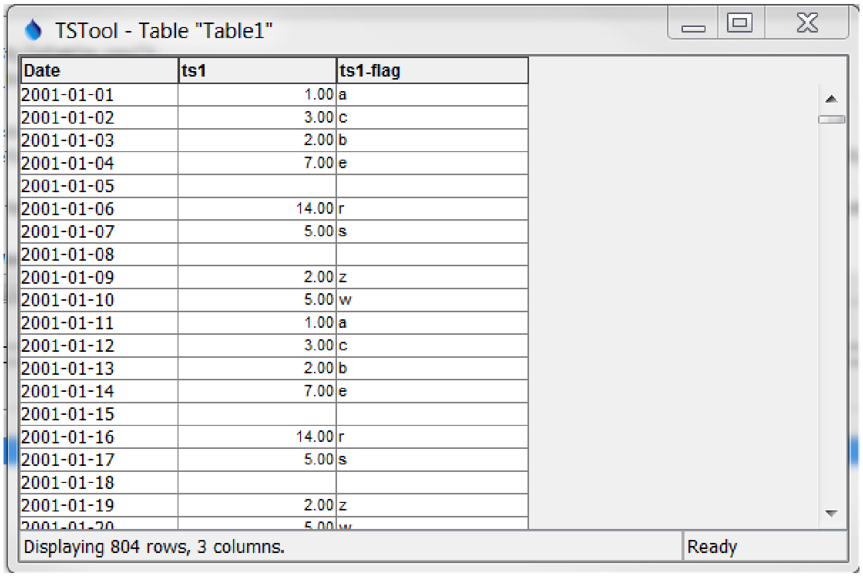
`ColumnIncludeFilters`	Indicate table column names and pattern to use to include rows. For example, include rows with blanks in columns. The format of the parameter is:
	`ColumnName1:Pattern1,`
	`ColumnName2:Pattern2,…`
	where patterns can contain `*` to match a substring.
		Include all rows.

`ColumnExcludeFilters`	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,`	Include all rows.

Parameter	Description	Default
	ColumnName2:Pattern2,… where patterns can contain * to match a substring.	
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	Default column with determined by Excel.

Parameter	Description	Default
	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)	
`ColumnDecimalPlaces`	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.



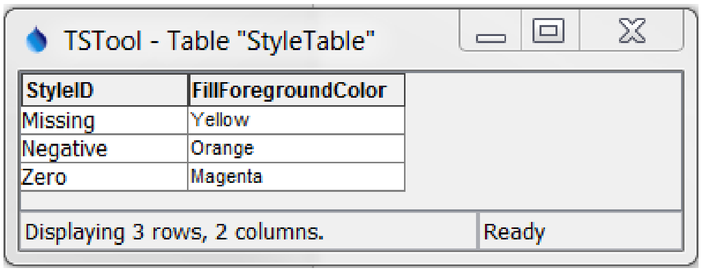
Date	ts1	ts1-flag
2001-01-01	1.00	a
2001-01-02	3.00	c
2001-01-03	2.00	b
2001-01-04	7.00	e
2001-01-05		
2001-01-06	14.00	r
2001-01-07	5.00	s
2001-01-08		
2001-01-09	2.00	z
2001-01-10	5.00	w
2001-01-11	1.00	a
2001-01-12	3.00	c
2001-01-13	2.00	b
2001-01-14	7.00	e
2001-01-15		
2001-01-16	14.00	r
2001-01-17	5.00	s
2001-01-18		
2001-01-19	2.00	z
2001-01-20	5.00	w

Displaying 804 rows, 3 columns. Ready

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.



TSTool - Table "StyleTable"

StyleID	FillForegroundColor
Missing	Yellow
Negative	Orange
Zero	Magenta

Displaying 3 rows, 2 columns. Ready

WriteTableToExcel_StyleTable

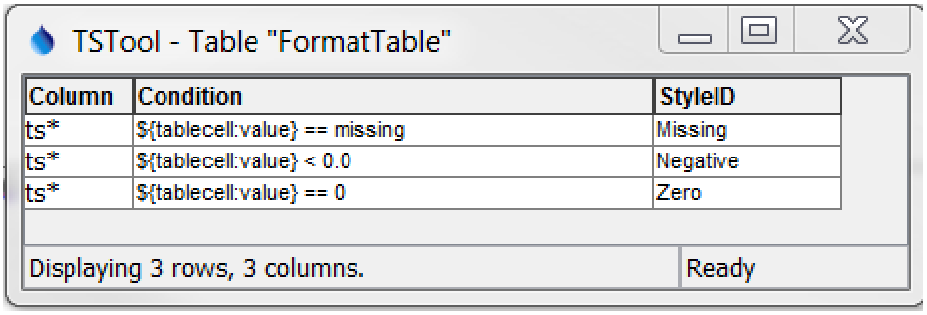
Style Table used with WriteTableToExcel() Command Style Formatting

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.



TSTool - Table "FormatTable"

Column	Condition	StyleID
ts*	${tablecell:value} == missing	Missing
ts*	${tablecell:value} < 0.0	Negative
ts*	${tablecell:value} == 0	Zero

Displaying 3 rows, 3 columns. Ready

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)

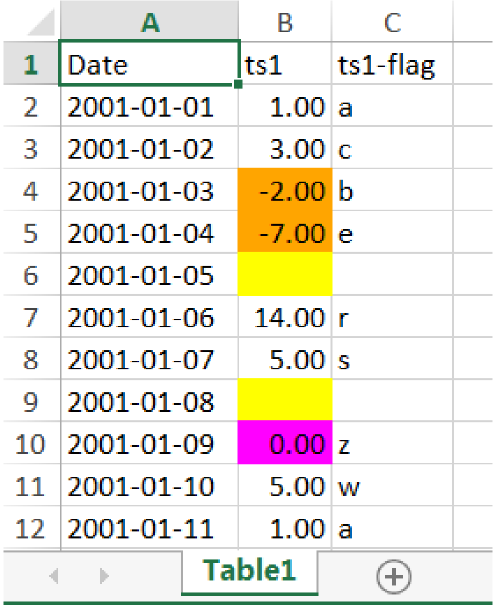
Values on the left and right of the operator must be separated with spaces to facilitate parsing the condition. The ***Condition*** column recognizes the following operators:

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.



	A	B	C
1	Date	ts1	ts1-flag
2	2001-01-01	1.00	a
3	2001-01-02	3.00	c
4	2001-01-03	-2.00	b
5	2001-01-04	-7.00	e
6	2001-01-05		
7	2001-01-06	14.00	r
8	2001-01-07	5.00	s
9	2001-01-08		
10	2001-01-09	0.00	z
11	2001-01-10	5.00	w
12	2001-01-11	1.00	a

WriteTableToExcel_Output

WriteTableToExcel() Command Example Output

This page is intentionally blank.