Command Reference: WriteTableToDelimitedFile()
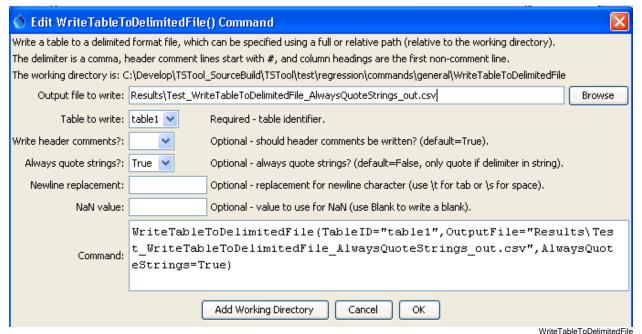
Write a table to a delimited file

Version 11.03.00, 2015-05-31

The `WriteTableToDelimitedFile()` command writes a table to a delimited file. Currently only the comma is supported as the delimiter. This command is the analog to the `ReadTableFromDelimitedFile()` command. It can be used to provide tabular data to other programs, such as spreadsheet programs and geographic information systems.

The default is to write a standard file header using comment lines that start with the # character. If available, column names will be written in double quotes as the first non-comment row. Formatting for cell values is limited and the default precision of floating point numbers may include too many digits – this will be addressed in future updates.

The following dialog is used to edit the command and illustrates the syntax for the command.

WriteTableToDelimitedFile() Command Editor

The command syntax is as follows:

```
WriteTableToDelimitedFile(Parameter=Value,…)
```

Command Parameters

Parameter	Description	Default
TableID	Identifier for the table to write. Can be specified using processor `${Property}`.	None – must be specified.
OutputFile	The name of the file to write, as an absolute path or relative to the command file location. Can be specified using processor `${Property}`.	None – must be specified.
WriteHeaderComments	Indicates whether to write the header comments, `True` or `False`. Some programs, such as Esri's ArcGIS do not handle delimited files with comments.	`True`
AlwaysQuoteStrings	Indicates whether values in string columns should always be surrounded by double quotes: • `False` – only quote strings that contain the delimiter • `True` – always quote strings An example of using `AlwaysQuoteStrings=True` is to quote identifiers that have a leading zero (e.g., `01234567`). Not quoting may cause the values to be interpreted as integers when read from the delimited file.	`False`
NewlineReplacement	The string to replace newlines in string values, necessary to prevent unexpected line breaks in output rows. In order to handle newlines from various systems, the following patterns are replaced in sequence: • `\r\n` • `\n` • `\r` The following special parameter values are recognized: • `\t` – replace newline with tab • `\s` – replace newline with space	Do not replace newlines.
NaNValue	The value to write for `NaN` data values. Specify `Blank` to write a blank (empty string).	`NaN`