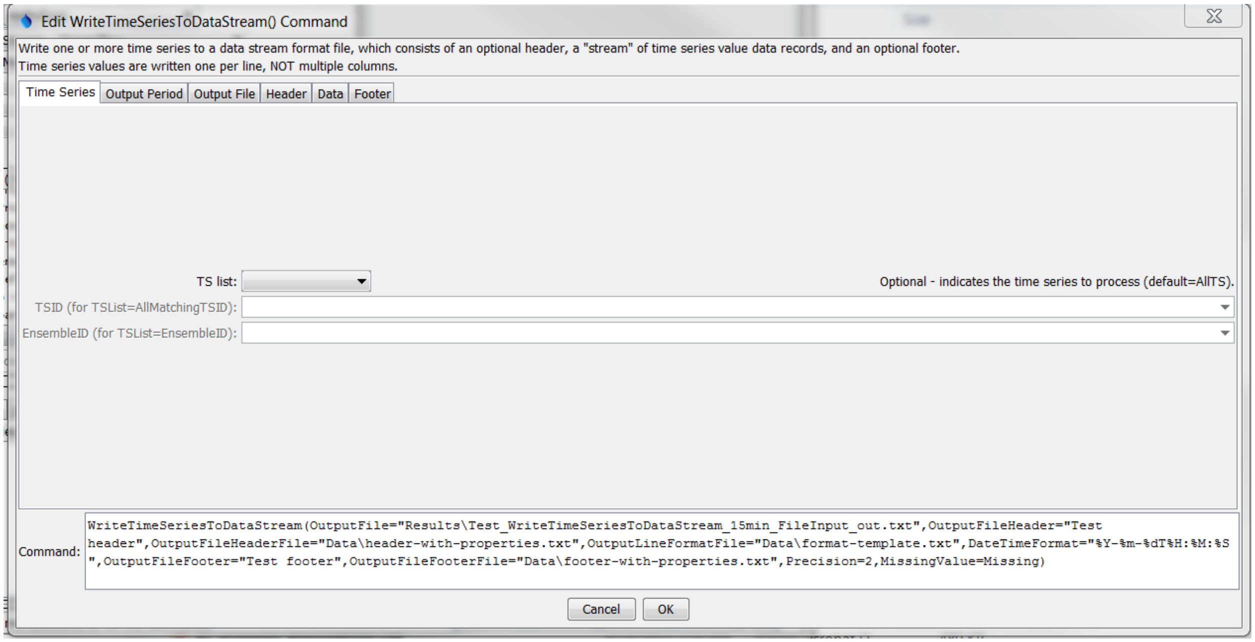
Command Reference: WriteTimeSeriesToDataStream()

Write time series to a file as a stream of data records

Version 11.07.03, 2015-10-23

The `WriteTimeSeriesToDataStream()` command writes time series to a file as a sequential "stream" of formatted data lines. This command is useful for processing data that are input to a data management system. Each time series is written as a single data value per line, not as columns for multiple time series.

The following dialog is used to edit the command and illustrates the syntax of the command, in this case writing time series to an XML file.



Edit WriteTimeSeriesToDataStream() Command

Write one or more time series to a data stream format file, which consists of an optional header, a "stream" of time series value data records, and an optional footer.
Time series values are written one per line, NOT multiple columns.

TS list: Optional - indicates the time series to process (default=AllTS).

TSID (for TSList=AllMatchingTSID):

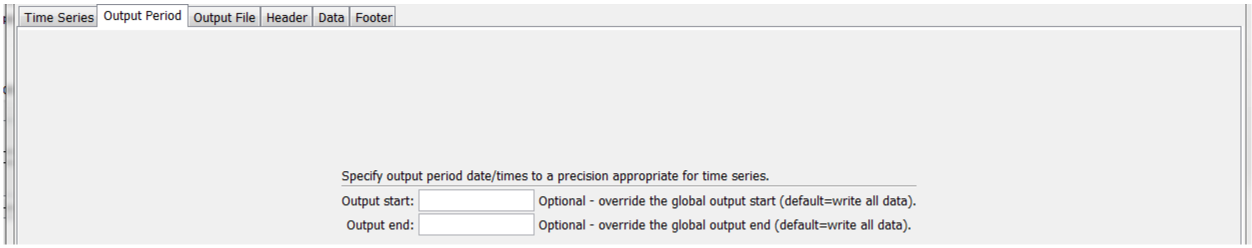
EnsembleID (for TSList=EnsembleID):

Command:

```
WriteTimeSeriesToDataStream(OutputFile="Results\Test_WriteTimeSeriesToDataStream_15min_FileInput_out.txt",OutputFileHeader="Test
header",OutputFileHeaderFile="Data\header-with-properties.txt",OutputLineFormatFile="Data\format-template.txt",DateTimeFormat="%Y-%m-%dT%H:%M:%S
",OutputFileFooter="Test footer",OutputFileFooterFile="Data\footer-with-properties.txt",Precision=2,MissingValue=Missing)
```

WriteTimeSeriesToDataStream

WriteTimeSeriesToDataStream() Command Editor for Time Series Parameters



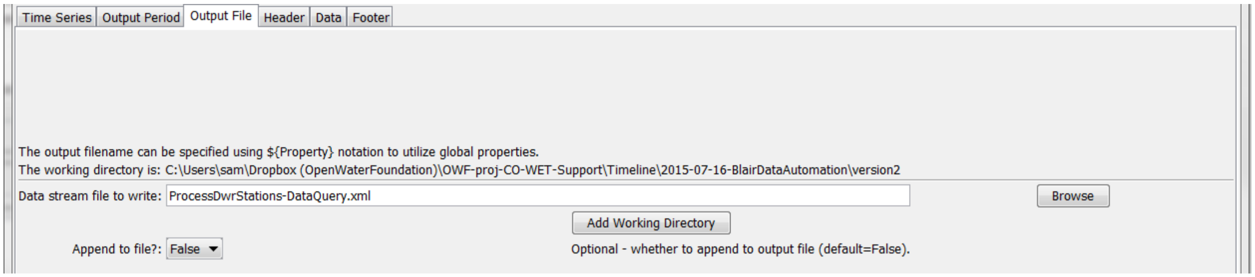
Specify output period date/times to a precision appropriate for time series.

Output start: Optional - override the global output start (default=write all data).

Output end: Optional - override the global output end (default=write all data).

WriteTimeSeriesToDataStream_Period

WriteTimeSeriesToDataStream() Command Editor for Output Period Parameters



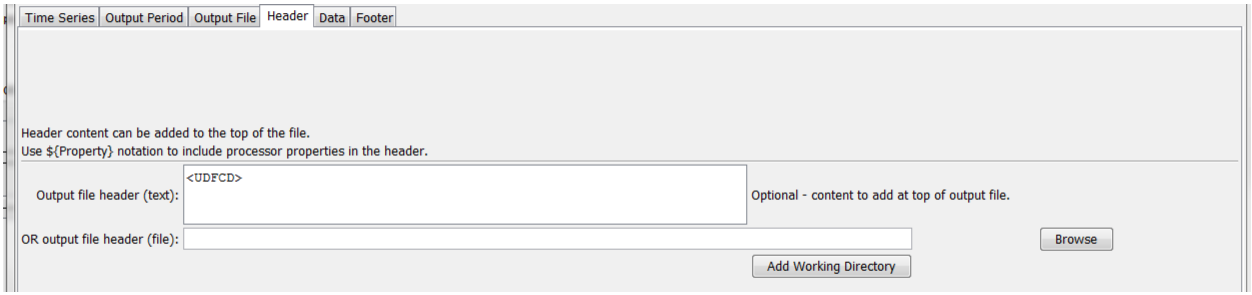
The output filename can be specified using ${Property} notation to utilize global properties.
The working directory is: C:\Users\sam\Dropbox (OpenWaterFoundation)\OWF-proj-CO-WET-Support\Timeline\2015-07-16-BlairDataAutomation\version2

Data stream file to write: ProcessDwrStations-DataQuery.xml

Append to file?: False ▾ Optional - whether to append to output file (default=False).

WriteTimeSeriesToDataStream_File

WriteTimeSeriesToDataStream() Command Editor for Output File Parameters



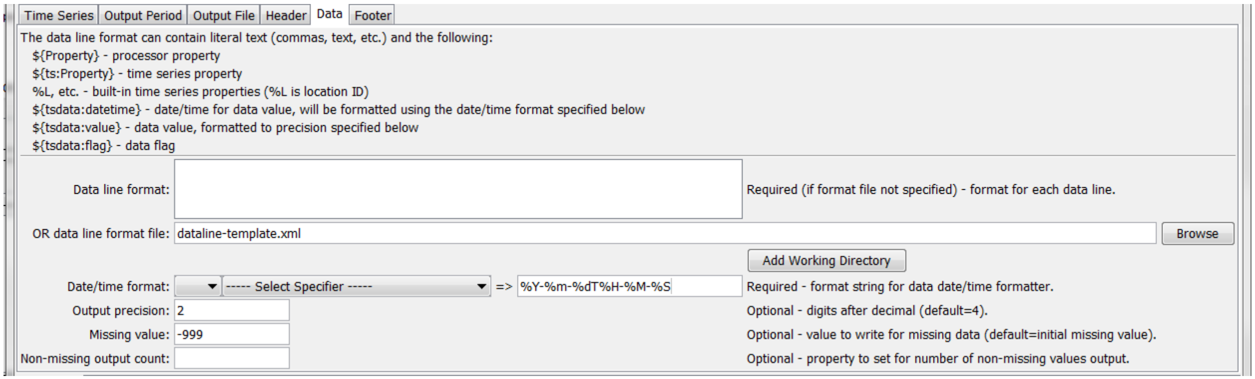
Header content can be added to the top of the file.
Use ${Property} notation to include processor properties in the header.

Output file header (text): <UDFCD> Optional - content to add at top of output file.

OR output file header (file):

WriteTimeSeriesToDataStream_Header

WriteTimeSeriesToDataStream() Command Editor for Output File Header Parameters



The data line format can contain literal text (commas, text, etc.) and the following:
${Property} - processor property
${ts:Property} - time series property
%L, etc. - built-in time series properties (%L is location ID)
${tsdata:datetime} - date/time for data value, will be formatted using the date/time format specified below
${tsdata:value} - data value, formatted to precision specified below
${tsdata:flag} - data flag

Data line format: Required (if format file not specified) - format for each data line.

OR data line format file: dataline-template.xml

Date/time format: => %Y-%m-%dT%H-%M-%S Required - format string for data date/time formatter.

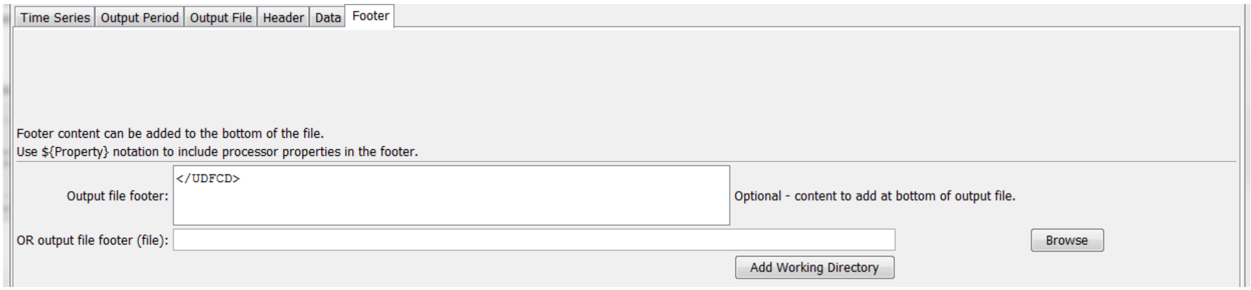
Output precision: 2 Optional - digits after decimal (default=4).

Missing value: -999 Optional - value to write for missing data (default=initial missing value).

Non-missing output count: Optional - property to set for number of non-missing values output.

WriteTimeSeriesToDataStream_Data

WriteTimeSeriesToDataStream() Command Editor for Data Parameters



Footer content can be added to the bottom of the file.
Use ${Property} notation to include processor properties in the footer.

Output file footer: </UDFCD> Optional - content to add at bottom of output file.

OR output file footer (file):

WriteTimeSeriesToDataStream_Footer

WriteTimeSeriesToDataStream() Command Editor for Footer Parameters

The above example uses a file to specify the format of data lines, as shown below (see the `OutputLineFormatFile` parameter):

```
<gage_wl id="${ts:NovaStarID}" last_rpt="${tsdata:datetime}"
${ts:NovaStarDataType}="${tsdata:value}" shef_id="${ts:SHEFID}"
```

```
name="${ts:NAME}" lat="${ts:LAT}" lon="${ts:LON}"
qtime="${RunStart}"/>
```

The command expands the output format line for each time series value being output, as follows:

- The notation `${Property}` is replaced with the corresponding TSTool global property. These properties can be set with `SetProperty()`, `FormatDateTimeProperty()`, `ReadPropertiesFromFile()`, and other commands.
- The notation `${ts:Property}` is replaced with the corresponding property from the time series that is being processed. Time series properties are set when a time series is read, depending on data that are available from a particular data source. The `SetTimeSeriesPropertiesFromTable()` command can also be used, for example in cases where time series metadata are available in a table.
- The notation `%L` (for location identifier), etc. also is replaced with standard time series properties. See read commands that have Alias parameters for a list of possible values. In the future corresponding standard `${ts:Property}` values may be defined.
- The notation `${tsdata:Property}` is replaced with the corresponding data from time series data records, in particular:
    - `${tsdata:datetime}` – corresponds to the date/time of the measurement (see the `DateTimeFormatterType` and `DateTimeFormat` command parameters)
    - `${tsdata:value}` – corresponds to the time series data value at a date/time (see the `Precision` and `MissingValue` command parameters)
    - `${tsdata:flag}` – corresponds to the time series flag at a date/time
- Any properties that are not found will result in output including the property name notation in the output.

The command syntax is as follows:

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

Command Parameters

Parameter	Description	Default
`TSList`	Indicates the list of time series to be processed, one of: • `AllMatchingTSID` – all time series that match the `TSID` (single TSID or TSID with wildcards) will be processed. • `AllTS` – all time series before the command. • `EnsembleID` – all time series in the ensemble will be processed. • `FirstMatchingTSID` – the first time series that matches the `TSID` (single TSID or TSID with wildcards) will be processed. • `LastMatchingTSID` – the last time series that matches the `TSID` (single TSID or TSID with wildcards) will be processed. • `SelectedTS` – the time series are those	`AllTS`

Parameter	Description	Default
	selected with the SelectTimeSeries() command.	
TSID	The time series identifier or alias for the time series to be processed, using the * wildcard character to match multiple time series.	Required if TSList=*TSID.
EnsembleID	The ensemble to be processed, if processing an ensemble.	Required if TSList= EnsembleID.
OutputFile	The output file. The path to the file can be absolute or relative to the working directory (command file location). Global properties can be used to specify the filename, using the ${Property} syntax.	None – must be specified.
Append	Indicate whether content should be appended to existing file. This may be appropriate but output files with header and footer text will need to be handled appropriately with the first and last additions to the output file.	False
OutputFile Header	Text that will be added to the top of the output file. Can contain ${Property}.	Do not include header content. See below.
OutputFile HeaderFile	Name of file containing text that will be added to the top of the output file. Can be specified with and contain ${Property}.	
OutputLine Format	Format that is used for each data value. Do not specify if OutputLineFormatFile is specified.	${tsdata:datetime} ${tsdata:value}
OutputLine FormatFile	The name of the file that contains the output line format specifiers. Do not specify if OutputLineFormat is specified. The output format file will be used as a template and expanded according to the explanation provided above this table. In particular, use a file for the format template if the template contains special characters that cannot be included in a parameter value.	
LastOutput LineFormat	Similar to OutputLineFormat, but if specified will be used only for the last data line.	
DateTime FormatterType	Specify the date/time formatter type, which indicates the syntax for DateTimeFormat. Currently, only C is supported, corresponding to the C programming language strftime() function, which is also used by other software (see Linux date command).	C
DateTime Format	The format used to expand the date/time corresponding to each time series data value. The format string can contain literal strings and specifiers supported by the DateTimeFormatterType.	
OutputFile Footer	Text that will be added to the bottom of the output file. Can contain ${Property}.	Do not include footer content. See below.
OutputFile FooterFile	Name of file containing text that will be added to the bottom of the output file. Can be specified with and contain ${Property}.	
Precision	The number of digits after the decimal for formatting	4 (in the future may

Parameter	Description	Default
	time series values.	default based on data type)
`MissingValue`	The value to write to the file to indicate a missing value in the time series. The value will be output literally and consequently string values are allowed. Specify the value as `Blank` to cause blanks to be used for missing values.	As initialized when reading the time series or creating a new time series, typically `-999`, `NaN`, or another value that is not expected in data.
`OutputStart`	The date/time for the start of the output.	Use the global output period.
`OutputEnd`	The date/time for the end of the output.	Use the global output period.
`NonMissing OutputCount`	The number of non-missing values to write. Specify as a negative number to write the values from the end (newest values). For example, -1 will write only the most recent non-missing value.	Write all the values.

The following example illustrates a simple JSON output file:

```
WriteTimeSeriesToDataStream(OutputFile="Results/Test_WriteTimeSeriesTo
DataStream_Irregular15min_jqPlot_out.txt",OutputFileHeader="var
timeseries[",OutputLineFormat="[\"${tsdata:datetime}\",${tsdata:value}
],",LastOutputLineFormat="[\"${tsdata:datetime}\",${tsdata:value}]",Da
teTimeFormat="%Y-%m-%d %H:%M:%S",OutputFileFooter="]")
```

```
var timeseries=[
["1950-01-01 00:00:00",5.0000],
["1950-01-01 00:15:00",10.0000],
["1950-01-01 00:30:00",12.0000],
["1950-01-01 00:45:00",13.0000],
…
["1950-01-05 11:45:00",10.0000],
["1950-01-05 12:00:00",12.0000]
]
```

This page is intentionally blank.