
Command Reference: TimeSeriesToTable()

Copy one or more time series into a table

Version 11.03.03, 2015-06-10

The `TimeSeriesToTable()` command copies one or more time series into a table. This command is useful when performing table analysis processing and outputting table formats (e.g., with the `WriteTableToDelimitedFile()` or `WriteTableToHTML()` commands). The command can be configured to output one of two table forms:

- Each time series in a separate column, with shared date/time column:
 - The time series must be regular interval (no irregular interval time series) and the intervals must match in order to allow alignment of the date/times.
 - Do not specify the `TableTSIDColumn` or `TableTSIDFormat` parameters.
- All time series values in a single column (useful for converting time series to a stream of data for loading into a database)
 - Any interval is allowed although mixing time series of varying precision is discouraged.
 - Specify the `TableTSIDColumn` and optionally `TableTSIDFormat` parameters.
 - See also the `WriteTimeSeriesToDataStream()` command.

Time series can be appended to an existing table. If time series are being appended in multi-column mode (one time series per column), the following checks are done to align the time series into the existing table:

1. If the output start and output end are specified, the specified period is used to write. If not specified, an overlapping period of the time series being written is determined. The first date/time in this period is matched with an existing date/time in the date/time column. If the time series being written do not overlap, new records are added to fill out the time sequence. Once the starting row is established, the time series are written as a “block” within the existing table.
2. The date/time values must be sequential. For example, using a `SortTable()` command prior to this command that results in date/time column values being reordered is not supported because it would be a major performance penalty to search the date/time column when setting each time series data value. A warning will be generated if the date/time column is determined to be out of order.
3. As an additional check, when adding time series to an existing table if the date/time column value does not equal the time series date/time, a warning will be generated.

The following dialog is used to edit the command and illustrates the syntax of the command when writing a multi-column data table while also outputting data flags. Note that the value columns can be specified using time series properties.

Copy time series date/time and value pairs to column(s) in a new table. If the table TSID column is specified, output will be to a single column.
The time series must have the same data interval if a multi-column table is created.
If the output window is specified, use a date/time precision consistent with data.

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

TSID (for TSList=AllMatchingTSID):

EnsembleID (for TSList=EnsembleID):

Table ID: Required - table identifier.

Date/time column in table: Required - column name for date/times.

Single-column output parameters

Table TSID column: Optional - column name for TSID (if values in single column).

Format of TSID: => Optional - can use if TableTSIDColumn is specified (default=alias or TSID).

Include missing values?: Optional - include missing values (default=True).

Data value column(s) in table: => Required - value column name(s) for 1+ time series.

Output precision: Optional - precision for value column(s) in table (default=2).

Flag column(s) in table: => Optional - flag column name(s) for 1+ time series.

Output start date/time: Optional (default=copy all).

Output end date/time: Optional (default=copy all).

☐ Output window: Start: Month: Day: Hour: Minute: End: Month: Day: Hour: Minute: Optional - output window within each year (default=full year).

Output window start \$(Property): Optional (default=full year).

Output window end \$(Property): Optional (default=full year).

Action if table not found: Optional - action if table not found (default=Warn).

Command:

```
TimeSeriesToTable(TableID="TestTable",DateTimeColumn="Year",ValueColumn="%L-%T",IfTableNotFound="Create")
```

Cancel OK

TimeSeriesToTable (TimeSeriesToTable_Year_Create)

TimeSeriesToTable() Command Editor to Create Multi-Column Data Table

The command syntax is as follows:

`TimeSeriesToTable (Parameter=Value, ...)`

Command Parameters

| Parameter | Description | Default |
|-----------|---|---------|
| TSList | Indicates the list of time series to be processed, one of: <ul style="list-style-type: none"> AllMatchingTSID – all time series that match the TSID (single TSID or TSID with wildcards). AllTS – all time series before the command. EnsembleID – all time series in the ensemble. FirstMatchingTSID – the first time series that matches the TSID (single TSID or TSID with wildcards). | AllTS |

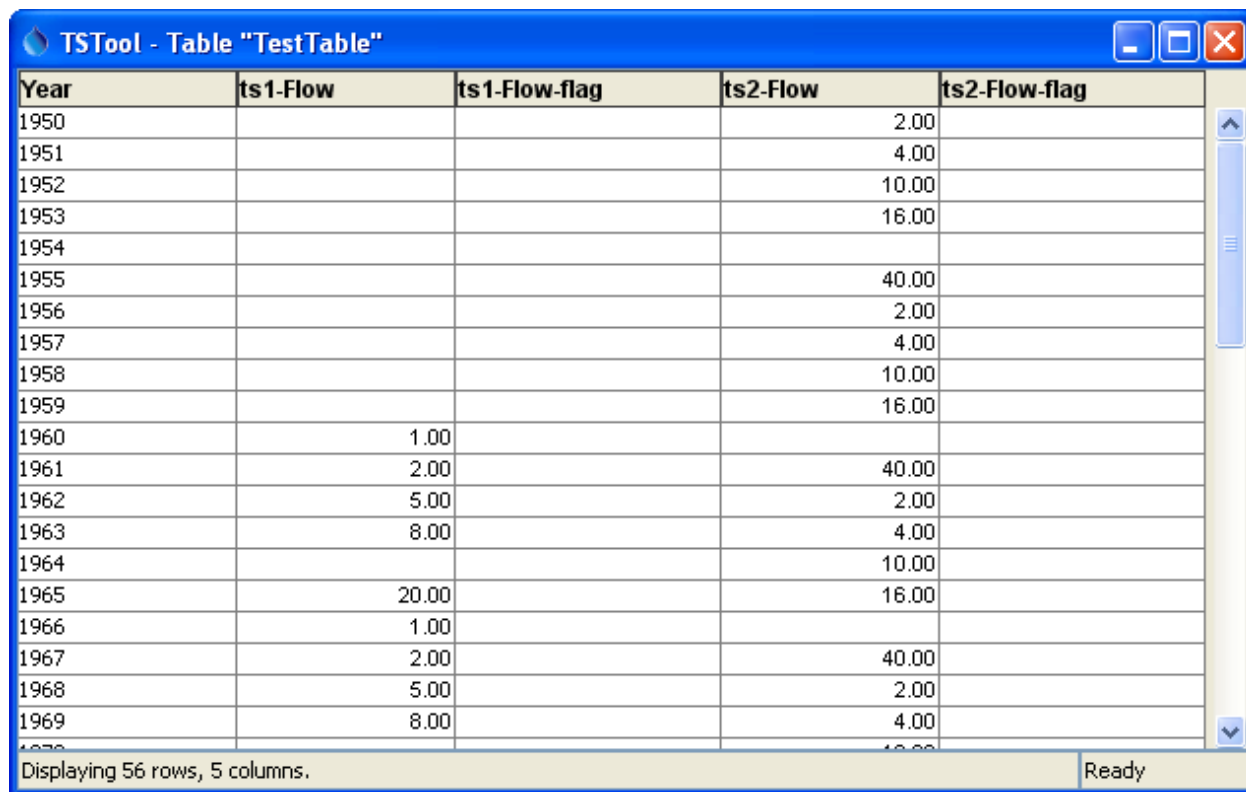
| Parameter | Description | Default |
|----------------------|---|--|
| | <ul style="list-style-type: none"> LastMatchingTSID – the last time series that matches the TSID (single TSID or TSID with wildcards). SelectedTS – the time series are those selected with the SelectTimeSeries () command. | |
| TSID | The time series identifier or alias for the time series to be modified, using the * wildcard character to match multiple time series. Can be specified using processor \${Property}. | Required when TSList=*TSID |
| EnsembleID | The ensemble to be modified, if processing an ensemble. Can be specified using processor \${Property}. | Required when TSList=EnsembleID. |
| TableID | The identifier for the table to copy data into (or the identifier for the new table to create if IfTableNotFound=Create). Can be specified using processor \${Property}. | None – must be specified. |
| DateTimeColumn | The table column name to receive date/time information. | None – must be specified. |
| TableTSIDColumn | For single-column output, the name of the column in the table for time series identifier information. The format of the identifier can be specified using the TableTSIDFormat parameter. | Optional – if specified will indicate single-column output. |
| TableTSIDFormat | For single-column output, indicates how to format the time series identifier that is inserted in the column specified by the TableTSIDColumn parameter. | Optional – if not specified the alias or full TSID will be used. |
| IncludeMissingValues | For single-column output, indicates whether missing values should be transferred to the table. This is useful to screen out missing values from sparse time series. | True |
| ValueColumn | <p>The data value column name(s) to receive time series data, specified as follows:</p> <ul style="list-style-type: none"> Multiple names separated by a comma. Time series property format specifiers, available in a list of choices. These specifiers are consistent with other commands and the legend formatter in the graphing tool. If a literal string is specified with multi-column output, names for columns 2+ will be generated by adding a sequential number to ValueColumn. | None – must be specified. |
| OutputPrecision | The number of digits after the decimal to use for data values in the output table. | 2 (in the future may auto-detect from units) |
| FlagColumn | The data flag column name(s) to receive time series flags, specified using the same syntax as | Do not output flags to the table. |

| Parameter | Description | Default |
|-------------------|--|-----------------------|
| | ValueColumn. A blank in the list will result in no transfer of flags for the specific time series. | |
| OutputStart | The starting date/time for the copy. Can be specified using processor <code>\${Property}</code> . | Available period. |
| OutputEnd | The ending date/time for the copy. Can be specified using processor <code>\${Property}</code> . | Available period. |
| OutputWindowStart | The calendar date/time for the output start within each year. Specify using the format MM, MM-DD, MM-DD hh, or MM-DD hh:mm, consistent with the time series interval precision. A year of 2000 will be used internally to parse the date/time. Use this parameter to limit data processing within the year, for example to output only a single month or a season. Can be specified using processor <code>\${Property}</code> by using the text field below the date editor. | Output the full year. |
| OutputWindowEnd | Specify date/time for the output end within each year. See OutputWindowStart for details. Can be specified using processor <code>\${Property}</code> . | Output the full year. |
| IfTableNotFound | Indicate action if the table identifier is not matched, one of: <ul style="list-style-type: none"> Create – create a new table Warn – warn that the table was not matched | Warn |

A sample command file is as follows (this command file is used to verify the command during testing):

```
# Test copying annual time series to a table, and also create the table
StartLog(LogFile="Results/Test_TimeSeriesToTable_Year_Create.TSTool.log")
RemoveFile(InputFile="Results/Test_TimeSeriesToTable_Year_Create_out.csv",
  IfNotFound=Ignore)
NewPatternTimeSeries(Alias="ts1",NewTSID="ts1..Flow.Year",SetStart="1960",
  SetEnd="2000",Units="ACFT",PatternValues="1,2,5,8,,20")
NewPatternTimeSeries(Alias="ts2",NewTSID="ts2..Flow.Year",SetStart="1950",
  SetEnd="2005",Units="ACFT",PatternValues="2,4,10,16,,40")
TimeSeriesToTable(TableID=TestTable,DateTimeColumn=Year,ValueColumn=%L-%T,
  FlagColumn="%L-%T-flag",IfTableNotFound="Create")
# Generate the results.
WriteTableToDelimitedFile(TableID="TestTable",
  OutputFile="Results/Test_TimeSeriesToTable_Year_Create_out.csv")
# Uncomment the following to recreate expected results
# WriteTableToDelimitedFile(TableID="TestTable",
#   OutputFile="ExpectedResults/Test_TimeSeriesToTable_Year_Create_out.csv")
CompareFiles(InputFile1=
  "ExpectedResults/Test_TimeSeriesToTable_Year_Create_out.csv",
  InputFile2="Results/Test_TimeSeriesToTable_Year_Create_out.csv",
  IfDifferent=Warn)
```

The resulting table will be listed in the **Tables** area of the TSTool interface and clicking on the `TestTable` identifier will display the table similar to the following:



| Year | ts1-Flow | ts1-Flow-flag | ts2-Flow | ts2-Flow-flag |
|------|----------|---------------|----------|---------------|
| 1950 | | | 2.00 | |
| 1951 | | | 4.00 | |
| 1952 | | | 10.00 | |
| 1953 | | | 16.00 | |
| 1954 | | | | |
| 1955 | | | 40.00 | |
| 1956 | | | 2.00 | |
| 1957 | | | 4.00 | |
| 1958 | | | 10.00 | |
| 1959 | | | 16.00 | |
| 1960 | 1.00 | | | |
| 1961 | 2.00 | | 40.00 | |
| 1962 | 5.00 | | 2.00 | |
| 1963 | 8.00 | | 4.00 | |
| 1964 | | | 10.00 | |
| 1965 | 20.00 | | 16.00 | |
| 1966 | 1.00 | | | |
| 1967 | 2.00 | | 40.00 | |
| 1968 | 5.00 | | 2.00 | |
| 1969 | 8.00 | | 4.00 | |

Displaying 56 rows, 5 columns. Ready

TimeSeriesToTable2

Multi-Column Data Table

The following example illustrates how to create a single data column table. Because a single column is being used for data, the data value and corresponding data flag column names are specified literally (not as time series properties). The column and format for the TSID also must be specified.

Edit TimeSeriesToTable() Command

Copy time series date/time and value pairs to column(s) in a new table. If the table TSID column is specified, output will be to a single column.
The time series must have the same data interval if a multi-column table is created.
If the output window is specified, use a date/time precision consistent with data.

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

TSID (for TSList=AllMatchingTSID):

EnsembleID (for TSList=EnsembleID):

Table ID: Required - table identifier.

Date/time column in table: Required - column name for date/times.

Single-column output parameters

Table TSID column: Optional - column name for TSID (if values in single column).

Format of TSID: => Optional - can use if TableTSIDColumn is specified (default=alias or TSID).

Include missing values?: Optional - include missing values (default=True).

Data value column(s) in table: => Required - value column name(s) for 1+ time series.

Output precision: Optional - precision for value column(s) in table (default=2).

Flag column(s) in table: => Optional - flag column name(s) for 1+ time series.

Output start date/time:

Output end date/time:

☒ Output window: Start: Month: Day: Hour: Minute: End: Month: Day: Hour: Minute: Optional - output window within each year (default=full year).

Output window start \${Property}: Optional (default=full year).

Output window end \${Property}: Optional (default=full year).

Action if table not found: Optional - action if table not found (default=Warn).

Command:

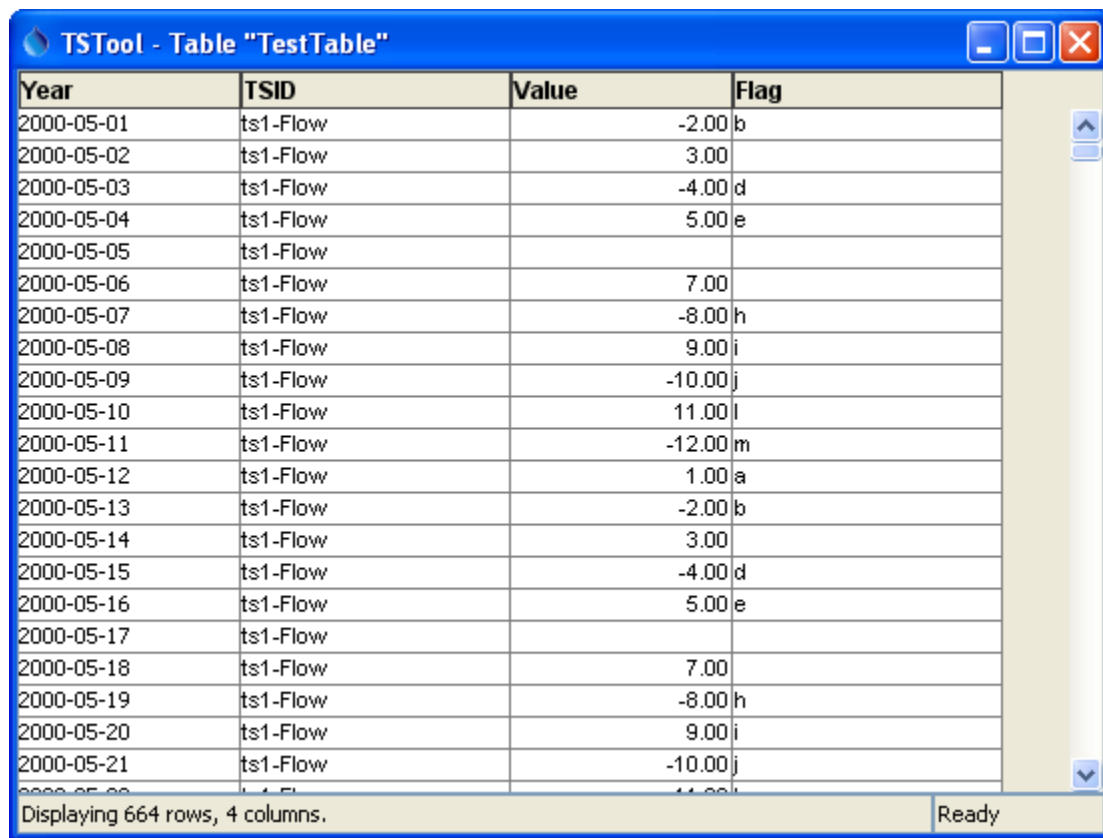
```
TimeSeriesToTable (TableID="TestTable", DateTimeColumn="Year", TableTSIDColumn="TSID", TableTSIDFormat="%L-%%", ValueColumn="Value", FlagColumn="Flag", OutputWindowStart="05-01", OutputWindowEnd="07-31", IfTableNotFound="Create")
```

Cancel OK

TimeSeriesToTable_Single (Day_MayJul_Single_Flag)

TimeSeriesToTable() Command Editor to Create Single Data Column Table

The resulting table is as shown in the following figure:



| Year | TSID | Value | Flag |
|------------|----------|--------|------|
| 2000-05-01 | ts1-Flow | -2.00 | b |
| 2000-05-02 | ts1-Flow | 3.00 | |
| 2000-05-03 | ts1-Flow | -4.00 | d |
| 2000-05-04 | ts1-Flow | 5.00 | e |
| 2000-05-05 | ts1-Flow | | |
| 2000-05-06 | ts1-Flow | 7.00 | |
| 2000-05-07 | ts1-Flow | -8.00 | h |
| 2000-05-08 | ts1-Flow | 9.00 | i |
| 2000-05-09 | ts1-Flow | -10.00 | j |
| 2000-05-10 | ts1-Flow | 11.00 | l |
| 2000-05-11 | ts1-Flow | -12.00 | m |
| 2000-05-12 | ts1-Flow | 1.00 | a |
| 2000-05-13 | ts1-Flow | -2.00 | b |
| 2000-05-14 | ts1-Flow | 3.00 | |
| 2000-05-15 | ts1-Flow | -4.00 | d |
| 2000-05-16 | ts1-Flow | 5.00 | e |
| 2000-05-17 | ts1-Flow | | |
| 2000-05-18 | ts1-Flow | 7.00 | |
| 2000-05-19 | ts1-Flow | -8.00 | h |
| 2000-05-20 | ts1-Flow | 9.00 | i |
| 2000-05-21 | ts1-Flow | -10.00 | j |
| 2000-05-22 | ts1-Flow | 11.00 | l |

Displaying 664 rows, 4 columns. Ready

TimeSeriesToTable_Single2

Single Data Column Table

This page is intentionally blank.