Command Reference: TimeSeriesToTable()

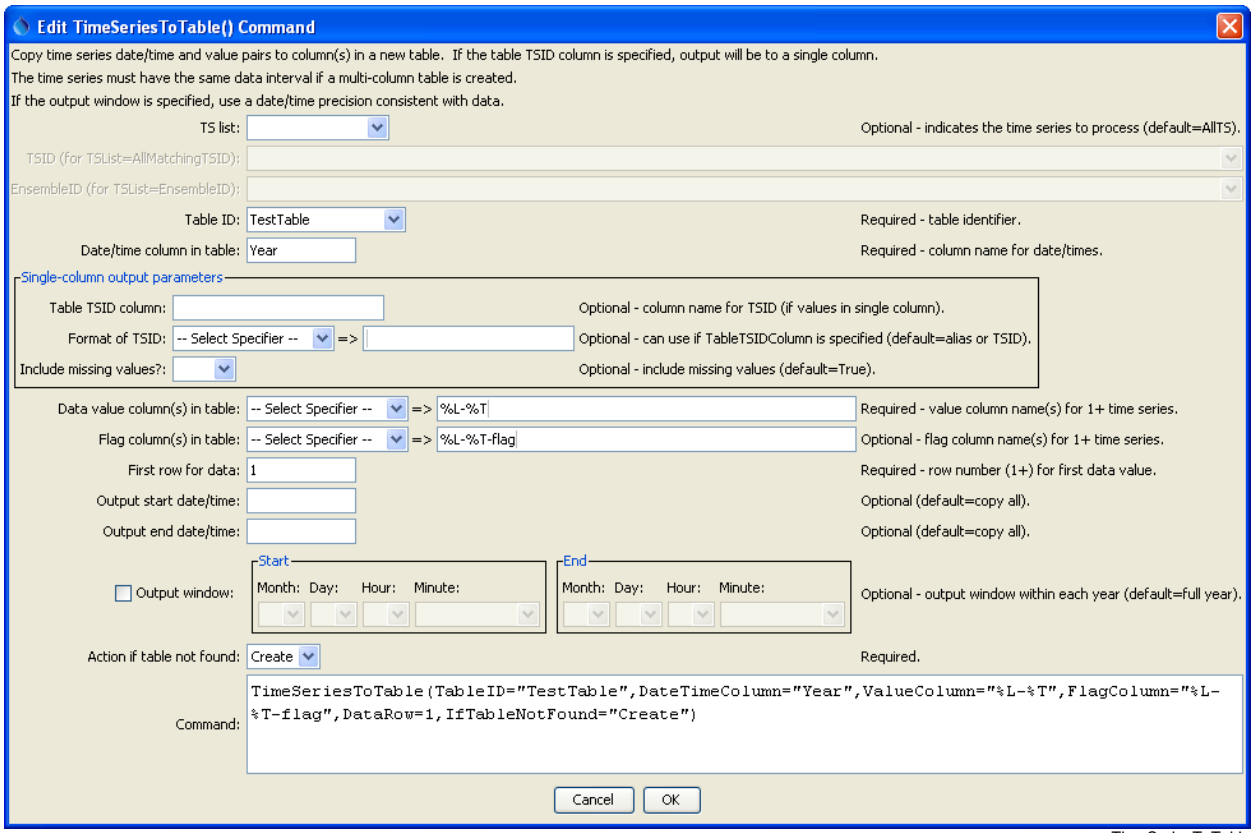
Copy one or more time series into a table

Version 10.21.00, 2013-06-26

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.

Currently the command can only be used to create a new table but in the future the command is envisioned to write into an existing table. 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.



TimeSeriesToTable

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: • 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). • 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.	AllTS
TSID	The time series identifier or alias for the time series to be modified, using the * wildcard character to match multiple time series.	Required when TSList=*TSID
EnsembleID	The ensemble to be modified, if processing an ensemble.	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).	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	The data value column name(s) to receive time	None – must be specified.

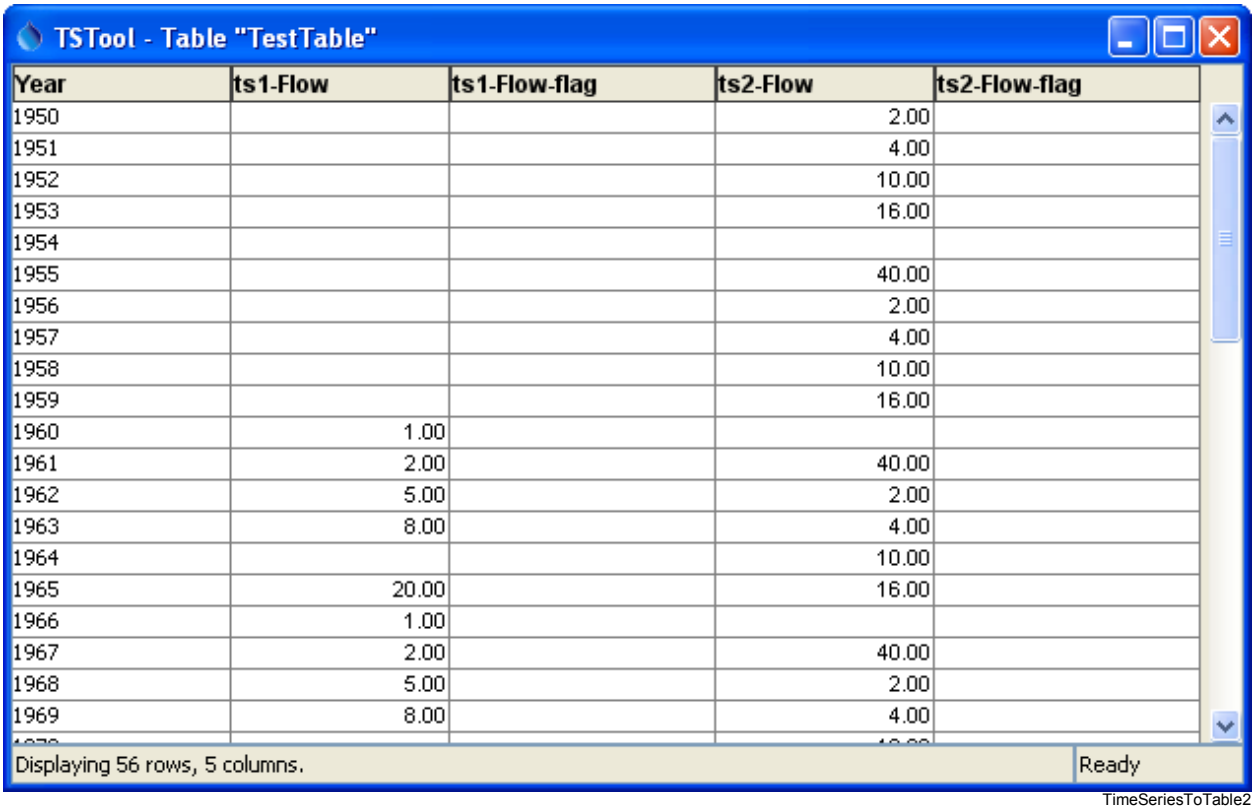
Parameter	Description	Default
	series data, specified as follows: • Multiple names separated by a comma. • Time series property format specifiers, available in a list of choices. These specifiers are consistent with other commands and the legend formatter in the graphing tool. • If a literal string is specified with multi-column output, names for columns 2+ will be generated by adding a sequential number to `ValueColumn`.	
`FlagColumn`	The data flag column name(s) to receive time series flags, specified using the same syntax as `ValueColumn`. A blank in the list will result in no transfer of flags for the specific time series.	Do not output flags to the table.
`DataRow1`	First table row for data (1+), where the row number is data only (column names are not considered a data row).	None – must be specified.
`OutputStart`	The starting date/time for the copy.	Available period.
`OutputEnd`	The ending date/time for the copy.	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.	Output the full year.
`OutputWindowEnd`	Specify date/time for the output end within each year. See `OutputWindowStart` for details.	Output the full year.
`IfTableNotFound`	Indicate action if the table identifier is not matched, one of: • `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",DataRow=1,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)
```

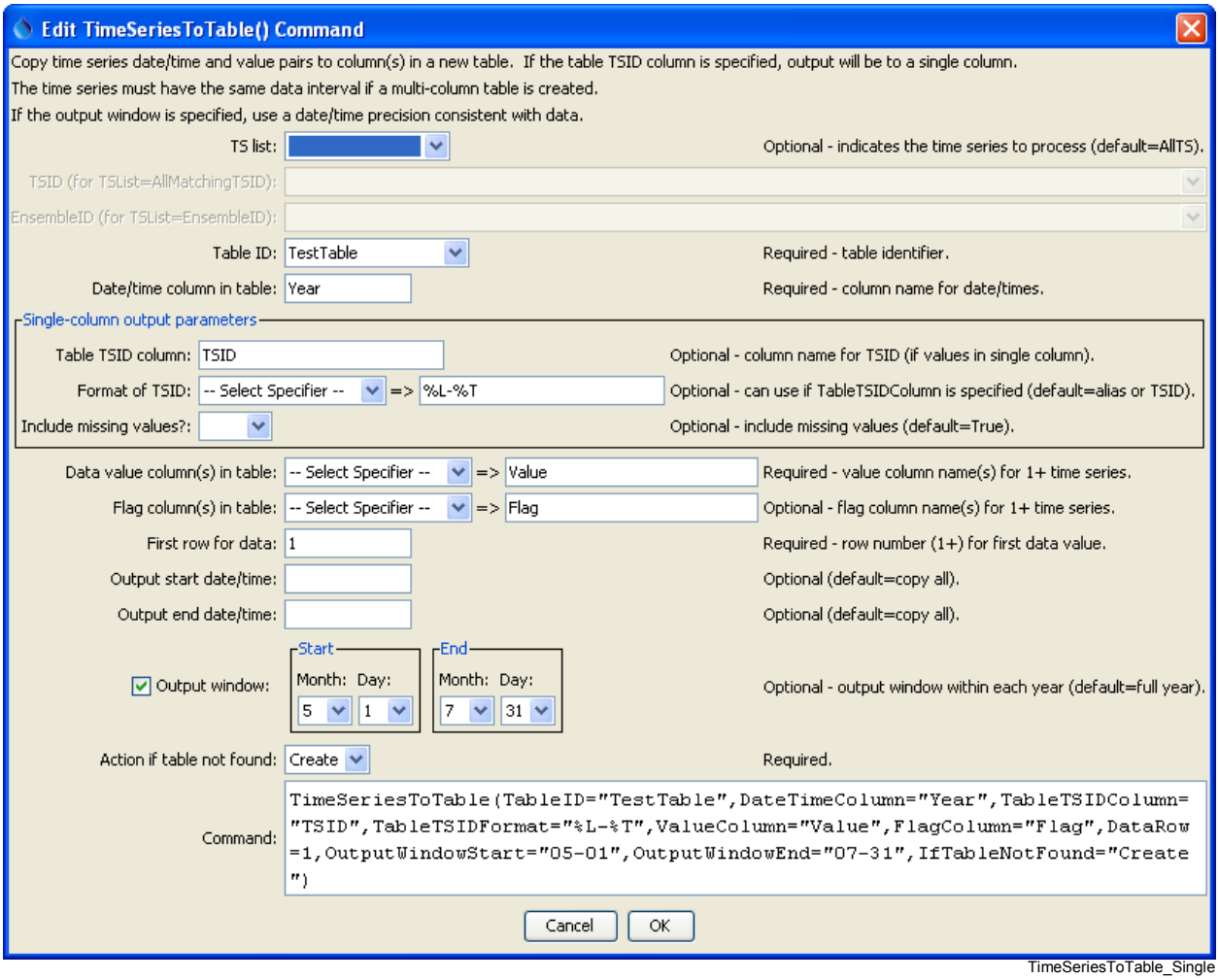
The resulting table will be listed in the ***Tables*** area of the TSTool interface and clicking on the `TestTable` identifier will display the table similar to the following:



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.



TimeSeriesToTable_Single

TimeSeriesToTable() Command Editor to Create Single Data Column Table

The resulting table is as shown in the following figure:

TSTool - Table "TestTable"

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

Displaying 664 rows, 4 columns. Ready

TimeSeriesToTable_Single2

Single Data Column Table