Command Reference: ReadTimeSeriesList()

Read one or more time series using location identifiers from a table

Version 11.03.02, 2015-06-08

The `ReadTimeSeriesList()` command reads one or more time series using location identifiers from a table, an example of which is shown below as a comma-separated value file (tables can also be read from Excel, delimited file, datastore, or other source):

```
# Example list file.  Comments start with the # character.
# Column headings can be specified in the first non-comment row using quotes.
“Structure ID”,”Structure Name”
500501,Ditch 501
500502,Ditch 502
# Invalid ID (see IfNotFound parameter)
509999,Ditch 9999
```

The command typically is used when reading time series from a single source and can streamline processing in the following situations:

- A list of identifiers may have been generated from a database query
- A list of identifiers may have been extracted from a model data set

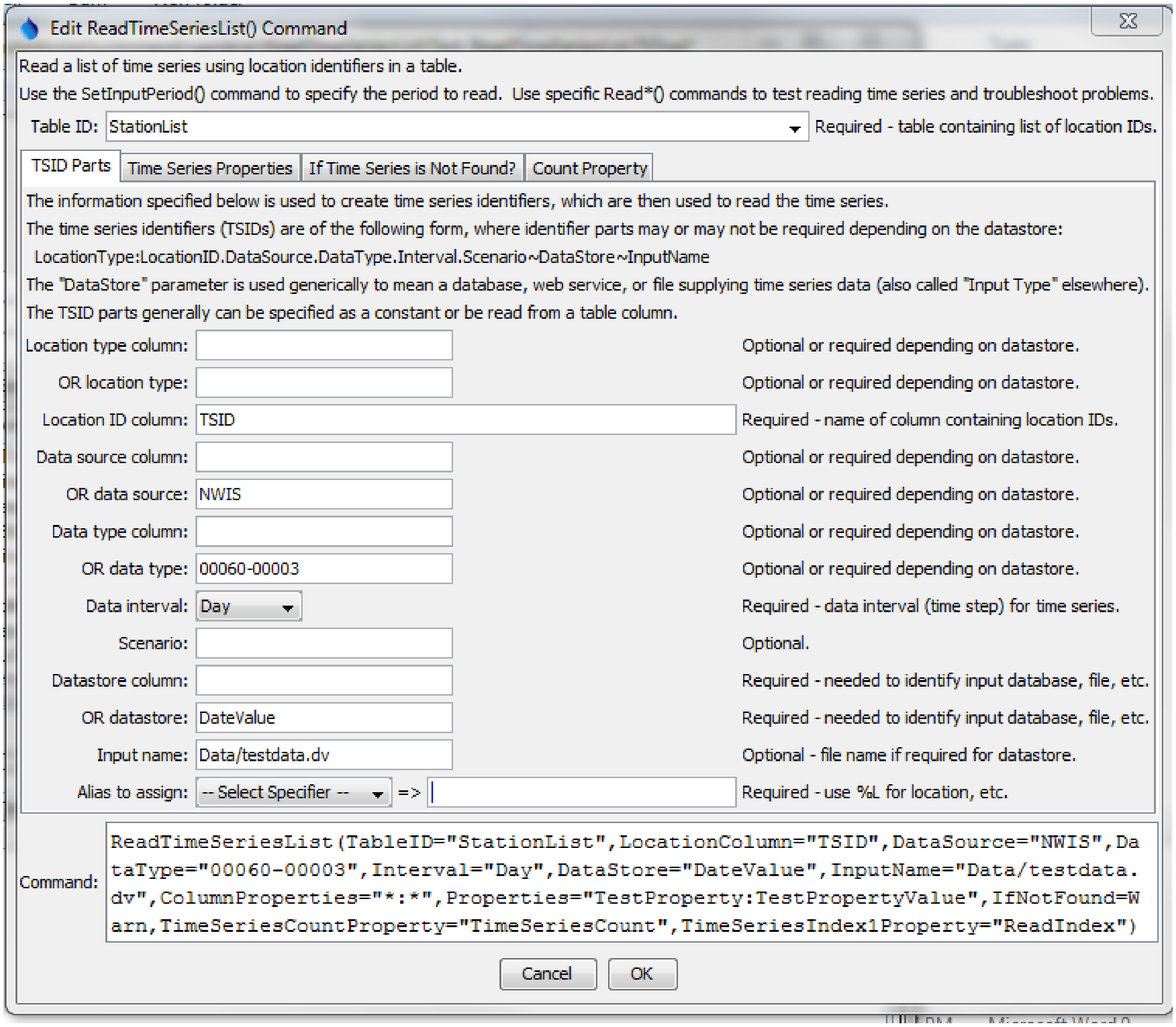
TSTool uses the location identifiers in the table with the command parameters and internally creates a list of time series identifiers. The time series are of the standard form (information in brackets is optional for basic use):

```
[LocationType:]Location.DataSource.DataType.Interval[.Scenario]~DataStore[~InputName]
```

TSTool then queries each time series using the time series identifier. See also the `ReadTimeSeries()` command, which performs essentially the same functionality but only reads one time series. Refer to the appendices for each datastore and file input type to understand specific time series identifier conventions.

Although it is possible to specify a datastore or input type that reads from files by also using the `InputName`, this may not be appropriate because the `ReadTimeSeriesList()` command can only specify one input file name and the file will be reopened for each time series read. Instead, read commands for specific file formats should be used because these commands typically are optimized to read multiple time series from the files. Use the `SetInputPeriod()` command to set the period to read.

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



Edit ReadTimeSeriesList() Command

Read a list of time series using location identifiers in a table.

Use the SetInputPeriod() command to specify the period to read. Use specific Read*() commands to test reading time series and troubleshoot problems.

Table ID: StationList — Required - table containing list of location IDs.

The information specified below is used to create time series identifiers, which are then used to read the time series.

The time series identifiers (TSIDs) are of the following form, where identifier parts may or may not be required depending on the datastore:

LocationType:LocationID.DataSource.DataType.Interval.Scenario~DataStore~InputName

The "DataStore" parameter is used generically to mean a database, web service, or file supplying time series data (also called "Input Type" elsewhere).

The TSID parts generally can be specified as a constant or be read from a table column.

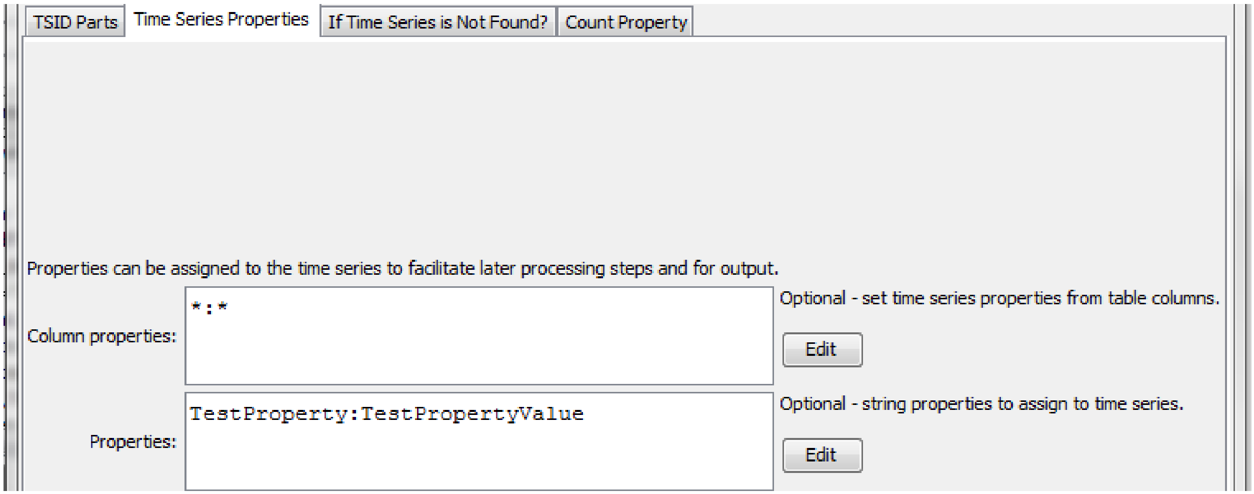
Location type column:		Optional or required depending on datastore.
OR location type:		Optional or required depending on datastore.
Location ID column:	TSID	Required - name of column containing location IDs.
Data source column:		Optional or required depending on datastore.
OR data source:	NWIS	Optional or required depending on datastore.
Data type column:		Optional or required depending on datastore.
OR data type:	00060-00003	Optional or required depending on datastore.
Data interval:	Day	Required - data interval (time step) for time series.
Scenario:		Optional.
Datastore column:		Required - needed to identify input database, file, etc.
OR datastore:	DateValue	Required - needed to identify input database, file, etc.
Input name:	Data/testdata.dv	Optional - file name if required for datastore.
Alias to assign:	-- Select Specifier -- =>	Required - use %L for location, etc.

Command:

```
ReadTimeSeriesList(TableID="StationList",LocationColumn="TSID",DataSource="NWIS",Da
taType="00060-00003",Interval="Day",DataStore="DateValue",InputName="Data/testdata.
dv",ColumnProperties="*:*",Properties="TestProperty:TestPropertyValue",IfNotFound=W
arn,TimeSeriesCountProperty="TimeSeriesCount",TimeSeriesIndex1Property="ReadIndex")
```

ReadTimeSeriesList

ReadTimeSeriesList() Command Editor for Main TSID Parameters



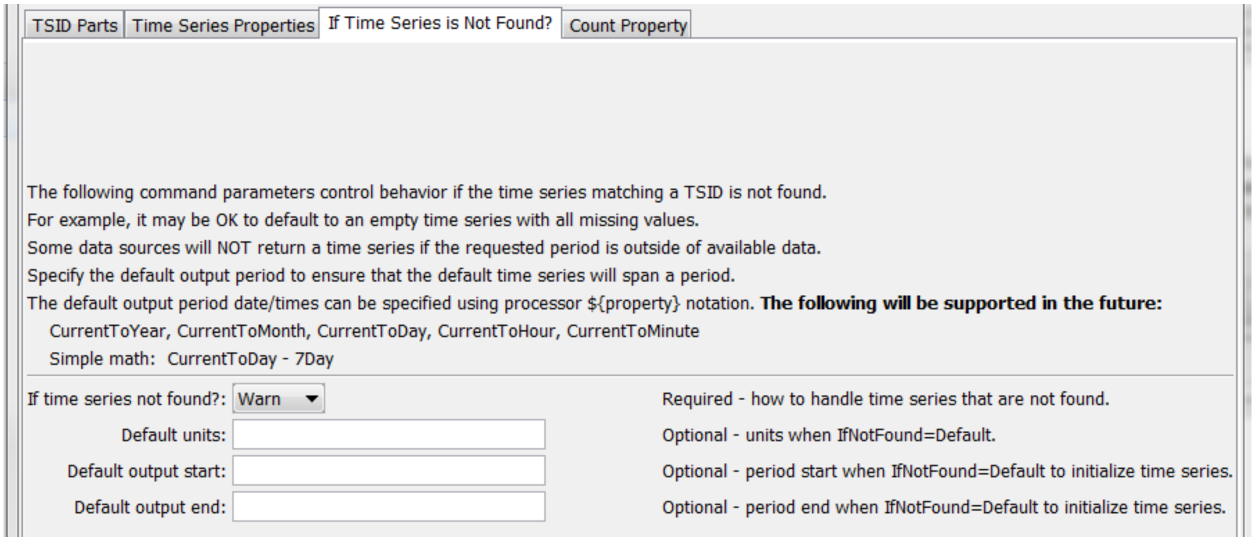
Properties can be assigned to the time series to facilitate later processing steps and for output.

Column properties: `*:*` Optional - set time series properties from table columns.

Properties: `TestProperty:TestPropertyValue` Optional - string properties to assign to time series.

ReadTimeSeriesList_TimeSeriesProperties

ReadTimeSeriesList() Command Editor Time Series Properties Parameters

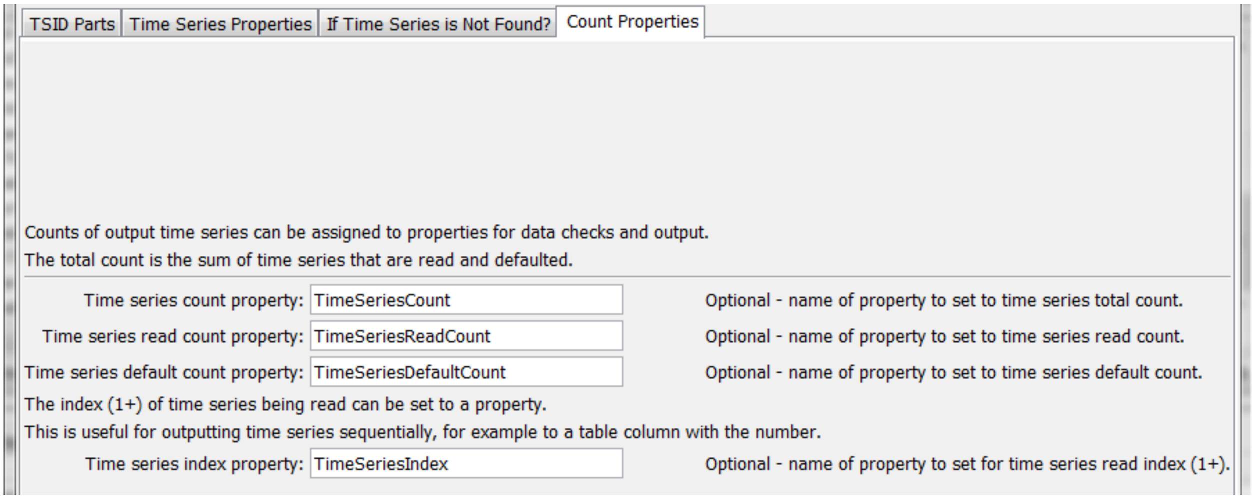


The following command parameters control behavior if the time series matching a TSID is not found.
For example, it may be OK to default to an empty time series with all missing values.
Some data sources will NOT return a time series if the requested period is outside of available data.
Specify the default output period to ensure that the default time series will span a period.
The default output period date/times can be specified using processor ${property} notation. **The following will be supported in the future:**
CurrentToYear, CurrentToMonth, CurrentToDay, CurrentToHour, CurrentToMinute
Simple math: CurrentToDay - 7Day

If time series not found?:	Warn	Required - how to handle time series that are not found.
Default units:		Optional - units when IfNotFound=Default.
Default output start:		Optional - period start when IfNotFound=Default to initialize time series.
Default output end:		Optional - period end when IfNotFound=Default to initialize time series.

ReadTimeSeriesList_IfNotFound

ReadTimeSeriesList() Command Editor for Parameters if Time Series is not Found



Counts of output time series can be assigned to properties for data checks and output.
The total count is the sum of time series that are read and defaulted.

Time series count property:	TimeSeriesCount	Optional - name of property to set to time series total count.
Time series read count property:	TimeSeriesReadCount	Optional - name of property to set to time series read count.
Time series default count property:	TimeSeriesDefaultCount	Optional - name of property to set to time series default count.

The index (1+) of time series being read can be set to a property.
This is useful for outputting time series sequentially, for example to a table column with the number.

Time series index property:	TimeSeriesIndex	Optional - name of property to set for time series read index (1+).

ReadTimeSeriesList_Check

ReadTimeSeriesList() Command Editor for Parameters Used in Checks

The command syntax is as follows:

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

Command Parameters

Parameter	Description	Default
`TableID`	The identifier for the table that provides the list of location identifiers. Can be specified using processor `${Property}`.	None – must be specified.
`LocationTypeColumn`	The column in the table containing the location type to use in time series identifiers. Specify `LocationTypeColumn` or `LocationType`.	May or may not be required, depending on the datastore or input type.
`LocationType`	The location type in the time series identifier. Specify `LocationTypeColumn` or `LocationType`.	May or may not be required, depending on the datastore or input type.
`LocationColumn`	The column in the table containing the location identifiers to use in time series identifiers.	None – must be specified.
`DataSourceColumn`	The column in the table containing the data source to use in time series identifiers. Specify `DataSourceColumn` or `DataSource`.	May or may not be required, depending on the datastore or input type.
`DataSource`	The data source(s) in the time series identifier, separated by commas. For example, if using the State of Colorado's HydroBase, `USGS` indicates that data are from the United States Geological Survey and `DWR` are from the Division of Water Resources. If multiple data sources are specified, each will be tried until a time series is found. This is enabled because sometimes gages change ownership. Specify `DataSourceColumn` or `DataSource`.	May or may not be required, depending on the datastore or input type
`DataTypeColumn`	The column in the table containing the data type to use in time series identifiers. Specify `DataTypeColumn` or `DataType`.	Data type is often required
`DataType`	The data type in the time series identifier. For example, if using the State of Colorado's HydroBase, `DivTotal` is used for diversion totals. Specify `DataTypeColumn` or `DataType`.	Data type is often required
`Interval`	Data interval in the time series identifier, using standard values such as `15Minute`, `6Hour`, `Day`, `Month`, `Year`.	None – must be specified.
`Scenario`	Scenario in the time series identifier.	Usually not required.
`DataStore`	The data store (or input type) in the time series identifier. Refer to the datastore and input type appendices or the TSTool main GUI for options.	None – must be specified.

Parameter	Description	Default
InputName	The input name in the time series identifier, when a file name is required.	Generally only required when reading from a file.
Alias	Time series alias to assign, using a combination of % specifiers and literal strings.	No alias is assigned.
ColumnProperties	Column names and matching time series property name to set, using syntax: Column1:Property1,Column2:Property2 Specify * for the column name to set all column values as properties. Specify * for the property value to use the column name for the time series property.	No time series properties will be set from the table.
Properties	String properties to be assigned to the time series using syntax Property1:Value1,Property2:Value2	
IfNotFound	Indicates how to handle missing time series, one of: • Warn – generate fatal warnings and do not include in output. • Ignore – generate non-fatal warnings and do not include in output. • Default – generate non-fatal warnings and create empty time series for those that could not be found. This requires that a SetOutputPeriod() command be used before the command to define the period for default time series.	Warn
DefaultUnits	Default units when IfNotFound=Default.	Blank – no units.
DefaultOutputStart	Specify the output period start when a default time series is read, using date/time string or ${Property}.	Uses global output start.
DefaultOutputEnd	Specify the output period start when a default time series is read, using date/time string or ${Property}.	Uses global output end.
TimeSeriesCountProperty	The name of the processor property to set with the total count of time series processed, including read and defaulted time series. Can be specified using processor ${Property}.	
TimeSeriesReadCountProperty	The name of the processor property to set with the count of time series read (not defaulted). Can be specified using processor ${Property}.	
TimeSeriesDefaultCountProperty	The name of the processor property to set with the count of time series that were defaulted. Can be specified using processor ${Property}.	
TimeSeriesIndex1Property	The name of the time series property to set with the index position (1+) for the time series read from the list, essentially a running count of read time series. Can be specified using processor ${Property}.	

A sample command file to process monthly diversion data from the State of Colorado’s HydroBase database is as follows:

```
# Read monthly diversion total from HydroBase for the structures in the list
# file. The data source is set to DWR because data source is saved in
# HydroBase.
ReadTimeSeriesList(TableID=”Diversions.csv",LocationColumn=”WDID”,
  DataSource=DWR,DataType=DivTotal,Interval=Month,InputType=HydroBase,
   IfNotFound=Default)
```