

---

# Command Reference: ReadTimeSeriesList()

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

Version 10.26.00, 2013-12-06

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 or a datastore):

```
# 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 is not generally recommended 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.  
 The information specified below is used with the location identifiers 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:  
 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).  
 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.
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.
Properties:		Optional - string properties to assign to time series.
		<input type="button" value="Edit"/>
If time series not found?:	Warn	Required - how to handle time series that are not found.
Default units:		Optional - units when IfNotFound=Default.
Command:	<pre>ReadTimeSeriesList (TableID="StationList", LocationColumn="TSID", DataSource=" NWIS", DataType="00060-00003", Interval="Day", DataStore="DateValue", InputName ="Data/testdata.dv", IfNotFound=Warn)</pre>	
<input type="button" value="Cancel"/> <input type="button" value="OK"/>		

ReadTimeSeriesList

### ReadTimeSeriesList() Command Editor

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.	None – must be specified.
LocationType Column	The column in the table containing the location type to use in time series identifiers. Specify <code>LocationTypeColumn</code> or <code>LocationType</code> .	May or may not be required, depending on the datastore or input type.
LocationType	The location type in the time series identifier. Specify <code>LocationTypeColumn</code> or <code>LocationType</code> .	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.
DataSource Column	The column in the table containing the data source to use in time series identifiers. Specify <code>DataSourceColumn</code> or <code>DataSource</code> .	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 <code>DataSourceColumn</code> or <code>DataSource</code> .	May or may not be required, depending on the datastore or input type
DataType Column	The column in the table containing the data type to use in time series identifiers. Specify <code>DataTypeColumn</code> or <code>DataType</code> .	Data type is often required
DataType	The data type in the time series identifier. For example, if using the State of Colorado's HydroBase, <code>DivTotal</code> is used for diversion totals. Specify <code>DataTypeColumn</code> or <code>DataType</code> .	Data type is often required
Interval	Data interval in the time series identifier, using standard values such as <code>15Minute</code> , <code>6Hour</code> , <code>Day</code> , <code>Month</code> , <code>Year</code> .	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.
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.

Parameter	Description	Default
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: <ul style="list-style-type: none"> <li>Warn – generate fatal warnings and do not include in output.</li> <li>Ignore – generate non-fatal warnings and do not include in output.</li> <li>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.</li> </ul>	Warn
DefaultUnits	Default units when IfNotFound=Default.	Blank – no units.

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)
```