Command Reference: ReadTimeSeriesList()

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

Version 10.21.00, 2013-05-17

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:

```
# 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

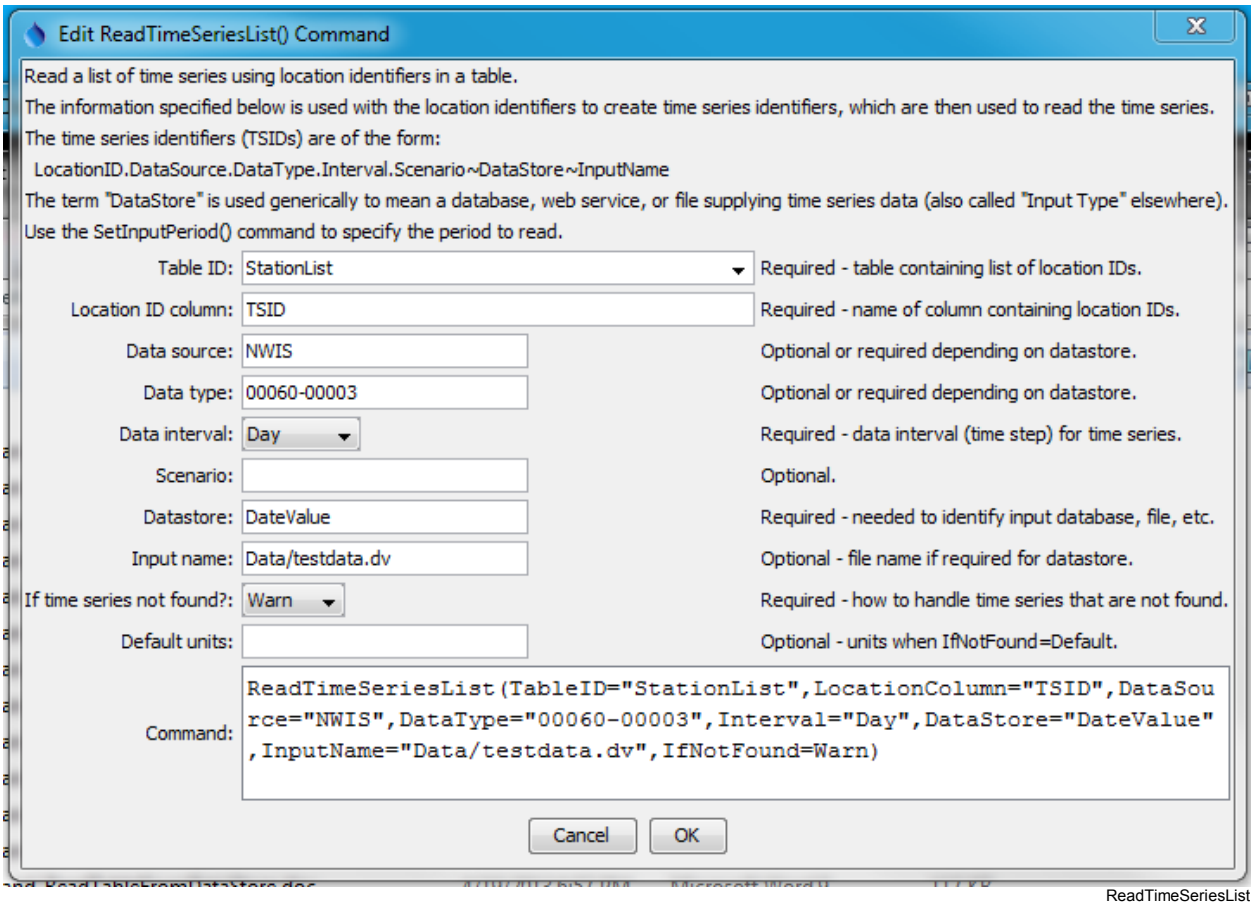
TSTool uses the location identifiers in the table with the command parameters and internally creates a list of time series identifiers. The time series are of the standard form:

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

where the brackets indicate optional information. TSTool then queries each time series, which can be processed further by other commands. See also the `ReadTimeSeries()` command, which performs essentially the same functionality but only reads one time series.

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 are typically 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.



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.
`LocationColumn`	The column in the table containing the location identifiers to use in time series identifiers.	None – must be specified.
`DataSource`	The data source in the time series identifier. For example, if using the State of Colorado's HydroBase, `USGS` indicates that data are from the United States Geological Survey. See the datastore and input type appendices for more information on available data types.	May or may not be required, depending on the datastore or input type. Refer to the input type appendices.
`DataType`	The data type in the time series identifier.	Usually required. Refer to the

Parameter	Description	Default
	For example, if using the State of Colorado's HydroBase, `DivTotal` is used for diversion totals. See the input type appendices for more information on available data types.	datastore and input type appendices.
`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.
`InputName`	The input name in the time series identifier, when a file name is required.	
`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.

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)
```