Command Reference: ResequenceTimeSeriesData()

Resequence time series data (shuffle years of data)

Version 08.15.00, 2008-05-11

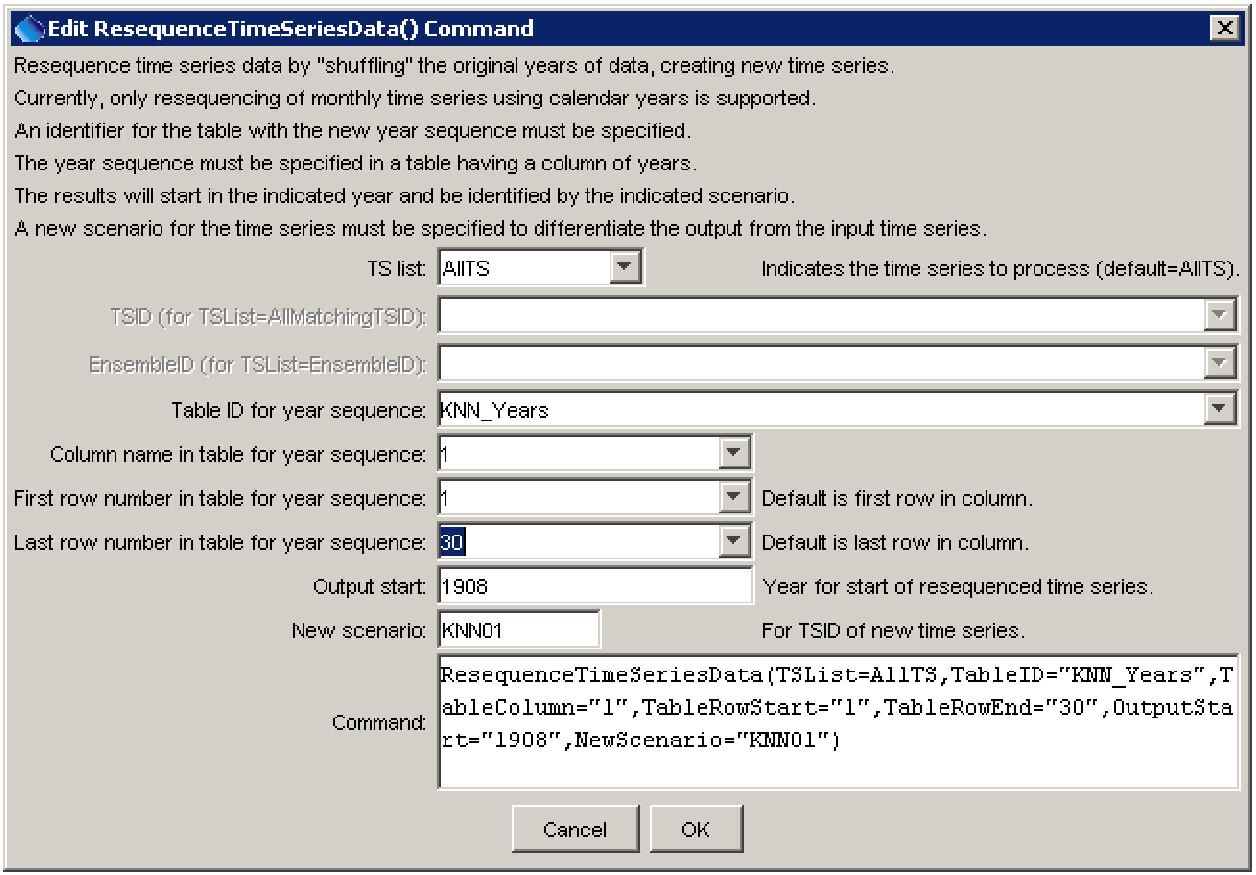
This command is experimental. It is functional; however, additional parameters and functionality may be required for operational implementation. The example that is shown is contrived and anticipates actual use.

The `ResequenceTimeSeriesData()` command resequences data in time series by shifting values from one period to another, creating new time series for each time series. For example, January 1950 might be shifted to January 1990. This command is useful for generating synthetic time series by resequencing historical data. The following constraints apply:

1. Processing occurs by calendar year.
2. Full start and end years are required.
3. For a daily data interval:
   a. If a short year (i.e., non-leap year with 365 days) is transferred to a long year (i.e., a leap year with 366 days), the first day after the short year is used for the 366th day during the transfer. What to do if the year being transferred is the last in the data set and no more years are available for the 366th day – repeat the last day?
   b. If a long year (i.e., leap year with 366 days) is transferred to a short year (i.e., a non-leap year with 365 days), the 366th day in the leap year is not transferred.
4. The original period is by default retained in the output time series. For example, if the original data are 1937 to 1997, the resequenced data will also be in a time series with a period 1937 to 1997. The `OutputStart` parameter can be used to shift the start year of output.

The command is designed to work with a table that provides sequence information. See the `ReadTableFromDelimitedFile()` command – additional table processing commands will be added in the future.

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



Resequence time series data by "shuffling" the original years of data, creating new time series.
Currently, only resequencing of monthly time series using calendar years is supported.
An identifier for the table with the new year sequence must be specified.
The year sequence must be specified in a table having a column of years.
The results will start in the indicated year and be identified by the indicated scenario.
A new scenario for the time series must be specified to differentiate the output from the input time series.

TS list: AllTS — Indicates the time series to process (default=AllTS).

TSID (for TSList=AllMatchingTSID):

EnsembleID (for TSList=EnsembleID):

Table ID for year sequence: KNN_Years

Column name in table for year sequence: 1

First row number in table for year sequence: 1 — Default is first row in column.

Last row number in table for year sequence: 30 — Default is last row in column.

Output start: 1908 — Year for start of resequenced time series.

New scenario: KNN01 — For TSID of new time series.

Command:

```
ResequenceTimeSeriesData(TSList=AllTS,TableID="KNN_Years",TableColumn="1",TableRowStart="1",TableRowEnd="30",OutputStart="1908",NewScenario="KNN01")
```

resequenceTimeSeriesData

ResequenceTimeSeriesData() Command Editor

The command syntax is as follows:

```
ResequenceTimeSeriesData(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) will be modified. • `AllTS` – all time series before the command. • `EnsembleID` – all time series in the ensemble will be modified. • `LastMatchingTSID` – the last time series that matches the `TSID` (single TSID or TSID with wildcards) will be modified. • `SelectedTS` – the time series are those selected with the `SelectTimeSeries()` command.	`AllTS`
`TSID`	The time series identifier or alias for the time series to	`TSID` or

Parameter	Description	Default
	be modified, using the * wildcard character to match multiple time series.	EnsembleID must be specified if identifiers are being matched.
EnsembleID	The ensemble to be modified, if processing an ensemble.	TSID or EnsembleID must be specified if identifiers are being matched.
TableID	The identifier for the sequence table to use, which indicates the dates to use when resequencing data (e.g., list of years for data sequence). For example, see the ReadTableFromDelimitedFile() command.	None – must be specified.
TableColumn	The column name containing the sequence information.	None – must be specified.
TableRowStart	The first data row number (1+) containing the first year in the new sequence.	Use all rows.
TableRowEnd	The last data row number (1+) containing the first year in the new sequence.	Use all rows.
OutputStart	The output start (as year, since processing full calendar years). The output end is relative to the output start and includes the number of years in the sequence.	Same as the original input data.
NewScenario	The new scenario to assign to the created time series.	Blank

The following example:

1. Reads a list of time series from a StateMod model file.
2. Reads a sequence of years from a delimited file.
3. Resequences the StateMod time series data.
4. Writes the resequenced file to a new StateMod file.
5. Runs StateMod (configured to run the resequenced data)
6. Reads StateMod output time series from the StateMod binary file.
7. Write a selected time series to a RiverWare file.

```
# Read all demand time series…
ReadStateMod(InputFile=”..\StateMod\gunnC2005.ddm”)
# Read the sequence of years to use…
Table 0001HK0101 = ReadTableFromDelimitedFile(InputFile=”0001HK0101.csv”)
# Resequence the StateMod time series…
ResequenceTimeSeriesData(TSList=AllTS,TableID=”0001HK0101”,
TableColumn=”Trace1”,NewScenario=”KNN0101”)
# Write the resequenced data for StateMod
WriteStateMod(TSList=AllMatchingTSID,TSID=”*.*.*.*.KNN0101”,
  OutputFile=”..\StateMod0101\gunnC2005.ddm”)
# Run StateMod
RunProgram(CommandLine=”statemod …”)
# Read StateMod results…
ReadStateModB(InputFile=”..\StateMod\gunnC2005C.B43”)
# Write RiverWare time series (repeat for all desired locations)…
WriteRiverWare(TSID=”SomeGage…Month”,OutputFile=”SomeGage_0001HK0101.rdf”)
```

The year sequence is specified in a file similar to the following.

```
Trace1,Trace2,…
1905,1967
1920,1943
etc.
```

Variations on the example can be implemented, for example, to process output products after the run.