2 Introduction

Version 09.08.00, 2010-09-08

TSTool can be thought of as a time series calculator. TSTool displays, manipulates, and analyzes time series data, either interactively or in batch (automated) mode. The TSTool GUI (Graphical User Interface) provides access to viewing and analysis features, command editors, and provides error feedback. Time series can be read from and written to a variety of file and database formats. Although a graphical user interface is provided, the heart of TSTool's analytical features is a command workflow processor. Depending on the task being performed, the command language may be used extensively or not at all. This flexibility makes TSTool useful for basic data viewing and advanced analysis. The documentation is divided into the following main sections.

Chapter 2 – Introduction provides background information on time series concepts and how TSTool processes time series.

Chapter 3 – Getting Started provides an overview of TSTool interface features.

Chapter 4 – Commands provides a summary of time series processing commands.

Chapter 5 – Tools provides information about analysis tools.

Chapter 6 – Examples of Use provides examples of how TSTool is commonly used.

Chapter 7 – Using the Map provides information about using the map interface to link time series with spatial data.

Chapter 8 – Troubleshooting provides troubleshooting information, including a list of obsolete commands.

Chapter 9 – Quality Control provides guidelines and examples for using TSTool to quality control data processing.

The **Command Reference** provides a complete command reference, with commands listed in alphabetical order. Because some commands are used in more than one situation, this allows the commands to be fully documented once, and referred to as needed.

The **Installation and Configuration** appendix provides information about installing and configuring TSTool.

The **Release Notes** appendix summarizes TSTool changes over time.

Several appendices provide information about supported input types (appendices are inserted as additional input types are added).

The **TSView Time Series Viewing Tools** appendix provides a general reference for time series viewing features. These features are used throughout TSTool and other software developed by RTi.

The **GeoView Mapping Tools** appendix provides a general reference for the GeoView map interface. The mapping interface is being phased in and is used by other software developed by RTi.

This documentation can be printed double-sided and is best viewed as PDF to use the navigable table of contents and bookmarks.

2.1 Time Series Objects and Identifiers

TSTool considers time series as objects that are queried, manipulated, viewed, and output. A time series is defined as a series of date/time versus data pairs. Data generally consist of floating point values; however, time series may contain other data (e.g., data quality flags). TSTool treats time series as either regular interval (equal spacing of date/time) or irregular interval (e.g., infrequent measurements, sometimes referred to as *observations*). Regular time series lend themselves to simpler storage and faster processing because date/time information only needs to be stored for the endpoints and processing is less complicated.

TSTool defines each time series as having an interval base and multiplier (e.g., `1Month`, `24Hour`). In many cases, the multiplier is 1 and is not shown in output (e.g., `Month` rather than `1Month` is shown). In addition to a period of record, interval, and data values, time series have attributes, or metadata, that include:

- Units (e.g., `CFS`)
- Data type (e.g., `Streamflow`)
- Data limits (the maximum, minimum, etc.)
- Description (generally a station or structure name)
- Missing data value (used internally to mark missing data and trigger data filling, often `-999` or in some cases `NaN` [Not a Number])
- Comments (often station comments, if available)
- Genesis history (a list of comments about how the time series was created and manipulated)

To manage time series, TSTool associates each time series with an identifier that uses the notation:

```
Location.Source.Type.Interval.Scenario[Seq]~InputType~InputName

Location.Source.Type.Interval.Scenario[Seq]~DataStoreName
```

The first five parts (`Location.Source.Type.Interval.Scenario`) are used to identify time series, with further explanation below:

- `Location` – typically a physical location identifier, such as a station, basin, or sensor identifier.
- `Source` – a data provider identifier, usually a government or system identifier (e.g., `USGS`, `NWS`), necessary because sometimes the provider for data changes for a location.
- `Type` – the data type, typically specific to the data (e.g., `Streamflow`, `Precip`) – TSTool does not try to institute global data type definitions).
- `Interval` – the data interval, indicating the time between data values (e.g., `6Hour`, `Day`, `Irregular`).
- `Scenario` – an optional item that indicates a scenario (e.g., `Hist`, `Filled`, `Max`, `Critical`).
- `Seq` – an optional item used in cases where multiple time series traces may be available, with all other identifier information being equal (e.g., for simulations where multiple versions of input are used or for cases when a historical time series is cut into annual traces, collectively known as

ensembles). Typically the sequence number is a four-digit year corresponding to the data input year.

The last part (`~InputType~InputName` or `DataStoreName`) is used indicate input information, which allows TSTool locate and read the time series from a file or database. The input information was introduced starting with TSTool version 5.04.00. The data store convention was introduced in TSTool version 9.08.00 and allows any name to be used to define a data store – the details of the configuration are defined in a properties file. This allows more flexibility in defining data connections. The data store convention will be phased in as the software is enhanced.

A summary of input types that are currently supported or under development is listed in the following table (see the input type appendices for more information about how time series identifiers are formatted for specific input types). Features for these input types may or may not be available, depending on the TSTool configuration (see the **TSTool Installation and Configuration Appendix**). The main constraint on whether an input type is considered for implementation in TSTool is whether the input type is a standard that is used in a relatively wide audience or for key applications. By supporting general formats, TSTool can support the largest group of users and provide the most useful general features.

Input Types for TSTool

Input Type	Description
ColoradoBNDSS	State of Colorado Basin Needs Decision Support System (BNDSS) database (under development).
ColoradoWaterHBGuest	Web service for State of Colorado’s historical data.
ColoradoWaterSMS	Web service for State of Colorado’s real-time data.
DateValue	General delimited date/value file with extended header information, able to store one or more time series.
Delimited	Generic column-delimited format (see the `ReadDelimitedFile()` command).
DIADvisor	DIADvisor real-time environmental monitoring software, from OneRain, Inc.
HEC-DSS	Army Corp of Engineers binary time series database file used with Hydrologic Engineering Center (HEC) software.
HydroBase	State of Colorado database.
MexicoCSMN	Hydrometeorological database for Mexico Coordinación Servicio Meteorologico Nacional (CSMN, similar to US National Weather Service).
MODSIM	Colorado State University MODSIM model, version 7.
NWSCard	National Weather Service River Forecast System (NWSRFS) card file format for hourly data.
NWSRFS_ESPTraceEnsemble	NWSRFS Ensemble Streamflow Prediction binary files.
NWSRFS_FS5Files	NWSRFS binary FS5Files preprocessor and processed database.
RiversideDB	Riverside Technology, inc. database used for real-time and historical time series data (e.g., use with RiverTrak® System software).
RiverWare	University of Colorado Center for Advanced Decision Support for Water and Environmental Systems (CADSWES) RiverWare model data format.
SHEF	Standard Hydrologic Exchange Format, a common data format used by United States government agencies.

Input Type	Description
StateCU	State of Colorado consumptive use model time series and report formats.
StateCUB	State of Colorado consumptive use model binary output file.
StateMod	State of Colorado StateMod model time series file format.
StateModB	State of Colorado StateMod model output binary file.
USGS NWIS	United States Geological Survey National Water Information System format

An example of a time series identifier for a monthly streamflow time series in HydroBase is:

```
09010500.USGS.Streamflow.Month~HydroBase
```

The same time series for a USGS NWIS input source might be identified using:

```
09010500.USGS.Streamflow.Month~USGSNWIS~C:\temp\09010500.txt
```

In this example, the optional scenario (fifth part) and sequence number are not used. This identifier string can be saved in a command file or time series product description file, which can be processed again later. The identifier string allows TSTool to determine how to re-query the time series. The time series identifier is useful for managing time series. The TSTool GUI typically handles creation of all time series identifiers; however, identifiers can be created with an editor once the format is familiar. The path to files can be absolute or relative to the command file. The latter is recommended to improve portability of files between computers.

Because time series identifiers are somewhat cumbersome to work with, TSTool allows a time series *alias* to be used instead. For example, the following command illustrates how a HydroBase time series can be read and associated with an alias:

```
TS X = ReadTimeSeries("09010500.USGS.Streamflow.Month~HydroBase")
```

This allows the time series to be referred to as `X` during further processing (e.g., when manipulated with commands). Whether full identifiers or aliases are used, the overall identifier must be unique during processing to guarantee that time series commands are processed as desired (duplicate aliases and identifiers can be present but the first one found will be used - see **Section 2.4 Time Series Commands and Processing Sequence** for an example). TSTool ignores upper/lower case when comparing identifiers, aliases, and other commands, although it is good practice to be consistent.

When editing commands, TSTool does not normally show the input type and input name parts of the identifier because this information is most appropriate for read commands. There are cases where two time series identifiers will be the same except for the input type and name. In these cases, an alias should be assigned when reading the time series and the alias used in later commands. If for some reason an alias cannot be used, the input type and name may need to be manually added if the command editors do not display by default (e.g., in cases where the identifiers cannot be determined without actually running a command).

Newer commands also allow the alias to be assigned even when reading multiple time series. This convention uses an `Alias` parameter rather than the `TS Alias =` syntax and may be phased in as the preferred syntax. For example, the alias may be assigned to the location identifier only.

2.2 Date/Time Conventions

TSTool uses date/time information in several ways:

1. Data values in time series are associated with a date/time and the precision of all date/time information should be consistent within the time series, as discussed below,
2. The data interval indicates the time spacing between data points and is represented as a multiplier (optional if `1`) and a base (e.g., `Day`, `24Hour`),
3. The period of a time series is defined by start and end date/time values, using an appropriate precision,
4. An analysis period may be used to indicate when data processing should occur,
5. Output is typically formatted for calendar year (January to December), water year (October to November), or irrigation year (November to October) – calendar year is the default but can be changed in some commands and output.

A date/time has a precision. For example, `2002-02` has a monthly precision and `2002-02-01` has a daily precision. Each date/time object knows its precision and "extra" date/time information is set to reasonable defaults (e.g., hour, minute, and second for a monthly precision date/time are set to zero and the day is set to 1). The date/time precision is important because TSTool uses the date/time objects to iterate through data, to compare dates, and to calculate a plotting position for graphs. Specifying date/time information with incorrect precision may cause inconsistent behavior.

The TSTool documentation and user interface typically use ISO 8601 International Standard formats for date/time information. For example, dates are represented using `YYYY-MM-DD` and times are represented using `hh:mm:ss`. A combined date/time is represented as `YYYY-MM-DD hh:mm:ss`. In order to support common use, TSTool also attempts to handle date/time information that uses United States and other date formats. In such cases, the length of the date/time string and the position of special characters are used to make a reasonable estimate of the format. Using ambiguous formats (e.g., two-digit years that may be confused with months) may cause errors in processing. Adhering to the ISO 8601, standard formats will result in the fewest number of errors. The appendices for various input types discuss issues with date/time formats.

Plotting positions are computed by converting dates to floating point values, where the whole number is the year, and the fraction is the fractional part of the year, considering the precision. The floating-point date is then interpolated to the screen pixels, as integers. In most cases, the high-precision date/time parts are irrelevant because they default to zero. However, in some cases the precision can impact plots significantly. For example, when plotting daily and monthly data on the same graph, the monthly data will be plotted ignoring the day whereas the daily values correspond days 1 to 31. The ability to plot monthly data mid-month or end-of-month has not been implemented. The **TSView Time Series Viewing Tools Appendix** provides examples of plots.

The date/time precision is very important when performing an analysis or converting between time series file formats. For example, a file may contain 6Hour data using a maximum hour of 24 (e.g., 6, 12, 18, 24). When reading this data, TSTool will convert the hour 24 values to hour 0 of the next day. Consequently, the hour and day of the original data will seemingly be shifted, even though the data are actually consistent. This shift may also be perceived when converting from hourly data to daily data because the hour can have a value of 0 to 23, whereas days in the month start with 1. The perceived shift is purely an artifact of time values having a minimum value of zero.

2.3 Time Scale for Time Series Data

The time scale for time series data gives an indication of how the data value were measured or computed. The time scale is generally determined from the data type (or the data type and interval) and can be one of the following (the abbreviations are often used in software choices):

- Instantaneous (INST): The data value represents the data observed at the time associated with the reading (e.g., instantaneous temperature, streamflow, or snow depth). Instantaneous data may be of irregular or regular interval, depending on the data collection system. If irregular, the precision of the date/time associated with the reading may vary (e.g., automated collection systems may have very precise times whereas infrequently recorded field measurements may only be recorded to the nearest day).
- Accumulated (ACCM): The data value represents the accumulation of the observed data over the preceding interval. The date/time associated with the data value corresponds to the end of the interval. For example, precipitation (rain or snow recorded as melt) is often recorded as an accumulation over some interval. Accumulated values are typically available as a regular time series, although this is not a requirement (e.g., precipitation might be accumulated between times that a rain gage is read and emptied).
- Mean (MEAN): The data value represents the mean value of observations during the preceding interval. The date/time associated with the data value corresponds to the end of the interval. The mean includes values after the previous timestamp and including the current timestamp. The computation of mean values may be different depending on whether the original data are irregular or regular. For example, if the original data are regular interval, then equal weight may be given to each value when computing the mean (a simple mean). If the original data are irregular interval, then the weight given to each irregular value may depend on the amount of time that a value was observed (a time-weighted mean, not a simple mean).

Without having specific information about the time scale for data, TSTool assumes that all data are instantaneous for displays. If time series are graphed using bars, an option is given to display the bar to the left, centered on, or to the right of the date/time. If time series are graphed using lines or points, the data values are drawn at the date/time corresponding to data values. This may not be appropriate for the time scale of the data. In most cases, this default is adequate for displays. Graphing data of different time scales together does result in visual inconsistencies. These issues are being evaluated and options may be implemented in future releases of the software. In particular, an effort to automatically determine the time scale from the data type and interval is being evaluated. This can be difficult given that data types are not consistent between input types and time scale may be difficult to determine when reading time series. Refer to the input type appendices for information about time scale.

The time scale is particularly important when changing the time interval of data. For example, conversion of instantaneous data to mean involves an averaging process. Conversion of instantaneous data to accumulated data involves summing the original data. Commands that change interval either operate only on data of a certain time scale or require that the time scale be specified to control the conversion. Refer to the command documentation for specific requirements.

2.4 Time Series Commands and Processing Sequence

Although TSTool can be run in batch mode (see **Chapter 3 – Getting Started**), you should be able to perform all time series viewing and manipulation within the GUI. Commands are used to read, manipulate, and output time series. Commands are processed sequentially from the first to the last commands using the steps described below. This section describes in detail the processing sequence. See the examples in **Chapter 6 – Examples of Use** for illustrations of the processing sequence.

Note that older versions of TSTool (before version 5.xx.xx) did not allow multi-step manipulation and therefore time series were read and manipulated in one step. This convention had limitations and has been changed to allow multi-step operations on time series, allowing more options for filling and manipulation. Old command files are supported as much as possible but some updates to old command files may be required.

TSTool commands fall into three main categories:

1. Time series identifiers (see Section **2.1 – Time Series Objects and Identifiers**), which are equivalent to time series "read" commands (where the identifier input type is used to determine which read command to use),
2. General commands, which are used to set properties like the period for output, and,
3. Time series commands, which are used to read and manipulate time series and output results.

Commands are processed sequentially and can be repeated as necessary. A typical user starts learning TSTool by performing simple queries and displaying results while gradually utilizing more commands. The current software uses command syntax as follows:

```
Command(Param1=Value1,Param2="Value",…)
```

Values that may contain spaces or commas are normally surrounded by double quotes. This notation is useful for the following reasons:

- The parameter names are included in the command, in order to make the command more readable.
- Because the parameter name is included, the parameters can generally be in any order. The command editor dialogs will enforce a default older.
- Parameters that have default values can be omitted from the parameter list, shortening commands.
- New parameters can be added over time, without requiring parameter order to change in existing commands.

The above notation is being used for new commands and older commands are being updated to the new syntax as new software releases are made. Command editor dialogs will update old commands to the new syntax and the processing code will recognize old and new command syntax. The **Command Reference** illustrates the current command syntax.

The following sequence occurs when processing commands:

1. **Parse the command**. A time series identifier or command is parsed to determine how to execute the command. Example commands are shown below. If the command is a general command, the action is taken and a new command is read in step 1 (general commands can be specified multiple times to change properties throughout a run). If the command results in reading or creating a time series, steps 2 - 4 are executed, as described below. If a command is a time series manipulation command, step 4 is executed.

```
# Example commands
08235350.USGS.Streamflow.Month~HydroBase
08236000.USGS.Streamflow.Month~HydroBase
Add(TSID="08235350.USGS.Streamflow.Month",HandleMissingHow=IgnoreMissing,
```

```
  TSList=SpecifiedTSID,AddTSID=”08236000.DWR.Streamflow.Month”)
08235350.USGS.Streamflow.Month~HydroBase
```

2. **Read Time Series.** TSTool recognizes that certain commands should read a new time series and will perform the appropriate action. For example, in the above example, the time series identifier `08235350.USGS.Streamflow.Month~HydroBase` indicates that the corresponding time series should be read from a HydroBase database. The input type in the identifier (information after the ~) is used to determine how to read the time series. Unless the `SetInputPeriod()` command has been used, t**he entire time series period is read in this step** because data filling steps may require the full period (e.g., to determine regression relationships or long-term monthly average).

   Commands that do not cause a time series to be read (but instead to be manipulated) are described in step 4.

   If the input type, and if needed, input name, are specified in the identifier, they are only used in the initial read. Additional manipulation commands only use the first five parts of the identifier or the time series alias to identify the time series. If the same time series needs to be read from two input types (e.g., to compare whether a time series was properly loaded into a database from a file), use a different time series alias for each time series to uniquely identify each time series. This may require using a specific variant of a read command that assigns an alias.

   At the end of this step, a new time series will exist in TSTool's memory.

3. **Compute Data Limits.** The time series data limits are computed because they may be needed later for filling. This information includes the long-term monthly averages. These limits are referred to as the original data limits.

4. **Access and Manipulate Time Series.** Commands that manipulate time series (fill, add, etc.) do not automatically read the time series or make another copy. Instead, time series that are in memory are located and manipulated. The following example illustrates how the time series identified by `08235350.USGS.Streamflow.Month` has its data values modified by adding the data from the time series identified by `08236000.USGS.Streamflow.Month.`

   ```
   # Example commands
   08235350.USGS.Streamflow.Month~HydroBase
   08236000.USGS.Streamflow.Month~HydroBase
   Add(TSID=”08235350.USGS.Streamflow.Month”,HandleMissingHow=IgnoreMissing,TSList
   =SpecifiedTSID,AddTSID=”08236000.DWR.Streamflow.Month”)
   08235350.USGS.Streamflow.Month~HydroBase
   ```

   To locate a time series so that it can be modified, TSTool first checks the alias of known time series (those that have been defined in previous commands) against the current time series of interest (`TSID=”08235350.USGS.Streamflow.Month”`), assuming that this string is an alias. If the alias is not found, it checks the full identifier of known time series against the current time series of interest. In this example, time series `08235350.USGS.Streamflow.Month` was read in the first step and is therefore found as a match for the identifier. Similarly, the second time series in the command (`08236000.USGS.Streamflow.Month`) is found and is used to process the command, resulting in a modification of the first time series. **Sequential manipulations of the same time series can occur (e.g., fill by one method, then fill by another)**.

To locate time series in memory, TSTool generally looks through the list of time series, searching backwards from the current command being processed. Alternatively, the `TSList` parameter for the command will be used if available. It is possible to use the same identifier more than once in a command file while allowing localized processing of each time series; however, this may lead to confusion and should be avoided. In the above example, the time series identified by `08235350.USGS.Streamflow.Month~HydroBase` is read twice, once to be acted on by the `Add()` command, and once with no manipulation (e.g., to compare the "before" and "after").

During processing, extra time series can accumulate and will be available for output. Use the `Free()` command to free time series that are no longer needed. This removes the time series from memory. See also the `DeselectTimeSeries()` and `SelectTimeSeries()` commands. Output commands also may use the `TSList` parameter to indicate which time series are to be output.

5. After processing the time series, a list of available time series that are in memory are listed in the GUI. One or more of these time series can be selected and viewed using the ***Results*** menu or analyzed using the ***Tools*** menu (also right click on time series listed in the results menu at the bottom of the main window). Time series can also be saved in some of recognized input type formats using the ***File...Save…Time Series As*** menus.

If running in batch mode using the `-commands` option, all of the above steps occur in sequence and the GUI interfaces are not displayed. Old command files should be updated to reflect the new processing sequence. Processing the example shown above results in three time series in memory:

1. A time series identified by `08235350.USGS.Streamflow.Month`, containing the sum of the two time series.
2. A time series identified by `08236000.USGS.Streamflow.Month`, containing the input to the `Add()` command.
3. A time series **also** identified by `08235350.USGS.Streamflow.Month`, containing the original data from the time series that is added to. This contains the original data because a time series identifier by itself in a command list will cause the time series to be read.

These time series can be graphed or saved in an output file.

2.5 Using Time Series Aliases

The previous sections discussed time series identifiers and processing time series. The concept of a time series alias was described as a “shortcut” when identifying a time series. Aliases are useful when creating more complicated lists of commands, where using full time series identifiers become cumbersome. Aliases are typically assigned when creating new time series using the following command syntax:

```
TS Alias = SomeCommandThatCreatesATimeSeries()

SomeCommandThatCreatesTimeSeries(Alias=”some pattern”)
```

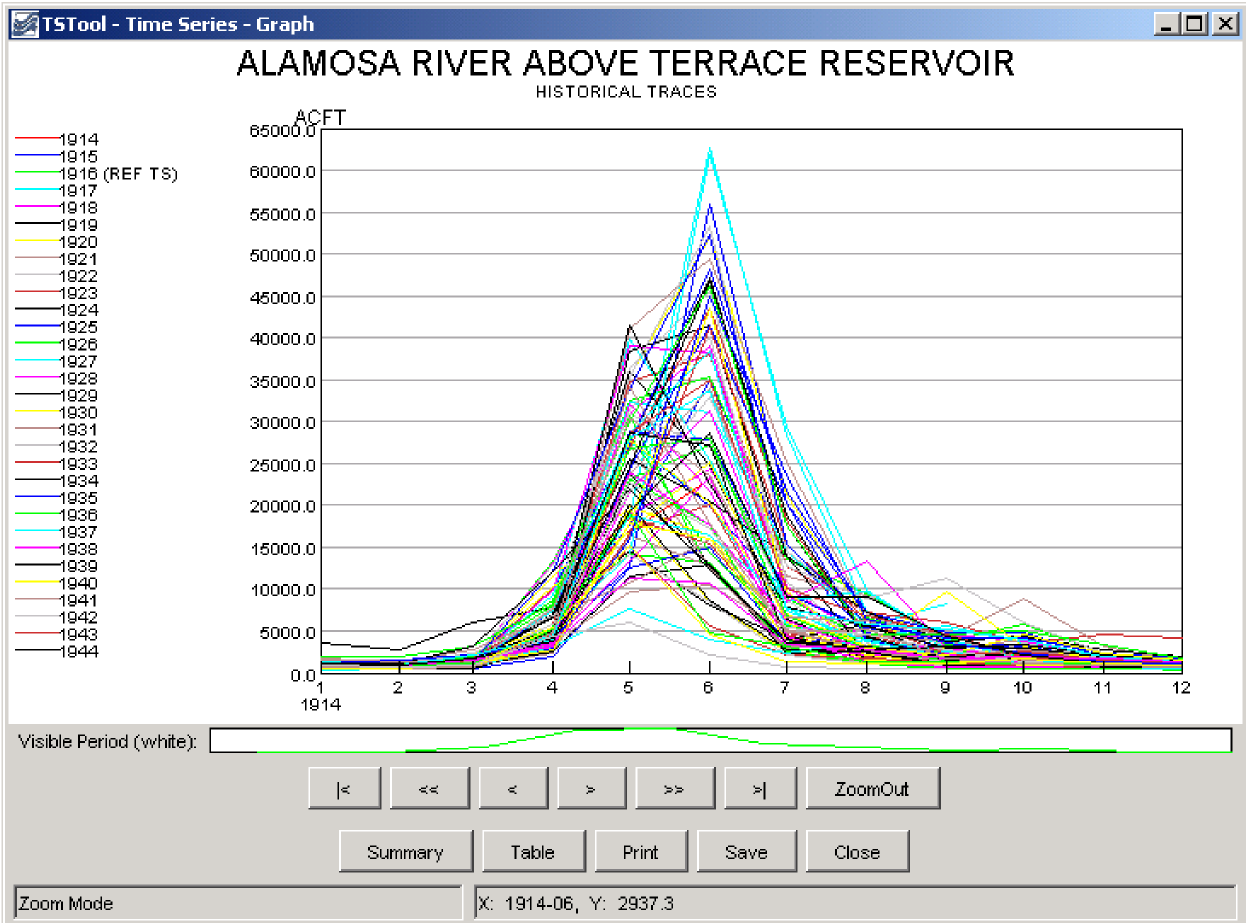
Most supported time series input types do not inherently use aliases (an exception is the DateValue format, which will initialize a time series’ alias if the input DateValue file specifies the information). Instead, time series typically are identified by the location part of the time series identifier (e.g., a station identifier). Although time series can use aliases to simplify processing, the location part of the identifier will generally be used when outputting time series to files or databases.

The time series manipulation features of TSTool facilitate using variations of an input time series for analysis. For example, a time series may be read and manipulated to produce several variants, which are then written for use in a model or analysis. The `TS Alias = Copy()` command could be used to create copies of the original time series. An alternative is to use the `TS Alias = NewTimeSeries()` command to create a new time series (specifying a location part of the time series identifier that is suitable for output), and then use the `SetFromTS()` command to copy all or part of the original time series into the new time series. These two commands therefore allow one time series to be read and copied into new time series, each of which has a new location in the time series identifier. Other commands also allow aliases to be assigned as time series are created.

If specified, time series aliases are generally output in the legends of graphs and other data products.

2.6 Time Series Ensembles

A time series ensemble is a group of related, typically overlapping, time series. Ensembles can be used to manage related scenarios (e.g., input and results of model scenarios) or as a way of shifting a historical time series so that years overlap. Many commands operate on a list of time series by using the parameters `TSList=EnsembleID` and `EnsembleID=”SomeID”`. Statistics time series can be derived from ensembles, for example to calculate the average condition over time (although care must be taken in whether this can be interpreted as a time series of related values, for use as input to a process). Ensembles are assigned unique identifiers and are displayed at the bottom of the TSTool main window in a separate results tab. The following figure illustrates an ensemble of annual time series created from a long historical time series.



TSView/TSView_Graph_Traces

Example Trace Ensemble Plot Showing Historical Years

This page is intentionally blank.