TSTool

– Time Series Tool –

Datastore Reference

Version 10.21.00, 2013-04-21

This page is intentionally blank.

This document is formatted for double-sided printing.

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Appendix: Data Store Overview

2012-05-15

Overview

Data stores are used to configure a "connection" to allow data access, and can be of the following type:

- Database Data Store – used when making connections to relational databases such as Microsoft SQL Server and Oracle. This type of data store is used when a software API (application programmer interface) library is available for direct database communication, typically in an enterprise computing environment. In this case the data store requires a database server name and database name. Database data stores typically are faster but have the disadvantage of requiring that a database be available.
- Web Service Data Store – used when making connections to web services. This type of data store is used when a web service API is available, such as using SOAP or REST web service technologies. Requests and data retrievals occur by making HTTP requests rather than direct connections to a database. Web service data stores typically are slower than database data stores but have the advantage of not requiring the database to be installed locally. One issue with web services is that it is more difficult to provide data browsing features because lists of stations, etc., must be retrieved with a performance penalty. TSTool compensates for this by caching data locally in memory or on disk; however, this can require more complex code. Internet access is required to use web service data stores and performance is impacted by other internet users. When using web service data stores it may be appropriate to retrieve data to a local format and then use the local data to perform additional processing.

Additional data store types may be added in the future. However, requiring a configuration file is cumbersome for simple file access and consequently data stores will be implemented only in cases where the data store configuration can easily be reused between TSTool sessions.

When opened by TSTool, the data store name is used on the end of time series identifiers (TSID) to indicate the storage location for data. For example, the end of the TSID is as follows:

```
Location.Source.DataType.Interval~DataStoreName
```

It is envisioned that some of the legacy "input types" supported by TSTool will be converted to data stores to take advantage of data store design features, including greater configuration ability.

Data Store Configuration Files

A data store is configured by enabling the data store type in the main *TSTool.cfg* configuration file, and creating a data store configuration file for each connection. Configurations are processed at software startup to enable data stores. An example of the TSTool configuration file for the UsgsNwisDaily data store is shown below. Multiple data stores can be defined using the `[DataStore:DataStoreName]` syntax.

```
# Configuration file for TSTool

[TSTool]

UsgsNwisDailyEnabled = true

# Startup data stores (note that data store name in config file takes precedence)

[DataStore:UsgsNwisDaily]

ConfigFile = "UsgsNwisDaily.cfg"
```

TSTool Configuration File with USGS NWIS Daily Data Store Properties

The following illustrates a data store configuration file format, which in this example is located in the same folder as the TSTool configuration file and configures the "UsgsNwisDaily" data store. Several standard properties are required and additional properties may be used for specific data stores.

```
# Configuration information for "UsgsNwisDaily" data store.
# Properties are:
#
# The user will see the following when interacting with the data store:
#
# Type - UsgsNwisDailyDataStore (required with no changes)
# Name - data store identifier used in applications, for example as the
#     input type information for time series identifiers (usually a short string)
# Description - data store description for reports and user interfaces (short phrase)
# Enabled - whether the data store is enabled (default=True)
#
# The following are specific to the USGS NWIS daily data store:
#
# ServiceRootURI - web service root URI, including the server name and root path
# ServiceAPIDocumentationURI - web service API documentation URI, describing
#     the syntax, input, and output
# ServiceOnlineURI - web service interactive page to query data, typically
#     "drill down" or form based

Type = "UsgsNwisDailyDataStore"
Name = "UsgsNwisDaily"
Description = "USGS NWIS Daily Value Web Service"
Enabled = True
ServiceRootURI = "http://waterservices.usgs.gov/nwis/dv"
ServiceAPIDocumentationURI = "http://waterservices.usgs.gov/rest/DV-Service.html"
ServiceOnlineURI = "http://waterservices.usgs.gov/rest/DV-Test-Tool.html"
```

Separate appendices are provided to describe each data store type that is supported by TSTool.

Using Multiple Data Stores to Construct a Complete Time Series

Organizations often provide access to data in different formats, quality, and availability. For example, streamflow may be available as "real-time" on one website and "historical" on another, typically for reasons related to data management, quality control, and other reasons. In many cases there is a concern about real-time data being used for important decisions and therefore such data are labeled as "provisional" with corresponding disclaimers. There also may be technical issues that complicate data access, such as real-time data using one identifier and historical data using another identifier. The use of maps that include layers for stations also can complicate access because a "streamflow station" symbol on a map may not clearly indicate whether corresponding data are available for real-time, daily, monthly, annual, or other representations. Unfortunately, the different data forms can make it difficult to access and use the data in an integrated analysis.

TSTool does not attempt to automatically merge historical and real-time data unless there is clear guidance from a data provider. Instead, separate data stores are provided where appropriate to allow access to historical and real-time data. TSTool can display different data intervals on graphs and in tabular displays; consequently, it generally is straightforward to query and display time series with different intervals. TSTool also provides commands to merge time series into a single time series. The following table illustrates how multiple data stores can be accessed to create a complete time series – this is a partial list meant for illustration. Data store appendices also reference other data stores as appropriate.

Examples of Data Stores and other Data Sources that can be Used in Conjunction to Create Complete Time Series

Data Type	Data Stores for Historical Data	Data Stores for Real-time Data
Climate (precipitation, temperature, etc.)	ColoradoWaterHBGuest (Colorado+), HydroBase (Colorado+), RccAcis (USA+)	ColoradoWaterSMS (minute, hour, day), HydroBase (limited), RccAcis (day)
Diversions	HydroBase (Colorado), UsgsNwisDaily	ColoradoWaterSMS (minute, hour, day), UsgsNwisDaily (day)
Snow	HydroBase (Colorado), Use TSTool WebGet() command with NRCS website (western USA)	Use TSTool WebGet() command with NRCS website (day)
Streamflow	ColoradoWaterHBGuest (Colorado+), HydroBase (Colorado+), UsgsNwisDaily (USA+)	ColoradoWaterSMS (minute, hour, day), HydroBase, UsgsNwisDaily (day)
Well levels	ColoradoWaterHBGuest (Colorado), HydroBase (Colorado)	UsgsNwisDaily?
Well Pumping	ColoradoWaterHBGuest (Colorado, use diversion records), UsgsNwisDaily (?)	

The above table indicates possible sources of information, as recognized by TSTool; however, the intricacies of determining specific stations, data types, etc., are left to the software user and involve an understanding of specific data offerings.

This page is intentionally blank.

Appendix: Colorado Satellite Monitoring System (SMS) Input Type

2010-03-03

Overview

The State of Colorado's Satellite Monitoring System (SMS) database stores observations, configuration information, processed data, and alarms, related to field observations collected from real-time stations throughout the State of Colorado. This database has links to HydroBase (see the **HydroBase Input Type Appendix**).

The database is currently only used to provide alarm information, for use as annotations on real-time data from HydroBase. In the future, additional capability may be added to read raw and processed time series from the Colorado SMS database.

See also the **ColoradoWaterSMS Input Type** appendix, which uses a web service to provide access to real-time streamflow and other data types.

This page is intentionally blank.

Appendix: Colorado Water HydroBase Guest (ColoradoWaterHBGuest) Data Store

2012-05-15

Overview

The State of Colorado's HydroBase database is the primary database for water data in Colorado. However, using the HydroBase input type in TSTool (see the **HydroBase Input Type** appendix) requires a direct connection to the database, and a local installation of the database may not be available. The ColoradoWaterHBGuest data store provides internet web service access to historical data and is described here:

> http://www.dwr.state.co.us/HBGuest/Documents/ColoradoHBGuestWebService.pdf
> http://www.dwr.state.co.us/HBGuest/Documents/HBGuestWebService Objects.pdf
> http://www.dwr.state.co.us/HBGuest/default.aspx

The ColoradoWaterHBGuest data store is being enhanced to read all time series that are available via the web service, but currently there are some limitations (see below).

The ColoradoWaterSMS data store (see the **Colorado Water SMS Data Store** appendix) provides access to real-time data using a web service.

ColoradoWaterHBGuest Web Service and Standard Time Series Properties

The standard time series identifier format for ColoradoWaterHBGuest time series is of the form:

```
Location.DataSource.DataType.Interval~DataStoreName
```

The meaning of the parts is as follows:

- The `Location` is set to the State of Colorado's water district identifier (WDID) for structures.
- The `DataSource` is set to the providing agency (e.g., `DWR` for diversion data).
- The `DataType` is set to the "measurement type" described in the State's web service documentation (e.g., `DivTotal` for total diversions through a structure). Refer to the **HydroBase Input Type** appendix for a full list of time series data available in HydroBase. In some cases, the data type used by TSTool will not exactly match HydroBase. For example, TSTool uses `ResMeasStorage` and HydroBase uses `ResMeas` to indicate reservoir measurements, which can contain several observations. The additional "Storage" is needed in TSTool to uniquely identify the time series for the specific data type.
- `Interval` is `Day`, `Month`, or `Year`, as requested. The interval string is converted from HydroBase conventions of `Daily`, `Monthly`, `Annual`, and `Random` (monthly and annual diversion data are stored together in HydroBase and are identified as `Annual` data). Real-time data can be retrieved using the ColoradoWaterSMS data store.
- The data store name (`ColoradoWaterHBGuest` by default) indicates that the data are being read from the ColoradoWaterHBGuest web service.

Limitations

The following limitations of the web service may impact users of the data.

- Data type – the following data types have been implemented, for all available intervals. Additional data types will be supported in the future (see the **HydroBase Input Type** appendix for a complete list of available data types). Note that creating a complete diversion time series, in particular for structures that are not frequently measured, may require considering several data types, including `DivTotal` (total through headgate), `IDivTotal` (infrequent measurement), and `DivComment` (indicates when ditches are out of service, not currently implemented in this data store). The following list of data types does not completely match data that are available in HydroBase, but additional data types will be enabled when they become available in the web services.
    - `AdminFlow – AdminFlow`
    - `Agriculture/GIS – CropAreaAllIrrigation` (pending)
    - `Agriculture/GIS – CropAreaDrip` (pending)
    - `Agriculture/GIS – CropAreaFlood` (pending)
    - `Agriculture/GIS – CropAreaFurrow` (pending)
    - `Agriculture/GIS – CropAreaSprinkler` (pending)
    - `Climate – EvapPan`
    - `Climate – FrostDate`
    - `Climate – Precip`
    - `Climate – Snow`
    - `Climate – SnowCourseDepth`
    - `Climate - SnowCourseSWE`
    - `Climate - Solar`
    - `Climate – TempMax`
    - `Climate – TempMean`
    - `Climate – TempMin`
    - `Climate – VaporPressure`
    - `Climate - Wind`
    - `Diversion - DivTotal`
    - `Diversion - IDivTotal`
    - `Reservoir - RelTotal`
    - `Reservoir - IRelTotal`
    - `Reservoir - ResEOM` – however, no data are returned from the service
    - `Reservoir - ResEOY` – however, no data are returned from the service
    - `Reservoir - ResMeasElev` – treated as a daily time series with many gaps
    - `Reservoir - ResMeasEvap` – treated as a daily time series with many gaps, not sure if daily is appropriate
    - `Reservoir - ResMeasFill` – treated as a daily time series with many gaps, not sure if daily is appropriate
    - `Reservoir - ResMeasRelease` – treated as a daily time series with many gaps, not sure if daily is appropriate
    - `Reservoir - ResMeasStorage` – treated as a daily time series with many gaps
    - `Stream - Streamflow`
    - `Well - WellLevelElev` – treated as a daily time series with many gaps
    - `Well - WellLevelDepth` – treated as a daily time series with many gaps
- Time series metadata – some metadata such as units and measurement counts currently may not available from the web service. This information will be displayed as blank in the time series listing.

- Roundoff – a comparison of data values read directly from HydroBase and from the web service may show very slight differences when values are rounded. This is due to numbers being read and formatted during processing. The differences should not be large enough to significantly impact final results.
- Performance – time series metadata (lists of location/data type/interval combinations) are retrievable from the web service by water district, water division, and single entry. In order for TSTool to provide the user with "drill down" capability starting with a full list of available data, it is necessary to request blocks of data from the web service. However, requesting too large a block results in performance problems due to the bandwidth necessary to transmit data across the network. Consequently, TSTool utilizes caching to store lists of time series metadata, grouped by water district, data type, and interval. The cache is populated based on user requests. Consequently, the first time that data are requested for a district, performance will be slower while the data are retrieved. Subsequent listing of the time series should be fast. Time series data are not currently cached and therefore there may be noticeable slowing for large queries. Additional optimization of data transfer will be evaluated as web service use increases.

Data Store Configuration File

A data store is configured by enabling ColoradoWaterHBGuest data stores in the main *TSTool.cfg* configuration file, and creating a data store configuration file for each data store connection. Configurations are processed at software startup. An example of the TSTool configuration file is shown below. Multiple data stores can be defined using the `[DataStore:DataStoreName]` syntax. For ColoradoWaterHBGuest, this would allow, for example, accessing different versions of the web services.

```
# Configuration file for TSTool
[TSTool]
ColoradoWaterHBGuestEnabled = true

# Data store for Colorado Water SMS web service (active if ColoradoWaterHBGuestEnabled=true above)
[DataStore:ColoradoWaterHBGuest]
ConfigFile = "ColoradoWaterHBGuest.cfg"
```

TSTool Configuration File with ColoradWaterHBGuest Data Store Properties

Properties for each data store are specified in an accompanying configuration file described below.

The following illustrates the ColoradoWaterHBGuest data store configuration file format, which in this example is located in the same folder as the TSTool configuration file and configures a data store named “ColoradoWaterHBGuest”.

```
# Configuration information for "ColoradoWaterHBGuest" web service data store.
# Properties are:
#
# The user will see the following when interacting with the data store:
#
# Name - data store identifier used in applications, for example as the
#     input type information for time series identifiers (usually a short string)
# Description - data store description for reports and user interfaces (short phrase)
# ServiceRootURI - web service root URI, including the server name and root path

Type = "ColoradoWaterHBGuestDataStore"
Name = "ColoradoWaterHBGuest"
Description = "Colorado HydroBase Guest Web Service"
ServiceRootURI = "http://www.dwr.state.co.us/HBGuest/HBGuestWebService.asmx?WSDL"
```

ColoradoWaterHBGuest Data Store Configuration File

Appendix: Colorado Water Satellite Monitoring System (ColoradoWaterSMS) Data Store

2012-05-16

Overview

The State of Colorado's Satellite Monitoring System (SMS) database stores observations, configuration information, processed data, and alarms, related to field observations collected from real-time stations throughout the State of Colorado. This database has links to HydroBase and HydroBase also includes real-time data (see the **HydroBase Input Type Appendix**). However, the HydroBase real-time time series are only available to State of Colorado personnel that have access to the State's internal servers. The ColoradoWaterSMS data store provides internet web service access to real-time data as described here:

http://www.dwr.state.co.us/SurfaceWater/help.aspx

Raw observations as well as hourly and daily aggregations can be requested. Because the data are considered provisional, time series data collected from external providers are NOT available and must be retrieved from the original provider (e.g., USGS). Time series with `DataSource=DWR` and several other data cooperators that rely on the State for data access are available from this web service.

ColoradoWaterSMS Web Services and Standard Time Series Properties

The standard time series identifier format for ColoradoWaterSMS time series is of the form:

```
Location.DataSource.DataType.Interval~DataStoreName
```

The meaning of the parts is as follows:

- The `Location` is set to the State of Colorado's station abbreviation, `ABBREV` (e.g., `PLAKERCO`), which typically is formed from the river/basin name (`PLA`=Platte River), station location (`KER`=Kersey), and state (`CO`=Colorado). Note that currently the web service does not provide the original identifier and therefore the abbreviation is always used.
- The `DataSource` is set to the providing agency. For example, although data are stored in the State's database, the data provider may be `USGS` or another agency. If different from `DWR`, then the data also may be available directly from the provider agency using the agency's station identifier. The State's web services only provide `DWR`–managed data and do not pass through provisional data from other agencies that are able to provide the data from their systems.
- The `DataType` is set to the "variable" described in the State's web service documentation. Streamflow is the primary data type, indicated by `DISCHRG`. Reservoir/lake level is indicated by `ELEV`, and storage by `STORAGE`. It is possible that a single gage collects data at more than one location, for example at branches of a confluence. Use TSTool to query the data store with a data type of "*" to list available data types at a location.
- `Interval` is `Irregular` for raw data, which generally is evenly spaced but may include more frequent observations during events. `Hour` and `Day` interval are also available as aggregated values.
- The data store name (`ColoradoWaterSMS` by default) indicates that the data are being read from the ColoradoWaterSMS web service.

Limitations

The following limitations of the web service may impact users of the data.

- Data type – a description of data types is not available from the web service and therefore cannot automatically be displayed by software. A query to determine a list of data types from time series is performed once and will result in a noticeable delay when the ColoradoWaterSMS input type is initially selected. Subsequent listing of time series will be relatively fast.
- Interval – the interval for raw observations is not provided by the web service and therefore real-time data interval is shown in TSTool as `Irregular`. Stations generally report at regular N-minute intervals; however, some also provide additional event-triggered values that result in more observations in a shorter timeframe. The web service does allow requesting hour and day interval aggregations, which are computed from the raw data by DWR.
- Station – provider identifiers, spatial data (county, state, HUC, lat/long, UTM) are not currently available for time series lists. `ABBREV` may not be assigned for all stations, in particular for external providers.
- Data units – currently are not available from the web service. Units are hard-coded for some data types, and are left blank for other data types, pending clarification from the State.
- Data – web service requests must be made with a query period. A default query period of the most recent 14 days is used to ensure that some data are returned. Optionally, use the `SetInputPeriod()` command in TSTool to specify a longer period. Some stations may not report data in off-season periods.

Data Store Configuration File

A data store is configured by enabling ColoradoWaterSMS data stores in the main *TSTool.cfg* configuration file and creating a data store configuration file for each data store connection. Configurations are processed at software startup. An example of the TSTool configuration file is shown below. Multiple data stores can be defined using the `[DataStore:DataStoreName]` syntax. For ColoradoWaterSMS, this would allow, for example, accessing different versions of the web services.

```
# Configuration file for TSTool

[TSTool]

ColoradoWaterSMSEnabled = true

# Data store for Colorado Water SMS web service (active if ColoradoWaterSMSEnabled=true above)
[DataStore:ColoradoWaterSMS]
ConfigFile = "ColoradoWaterSMS.cfg"
```

TSTool Configuration File with ColoradoWaterSMS Data Store Properties

Properties for each data store are specified in an accompanying configuration file described below.

The following illustrates the ColoradoWaterSMS data store configuration file format, which in this example is located in the same folder as the TSTool configuration file and configures a data store named "ColoradoWaterSMS".

```
# Configuration information for "ColoradoWaterSMS" web service data store.
# Properties are:
#
# The user will see the following when interacting with the data store:
#
# Name - data store identifier used in applications, for example as the
#     input type information for time series identifiers (usually a short string)
# Description - data store description for reports and user interfaces (short phrase)
# ServiceRootURI - web service root URI, including the server name and root path

Type = "ColoradoWaterSMSDataStore"
Name = "ColoradoWaterSMS"
Description = "Colorado Satellite Monitoring System Web Service"
ServiceRootURI = "http://www.dwr.state.co.us/SMS_WebService/ColoradoWaterSMS.asmx?WSDL"
```

ColoradoWaterSMS Data Store Configuration File

This page is intentionally blank.

Appendix: DateValue Input Type

2012-04-05

Overview

The DateValue time series file format can be used to store one or more time series of consistent time interval. The format has been developed to facilitate data management and processing with the TSTool software. The example below shows the format of the file. Important comments about the file format are:

- The file is divided into a header section with time series properties (top) and data section (bottom). Comments can occur anywhere in the file and are lines that start with #.
- The default delimiter between property columns and data columns is a space. Use the `Delimiter` property to reset the delimiter (e.g., to a tab). Adjacent delimiters WILL NOT be merged into one column; consequently, do not use extra delimiters (i.e., whitespace) to format output.
- If not specified, many of the header properties will be set to reasonable defaults as data are read by software such as TSTool. However, as much information as possible should be specified to allow complete time series handling. Header information is displayed by applications like TSTool to allow selection of time series before the data section is read.
- If multiple irregular interval time series are stored in the file, non-overlapping date/times will result in blanks rather than the missing value indicator (missing value indicators indicate actual values in the irregular time series; “missing” values are allowed to have data flags, whereas “no value” are represented as blanks).
- Properties names are case-independent.
- The `TSID`, `Start`, `End`, and `Units` properties are important for basic time series handling.
    - The interval part of the TSID is used to determine how memory should be allocated for data.
    - The `Start` and `End` values are used to allocate memory for regular interval time series. Dates associated with data values are used to allocate memory for irregular interval time series.
    - For regular interval time series, if data lines between the start and end dates are omitted, the unspecified values are set to the missing data value for the time series.

```
# DateValueTS 1.1 file
#
# This is a sample of a typical DateValue minute time series.  An
# example file is as follows and conforms to the following guidelines:
#
# * Comments are lines that start with #.
# * Applications often add a comment section at the top indicating how the
#   file was created
# * Any line that starts with a number is assumed to be a data line.
# * Time hours should be in the range 0 to 23 (an hour of 24 will be
#   converted to hour 0 of the next day).
# * If a time is necessary, the date/time may be separated by a space, T, :, or
#   @.  If a space is used, use date and time column headings.
# * If multiple time series are written, header variables are delimited with
#   space or tab characters.  Data are delimited as per the Delimiter property.
# * Time series are required to have the same data interval.
# * Header property values and column headers can be enclosed in double quotes
#   if the data contain spaces.
# * Missing data can either be coded as the missing data value or no value.
# * Missing records will result in missing data being used when read.
#
# The following header variables are recognized.  This information can be
```

```
# used by software.
Version = 1.1                 # Optional.  File format version
                              # (to handle format changes)
Delimiter = " "               # Optional.  Delimiter for property and data lines
                              # (default is space)
NumTS = 2                     # Optional.  Number of time series in file
                              # (default is 1)
TSID = “XXX.USGS.Streamflow.15MINUTE” “YYY.USGS.Streamflow.15Minute”
                              # Required.
                              # List of time series identifiers in file
                              # Location.Source.DataType.Interval.Scenario
                              # Do not include input type and name in identifier
Alias = “XXXX-Streamflow” “YYYY-Streamflow”
                              # Optional – assign the time series alias
SequenceNum = 1950 1951       # Optional – used with time series ensembles.
                              # Indicates the year for each trace
Description = "Flow at XXX" "Flow at Y"
                              # Optional.  Description for each time series.
DataFlags = true false        # Optional.  Indicates whether data flags for the
                              # data values are provided.  If flags are used,
                              # the data value column a time series with data
                              # flags is followed by a column for the data flag.
                              # Surround the flags by “” if a flag is not
                              # specified or is a space.
DataType = Streamflow Streamflow
                              # Optional.  Data types for each time series
                              # (consistent with TSID if specified).
                              # The default is to use the data type in the TSID
                              # Supplied to simplify use by other programs.
Units = CFS CFS               # Optional.  Units for each time series
                              # (default is no units).
MissingVal = -999 -999        # Optional. Missing data value for each
                              # time series (default is -999).
                              # NaN (not a number) is recommended
IncludeCount = true           # Optional.  If true, column after date/time
                              # is record count (1...) (default is false).
IncludeTotalTime = true       # Optional.  If true, column after date
                              # is cumulative time (0...) (default is false).
# Both of above can be true, and both columns will be added after the date
Start = 1996-10-18:00:00      # Required.  Start date for time series
End = 1997-06-14:00:00        # Required.  End date for time series
                              # Period dates should be of a precision consistent
                              # with the dates used in the data section below.
# Optional.  The following line can be read into a spreadsheet or database for
# headers.  The lines above this line can be ignored in a spreadsheet import.
# The number of headings should agree with the number of columns.
Date "Time" "Count" "TotalTime" "Description 1" “DataFlag1” "Description 2"
1996-10-18 00:00 1 0 110.74 “m” 14.2
1996-10-18 00:15 2 15 113.24 “” 13.7
...
```

DateValue Files and Standard Time Series Properties

The standard time series identifier for DateValue files is as follows:

```
Location.DataSource.DataType.Interval.Scenario~DateValue~PathToFile
```

Because DateValue time series files are a persistent storage format for in-memory time series objects, the properties stored in the file closely match the standard time series properties of the objects. In particular, the time series data type, units, and missing data value are consistent with time series header information. The `TSID` property in a DateValue file is directly applied to time series objects read from the file, allowing explicit identification of the time series in the file, regardless of the name of the file. This allows multiple time series to be saved in a single file. The data source typically agrees with that determined from a data-supplying agency or model that generates the data.

Limitations

DateValue files have the following limitations:

- The header information in DateValue files may be too technical for some general tools. However, simple delimited files require additional information for TSTool to properly handle the data as time series with properties. Spreadsheets can import DateValue files easily by ignoring the header lines.
- Because date/time values are included on every data line, processing DateValue time series files requires more disk space and processing time. However, using the dates on each line also allows gaps in data to be omitted from the file. Inclusion of the date/times for each data point is considered a reasonable trade-off to ensure data quality and readability. Many other time series file formats also include the date/time on each line.

Examples

The following is an example file for day interval data with data flags:

```
#
#
Delimiter   = " "
NumTS       = 1
TSID        = "MyLoc..MyData.Day"
Alias       = "MyLoc"
Description = "Test data, pattern"
DataType    = "MyData"
Units       = "CFS"
MissingVal  = -999.0000
DataFlags   = true
Start       = 1950-01-01
End         = 1951-03-12
#
# Time series comments/histories:
#
#
# Creation history for time series 1 (TSID=MyLoc..MyData.Day Alias=MyLoc):
#
#   Created new time series with interval determined from TSID "MyLoc..MyData.Day"
#   Set 1950-01-01 to 1951-03-12 to pattern=5.000,10.000,12.000,13.000,75.000,
#                        flags=Flag1,Flag2,,Flag4,Flag5
#
#EndHeader
```

```
Date "MyLoc, CFS" DataFlag
1950-01-01 5.0000 "Flag1"
1950-01-02 10.0000 "Flag2"
1950-01-03 12.0000 ""
1950-01-04 13.0000 "Flag4"
1950-01-05 75.0000 "Flag5"
1950-01-06 5.0000 "Flag1"
1950-01-07 10.0000 "Flag2"
1950-01-08 12.0000 ""
1950-01-09 13.0000 "Flag4"
1950-01-10 75.0000 "Flag5"
…
```

The following is an example file for hour interval data (no data flags):

```
#
#
Delimiter   = " "
NumTS       = 1
TSID        = "MyLoc..MyData.Hour"
Alias       = "MyLoc"
Description = "Test data, pattern"
DataType    = "MyData"
Units       = "CFS"
MissingVal  = -999.0000
Start       = 1950-01-01 00
End         = 1950-01-03 12
#
# Time series comments/histories:
#
#
# Creation history for time series 1 (TSID=MyLoc..MyData.Hour Alias=MyLoc):
#
#   Created new time series with interval determined from TSID "MyLoc..MyData.Hour"
#   Set 1950-01-01 00 to 1950-01-03 12 to pattern=5.000,10.000,12.000,13.000,75.000
#
#EndHeader
Date Time "MyLoc, CFS"
1950-01-01 00 5.0000
1950-01-01 01 10.0000
1950-01-01 02 12.0000
1950-01-01 03 13.0000
1950-01-01 04 75.0000
1950-01-01 05 5.0000
1950-01-01 06 10.0000
…
```

Appendix: Generic Database Datastore

2012-11-09

Overview

The generic database datastore can be used to provide general access to database tables and views, for example with the `ReadTableFromDataStore()` command. The trade-off is that although tables and views can be accessed, there is no application programming interface (API) to deal with the intricacies of the database and converting tables to more complex data objects like time series. Consequently, subsequent processing must operate on the returned tables. The `TableToTimeSeries()` command is available to convert a table into time series.

The datastore internally corresponds to an Open Database Connectivity (ODBC) connection. The connection can be defined one of two ways:

- Define an ODBC connection using Windows tools. The advantage of this approach is that database authentication occurs through the ODBC connection. The disadvantage is that the connection may use a generic database driver that does not perform as well as vendor drivers. This approach is used when the `DatabaseEngine` and `OdbcName` configuration properties are defined for the datastore.
- Provide connection information via `DatabaseEngine`, `DatabaseServer`, `DatabaseName`, and potentially login configuration properties, and allow the software to use a vendor-specific JDBC (Java Database Connectivity) driver, which is generally optimized for the database software. The disadvantage of this approach is that advanced authentication interfaces have not been implemented (this may or not be an issue depending on the security enabled for the database).

Limitations

The following limitations apply to the generic database datastore:

- Database permissions control which tables and views are accessible and consequently protected tables may not be visible in software or may generate errors if attempts are made to manipulate outside of permissions.
- An attempt is made in the `ReadTableFromDataStore()` command to list tables and views for selection. However, the ability to filter out system tables is limited because some database drivers do not implement required functionality. For example, the SQL Server JDBC driver does not allow generic filtering of system tables and a work-around has been implemented to remove known system table and view names from lists displayed to users.
- Table column properties in TSTool are determined from database column metadata. Although support for common data types has been implemented, some data types may not be fully supported. If a database column type is not supported, the default is to translate the column data to strings in the output table.
- Although database column properties can specify the width and precision for floating point data, some database metadata is inaccessible, causing data-handling or visualization issues. For example, the SQL Server metadata defaults result in the precision of floating point numbers (called “precision” in TSTool and “scale” in SQL Server column properties) to be set to zero.

The work-around is that any floating point data column that has a precision of zero is treated as having a precision of 6 digits after the decimal point.

Datastore Configuration Files

A datastore is configured by enabling the datastore in the main *TSTool.cfg* configuration file, and creating a datastore configuration file for each connection. Configurations are processed at software startup to enable datastores. An example of the TSTool configuration file is shown below. Multiple datastores can be defined using the `[DataStore:DataStoreName]` syntax. Properties for each datastore are specified in an accompanying configuration file described after the following example.

```
# Configuration file for TSTool

...properties omitted…

# Startup datastores (note that datastore name in config file takes precedence)

[DataStore:SomeDatabaseDataStore]
ConfigFile = "SomeDatabaseDataStore.cfg"
```

TSTool Configuration File with Generic Database Datastore Properties

The following illustrates the generic database datastore configuration file format, which in this example is located in the same folder as the TSTool configuration file.

```
# Configuration information for "SomeDatabaseDataStore" datastore (connection).
#
# The user will see the following when interacting with the datastore:
#
# Type - GenericDatabaseDataStore (required as indicated)
# Name - database identifier for use in applications, for example as the
#     input type/name information for time series identifiers (usually a short string)
# Description - database description for reports and user interfaces (a sentence)
# Enabled - whether the datastore is enabled (default=True)
#
# The following are needed to make the low-level data connection:
#
# DatabaseEngine - the database software (SqlServer)
# OdbcName - ODBC name (specify this OR the following properties)
# DatabaseServer - IP or string address for database server
# DatabaseName - database name used by the server
# SystemLogin - the login to be used for the database connection
# SystemPassword - the password to be used for the database connection
#
# Property values can use the notation “Env:xxxx” to use an environment variable,
# “SysProp:xxxx” to use a JRE system property, or “Prompt” to prompt the user for
# the property value (system console is used - not suitable for TSTool startup from
# the Start menu)

Type = "GenericDatabaseDataStore"
Name = "SomeDatabaseDataStore"
Description = "Database on some server"
Enabled = True
DatabaseEngine = "SqlServer"
# Specify OdbcName…
OdbcName = "OdbcName"
# Or, specify the following…
DatabaseServer = "ServerName"
DatabaseName = "DatabaseName"
SystemLogin = "LoginForConnection"
SystemPassword = "PasswordForConnection"
```

Generic Database Datastore Configuration File

The `DatabaseEngine` can be one of the following values, and is used to control internal database interactions:

- `Access` – Microsoft Access database
- `Excel` – Microsoft Excel workbook (first row of worksheet should be the column names, column types are determined by scanning rows (independent of the ***Rows to Scan*** value in the ODBC DNS setup); refer to sheet in SQL as `"Select * from [Sheet1$]"`)
- `H2` – H2 database
- `Informix` – INFORMIX database
- `MySQL` – MySQL database
- `Oracle` – Oracle database
- `PostgreSQL` – PostgreSQL database
- `SQLServer` – Microsoft SQL Server database

TSTool, which is written in Java, is distributed with the software drivers for the above databases only if datastores have been implemented that use a database product. For example, the State of Colorado's HydroBase database is implemented in SQL Server and consequently the SQL Server driver is distributed with TSTool. Other drivers (e.g., Access via ODBC) depend on installation of the database software, which typically includes the ODBC drivers. Some of the databases listed above have only been used in development and software support may be out of date. If in doubt, contact the software developers and the issue will be evaluated. More databases can be supported if the number of users increases.

The following example illustrates how to configure a datastore for an ODBC DSN connection to an Access database:

```
# Configuration information for Nebraska DNR development database.
# Properties are:
#
# The user will see the following when interacting with the datastore:
#
# Type - required to be GenericDatabaseDataStore
# Name - datastore identifier used in applications, for example as the
#     input type information for time series identifiers (usually a short string)
# Description - datastore description for reports and user interfaces (short phrase)
# DatabaseEngine - the database software
# OdbcName - the Open Database Connectivity Data Source Name (ODBC DSN), configured
#     in Windows Control Panel
#

Type = "GenericDatabaseDataStore"
Name = "ElSalvador"
Description = "El Salvador Database"
DatabaseEngine = "Access"
OdbcName = "ElSalvador"
```

Generic Database Datastore Configuration File Using ODBC DSN Properties

The following example illustrates how to configure a generic datastore for a SQL Server database, using separate database connection properties (NOT using an ODBC DSN). Such configurations may not be suitable because it may be desirable to configure login information in an ODBC DSN. The following is appropriate if a generic read-only service account is configured.

```
# Configuration information for Nebraska DNR development database.
# Properties are:
#
# The user will see the following when interacting with the datastore:
#
# Name - datastore identifier used in applications, for example as the
#     input type information for time series identifiers (usually a short string)
# Description - datastore description for reports and user interfaces (short phrase)
#

Type = "GenericDatabaseDataStore"
Name = "NDNR-Cascade-WaterRights"
Description = "INSIGHT Development Database"
DatabaseEngine = "SqlServer"
DatabaseServer = "xxxxx"
DatabaseName = "WaterRights"
SystemLogin = "guest"
SystemPassword = "guest"
```

Generic Database Datastore Configuration File Using Database Connection Properties

Appendix: HEC-DSS Input Type

2009-01-14

Overview

HEC-DSS input type refers to the United States Army Corps of Engineers' Hydrologic Engineering Center (HEC) Data Storage System (DSS). Refer to the following web sites for more information:

http://www.hec.usace.army.mil (main web site)

http://www.hec.usace.army.mil/software/legacysoftware/hec-dss_pc-dos/documentation/overview.pdf
(HEC-DSS User's Guide and Utility Program Manuals overview)

HEC-DSS Files and Standard Time Series Properties

The standard time series identifier used with TSTool and other software is of the form:

```
Location.DataSource.DataType.Interval.Scenario~InputType~PathToFile
```

The implementation of the identifier for HEC-DSS files is of the form:

```
Apart:Bpart.HEC-DSS.Cpart.Epart.Fpart~HEC-DSS~PathToFile
```

HEC-DSS time series identifier information is taken from the A-F "pathname parts" used to identify HEC-DSS time series. The following assignments are made:

- The location part of the identifier is set to the A-part, a colon, and the B-part. This retains the original HEC-DSS location information. **The colon and period characters cannot be used in the original HEC-DSS A- and B- parts because they conflict with the identifier implementation described above. Instead, it is recommended that dashes, underscores, and other delimiting characters are used within the parts if the HEC-DSS data will be used extensively with TSTool.**
- The data source part of the identifier by default is set to `HEC-DSS` to indicate that "HEC-DSS" is the provider of the data. In the future this could be used to store the data source (data provider) such as "USGS", "NWS", etc., if such information could be obtained from the HEC-DSS time series pathname or supplemental data.
- The data type is set to the C-part. In HEC-DSS, this is referred to as the "parameter".   In HEC-DSS the term "type" is used to indicate whether a time series is instantaneous, mean, or accumulated. **The period character cannot be used in the original HEC-DSS C-part when used with TSTool.**
- The data interval is set to the E-part. **The period character cannot be used in the original HEC-DSS E-part when used with TSTool. Currently the irregular interval is not supported, but will be added in the future.**
- The scenario is set to the F-part. **The period character cannot be used in the original HEC-DSS E-part when used with TSTool.**
- The data units are determined from time series information.
- The D-part is initially used to assign the time series period, but is reset with information from the time series data records, if available. The D-part dates are of the form `DDMonYYYY` or `DDMMYYYY – DDMonYYYY`. A single date indicates the start of a data block in the HEC-DSS data management scheme (see the HEC-DSS documentation referenced above). Two dates indicates the starting and ending data blocks for a condensed catalog (list of time series); however, in this case the ending date

is actually the start of the last data block. The size of each data block depends on the time series interval (e.g., year interval data are stored in blocks of centuries). The HEC-DSS standards for block size are used in conjunction with the time series interval to set the time series end date as appropriate.

- The missing data value is assigned to the internal representation for HEC-DSS files (a large negative number).
- The description is by default set to the location, a comma, followed by the data type.

Limitations

The following limitations are known with the HEC-DSS input type:

- A- and B- parts that include colons will cause an error when converting HEC-DSS identifiers to/from the convention described above – avoid using colons in the A- and B- parts.
- A-, B-, C-, E-, and F- parts that include periods will cause an error when converting to/from HEC-DSS identifiers to the convention described above – avoid using colons in the location identifier. If encountered, the alias for the time series is set to the full identifier with periods. The periods are then removed before setting the full identifier.
- Irregular data are not fully implemented but are supported by the software architecture – they will be enabled in the future.
- Paired data are not currently read – the data are envisioned to be read into the table objects supported by TSTool.
- Access to supplemental data such as station comments is not implemented but is supported by the software architecture.
- Data units from HEC-DSS time series are read and used with the time series. However, conversion between units may not be supported if the units are not included in the data units file used by software. Additional data unit definitions can be added if necessary to facilitate conversions.
- HEC-DSS can store the observation date/time for data to a higher precision than the time series data interval. For example, the observation date/time for an annual value may be July 30, 2400 of the year. The additional precision is ignored when read. An observation time at the end of an interval that might result in a different base date/time (e.g., hour 2400 causing the day to increment for daily interval data), is handled to prevent the rollover.
- Plotting capabilities do not recognize the data scale (instantaneous, mean, accumulated) – line graphs are always drawn as if data were instantaneous. This could be addressed by enhancing the software to utilize a data type file and display styles to determine when data types are instantaneous, mean, or accumulated.

Appendix: HydroBase Datastore

2012-12-18

Overview

This appendix describes the HydroBase datastore, which is being phased in to replace the HydroBase "input type". The newer datastore design is more flexible. Some issues remain before moving completely to the datastore concept and consequently the configuration of a HydroBase datastore is an optional feature that is not enabled by default (to enable, remove the `Enabled=False` property in the example configuration files provided with the software). Technical issues mainly involve how to change the traditional HydroBase login dialog and best practices for naming datastores when dealing with dated versions of HydroBase.

The HydroBase database contains stored procedures and views suitable for the following users (accounts):

- CDSS user (SQL Server login `cdss`, password `cdss%tools`) is a service account that exposes HydroBase data through stored procedures and views appropriate for decision support system projects. HydroBase CDSS view names have the prefix *vw_CDSS*. This is the default account when using the HydroBase datastore.
- HBGuest user (SQL Server login `HBGuest`, password `1HBGuest`) is a service account that exposes HydroBase data through stored procedures and views appropriate for general database users. HydroBase HBGuest view names have the prefix *vw_CDSS*. See the *C:\CDSS\doc\HydroBase\UserManual\HBGuest.pdf* documentation (available after installing the HydroBase DVD) for an overview of HBGuest configuration and database access.

A general rule of thumb is that the CDSS account should be used for full TSTool functionality; however, the HBGuest account can be used to query views that are not available via the CDSS account (e.g., the *vw_HBGuest_StructureAssocWDID* table provides information about augmentation plans and other structure groups).

TSTool connections for the above HydroBase accounts can be figured in several ways:

- CDSS user:
  - Configure as the HydroBase "input type". This is the default legacy behavior when HydroBase is enabled, and relies on the user selecting a HydroBase database via the HydroBase selection dialog. This approach may be phased out in the future in favor of the more generic datastore approach.
  - Configure a HydroBase CDSS datastore using a configuration file and NO Open Database Connectivity (ODBC) Data Source Name (DSN). If the datastore name is configured as "HydroBase", this connection will supersede the "input type" when encountered in time series identifiers. The account will default to the CDSS service account. An example configuration file is shown in the **HydroBase Datastore Configuration** section below.
  - Configure a HydroBase CDSS datastore using a configuration file with an ODBC DSN. In this case, the ODBC DSN must be configured using a SQL Server driver with login `cdss` and password `cdss%tools`. An example configuration file is shown in the **HydroBase Datastore Configuration** section below.

- HBGuest user:
  - Configure a Generic Database datastore using a configuration file and NO Open Database Connectivity (ODBC) Data Source Name (DSN). See the **HydroBase Datastore Configuration** section below). This approach may be needed if the ODBC DSN configuration (next item) results in errors, and the text configuration file may be more transparent than the ODBC DSN.
  - Configure a HydroBase datastore using a configuration file with an ODBC DSN. In this case, the ODBC DSN must be configured using a SQL Server driver with login `HBGuest` and password `1HBGuest`. See the example for the CDSS account in the **HydroBase Datastore Configuration** section below.

The State of Colorado's HydroBase database stores a variety of time series data. The time series conventions described here, in particular for time series identifiers, are consistent for major CDSS software components including TSTool, StateView/CWRAT, StateDMI, and StateMod GUI. This allows for consistent features and sharing of data between software tools.

The current database design splits time series into three main categories:

1. Data related to structures or administrative data maintained by the State of Colorado (e.g., diversions, reservoirs). Structure locations are typically identified using a water district identifier (WDID), consisting of a two-digit State of Colorado water district number and a trailing structure identifier (which in the past was four digits but has been increased to five or more digits to support longer identifiers). Although a single WDID identifier is used when identifying time series, the separate WD and ID fields are generally needed to find information in HydroBase.
2. Data for stations, consisting mainly of location information and time series (e.g., NOAA precipitation data, USGS streamflow). Station locations are typically identified using a station identifier from the data source. For example, stations can use a USGS identifier, a State of Colorado Satellite Monitoring System abbreviation, or other identifier.
3. Data recorded at locations that are not stations or structures. For example, Water Information Sheet (WIS) are daily spreadsheets used to administer water. Although WIS contain data values for structures an stations, the time series are extracted from database tables that are not directly associated with structure or station database tables. Other examples include Colorado and national agricultural crop statistics.

A structure or station may have more than one identifier depending on the number of agencies involved with data collection, etc. For example, a reservoir may have a State of Colorado WDID because it has water rights, a Bureau of Reclamation identifier, a US Geological Survey identifier, and a second State of Colorado identifier corresponding to a real-time data station. HydroBase collects data from many sources; however, the State has not attempted in all cases to cross-references the identifiers. For example, a streamflow station may have a partial time series record with a "USGS" data source and identifier and a partial time series record with a "DWR" (Division of Water Resources) data source and identifier – the user must recognize that this may be the same station, under different management at different times.

HydroBase is updated for release to the public approximately once per year, although internal updates may occur year-round. Time series are used with CDSS (Colorado's Decision Support Systems) applications and follow basic time series standards when used by TSTool and other software.

HydroBase and Standard Time Series Properties

The standard time series identifier format for HydroBase datastore time series is of the form:

```
Location.DataSource.DataType.Interval~DataStoreName
```

Due to the variety of data types, sources, and formats in HydroBase, time series properties can be set a number of ways. General guidelines are as follows:

- The location part of the time series identifier is set to a station or structure identifier, which is typically the identifier used by the managing agency. For example, USGS stream gages use the 8-digit USGS identifier and State of Colorado diversions use a7-character zero-padded structure WDID (water district identifier). Wells often use the latitude and longitude merged together (see additional comments in the wells section).
- The source part of the time series identifier corresponds to a data provider abbreviation. For example, if the current provider for a time series is the USGS, then the data source will be `USGS`. If the State of Colorado has at some point taken over maintenance of a station from the USGS, then the data source will be `DWR`. Individual data records may indicate a variety of data sources. The convention in HydroBase is to store the data records under the current data source, rather than force the user to query more than one time series and merge the time series. If, however, a station has moved, then separate time series typically will be available, likely under different identifiers. For this reason, the location of the station or structure is important in understanding historical records.
- The data type part of the time series identifier as much as possible uses the "measurement type" information in HydroBase or a readable and reasonable data type phrase. For example `Precip` is a measurement type for station data and `DivTotal` (diversion total) is a measurement type for diversion data. In some cases, especially with real-time data, the data type may not exactly match HydroBase. For example, HydroBase uses a measurement type `RT_Rate` for multiple stream related data types. TSTool uses a data type of `Streamflow`. The tables at the end of this appendix describe all available HydroBase data types and provides guidance for upgrading from old data types.
- Data intervals are set based on the tables that are being queried. In most cases, a regular interval like `Day` or `Month` is used. `Irregular` is used for real-time data because there currently is no way to know without doubt what the regular data interval is (e.g., `15MIN`). Data that are measured infrequently (e.g., reservoir field measurements) typically are stored as a regular interval time series with interval `Day`. This allows more flexibility in data processing and filling.
- In older versions of TSTool, the scenario part of the identifier (after the interval) was sometimes used to supplement the data type information. For example, real-time flow data in the database has a number of attributes (`Streamflow`, `RT_Rate`, `DISCHRG`) that cannot easily fit into the standard time series identifier. The current version of TSTool uses datatype-subdatatype where necessary and generally does not use the scenario for normal time series identifiers (WIS time series are an exception) and this field is being reserved to possibly indicate historical data, filled data, etc.
- Units are set based on the database table definitions.
- Period of record is set based on the available database contents. Periods displayed when listing stations and time series typically are not determined by checking the data because this would require querying large amounts of data. Instead, periods are determined from summary information available in the database. In some cases, the period of record information is not saved at a precision sufficient to accurately represent the true period (e.g., the database may indicate data for years but not months). Therefore, the true period will only be available when time series data values are queried.
- Missing data typically are set to `-999` in time series but are stored as nulls in the database. TSTool does allow `NaN` (not a number) to be used but `-999` continues to be used to match legacy conventions.
- The input type of the time series identifier is set to the datastore name, which typically is `HydroBase` for a default connection. If multiple HydroBase connections are needed, the datastore configuration information is used to assign an appropriate name.

- The time scale for data (whether accumulated [ACCM], instantaneous [INST], or mean [MEAN]) is not automatically determined from the data type and interval but must be understood from the data type.

Diversion data may be retrieved from several tables in HydroBase, including daily and monthly detailed records, infrequent values, diversion comments, and currently in use values. The TSTool `ReadHydroBase()` command provides several options for handling data and the `FillUsingDiversionComments()` command can be used to fill with additional zero values. When using time series identifiers to read time series, the following defaults are used:

- Daily `DivClass` and `DivTotal` time series are filled using the carry-forward technique implemented by the State of Colorado. Missing irrigation years remain missing. Years with data are filled with zeros at the start and values are carried forward until another observation is found, or to the end of the irrigation year.
- Diversion comments and "currently in use" flag are NOT automatically applied. This default may change in the future but is retained for historical data processing reasons.

Limitations

HydroBase has the following limitations related to time series storage:

- The station and structure measurement types and time series tables defined in HydroBase do not always allow information to be determined from database records. Instead, some time series properties must be hard-coded based on the table design. For example, the *meas_type* table has a `MeanTemp`, `MaxTemp`, `MinTemp` types defined, but these refer primarily to the separate daily tables for such data. The *monthly_temp* table includes *avg_max_t*, *avg_min_t*, and *mean_t* fields that do not correspond one-to-one with *meas_type* values. Therefore, applications like TSTool use data types that are not specifically defined as strings in HydroBase, which have consequently been hard-coded. This is an issue with station and structure time series.
- Real-time data types in HydroBase do not directly translate to time series data types used in TSTool. An effort has been made to be as consistent as possible while using data types that can be understood by users.
- Data units are not defined consistently in tables. Some tables have a units string and others do not and the units abbreviations are not always consistent (units of "A" are often used for acre-feet and "C" for CFS). A master units table is not used in HydroBase to enforce data units consistency throughout the database.
- The time scale for time series (whether accumulated, instantaneous, or mean) is not automatically determined from the data type and interval. Users much understand how to interpret the data, in particular when changing the data interval.

Datastore Configuration Files

A datastore is configured by enabling a HydroBase datastore in the main *TSTool.cfg* configuration file and creating a datastore configuration file for each connection. Configurations are processed at software startup to enable datastores. An example of the TSTool configuration file is shown below. Multiple datastores can be defined using the `[DataStore:DataStoreName]` syntax. Properties for each datastore are specified in an accompanying configuration file described after the following example.

```
# Configuration file for TSTool
[TSTool]

HydroBaseEnabled = true

# Startup datastores (note that datastore name in config file takes precedence)

# Datastore for Colorado's HydroBase database, CDSS account
# (active if HydroBaseEnabled=true above)
[DataStore:HydroBase2012]
ConfigFile = "HydroBase2012.cfg"

# Datastore for Colorado's HydroBase database, HBGuest account
# (active if HydroBaseEnabled=true above)
[DataStore:HydroBase-HBGuest]
ConfigFile = "HydroBase-HBGuest.cfg"
```

TSTool Configuration File with HydroBase Properties

HydroBase CDSS Account (No ODBC DSN)

The following illustrates the HydroBase datastore configuration file format for the CDSS account NOT using an ODBC DSN. The configuration file is located in the same folder as the TSTool configuration file and configures a datastore named "HydroBase". The default "cdss" service account is used for authentication and allows read-only access to the database.

```
# Configuration information for HydroBase database datastore, CDSS account.
#
# The user will see the following when interacting with the datastore:
#
# Name - datastore identifier used in applications, for example as the
#     input type information for time series identifiers (usually a short string)
# Description - datastore description for reports and user interfaces (short phrase)
#
# The following are needed to make database connections in the software
#
# Type - must be HydroBaseDataStore
# DatabaseEngine - the database software (SqlServer is current standard)
# DatabaseServer - IP or string address for database server, with instance name
#                  (e.g., "localhost\CDSS" can be used for local computer).  Omitting
#                  the instance name will result in default port numbers being tried
#                  until one works.
# DatabaseName - database name used by the server (e.g., HydroBase_CO_20120722)
# SystemLogin - service account login (omit for default)
# SystemPassword - service account password (omit for default)
# Enabled - if True then datastore will be enabled when software starts, False to disable
Type = "HydroBaseDataStore"
Name = "HydroBase"
Description = "HydroBase Datastore"
DatabaseEngine = "SqlServer"
# Local SQL Server Express installation...
```

```
DatabaseServer = "amazon\CDSS"
DatabaseName = "HydroBase_CO_20120722"
```

HydroBase Datastore Configuration File for Default CDSS Account (no ODBC DSN)

HydroBase CDSS Account Using ODBC DSN

The following illustrates the HydroBase datastore configuration file format for the CDSS account using an ODBC DSN. The configuration file is located in the same folder as the TSTool configuration file and configures a datastore named “HydroBase-DSN”. The ODBC DSN must have been configured using SQL Server authentication and the `cdss` account information. This approach can be problematic if the ODBC driver, authentication, method, or other SQL Server configuration information is in error.

```
# Configuration information for HydroBase database datastore, CDSS account, using ODBC DSN.
#
# The user will see the following when interacting with the datastore:
#
# Name - datastore identifier used in applications, for example as the
#     input type information for time series identifiers (usually a short string)
# Description - datastore description for reports and user interfaces (short phrase)
#
# The following are needed to make database connections in the software
#
# Type - must be GenericDatabaseDataStore
# DatabaseEngine - the database software (SqlServer is current standard)
# OdbcName - the Open Database Connectivity Data Source Name (ODBC DSN), configured
#     in Windows Control Panel
# Enabled - if True then datastore will be configured when software starts, False to disable

Enabled = True
Type = "GenericDatabaseDataStore"
Name = "HydroBase-DSN"
Description = "HydroBase Datastore using DSN"
DatabaseEngine = "SqlServer"
OdbcName = "HydroBase-DSN"
```

HydroBase Datastore Configuration File for CDSS Account Using ODBC DSN

HydroBase HBGuest Account using Generic Database Datastore (No ODBC DSN)

The following illustrates the Generic Database datastore configuration file format for the HBGuest account NOT using an ODBC DSN (see also the **Generic Database Datastore** appendix). The configuration file is located in the same folder as the TSTool configuration file and configures a datastore named “HydroBase-HBGuest”. The “HBGuest” service account is used for authentication and allows read-only access to the database.

```
# Configuration information for HydroBase database HBGuest datastore,
# using generic database datastore.
#
# The user will see the following when interacting with the datastore:
#
# Name - datastore identifier used in applications, for example as the
#     input type information for time series identifiers (usually a short string)
# Description - datastore description for reports and user interfaces (short phrase)
#
# The following are needed to make database connections in the software
#
# Type - must be GenericDatabaseDataStore
```

```
# DatabaseEngine - the database software (SqlServer is current standard)
# DatabaseServer - IP or string address for database server, with instance name
#                  (e.g., "localhost\CDSS" can be used for local computer)
# DatabaseName - database name used by the server (e.g., HydroBase_CO_20120722)
# SystemLogin - service account login (specify for HBGuest account)
# SystemPassword - service account password (specify for HBGuest account)

Enabled = True
Type = "GenericDatabaseDataStore"
Name = "HydroBase-HBGuest"
Description = "HydroBase HBGuest Datastore"
DatabaseEngine = "SqlServer"
# Local SQL Server Express installation...
DatabaseServer = "localhost\CDSS"
DatabaseName = "HydroBase_CO_20121126"
# Specify HBGuest login...
SystemLogin = "HBGuest"
SystemPassword = "1HBGuest"
```

Generic Database Datastore Configuration File for HydroBase HBGuest Account

Troubleshooting

The following table lists errors that may result from connecting to or using HydroBase, and potential solutions to address the errors.

Errors and Possible Solutions

Error	Possible solutions
A HydroBase datastore configuration generates errors.	1. Verify that the configuration information specified in the HydroBase datastore configuration file is correct by using another tool to connect to the database: a. Sequel Server Management Studio (SSMS) b. Create an Open Database Connectivity Data Source Name (ODBC DSN). c. Use a database tools such as SQuirreL (see http://squirrel-sql.sourceforge.net/). 2. If the DatabaseServer has been specified with an instance name (e.g., “ServerName\CDSS”, then the server computer (even if the local computer) must be running the SQL Server Browser process. Otherwise, the port number cannot be determined from the instance name. The SQL Server Browser process is started as a service under ***Windows Control Panel… Administrative Tools…Services***. Locate the ***Sql Server Browser*** entry, edit its properties, and set the ***Startup type*** to ***Automatic*** or ***Manual*** and press ***Start*** to start the service. You must have administrator privileges to make this change.
A specific requested time series is not returned from the HydroBase database.	Time series in HydroBase are associated with the data source (e.g., `USGS`). These data source abbreviations or their handling by software may have changed over time and a data source in a time series identifier may not be valid. Current software requires the data source for HydroBase time series, if a data source is used with the data type in HydroBase. Try interactively querying the time series to see if the data source has changed.
Bulk time series queries do not	1. By default all data are returned. However, a filter or previous command may have specified limiting criteria, such as the input period fo r the query.

Error	Possible solutions
returned data from HydroBase.	2. Verify that the database includes the water districts of interest using the ***File…Properties…HydroBase*** Information menu. HydroBase is distributed for the State of Colorado and water divisions and the version being used may be for a division. Subsets of the entire state were distributed in older versions of the database.

Available Time Series by Data Type Categories

The following tables present a summary of time series identifier fields for the HydroBase data types. Data sources may be added and/or removed with data updates. Data types are listed by major group and are alphabetized by the data type description within the group. The time scale is provided to facilitate data use, in particular when changing the time interval.

HydroBase Time Series Types and Standard Time Series Identifier Fields
Agricultural Crop and Livestock Data

Data Group	Data Type Description	Location	Data Source	Data Type	Available Intervals and Time Scale	Comments
Agricultural/ CASS	Colorado Agricultural Statistics Service crop area harvested	County Name	CASS	CropAreaHarvested-Commodity_ Practice Commodity and practice are from available values in HydroBase.	Year INST	See NASS data for orchards, pasture, and vegetables. Perennial crops usually have only harvested value.
	CASS area planted	County Name	CASS	CropAreaPlanted-Commodity_ Practice Commodity and practice are from available values in HydroBase.	Year INST	Annual crops should have planted value but use maximum of planted and harvested if necessary.
	CASS livestock head	County Name	CASS	LivestockHead-Commodity_ Type Commodity and type are from available values in HydroBase.	Year INST	For each commodity (e.g., sheep), multiple types (e.g. sheep at various maturity levels).
Agricultural/ GIS	CDSS irrigated lands assessment result. See also Diversion Comments below.	WDID	CDSSGIS	CropAreaAllIrrigation-CropType CropAreaDrip-CropType CropAreaFlood-CropType CropAreaFurrow-CropType CropAreaSprinkler-CropType CropType is taken from available values in HydroBase.	Year INST	Data are only available for years where DSS projects or data refreshes have occurred. Partial data for intermediate years may be available in spatial data layer attributes but not HydroBase. Data are available for lands served by surface water structures, listed by crop/year/irrigation type.
Agricultural/ NASS	CropArea	County Name	NASS	CropArea-Commodity Commodity is taken from available values in HydroBase.	Year INST	See CASS data where available. NASS does not distinguish between planted and harvested. NASS data are useful for orchards, pasture, and vegetables, which may not be reported in CASS.

HydroBase Time Series Types and Standard Time Series Identifier Fields (Climate Data)
Climate Group Table 1 of 2

Data Group	Data Type Description	Location	Data Source	Data Type	Available Intervals and Time Scale	Comments
Climate	Evaporation (Pan)	Station ID	NOAA	EvapPan Old (obsolete) data type was EPAN.	Day ACCM, Month ACCM	
	Frost Dates (derived from temperatures)	Station ID	COAGM, NOAA	FrostDateL28S, FrostDateL32S, FrostDateF28F, FrostDateF32F Old (obsolete) data type was FrostDate or FrostDates.	Year INST	Time series in software are the Julian day of the year (1-366) to allow graphing, filling, and manipulation.
	Precipitation	Station ID	COAGM, NOAA	Precip Old (obsolete) data type was PTPX.	Day ACCM, Month ACCM, Irregular ACCM	Irregular data are real-time increments.
	Snow (accumulation on ground during interval).	Station ID	NOAA	Snow Old (obsolete) data type was SNOG.	Day ACCM, Month ACCM	
	Snow course depth and snow water equivalent	Station ID	SCS	SnowCourseDepth, SnowCourseSWE Old (obsolete) data type was SnowCrse, SNWE.	Day INST	Values are recorded on a day, with one or more times a month.
	Solar radiation	Station ID	COAGM	Solar Old (obsolete) data type was RADS.	Day ACCM	
	Temperature (instantaneous)	Station ID	various	Temp	Irregular INST	
	Temperature (maximum)	Station ID	COAGM, NOAA	TempMax Old (obsolete) data type was MaxTemp, TAMN.	Day INST	
	Temperature (mean of maximum daily values)	Station ID	COAGM, NOAA	TempMeanMax Old (obsolete) data type was MaxTemp, TAMX with monthly interval.	Month MEAN	
	Temperature (mean)	Station ID	COAGM, NOAA	TempMean Old (obsolete) data type was MeanTemp, TAVG.	Month MEAN	
	Temperature (minimum)	Station ID	COAGM, NOAA	TempMin Old (obsolete) data type was MinTemp, TAMN.	Day INST	
	Temperature (mean of minimum daily values)	Station ID	COAGM, NOAA	TempMeanMin Old (obsolete) data type was MinTemp, TAMN with monthly interval.	Month MEAN	

HydroBase Time Series Types and Standard Time Series Identifier Fields (Climate Data)
Climate Group Table 2 of 2

Data Group	Data Type Description	Location	Data Source	Data Type	Available Intervals and Time Scale	Comments
Climate	Vapor pressure (mean daily)	StationID	COAGM	VaporPressure Old (obsolete) data type was VP, MVP.	Day MEAN	
	Wind run	Station ID	AGRO, COAGM	Wind Old (obsolete) data type was UDIS.	Day ACCM	

HydroBase Time Series Types and Standard Time Series Identifier Fields (Demographic Data)

Demographic data are related to human population. See the Agricultural Data above for livestock population.

Data Group	Data Type Description	Location	Data Source	Data Type(s)	Available Intervals and Time Scale	Comments
Demographics	Human population (persons)	Area_type-Area_name The type indicates whether a county, municipality, state, etc. The name agrees with the type. The combination defines a unique location.	(blank) This could be assumed from the Pop_type part of the data type; however, the data source is not readily available in HydroBase.	HumanPopulation-Pop_type The population type is Census, Estimated, etc.	Year INST	See CDSS documents for information on how population estimates are determined.

HydroBase Time Series Types and Standard Time Series Identifier Fields (Diversion Data)

Data Group	Data Type Description	Location	Data Source	Data Type	Available Intervals and Time Scale	Comments
Diversion May include records for reservoir and well structures, as per State of Colorado administration practices. See also reservoir data.	Diversion Class (showing water color)	`WDID`	`DWR`	`DivClass-SFUT` Old (obsolete) data type was `DQME`, `Div`, or `Diversion`.	`Day MEAN`, `Month INST or ACCM`, `Year INST` or `ACCM`	`SFUT` is encoded as: `S:s F:f U:u T:t` `s` = source `f` = from `u` = use `t` = type Annual values are for irrigation year (Nov-Oct).
	Diversion Comment (the acreage for a diversion and string data flag indicating whether a structure irrigated in a year)	`WDID`	`DWR`	`DivComment`	`Year INST or ACCM`	The numerical time series value is set to the acreage for the year. The data quality flag is set to the HydroBase *diversion_comment. not_used* flag. Therefore, this time series can be used to extract total acreage for a structure and determine if diversions should be zero for a year. Annual values are for irrigation year (Nov-Oct).
	Diversion Total (sum of all DivClass records for a structure).	`WDID`	`DWR`	`DivTotal` Old (obsolete) data type was `DQME`, `Div`, or `Diversion`.	`Day MEAN`, `Month INST or ACCM`, `Year INST or ACCM`	Annual values are for irrigation (Nov-Oct) year.

The above table summarizes how diversion records are available as time series. However, to determine a complete diversion time series, it is necessary to understand the various ways that diversion records can be stored. See also the **State of Colorado's Water Commissioner Manual**.

Raw data observations for a diversion structure are stored as one or more of the following forms in HydroBase:

- Daily water class time series. These data are recorded using irrigation year (November to October). If one or more values have been entered in a month, then HydroBase will include a full month of data. Days at the beginning of the irrigation year that have no observed values at the start of the year should be considered to be zero, regardless of values found in previous irrigation years. Once an observation occurs, then days within the month where an observation was not recorded are set to the last observed value. Therefore, if an irrigation year contains at least one value, that irrigation year will have at least one month of values (with no missing in the month). To preserve space in HydroBase, months with no observations are not included in the daily data in the database. If a year has no observation, then no data are available in HydroBase for the year and a determination of whether the data values should be zero or other must be determined using other data (see below) or engineering judgment. **TSTool and StateView by default implement the carry-forward procedure within irrigation years.**
- Diversion comments. Diversion comments may be included for an irrigation year. The not_used flag indicates if a diversion was not used in a year. If this is the case, then daily diversion records should not be available and a zero value can be assumed for the water year. **TSTool and StateView DO NOT by default use diversion comments when providing daily or monthly time series.**
- Infrequent water class. Infrequent water class values can be entered as an annual value for the irrigation year, or as a monthly value. The data can be accessed as time series in TSTool, although no specific capabilities have been implemented to supplement the daily or monthly time series.

Processed (derived) data records are created as follows:

- Daily total diversion. Daily water class values are accumulated to daily total records. Similar to the daily water class, any month that has at least one value will result in a month with no missing data. To preserve space in HydroBase, only months that include an observation are included in HydroBase. Other months in the same irrigation year should be carried forward. Irrigation years with no observation have no records in HydroBase and a determination of whether the data values should be zero or other must be made using other data (see below) or engineering judgment. **TSTool and StateView by default implement the carry-forward procedure within irrigation years.**
- Monthly water class. Monthly water class is computed by converting the daily water class values (average CFS) to ACFT for each day of the month, and adding the values. Because of the way that daily data are treated, a month will either have all daily values or none. A month with no data will have its value set to missing in the database. Full irrigation years with no observation will result in a full year of missing values, and a determination of whether the data values should be zero or other must be determined using other data (see below) or engineering judgment. Unlike daily data, monthly diversion records are included in HydroBase for the full data period. Full years of missing values may be included in the database.
- Monthly total diversion. This is derived using the same procedure as monthly water class; however, the daily total diversion is used as input.
- Infrequent data are not considered when producing the monthly total time series.

Therefore, to determine a complete time series, the following must be performed, using TSTool or other software:

Daily time series:

1. Read the daily time series from HydroBase. The default in TSTool and StateView is now to carry forward daily diversion time series within the irrigation year.
2. Utilize the diversion comments to set additional years of data to zero. Using diversion comments is an option with TSTool and StateDMI time series read commands.
3. For years with no data, use an appropriate fill technique. If it is known that the ditch did not operate, then zeros should be used. If it is known that the ditch did operate, use historical averages or some other method to fill the data.
4. HydroBase infrequent diversions could be used to supplement the data, but currently there is no software to help users with this process.

Monthly time series:

1. Read the monthly time series from HydroBase. Any irrigation year with at least one daily observation results in 12 monthly time series values.
2. Utilize the diversion comments to set additional years of data to zero. Using diversion comments is an option with TSTool and StateDMI time series read commands.
3. For years with no data, use an appropriate fill technique. If it is known that the ditch did not operate, then zeros should be used. If it is known that the ditch did operate, use historical averages or some other method to fill the data.
4. HydroBase infrequent diversions could be used to supplement the data, but currently there is no software to help users with this process.

Yearly time series:

1. Infrequent time series can be read by TSTool and can supplement the above data. However, currently there is no software to help users with this process. General TSTool commands must be used as appropriate.

HydroBase Time Series Types and Standard Time Series Identifier Fields (Hardware Data)

Data Group	Data Type Description	Location	Data Source	Data Type(s)	Available Intervals and Time Scale	Comments
Hardware	Battery voltage	`Station ID`	`DWR`	Battery	`Irregular INST`	Limited data are available. This data type allows remote system maintenance checks.

Hardware data types are not commonly available have been implemented as a test and to allow for greater future use.

HydroBase Time Series Types and Standard Time Series Identifier Fields (Reservoir Data)
Reservoir Group Table 1 of 2

Data Group	Data Type Description	Location	Data Source	Data Type	Available Intervals and Time Scale	Comments
Reservoir	Field Measurements	WDID	DWR, other	ResMeasElev, ResMeasEvap, ResMeasFill, ResMeas Release, ResMeas Storage Old (obsolete) data type was RSTO.	Day INST, Day ACCM, Day ACCM, Day ACCM, Day ACCM	Reservoir measurements are often recorded at the beginning or end of the month.
	Pool Elevation	Station ID or State of CO Abbrev.	DWR, other	PoolElev	Irregular INST	Real-time data for reservoirs are recorded using a station abbreviation that does not match a WDID.
	Release Class (showing water color)	WDID	DWR	RelClass-SFUT	Day MEAN, Month INST or ACCM, Year INST or ACCM	SFUT is encoded as: S:s F:f U:u T:t s = source f = from u = use t = type Annual values are for irrigation year (Nov-Oct).
	Release Comment (the acreage for a release and string data flag)	WDID	DWR	RelComment	Year INST or ACCM	See DivComment comments. Sometimes acreage is associated with reservoirs. Annual values are for irrigation year (Nov-Oct).
	Release Total (sum of all RelClass records for a structure).	WDID	DWR	RelTotal	Day MEAN, Month INST or ACCM, Year INST or ACCM	Annual values are for irrigation year (Nov – Oct).

HydroBase Time Series Types and Standard Time Series Identifier Fields (Reservoir Data)
Reservoir Group Table 2 of 2

Data Group	Data Type Description	Location	Data Source	Data Type	Available Intervals and Time Scale	Comments
Reservoir	Release (instantaneous)	`Station ID`	`DWR`, `other`	`Release`	`Irregular INST`	Real-time data for reservoirs are recorded using a station abbreviation that does not match a WDID.
	Reservoir Storage (end of month).	`WDID`	`USBR`, `DWR`, `other`	`ResEOM` Old (obsolete) data type was `RSTO`.	`Month INST`	Few time series are available.
	Reservoir Storage (end of year).	`WDID`	`USBR`, `DWR`, `other`	`ResEOY`	`Year INST`	From *annual_res* table. Annual value is for irrigation year (Nov-Oct).
	Storage (instantaneous)	`Station ID` or `State of CO Abbrev.`	`DWR`, `other`	`Storage`	`Irregular INST`	Real-time data for reservoirs are recorded using a station abbreviation that does not match a WDID.

HydroBase Time Series Types and Standard Time Series Identifier Fields (Stream Data)

Data Group	Data Type Description	Location	Data Source	Data Type	Available Intervals and Time Scale	Comments
Stream	Natural Flow	Station ID	USBR	NaturalFlow Old (obsolete) data type was Nat_flow, NQME	Month INST or ACCM	
	Stage	Station ID	DWR, other	Stage	Irregular INST	Real-time data.
	Streamflow	DWR Abbrev. or USGS station ID	DWR, USGS, other	Streamflow Old (obsolete) daily, monthly data type was QME. Old real-time data type used RT_rate and scenario DISCHRG or other *VAXfield* to indicate channel.	Day MEAN, Month INST or ACCM, Irregular INST	Real-time data use Irregular time interval.
	Streamflow (maximum of daily mean)	Station ID	DWR, USGS	StreamflowMax Old (obsolete) data type was Maxq, Maxflow.	Month INST	
	Streamflow (minimum of daily mean)	Station ID	DWR, USGS	StreamflowMin Old (obsolete) data type was Minq, Minflow.	Month INST	
	Water temperature (instantaneous)	Station ID, State of CO Abbrev.	DWR, other	WatTemp	Irregular INST	Real-time data, using identifier that does not match USGS or other identifier for historical data.

HydroBase Time Series Types and Standard Time Series Identifier Fields (Water Information Sheet Data)

Data Group	Data Type Description	Location	Data Source	Data Type	Available Interval and Time Scale	Comments
WIS	Water Information Sheet (WIS) cell values, over time	WIS row identifier. For example, structures have an identifier wdid:NNNNNNN, where the leading "wdid:" is a literal string and the following information is the actual WDID. Similarly, stations start with "stat:", followed by a station ID; confluences with "conf:", followed by the HydroBase *wd_water* numbers for the tributary and the larger stream; other row types with "othr:", followed by a sequential number in the WIS.	`DWR`	Data types match the WIS columns, as follows: `WISPointFlow`, `WISNaturalFlow`, `WISDeliveryFlow`, `WISGainLoss`, `WISRelease`, `WISPriorityDiversion`, `WISDeliveryDiversion`, `WISTribNaturalFlow`, `WISTribDeliveryFlow`, `WISDryRiver` (not currently implemented – may be implemented as a data flag in the future).	`Day MEAN`	The scenario part of the time series identifier is set to the sheet name. Over time, WIS with a particular sheet name may be modified in format. The combination of sheet name and row identifier can be used to find data. The time series description is set to the row label. Data values are as stored for the WIS, which reflect the gain method used when the sheet was stored.

HydroBase Time Series Types and Standard Time Series Identifier Fields (Well Data)

Data Group	Data Type Description	Location	Data Source*	Data Type	Available Intervals and Time Scale	Comments
Well	Well level (elevation and depth)	Location identifier, based on the current data source. For example, if the data source is USGS, the location identifier will be the USGS identifier.	BJORKLUND CH2MHILL CSU CWSD DWR FOX HALAPASKA HILLIER MCCONAGHY NELSON ROBSON ROBSONBANT SCHNEIDER SEO SMITH SOUTHMETRO SPDSS USGS USGS_NAWQA WILSON *as of 2005-06-16	WellLevel (phasing out in favor of WellLevelElev), WellLevelDepth, WellLevelElev	Day INST, Irregular INST	Daily data are historical measurements, often at the ends of a month. A well may have multiple identifiers. However, the identifier presented in TSTool is that corresponding to the primary data source. See information below this table. Irregular data are real-time using state station abbreviations, which do not match the identifier for historical data.

The vw_CDSS_GroundwaterWellsGroundwaterWellsMeasType view in HydroBase contains the metadata for well level data. Well identifiers are used in TSTool time series identifiers as follows:

1. If the view identifier column has a value, the value is used for the well identifier. Typically this is the identifier for the primary data provider.
2. If the WD ID values are available, they are used.
3. If the latitude and longitude are available, the location identifier is set to LL:LatLong, where latitude is format DDMMSS and longitude is DDDMMSS (positive value). The conversion of decimal degrees to degrees, minutes, and seconds truncates the remainder on the seconds. Subsequent reverse lookups of the well occur by querying a box around the location from the identifier and then regenerating the LL: identifier to find a match. This may result in duplicate identifiers if wells are very close together, and will generate an error. The use of the temporary identifier can be minimized by reviewing original data and ensuring that a valid identifier column value is defined in HydroBase.
4. If none of the above methods can be used to assign a location identifier to the time series, an error will result.

This page is intentionally blank.

Appendix: HydroBase Input Type

2012-09-10

Overview

The State of Colorado's HydroBase database stores a variety of time series data. The time series conventions described here, in particular for time series identifiers, are consistent for major CDSS software components including TSTool, StateView/CWRAT, StateDMI, and StateMod GUI. This allows for consistent features and sharing of data between software tools. See also the **HydroBase DataStore Appendix** – HydroBase connectivity is being converted to the datastore design and corresponding documentation will be more up to date going into the future.

The current database design splits time series into three main categories:

1. Data related to structures or administrative data maintained by the State of Colorado (e.g., diversions, reservoirs). Structure locations are typically identified using a water district identifier (WDID), consisting of a two-digit State of Colorado water district number and a trailing structure identifier (which in the past was four digits but has been increased to five or more digits to support longer identifiers). Although a single WDID identifier is used when identifying time series, the separate WD and ID fields are generally needed to find information in HydroBase.
2. Data for stations, consisting mainly of location information and time series (e.g., NOAA precipitation data, USGS streamflow). Station locations are typically identified using a station identifier from the data source. For example, stations can use a USGS identifier, a State of Colorado Satellite Monitoring System abbreviation, or other identifier.
3. Data recorded at locations that are not stations or structures. For example, Water Information Sheet (WIS) are daily spreadsheets used to administer water. Although WIS contain data values for structures an stations, the time series are extracted from database tables that are not directly associated with structure or station database tables. Other examples include Colorado and national agricultural crop statistics.

A structure or station may have more than one identifier depending on the number of agencies involved with data collection, etc. For example, a reservoir may have a State of Colorado WDID because it has water rights, a Bureau of Reclamation identifier, a US Geological Survey identifier, and a second State of Colorado identifier because real-time data are collected. HydroBase collects data from many sources; however, the State has not attempted in all cases to cross-references the identifiers. For example, a streamflow station may have a partial time series record with a "USGS" data source and identifier and a partial time series record with a "DWR" (Division of Water Resources) data source and identifier – the user must recognize that this may be the same station, under different management at different times.

HydroBase is updated for release to the public approximately once per year, although internal updates may occur year-round. Time series are used with CDSS (Colorado's Decision Support Systems) applications and follow basic time series standards when used by TSTool and other software.

HydroBase and Standard Time Series Properties

The standard time series identifier format for HydroBase time series is of the form:

```
Location.DataSource.DataType.Interval~HydroBase
```

Due to the variety of data types, sources, and formats in HydroBase, time series properties can be set a number of ways. General guidelines are as follows:

- The location part of the time series identifier is set to a station or structure identifier, which is typically the identifier used by the managing agency. For example, USGS stream gages will use the 8-digit USGS identifier and State of Colorado diversions will use a structure WDID.
- The source part of the time series identifier corresponds to the current source of the data. For example, if the current provider for a time series is the USGS, then the data source will be USGS. If the State of Colorado has at some point taken over maintenance of a station from the USGS, then the data source will be DWR. Individual data records may indicate a variety of data sources. The convention in HydroBase is to store the data records under the current data source, rather than force the user to query more than one time series and merge the time series. If, however, a station has moved, then separate time series will typically be stored, likely under different identifiers.
- The data type part of the time series identifier as much as possible uses the "measurement type" information in HydroBase or a readable and reasonable data type phrase. For example "Precip" is a measurement type for station data and "DivTotal" (diversion total) is a measurement type for diversion data. In some cases, especially with real-time data, the data type may not exactly match HydroBase. For example, HydroBase uses a measurement type "RT_Rate" for multiple stream related data types. TSTool uses a data type of "Streamflow". In the past, TSTool and other software used data types that did not as closely match the measurement types in HydroBase. For example, daily streamflow was identified as QME (a National Weather Service notation) because that is how it was defined in CRDSS modeling efforts. The table at the bottom of this appendix describes all available HydroBase data types and provides guidance for upgrading from old data types.
- Data intervals are set based on the tables that are being queried. In most cases, a regular interval like DAY or MONTH is used. IRREGULAR is used for real-time data because there is currently no way to know without doubt what the regular data interval is (e.g., 15MIN). Data that are measured infrequently (e.g., reservoir field measurements) are typically stored as a regular interval time series with interval DAY. This allows more flexibility in data processing and filling.
- In older versions of TSTool, the scenario part of the identifier was sometimes used to supplement the data type information. For example, real-time flow data in the database has a number of attributes (`Streamflow`, `RT_Rate`, `DISCHRG`) that cannot easily fit into the standard time series identifier. The current version of TSTool uses datatype-subdatatype where necessary and generally does not use the scenario for normal time series identifiers (WIS time series are an exception) and this field is being reserved to possibly indicate historical data, filled data, etc.
- Units are set based on the database table definitions.
- Period of record is set based on the available database contents. Periods are typically not determined by checking the data because this would require querying large amounts of data. When listing time series, periods are normally determined from summary information available in the database. In some cases, the period of record information is not saved at a precision sufficient to accurately represent the true period (e.g., the database may indicate data for years but not months). Therefore, the true period will only be available when data are actually queried.
- Missing data are typically set to `-999` in time series but are typically stored as nulls in the database.
- The input type of the time series identifier may not be used for older applications. The new convention is being phased in and uses an input type of `HydroBase` (e.g.,

12345678.USGS.QME.DAY~HydroBase). If multiple HydroBase connections are needed, the input name may also be added (e.g., 12345678.USGS.QME.DAY~HydroBase~ServerName), although this capability is only in the evaluation stage.
- The time scale for data (whether accumulated [ACCM], instantaneous [INST], or mean [MEAN]) is not automatically determined from the data type and interval.

Diversion data may be retrieved from several tables in HydroBase, including daily and monthly detailed records, infrequent values, diversion comments, and currently in use values. The TSTool ReadHydroBase() command gives several options for handling data and the FillUsingDiversionComments() command can be used to fill with additional zero values. When using time series identifiers to read time series, the following defaults are used:

- Daily DivClass and DivTotal time series are filled using the carry-forward technique implemented by the State of Colorado. Missing irrigation years remain missing. Years with data are filled with zeros at the start and values are carried forward until another observation is found, or to the end of the irrigation year.
- Diversion comments and "currently in use" flag are NOT automatically applied. This default may change in the future but is retained for historical data processing reasons.

The following tables present a summary of time series identifier fields for the HydroBase data types. Data sources may be added and/or removed with data updates. Data types are listed by major group and are alphabetized by the data type description within the group. The time scale is provided to facilitate data use, in particular when changing the time interval.

HydroBase Time Series Types and Standard Time Series Identifier Fields
Agricultural Crop and Livestock Data

Data Group	Data Type Description	Location	Data Source	Data Type	Available Intervals and Time Scale	Comments
Agricultural/ CASS	Colorado Agricultural Statistics Service crop area harvested	`County Name`	`CASS`	`CropAreaHarvested-Commodity_Practice` Commodity and practice are from available values in HydroBase.	`Year INST`	See NASS data for orchards, pasture, and vegetables. Perennial crops usually have only harvested value.
	CASS area planted	`County Name`	`CASS`	`CropAreaPlanted-Commodity_Practice` Commodity and practice are from available values in HydroBase.	`Year INST`	Annual crops should have planted value but use maximum of planted and harvested if necessary.
	CASS livestock head	`County Name`	`CASS`	`LivestockHead-Commodity_Type` Commodity and type are from available values in HydroBase.	`Year INST`	For each commodity (e.g., sheep), multiple types (e.g. sheep at various maturity levels).

Agricultural/ GIS	CDSS irrigated lands assessment result.
	
	See also Diversion Comments below.
		`WDID`	`CDSSGIS`	`CropAreaAllIrrigation-CropType`
				
				`CropAreaDrip-CropType`
				
				`CropAreaFlood-CropType`
				
				`CropAreaFurrow-CropType`
				
				`CropAreaSprinkler-CropType`
				
				CropType is taken from available values in HydroBase.
					`Year INST`	Data are only available for years where DSS projects or data refreshes have occurred. Partial data for intermediate years may be available in spatial data layer attributes but not HydroBase. Data are available for lands served by surface water structures, listed by crop/year/irrigation type.

Agricultural/ NASS	CropArea	`County Name`	`NASS`	`CropArea-Commodity` Commodity is taken from available values in HydroBase.	`Year INST`	See CASS data where available. NASS does not distinguish between planted and harvested. NASS data are useful for orchards, pasture, and vegetables, which may not be reported in CASS.

HydroBase Time Series Types and Standard Time Series Identifier Fields (Climate Data)
Climate Group Table 1 of 2

Data Group	Data Type Description	Location	Data Source	Data Type	Available Intervals and Time Scale	Comments
Climate	Evaporation (Pan)	Station ID	`NOAA`	`EvapPan` Old (obsolete) data type was `EPAN`.	`Day ACCM`, `Month ACCM`	
	Frost Dates (derived from temperatures)	Station ID	`COAGM`, `NOAA`	`FrostDateL28S`, `FrostDateL32S`, `FrostDateF28F`, `FrostDateF32F` Old (obsolete) data type was `FrostDate or FrostDates`.	`Year INST`	Time series in software are the Julian day of the year (1-366) to allow graphing, filling, and manipulation.
	Precipitation	Station ID	`COAGM`, `NOAA`	`Precip` Old (obsolete) data type was `PTPX`.	`Day ACCM`, `Month ACCM`, `Irregular ACCM`	Irregular data are real-time increments.
	Snow (accumulation on ground during interval).	Station ID	`NOAA`	`Snow` Old (obsolete) data type was `SNOG`.	`Day ACCM`, `Month ACCM`	
	Snow course depth and snow water equivalent	Station ID	`SCS`	`SnowCourseDepth`, `SnowCourseSWE` Old (obsolete) data type was `SnowCrse`, `SNWE`.	`Day INST`	Values are recorded on a day, with one or more times a month.
	Solar radiation	Station ID	`COAGM`	`Solar` Old (obsolete) data type was `RADS`.	`Day ACCM`	
	Temperature (instantaneous)	Station ID	`various`	`Temp`	`Irregular INST`	
	Temperature (maximum)	Station ID	`COAGM`, `NOAA`	`TempMax` Old (obsolete) data type was `MaxTemp`, `TAMN`.	`Day INST`	
	Temperature (mean of maximum daily values)	Station ID	`COAGM`, `NOAA`	`TempMeanMax` Old (obsolete) data type was `MaxTemp`, `TAMX` with monthly interval.	`Month MEAN`	
	Temperature (mean)	Station ID	`COAGM`, `NOAA`	`TempMean` Old (obsolete) data type was `MeanTemp`, `TAVG`.	`Month MEAN`	
	Temperature (minimum)	Station ID	`COAGM`, `NOAA`	`TempMin` Old (obsolete) data type was `MinTemp`, `TAMN`.	`Day INST`	
	Temperature (mean of minimum daily values)	Station ID	`COAGM`, `NOAA`	`TempMeanMin` Old (obsolete) data type was `MinTemp`, `TAMN` with monthly interval.	`Month MEAN`	

HydroBase Time Series Types and Standard Time Series Identifier Fields (Climate Data)
Climate Group Table 2 of 2

Data Group	Data Type Description	Location	Data Source	Data Type	Available Intervals and Time Scale	Comments
Climate	Vapor pressure (mean daily)	StationID	`COAGM`	`VaporPressure` Old (obsolete) data type was `VP, MVP`.	`Day MEAN`	
	Wind run	Station ID	`AGRO, COAGM`	Wind Old (obsolete) data type was `UDIS`.	`Day ACCM`	

HydroBase Time Series Types and Standard Time Series Identifier Fields (Demographic Data)

Demographic data are related to human population. See the Agricultural Data above for livestock population.

Data Group	Data Type Description	Location	Data Source	Data Type(s)	Available Intervals and Time Scale	Comments
Demographics	Human population (persons)	`Area_type-Area_name` The type indicates whether a county, municipality, state, etc. The name agrees with the type. The combination defines a unique location.	`(blank)` This could be assumed from the `Pop_type` part of the data type; however, the data source is not readily available in HydroBase.	`HumanPopulation-Pop_type` The population type is `Census`, `Estimated`, etc.	`Year INST`	See CDSS documents for information on how population estimates are determined.

HydroBase Time Series Types and Standard Time Series Identifier Fields (Diversion Data)

Data Group	Data Type Description	Location	Data Source	Data Type	Available Intervals and Time Scale	Comments
Diversion May include records for reservoir and well structures, as per State of Colorado administration practices. See also reservoir data.	Diversion Class (showing water color)	`WDID`	`DWR`	`DivClass-SFUT` Old (obsolete) data type was `DQME`, `Div`, or `Diversion`.	`Day MEAN`, `Month INST or ACCM`, `Year INST` or `ACCM`	`SFUT` is encoded as: `S:s F:f U:u T:t` `s` = source `f` = from `u` = use `t` = type Annual values are for irrigation year (Nov-Oct).
	Diversion Comment (the acreage for a diversion and string data flag indicating whether a structure irrigated in a year)	`WDID`	`DWR`	`DivComment`	`Year INST or ACCM`	The numerical time series value is set to the acreage for the year. The data quality flag is set to the HydroBase *diversion_comment. not_used* flag. Therefore, this time series can be used to extract total acreage for a structure and determine if diversions should be zero for a year. Annual values are for irrigation year (Nov-Oct).
	Diversion Total (sum of all DivClass records for a structure).	`WDID`	`DWR`	`DivTotal` Old (obsolete) data type was `DQME`, `Div`, or `Diversion`.	`Day MEAN`, `Month INST or ACCM, Year INST or ACCM`	Annual values are for irrigation (Nov-Oct) year.

The above table summarizes how diversion records are available as time series. However, to determine a complete diversion time series, it is necessary to understand the various ways that diversion records can be stored. See also the **State of Colorado's Water Commissioner Manual**.

Raw data observations for a diversion structure are stored as one or more of the following forms in HydroBase:

- Daily water class time series. These data are recorded using irrigation year (November to October). If one or more values have been entered in a month, then HydroBase will include a full month of data. Days at the beginning of the irrigation year that have no observed values at the start of the year should be considered to be zero, regardless of values found in previous irrigation years. Once an observation occurs, then days within the month where an observation was not recorded are set to the last observed value. Therefore, if an irrigation year contains at least one value, that irrigation year will have at least one month of values (with no missing in the month). To preserve space in HydroBase, months with no observations are not included in the daily data in the database. If a year has no observation, then no data are available in HydroBase for the year and a determination of whether the data values should be zero or other must be determined using other data (see below) or engineering judgment. **TSTool and StateView by default implement the carry-forward procedure within irrigation years.**
- Diversion comments. Diversion comments may be included for an irrigation year. The not_used flag indicates if a diversion was not used in a year. If this is the case, then daily diversion records should not be available and a zero value can be assumed for the water year. **TSTool and StateView DO NOT by default use diversion comments when providing daily or monthly time series.**
- Infrequent water class. Infrequent water class values can be entered as an annual value for the irrigation year, or as a monthly value. The data can be accessed as time series in TSTool, although no specific capabilities have been implemented to supplement the daily or monthly time series.

Processed (derived) data records are created as follows:

- Daily total diversion. Daily water class values are accumulated to daily total records. Similar to the daily water class, any month that has at least one value will result in a month with no missing data. To preserve space in HydroBase, only months that include an observation are included in HydroBase. Other months in the same irrigation year should be carried forward. Irrigation years with no observation have no records in HydroBase and a determination of whether the data values should be zero or other must be made using other data (see below) or engineering judgment. **TSTool and StateView by default implement the carry-forward procedure within irrigation years.**
- Monthly water class. Monthly water class is computed by converting the daily water class values (average CFS) to ACFT for each day of the month, and adding the values. Because of the way that daily data are treated, a month will either have all daily values or none. A month with no data will have its value set to missing in the database. Full irrigation years with no observation will result in a full year of missing values, and a determination of whether the data values should be zero or other must be determined using other data (see below) or engineering judgment. Unlike daily data, monthly diversion records are included in HydroBase for the full data period. Full years of missing values may be included in the database.
- Monthly total diversion. This is derived using the same procedure as monthly water class; however, the daily total diversion is used as input.
- Infrequent data are not considered when producing the monthly total time series.

Therefore, to determine a complete time series, the following must be performed, using TSTool or other software:

Daily time series:

1. Read the daily time series from HydroBase. The default in TSTool and StateView is now to carry forward daily diversion time series within the irrigation year.
2. Utilize the diversion comments to set additional years of data to zero. Using diversion comments is an option with TSTool and StateDMI time series read commands.
3. For years with no data, use an appropriate fill technique. If it is known that the ditch did not operate, then zeros should be used. If it is known that the ditch did operate, use historical averages or some other method to fill the data.
4. HydroBase infrequent diversions could be used to supplement the data, but currently there is no software to help users with this process.

Monthly time series:

1. Read the monthly time series from HydroBase. Any irrigation year with at least one daily observation results in 12 monthly time series values.
2. Utilize the diversion comments to set additional years of data to zero. Using diversion comments is an option with TSTool and StateDMI time series read commands.
3. For years with no data, use an appropriate fill technique. If it is known that the ditch did not operate, then zeros should be used. If it is known that the ditch did operate, use historical averages or some other method to fill the data.
4. HydroBase infrequent diversions could be used to supplement the data, but currently there is no software to help users with this process.

Yearly time series:

1. Infrequent time series can be read by TSTool and can supplement the above data. However, currently there is no software to help users with this process. General TSTool commands must be used as appropriate.

HydroBase Time Series Types and Standard Time Series Identifier Fields (Hardware Data)

Data Group	Data Type Description	Location	Data Source	Data Type(s)	Available Intervals and Time Scale	Comments
Hardware	Battery voltage	`Station ID`	`DWR`	Battery	`Irregular INST`	Limited data are available. This data type allows remote system maintenance checks.

Hardware data types are not commonly available have been implemented as a test and to allow for greater future use.

HydroBase Time Series Types and Standard Time Series Identifier Fields (Reservoir Data)
Reservoir Group Table 1 of 2

Data Group	Data Type Description	Location	Data Source	Data Type	Available Intervals and Time Scale	Comments
Reservoir	Field Measurements	WDID	DWR, other	ResMeasElev, ResMeasEvap, ResMeasFill, ResMeas Release, ResMeas Storage Old (obsolete) data type was RSTO.	Day INST, Day ACCM, Day ACCM, Day ACCM, Day ACCM	Reservoir measurements are often recorded at the beginning or end of the month.
	Pool Elevation	Station ID or State of CO Abbrev.	DWR, other	PoolElev	Irregular INST	Real-time data for reservoirs are recorded using a station abbreviation that does not match a WDID.
	Release Class (showing water color)	WDID	DWR	RelClass-SFUT	Day MEAN, Month INST or ACCM, Year INST or ACCM	SFUT is encoded as: S:s F:f U:u T:t s = source f = from u = use t = type Annual values are for irrigation year (Nov-Oct).
	Release Comment (the acreage for a release and string data flag)	WDID	DWR	RelComment	Year INST or ACCM	See DivComment comments. Sometimes acreage is associated with reservoirs. Annual values are for irrigation year (Nov-Oct).
	Release Total (sum of all RelClass records for a structure).	WDID	DWR	RelTotal	Day MEAN, Month INST or ACCM, Year INST or ACCM	Annual values are for irrigation year (Nov – Oct).

HydroBase Time Series Types and Standard Time Series Identifier Fields (Reservoir Data)
Reservoir Group Table 2 of 2

Data Group	Data Type Description	Location	Data Source	Data Type	Available Intervals and Time Scale	Comments
Reservoir	Release (instantaneous)	Station ID	DWR, other	Release	Irregular INST	Real-time data for reservoirs are recorded using a station abbreviation that does not match a WDID.
	Reservoir Storage (end of month).	WDID	USBR, DWR, other	ResEOM Old (obsolete) data type was RSTO.	Month INST	Few time series are available.
	Reservoir Storage (end of year).	WDID	USBR, DWR, other	ResEOY	Year INST	From *annual_res* table. Annual value is for irrigation year (Nov-Oct).
	Storage (instantaneous)	Station ID or State of CO Abbrev.	DWR, other	Storage	Irregular INST	Real-time data for reservoirs are recorded using a station abbreviation that does not match a WDID.

HydroBase Time Series Types and Standard Time Series Identifier Fields (Stream Data)

Data Group	Data Type Description	Location	Data Source	Data Type	Available Intervals and Time Scale	Comments
Stream	Natural Flow	Station ID	USBR	NaturalFlow Old (obsolete) data type was Nat_flow, NQME	Month INST or ACCM	
	Stage	Station ID	DWR, other	Stage	Irregular INST	Real-time data.
	Streamflow	DWR Abbrev. or USGS station ID	DWR, USGS, other	Streamflow Old (obsolete) daily, monthly data type was QME. Old real-time data type used RT_rate and scenario DISCHRG or other *VAXfield* to indicate channel.	Day MEAN, Month INST or ACCM, Irregular INST	Real-time data use Irregular time interval.
	Streamflow (maximum of daily mean)	Station ID	DWR, USGS	StreamflowMax Old (obsolete) data type was Maxq, Maxflow.	Month INST	
	Streamflow (minimum of daily mean)	Station ID	DWR, USGS	StreamflowMin Old (obsolete) data type was Minq, Minflow.	Month INST	
	Water temperature (instantaneous)	Station ID, State of CO Abbrev.	DWR, other	WatTemp	Irregular INST	Real-time data, using identifier that does not match USGS or other identifier for historical data.

HydroBase Time Series Types and Standard Time Series Identifier Fields (Water Information Sheet Data)

Data Group	Data Type Description	Location	Data Source	Data Type	Available Interval and Time Scale	Comments
WIS	Water Information Sheet (WIS) cell values, over time	WIS row identifier. For example, structures have an identifier wdid:NNNNNNN, where the leading "wdid:" is a literal string and the following information is the actual WDID. Similarly, stations start with "stat:", followed by a station ID; confluences with "conf:", followed by the HydroBase *wd_water* numbers for the tributary and the larger stream; other row types with "othr:", followed by a sequential number in the WIS.	`DWR`	Data types match the WIS columns, as follows: `WISPointFlow`, `WISNaturalFlow`, `WISDeliveryFlow`, `WISGainLoss`, `WISRelease`, `WISPriorityDiversion`, `WISDeliveryDiversion`, `WISTribNaturalFlow`, `WISTribDeliveryFlow`, `WISDryRiver` (not currently implemented – may be implemented as a data flag in the future).	`Day MEAN`	The scenario part of the time series identifier is set to the sheet name. Over time, WIS with a particular sheet name may be modified in format. The combination of sheet name and row identifier can be used to find data. The time series description is set to the row label. Data values are as stored for the WIS, which reflect the gain method used when the sheet was stored.

HydroBase Time Series Types and Standard Time Series Identifier Fields (Well Data)

Data Group	Data Type Description	Location	Data Source*	Data Type	Available Intervals and Time Scale	Comments
Well	Well level (elevation and depth)	Location identifier, based on the current data source. For example, if the data source is USGS, the location identifier will be the USGS identifier.	BJORKLUND CH2MHILL CSU CWSD DWR FOX HALAPASKA HILLIER MCCONAGHY NELSON ROBSON ROBSONBANT SCHNEIDER SEO SMITH SOUTHMETRO SPDSS USGS USGS_NAWQA WILSON *as of 2005-06-16	WellLevel (phasing out in favor of WellLevelElev), WellLevelDepth, WellLevelElev	Day INST, Irregular INST	Daily data are historical measurements, often at the ends of a month. A well may have multiple identifiers. However, the identifier presented in TSTool is that corresponding to the current data source. Use StateView to see alternate identifiers for the location, to cross-reference with data outside of HydroBase. Irregular data are real-time using state station abbreviations, which do not match the identifier for historical data.

Limitations

HydroBase has the following limitations related to time series storage:

- The station and structure measurement types and time series tables defined in HydroBase do not always allow information to be determined from database records. Instead, some time series properties must be hard-coded based on the table design. For example, the *meas_type* table has a MeanTemp, MaxTemp, MinTemp types defined, but these refer primarily to the separate daily tables for such data. The *monthly_temp* table includes *avg_max_t*, *avg_min_t*, and *mean_t* fields that do not correspond one-to-one with *meas_type* values. Therefore, applications like TSTool use data types that are not specifically defined as strings in HydroBase, which have consequently been hard-coded. This is an issue with station and structure time series.
- Real-time data types in HydroBase do not directly translate to time series data types used in TSTool. An effort has been made to be as consistent as possible while using data types that can be understood by users.
- Data units are not defined consistently in tables. Some tables have a units string and others do not and the units abbreviations are not always consistent (units of "A" are often used for acre-feet and

"C" for CFS). A master units table is not used in HydroBase to enforce data units consistency throughout the database.

- The time scale for time series (whether accumulated, instantaneous, or mean) is not automatically determined from the data type and interval. Users much understand how to interpret the data, in particular when changing the data interval.

This page is intentionally blank.

Appendix: MODSIM Input Type

2008-10-30

Overview

MODSIM is a computer model developed at Colorado State University that is used for water allocation modeling. MODSIM uses several time series formats for input and output. The MODSIM input type for TSTool is the output file format, which normally contains model output time series and an echo of input time series. This file format is used with MODSIM version 7 output. To generate version 7 output in MODSIM version 8, check the output control option ***MODSIM…Output Control…Enable MODSIM Version 7 Output*** (it is not enabled by default).

The example below shows the format of a MODSIM output file. Important comments about the file format are:

- The file extension for MODSIM output time series files indicate the contents of the file. Contents are specific to model feature types (nodes or links). The FLO file contains flow link information. The RES file contains reservoir information. The DEM file contains demand information. The GW file contains ground water information. The ACC file contains link account (storage) information.
- The first column in the file is either `LINK` or `NODE`, depending on the feature type.
- The node/link number is assigned automatically by MODSIM and should not be used externally because the order can change.
- The name corresponds to that shown in the graphical user interface, input, and output.
- A monthly data file has columns for `YEAR` and `MONTH`, which correspond to the model years and therefore may be water years, calendar years, irrigation years, etc. A daily data file has columns for `WEEK` and `DAY`. Monthly and daily data have columns for `CALENDAR_YEAR`, `CALENDAR_MONTH`, and `CALENDAR_DATE`. For monthly data, the `CALENDAR_DATE` may be the mid- (for flow data) or end-month (for storage data) position. Five date columns will be present in any case.
- The seventh and greater columns contain time series data.
- Each node/link is listed for the first model year's data. This pattern is repeated for each model year. Therefore, it is necessary to scan the entire file to read the full period for any time series.

```
"LINK", "NAME", "YEAR", "MONTH", "CALENDAR_YEAR", "CALENDAR_MONTH", "CALENDAR_DATE", "FLOW", "LOSS", "NATFLOW"
1, "", 1990, 1, 1989, 11, 15, 3500, 0, 3500
1, "", 1990, 2, 1989, 12, 15, 0, 0, 0
1, "", 1990, 3, 1990, 1, 15, 0, 0, 0
1, "", 1990, 4, 1990, 2, 14, 0, 0, 0
1, "", 1990, 5, 1990, 3, 15, 0, 0, 0
1, "", 1990, 6, 1990, 4, 15, 0, 0, 0
1, "", 1990, 7, 1990, 5, 15, 0, 0, 0
1, "", 1990, 8, 1990, 6, 15, 0, 0, 0
1, "", 1990, 9, 1990, 7, 15, 0, 0, 0
1, "", 1990, 10, 1990, 8, 15, 0, 0, 0
1, "", 1990, 11, 1990, 9, 15, 0, 0, 0
1, "", 1990, 12, 1990, 10, 15, 0, 0, 0
...
```

```
80, "HGLOVE", 1990, 1, 1989, 11, 15, 474, 0, 474
80, "HGLOVE", 1990, 2, 1989, 12, 15, 485, 0, 485
80, "HGLOVE", 1990, 3, 1990, 1, 15, 481, 0, 481
80, "HGLOVE", 1990, 4, 1990, 2, 14, 456, 0, 456
80, "HGLOVE", 1990, 5, 1990, 3, 15, 540, 0, 540
80, "HGLOVE", 1990, 6, 1990, 4, 15, 540, 0, 540
80, "HGLOVE", 1990, 7, 1990, 5, 15, 579, 0, 579
80, "HGLOVE", 1990, 8, 1990, 6, 15, 562, 0, 562
80, "HGLOVE", 1990, 9, 1990, 7, 15, 579, 0, 579
80, "HGLOVE", 1990, 10, 1990, 8, 15, 389, 0, 389
80, "HGLOVE", 1990, 11, 1990, 9, 15, 378, 0, 378
80, "HGLOVE", 1990, 12, 1990, 10, 15, 548, 0, 548
```

MODSIM Output Files and Standard Time Series Properties

The standard time series identifier for MODSIM data files is of the form:

```
Location..DataType.Interval~MODSIM~PathToFile
```

MODSIM files contain limited information to assign to standard time series properties. The following assignments are made:

- The location part of the identifier is assigned to the node/link name. In cases where the name is an empty string (e.g., for internal nodes and links) or cases where the node/link name contains a period, the name is assigned to `Node_#` or `Link_#`, where `#` is the node/link number. This identifier is used to uniquely identify locations but should not be used extensively because node and link numbers can be changed in the MODSIM data set changes. Using node/link identifiers without periods in a MODSIM data set will avoid conflict with the standard time series identifier convention, which uses periods.
- The source part of the identifier is left blank.
- The data type is taken from the column name.
- The data interval is determined from the file header. If the header contains `WEEK`, it is assumed that the data are daily and the interval is set to `Day`. Otherwise, monthly data are assumed.
- The units are not set.
- The missing data value is set to `–999`.
- The description is set to the location, a comma, followed by the data type.

Limitations

MODSIM files have the following limitations:

- The files contain the full period of output for all time series, organized by model year. It is somewhat inefficient to extract a single time series from the file.
- Some internal node/link names are set to blank and therefore it may be difficult to associate the time series data with the model feature.
- Units are not indicated in the files.
- Time series properties like the description and data source are not available in the file.

Appendix: NRCS AWDB Web Service Datastore

2012-11-08

Overview

The NRCS AWDB datastore corresponds to the Natural Resources Conservation Service (NRCS) Air and Water Database (AWDB) web service, as described on the following page:

http://www.wcc.nrcs.usda.gov/web_service/AWDB_Web_Service_Reference.htm

Although the NRCS AWDB web services are largely compatible with TSTool conventions, there are a number of limitations, which are discussed below. The web services are implemented using the SOAP protocol and therefore it is not possible to access web services directly with a web browser.

NRCS AWDB and Standard Time Series Properties

The standard time series identifier for NRCS AWDB time series in TSTool is of the form:

```
Location.DataSource.DataType.Interval~DataStoreName
```

More specifically, the identifier adheres to the following convention:

```
State-StationID.NetworkCode.ElementCode.Day~DataStoreName
State-StationID.NetworkCode.ElementCode.Month~DataStoreName
State-StationID.NetworkCode.ElementCode.Year~DataStoreName
State-StationID.NetworkCode.ElementCode.Irregular~DataStoreName
```

where identifier parts and other time series properties are described as follows, using terms described in the NRCS AWDB web service documentation):

- The NRCS AWDB uses a "station triplet" to uniquely identify stations, using the convention `StationID:State:NetworkCode` (e.g., `471:ID:SNTL`). This information is mapped to TSTool conventions as follows:
  - The state abbreviation forms the first part of the TSID location.
  - The `StationID` forms the second part of the TSID location, separated from the state with a dash.
  - The `NetworkCode` corresponds to the TSID data source.
- `ElementCode` is assigned from the `StationElement.elementCd` value returned by the `getStationElements` web service method.
- The interval is assigned from the `StationElement.duration` value returned by the `getStationElements` web service method, as follows:
  - `Duration.DAILY` translates to TSID `Day`
  - `Duration.MONTHLY` translates to TSID `Month`
  - `Duration.SEMIMONTHLY` is not supported (could treat as irregular)
  - `Duration.ANNUAL` translates to TSID `Year`
  - `Duration.SEASONAL` is not supported (could treat as irregular)
  - `Duration.INSTANTANEOUS` translates to TSID `Irregular`
- `DataStoreName` is the user-defined data store name from the configuration information.

- Data units are assigned from the `StationElement.storedUnitsCd` value returned by the `getStationElements` web service method.
- Missing numerical values are internally represented as `NaN` and are assigned to any date/times in the period that do not have values.
- Data value flags, if encountered, are retained in the time series.
- Properties are extracted from the `StationMetaData` and `StationElement` (and `ReservoirMetadata` if a reservoir) objects and are saved as time series properties. For example, the station longitude and latitude are assigned from station metadata.

Limitations

NRCS AWDB data store limitations relative to TSTool standard features are as follows:

- Although the `getStations` web service request allows element codes to be specified for the query, the returned list provides only the station triplet, but not the accompanying element. Consequently, it is not possible to cross-rereference the original request with the returned information. Instead, an additional `getStationElements` request must occur using the triplet in order to find matching elements, which adds to the query time. The requested interval also is checked against the available durations.
- Instantaneous data are handled using the `Irregular` interval because it is not possible to determine from the web service whether returned values are spaced regularly (e.g., `15Min`).
- All values are returned for instantaneous data, even though the web service allows first of day or midnight values to be requested.
- The `getStationElements` method does not appear to return combinations that include `duration=INSTANTANEOUS`, and sometimes null is returned for the duration. The work-around that has been implemented detects daily duration data and assumes that instantaneous is also available in that case. This may lead to warnings when instantaneous data are actually not available.

Datastore Configuration File

A datastore is configured by enabling NRCS AWDB datastores in the main *TSTool.cfg* configuration file, and creating a datastore configuration file for each datastore. Configurations are processed at software startup. An example of the TSTool configuration file is shown below. Multiple data stores can be defined using the `[DataStore:DataStoreName]` syntax. For example, this can be used to access different versions of the web services.

```
# Configuration file for TSTool

[TSTool]

NrcsAwdbEnabled = true

# Startup datastores (note that datastore name in config file takes precedence)

[DataStore:NrcsAwdb]

ConfigFile = "NrcsAwdb.cfg"
```

TSTool Configuration File with NRCS AWDB Datastore Properties

Properties for each datastore are specified in an accompanying data store configuration file (see below), which in the following example is located in the same folder as the TSTool configuration file and configures a data store named “NrcsAwdb”.

```
# Configuration information for NRCS AWDB web service data store.
# Properties are:
#
# Type - must be NrcsAwdbDataStore
# Name - datastore identifier used in applications
# Description - datastore description for reports and user interfaces (short phrase)
# ServiceRootURI - web service root URI, including the server name path to WSDL
# ServiceAPIDocumentationURI - documentation for web service
# ServiceOnlineURI - on-line test tool

Type = "NrcsAwdbDataStore"
Name = "NrcsAwdb"
Description = "NRCS AWDB Web Service"
ServiceRootURI = "http://www.wcc.nrcs.usda.gov/awdbWebService/services?WSDL"
ServiceAPIDocumentationURI =
"http://www.wcc.nrcs.usda.gov/web_service/AWDB_Web_Service_Reference.htm"
# ServiceOnlineURI =
```

NRCS AWDB Datastore Configuration File

This page is intentionally blank.

Appendix: NWSCard Input Type

2005-05-11, Acrobat Distiller

Overview

The NWSCard time series file format (also referred to as NWS DATACARD) can be used to store 1 to 24 hour time series data for:

- A single time series, referred to as the "NWS Card single time series" format.
- One or more time series traces, as written by the ESPADP software, referred to as the "NWS Card trace" format. In this case, additional header information is included in the file and the remainder of the file contains a sequence of sections consistent with the single time series format. The initial header is used to allocate the memory for the time series, while the subsequent headers indicate historical trace years that are identified with a sequence number after reading.

The format has been developed by the National Weather Service and the single time series format is often used with the NWSRFS (National Weather Service River Forecast System) for historical data (e.g., for model calibration). The examples below shows the format of the single time series and trace files. Important comments about the file format are:

- The file is divided into a header section (top) and data section (bottom). Comments can occur only at the top and are lines that begin with `$`. (For the trace file, comments can occur throughout the file, before each trace.) The `#` comments shown below are for illustration.
- If possible, all header information should be included. Header information is displayed by applications like TSTool to allow selection of time series before the data section is read. Omitting station descriptions can lead to confusion when reviewing output.
- The data units and dimension (e.g., `L3/T`) are NWS standards and help general applications like TSTool determine how to convert units and display data to appropriate precision.
- Formatting information in the file was originally implemented to help FORTRAN software read the files. This information is key to interpreting the fixed-format data records.

The following example illustrates the format of an NWSCard single time series file.

```
#NWSCard
#
# This is an example of a typical National Weather Service (NWS) CARD format
# time series, which can be used for hourly data (1-24 hours).  This format
# is commonly used by the NWS.  The NWS Card file uses hours 1 to 24 whereas
# in-memory time series storage uses 0-23.  The translation of date/times
# from the CARD file to in-memory time series occurs as follows as the file
# is read (using a single 31-day month).  The inverse occurs when writing.
#
# Data     | CARD         | Time Series | CARD          | Time Series
# Interval | Start        | Start       | End           | End
# ---------|--------------|-------------|---------------|--------------------
# 6-Hour   | Day 1, Hr 6  | Day 1, Hr 6 | Day 31, Hr 24 | Mon 2, Day 1, Hr 0
# 24-Hour  | Day 1, Hr 24 | Day 2, Hr 0 | Day 31, Hr 24 | Mon 2, Day 1, Hr 0
#
# If, for example, a DateValue time series is read and then is written as a
# CARD file, then use a 1Day interval DateValue file and don't specify hour
# in the dates, OR, use an hourly file and specify hours in the date/times.
# Otherwise, the precision of the input data may not translate correctly.
#
# An example file is as follows and conforms to the following guidelines:
# * Only one time series per file.
# * The sequence number in data lines (field 3) has a maximum value of 9999.
# * Full months are included, with missing values as needed.
# * See the header below for more information.
# * Data are fixed format.
# * Comments in the file start with $ (these #-comments are for illustration
#   only)
# * Data lines are printed using the specified format.
# * Data lines have station, month, year (2 digit), count, data values.
#
$  IDENTIFIER=STATIONX       DESCRIPTION=RIVER Y BELOW Z
$  PERIOD OF RECORD=08/1978 THRU 11/1995
$  SYMBOL FOR MISSING DATA=-999.00   SYMBOL FOR ACCUMULATED DATA=-998.00
$  TYPE=SQIN   UNITS=CMS    DIMENSIONS=L3/T   DATA TIME INTERVAL= 6 HOURS
$  OUTPUT FORMAT=(3A4,2I2,I4,6F10.2)
DATACARD      SQIN L3/T CMS   6    26433
 8  1984 10   1984  6   F10.2
STATIONX     884   1     91.66     88.95     86.24     83.53     81.14     78.74
STATIONX     884   2     76.35     73.96     73.00     72.04     71.07     70.11
...
STATIONX     884  20    299.88    296.23    273.81    251.39    228.97    206.55
STATIONX     884  21    192.56    178.56    164.57    150.57
STATIONX     984   1    145.28    139.99    134.70    129.41    123.45    117.50
STATIONX     984   2    111.54    105.58    102.26     98.94     95.63     92.31
STATIONX     984   3    163.89    235.48    307.07    378.65   1032.13   1685.60
...
```

The following example illustrates the format of an NWSCard trace file.

```
$
$ OUTPUT FROM THE ESPADP TRACEFILE EXPORT COMMAND
$  VERSION 0.00
$  SEGID=TDAO3    TSID=TDAO3W    DTYPE=SQIN  IDT=24  UNITS=CFS
$  SIMFLAG=0  (CONDITIONAL SIMULATION)
$  CREATION TIME=04/20/2005 01:12:43.00
$  HISTORICAL RUN PERIOD= 4/18/2005 24 - 11/30/2005 24
$  NUMBER OF TRACES=44
$  MONTHS PER TRACE=0
$  INDIVIDUAL TRACE DATA FOLLOWS
$  ----------------------------
$
$  IDENTIFIER=TDAO3W   DESCRIPTION=ESP TRACE YEAR 1949
$  PERIOD OF RECORD =  4/1949 THRU 11/1949
$  SYMBOL FOR MISSING DATA = -999.00, SYMBOL FOR ACCUMULATED DATA = -998.00
$  TYPE=SQIN  UNITS=CFS  DIMENSIONS=L3/T  DATA TIME INTERVAL=24 HOURS
$  OUTPUT FORMAT=(A12,2I2,I4, 1F9.0)
$
DATACARD      SQIN L3/T CFS  24   TDAO3W      ESP TRACE YEAR 1949
 4  1949 11   1949  1   F9.0
TDAO3W       449   1     -999
TDAO3W       449   2     -999
TDAO3W       449   3     -999
TDAO3W       449   4     -999
TDAO3W       449   5     -999
TDAO3W       449   6     -999
TDAO3W       449   7     -999
TDAO3W       449   8     -999
… additional data records omitted…
TDAO3W      1149 242    99012
TDAO3W      1149 243   132273
TDAO3W      1149 244   141764
$
$  IDENTIFIER=TDAO3W   DESCRIPTION=ESP TRACE YEAR 1950
$  PERIOD OF RECORD =  4/1950 THRU 11/1950
$  SYMBOL FOR MISSING DATA = -999.00, SYMBOL FOR ACCUMULATED DATA = -998.00
$  TYPE=SQIN  UNITS=CFS  DIMENSIONS=L3/T  DATA TIME INTERVAL=24 HOURS
$  OUTPUT FORMAT=(A12,2I2,I4, 1F9.0)
$
DATACARD      SQIN L3/T CFS  24   TDAO3W      ESP TRACE YEAR 1950
 4  1950 11   1950  1    F9.0
TDAO3W       450 245     -999
TDAO3W       450 246     -999
TDAO3W       450 247     -999
TDAO3W       450 248     -999
TDAO3W       450 249     -999
TDAO3W       450 250     -999
TDAO3W       450 251     -999
TDAO3W       450 252     -999
TDAO3W       450 253     -999
TDAO3W       450 254     -999
TDAO3W       450 255     -999
TDAO3W       450 256     -999
TDAO3W       450 257     -999
TDAO3W       450 258     -999
TDAO3W       450 259     -999
TDAO3W       450 260     -999
TDAO3W       450 261     -999
TDAO3W       450 262   132719
TDAO3W       450 263   144539
…remainder of file omitted…
```

Note that the period indicated for each historical trace uses the historical period (i.e., the historical period used to generate an ESP forecast trace in NWSRFS). However, when reading the file for analysis or visualization, the traces should be aligned using the `HISTORICAL RUN PERIOD` information in the main header. The historical traces are treated as sequential data, regardless of the specific dates. In particular, differences due to leap year are ignored. Consequently, the sequence of values from the starting date in `HISTORICAL RUN PERIOD` is used to generate a trace.

NWSCard Files and Standard Time Series Properties

The standard time series identifier for NWSCard single time series files is of the form:

```
Location..DataType.Interval~NWSCard~PathToFile
```

The standard time series identifier for NWS Card trace files is of the form:

```
Location..DataType.Interval[Year]~NWSCard~PathToFile
```

Most standard time series properties can be properly assigned from an NWSCard file:

- The identifier is used for the location.
- For the time series identifier, the data source is left blank.
- The data type and interval are determined from the file header.
- For trace files, the first year in the historical trace is used for the sequence number (`[Year]` in the above example).
- `NWSCard` is used for the input type.
- The filename is used for the input name.

Limitations

NWSCard files have the following limitations:

- Only hourly data can be saved. Daily data can be saved by treating as 24-hour values but can cause confusion due to the hour 24/0 conversion issue. For example, hour 24 of day N is automatically converted to hour 0 of day N + 1 when reading the file.
- The identifier is used for the location part of the time series identifier and should not contain '.' characters because this conflicts with the time series identifier standards.
- NWSCard data are often used with the NWSRFS. The NWSRFS uses Zulu (GMT) time to store data in its database. However, historic and real-time data are often viewed as local time. Additionally, systems may be defined where, although the data interval is 6 hour, the start of the computational day may be shifted so that it starts at 8 AM local time (or local standard time). All of these factors can complicate managing time series data. Unfortunately, the NWSCard file format does not support specifying a time zone. Therefore, the end-user must understand how the times in the file relate to the current use of the data.
- Only one time series can be saved per file, unless the file contains traces. In other words, a card file is not suitable for storing multiple time series (e.g., time series at different locations).
- Years within the body of the file are not 4-digit (the header does use 4-digit years). To increase processing speed and avoid converting to 4-digit values, years within the body of the file are ignored (the sequence is used to store data). Therefore, errors in the dates in the data section may result in data values being placed in the wrong dates in the result.
- A record count is included in each record after the date. When writing large files, the record count should be limited to less than or equal to `9999` because of a 4-digit limit in the field.
- Trace files have often been noted to contain formatting that is different from the documented NWSCard standard. Files may need to be edited to allow software to read the files.

Appendix: NWSRFS ESP Trace Ensemble Input Type

2013-04-01

Note: The binary NWSRFS ESP ensemble format is a legacy format and may be phased out in the future as alternative formats are implemented.

Overview

The NWSRFS ESP trace ensemble file format stores one or more time series traces of consistent time interval (1 to 24 hours), data type, units, etc. The format has been developed by the National Weather Service (NWS) to support its Ensemble Streamflow Prediction (ESP) software. The ESP system is that portion of the National Weather Service River Forecast System (NWSRFS) that provides the capability of making long-range probabilistic forecasts of streamflow and streamflow related variables. ESP uses conceptual hydrologic/hydraulic models to forecast future streamflow using the current soil moisture, river, and reservoir conditions, with historical meteorological data as input. The ESP procedure assumes that meteorological events that occurred in the past are representative of events that may occur in the future. Each year of historical meteorological data is assumed to be a possible representation of the future and is used to simulate a streamflow trace. ESP produces a probabilistic forecast for each streamflow variable and period of interest. Although often applied to streamflow, the ESP approach and data format can be utilized for other time series data types.

The ESP system can produce two types of stream flow simulations: Conditional and Historic Simulations (CS and HS respectively). Conditional simulation files are produced when ESP repeatedly runs a streamflow simulation, using consistent starting conditions (e.g., the current system states) with a sequence of historical data as inputs. The resulting ensemble of time series traces can be analyzed to determine probabilistic qualities of potential streamflow forecasts. Although often applied to streamflow forecasts, the trace ensembles can also contain other data types (e.g., reservoir levels and releases). For the historical simulation, ESP simulates the entire historical period of record continuously without resetting the initial conditions. The analysis of the historical simulation can be compared to the analysis of the observed flows to assess any bias that might exist in the system. This information is often used to subjectively adjust the conditional simulation before a forecast is made. HS (historical) simulations are used to overcome the limited forecast window in the operational system. Currently the only ESP trace ensemble simulation files compatible with TSTool are conditional simulation files. Therefore, the information below applies to CS files only and do not apply to HS files.

ESP trace ensemble files, when used with NWSRFS, are typically named according to the convention:

Segment.TimeSeries.DataType.Interval.Scenario

For example:

GRCCH.GRCCH.QINE.06.HS
OWSN6.OWSSIMEL.SPEL.06.CS

The data type is specified using standard NWS data types, as found in the NWSRFS *DATAUNIT* system file. The interval is hours. The scenario is typically HS for a historical ESP run and CS for a conditional simulation.

The ESP trace ensemble file is a FORTRAN direct access binary file consisting of header records and time series data. The format of the file is conducive to the operation of ESP software. The file header section consists of one 496-byte record, stored as 124 words of 4-byte data, in the format below. Character data in the header record are stored with spaces padding them out to their maximum length. Consequently, a value of '`ETC`' in the Segment ID, which is an 8-byte character field, will be stored in the file as '`ETC     `'. Null characters are not used to pad strings.

The header record only has data through the 412th byte, beyond which the record consists of null values (byte value 00), which should be ignored. The specific format of the header record is described in the following table (data members refer to code variables, to aid developers).

ESP Trace Ensemble File Field Description

Record	Field Number (>= 1)	Field Type	Bytes	Field Starting Byte (in Record)	Data member (from code)	Data Description
1	1	F4	4	0	_format_ver	Format version. e.g., 1.01. There is no logic in existing ESPADP code to do anything differently based on the format version.
1	2	C8	8	4	_seg_id	Segment ID (e.g., '`OWSN6  `').
1	4	C8	8	12	_ts_id	TS ID (e.g., '`OWSSIMEL`').
1	6	C4	4	20	_ts_type	NWS data type (e.g., '`QINE`').
1	7	I4	4	24	_ts_dt	TS time interval in hours (e.g., 6).
1	8	I4	4	28	_simflag	Simulation flag, set to one of: 0 – SIMFLAG_CONDITIONAL 1 – SIMFLAG_HISTORICAL 2 – SIMFLAG_OBSERVED
1	9	C4	4	32	_ts_unit	NWS data units (e.g., '`CMS `')
1	10	I4	4	36	now[0]	Trace file creation month (1-12).
1	11	I4	4	30	now[1]	Trace file creation day (1-31).
1	12	I4	4	44	now[2]	Trace file creation year (4-digit).
1	13	I4	4	48	now[3]	Trace file creation date hours (1-24) and minutes (0-59). The hour is (int)(now[3]/100), the minutes are (int)(now[3]%100).
1	14	I4	4	52	now[4]	Trace file creation date seconds (0-59) and hundredths of seconds 0-99). The seconds are (int))(now[4]/100), the hundredths of seconds are (int)(now[4]%100).
1	15	I4	4	56	_im	Month of the first day of the first time series in the file (1-12).
1	16	I4	4	60	_iy	Year of the first day of the first time series in the file (4-digit).
1	17	I4	4	64	_idarun	Start if the traces as Julian day relative to December 31, 1899, in Zulu time.
1	18	I4	4	68	_ldarun	End of the traces as Julian Day format relative to December 31, 1899, in Zulu time.

Record	Field Number (>= 1)	Field Type	Bytes	Field Starting Byte (in Record)	Data member (from code)	Data Description
1	19	I4	4	72	_ijdlst	Carryover (states) day (1-31)
1	20	I4	4	76	_ihlst	Carryover (states) hour (1-24)
1	21	I4	4	80	_ljdlst	Last day of forecast in Julian Day format relative to December 31, 1899, in Zulu time.
1	22	I4	4	84	_lhlst	Last hour of forecast (1-24), in Zulu time.
1	23	I4	4	88	_n_traces	Number of traces in ESP file.
1	24	I4	4	92	_ncm	Number of conditional months.
1	25	I4	4	96	_nlstz	NWSRFS time zone relative to Zulu time. E.g. –6 = Mountain Standard Time.
1	26	I4	4	100	noutds	NWSRFS daylight savings flag. Either: 0 – No daylight savings time 1 – Daylight savings time
1	27	I4	4	104	_irec	Record number of first trace data, typically 2 (allows for future growth of the header information).
1	28	C4	4	108	_dim	Unit dimensions from NWS data units.
1	29	C4	4	112	_tscale	Time scale of code (INST, MEAN, ACCM)
1	30	C20	20	116	seg_desc	Segment description (from NWSRFS database).
1	35	F4	4	136	xlat	Latitude of segment in decimal degrees
1	36	F4	4	140	xlong	Longitude of segment in decimal degrees
1	37	C8	8	144	fg	Forecast group (from NWSRFS database).
1	39	C8	8	152	cg	Carryover group (from NWSRFS database).
1	41	C8	8	160	rfcname	RFC name (from NWSRFS database).
1	43	C80	80	168	_espfname	Trace file name, without path. (e.g., `GRCCH.GRCCH.QINE.06.HS`).
1	63	C80	80	248	prsf_string	This is the PRSF flag value that is used by the ESPADP NWSRFS application to display hours on the X-axis.
1	83	C80	80	328	_esptext	User comments
1	103	I4	4	408	adjcount	Adjustment counter 0-9. No further information can be determined about this data at this time.
1	104	C84	84	412	Dummy	Dummy contains 84 bytes of null (to be ignored)
2-EOF						Data records – see below.

Some ESP trace ensemble files contain additional records before data records, including analysis information. However, in most cases, the single header record is immediately followed by data records. The data section of the ESP file contains multiple records with 496 bytes of data in 124 4-byte floats. Records are stored in the file by trace in ascending date order. In other words, the first group of data records is for trace1, the second group of data records is for trace2, etc.

Each record contains 124 words of 4-byte floating-point numbers, regardless of the type of trace data interval (24-hour, 6-hour, hourly). Consequently, for 24-hour time series one record will contain:

124 values/1 value-per-day = 124 floating point numbers = 4 months of data per record

For hourly time series, one record will contain:

124 values/24 values-per-day = 5 days of data per record

If the trace period of record covers an entire year (365 or 366 days of data), the number of records for one trace of 24-hour time series will be:

365 days/124 days per record = 3 records per trace

For hourly time series, the number of records will be:

365 days/5 days per record = 73 records per trace

There are some qualifications on the format of the data records. These include:

- All the data values within a trace are sequential. This means that the May 1 data always follow the April 30 data.
- Traces consist of sequential data. February 28 is always followed by data values, whether taken from February 29 or March 1 of historical data.
- Each trace begins at the start of a new record. Therefore, a trace may not fill its final record to the 496th byte.
- If a trace does not fill the end of its final record, the end of the record will be filled with null bytes.

Due to these complexities, it is recommended that users not try to read data values from an ESP trace ensemble file – software like TSTool can read the trace file and translate the data into a format that can be more easily manipulated.

ESP Trace Ensemble Files and Standard Time Series Properties

The standard time series identifier for ESP trace ensemble files is of the form:

```
Location.NWSRFS.DataType.Interval[Trace]~NWSRFS_ESPTraceEnsemble~PathToFile
```

Similar to other input types (e.g., `DateValue`), ESP trace ensemble files support storing multiple time series in the same file. However, whereas other input types typically store time series for different stations or data type, ESP trace ensemble files store one or more traces of time series at the same station, where header information is the same for each time series, and each trace corresponds to a different historical input period (or year). To uniquely identify each trace, a time series sequence number (trace) is used in the time series identifier. This sequence number is the historical year for the start of each trace. For example, valid time series identifiers are:

```
GRCCH.NWSRFS.SQIN.6[1950]~NWSRFS_ESPTraceEnsemble~PathToFile
GRCCH.NWSRFS.SQIN.6[1951]~NWSRFS_ESPTraceEnsemble~PathToFile
…
GRCCH.NWSRFS.SQIN.6[2003]~NWSRFS_ESPTraceEnsemble~PathToFile
```

When reading a trace ensemble file:

- Each trace is converted to a separate time series, where the period is consistent with the overlapping period (e.g., the carryover date +1 interval forward through the end of a simulation).
- Each time series has redundant header information matching the ensemble header information.
- Each time series is given a sequence number (a time series property) corresponding to the starting year for the trace input data. For example, if the period that the traces are being generated is 2002-01-01 to 2002-12-31 and historical data starting in 1950 are being used, then the first trace time series will have a period 2002-01-01 to 2002-12-31 and a sequence number 1950. The second trace time series will have a period 2002-01-01 to 2002-12-31 and a sequence number 1951.
- Each time series is assigned an alias of the format `Segment_Trace_Seq`, where the segment identifier is taken from the file, "`_Trace_`" is a literal string, and sequence number is as described above.
- The location is set to the segment identifier.
- The data source is set to NWSRFS, because the NWSRFS system has generated the simulated time series.
- Data type is assigned based on the value in the file.
- Data units are assigned based on the value in the file.
- The interval is set based on the value in the file.
- The input type is set to `NWSRFS_ESPTraceEnsemble`.
- The input name is set to the ESP trace ensemble file name.

Some header information may be lost when converting between different time series input types (i.e., if the time series traces read from an ESP trace ensemble file are saved to a DateValue file. The following table illustrates how the internal ESP trace header information is stored in time series data after the trace file is read. In many cases, the specific trace ensemble header data are simply saved to the text comments in the time series.

Similarly, converting other data formats to an ESP trace file any header requires calculating some values and defaulting others. Although most values (such as _ncm, and _idarun) are computed and inserted into the trace header, some header information cannot be assigned without user input. TSTool does allow for some of these lost header fields to be inserted through options in the `WriteNwsrfsEspTraceEnsemble()` command.

ESP Trace Ensemble File Data and Associated Time Series Data

Data	Time Series Trace Data
_format_ver	Stored in TS comments and general time series property.
_seg_id	Stored in TS comments and general time series property.
_ts_id	Stored in TS comments and general time series property.
_ts_type	Stored in TS comments and general time series property.

Data	Time Series Trace Data
_ts_dt	Data Interval, and general time series property
_simflag	Stored in TS comments and general time series property.
_ts_unit	Data Units, Data Units Orig, and general time series property
now[0]	Stored in TS comments and general time series property.
now[1]	Stored in TS comments and general time series property.
now[2]	Stored in TS comments and general time series property.
now[3]	Stored in TS comments and general time series property.
now[4]	Stored in TS comments and general time series property.
_im	Stored in TS comments and general time series property.
_iy	Stored in TS comments and general time series property.
_idarun	Stored in TS comments and general time series property, used to calculate Date1 and Date1 Original: Convert Julian Hours from [((_idarun – 1) * 24) + (_ihlst – 1)] to DateTime
_ldarun	Stored in TS comments and general time series property, used to calculate Date2 and Date2 Original: Convert Julian Hours from [((_ldarun – 1) * 24) + (_lhlst – 1)] to DateTime
_ijdlst	Stored in TS comments.
_ihlst	Stored in TS comments and general time series property, used to calculate Date1 and Date1 Original: Convert Julian Hours from [((_idarun – 1) * 24) + (_ihlst – 1)] to DateTime
_ljdlst	Stored in TS comments.
_lhlst	Stored in TS comments and general time series property, used to calculate Date2 and Date2 Original: Convert Julian Hours from [((_ldarun – 1) * 24) + (_lhlst – 1)] to DateTime
_n_traces	Implicitly stored as the number of time series and general time series property.
_ncm	Stored in TS comments and general time series property.
_nlstz	Stored in TS comments and general time series property.
noutds	Stored in TS comments and general time series property.
_irec	Stored in TS comments and general time series property.
_dim	Stored in TS comments and general time series property.
_tscale	Stored in TS comments and general time series property.
seg_desc	Description, and general time series property
xlat	Stored in TS comments and general time series property.
xlong	Stored in TS comments and general time series property.
fg	Stored in TS comments and general time series property.
cg	Stored in TS comments and general time series property.
rfcname	Stored in TS comments and general time series property.
_espfname	TS Identifier and general time series property.
prsf_string	Stored in TS comments and general time series property.
_esptext	Stored in TS comments and general time series property.
adjcount	Stored in TS comments and general time series property.

Limitations

ESP trace ensemble files have the following limitations:

- The format is specific to hourly data.
- Multi-year trace periods are difficult to process.
- The binary, fixed record length format is not as flexible as text-based time series structures.

Appendix: NWSRFS FS5Files Input Type

2004-10-14 Acrobat Distiller

Overview

The NWSRFS FS5Files time series input type refers to time series read from the binary FORTRAN database files used with the National Weather Service River Forecast System (NWSRFS). This system typically runs on Linux workstations. Time series within the NWSRFS consist of the following:

- Preprocessor database time series. Raw observation data (e.g., in SHEF format – see the SHEF input type documentation) are processed by the **shefpars** and **shefpost** software with the results being stored in the preprocessor database as 1, 3, 6, 12, or 24-hour values.
- Processed database time series. Time series from the preprocessor database, and other input, are processed to create regular-interval 1, 3, 6, 12, or 24-hour time series in the processed database. The *DATATYPE* file indicates a PP if a preprocessor routine is the only component to write the processed database time series. For example, the MAP preprocessor uses point precipitation time series in the preprocessor database to compute mean areal precipitation (MAP) in the processed database. Processed database time series are referenced using "external" identifiers, which correspond to time series identifiers in the NWSRFS operations tables. Although NWSRFS models may use numerous time series, only those identified as "output" time series are written to the processed database and are therefore available after simulation. Output time series can have the same location identifier and data type, but must have different intervals.
- Calibration database time series. Historical time series, which are used in model calibration, are typically stored in NWS Card format files. These time series are not discussed here – see the **NWSCard Input Type Appendix**.
- Ensemble streamflow prediction (ESP) time series. ESP trace ensemble time series are produced by the ESP software, which uses current conditions in the forecast system, and historical input from the calibration database, to generate probabilistic forecasts. These time series are not discussed here – see the **NWSRFS ESP Trace Ensemble Input Type Appendix**.
- Other time series, including time-stamped gridded data are not considered in this documentation.

The remainder of this documentation focuses on the preprocessor and processed database time series.

Important comments about the NWSRFS FS5Files input type are:

- Because the NWSRFS is a real-time forecasting system that is used for large regions, recognition of time zone is important. Therefore, the NWSRFS database uses Z time (Zulu, or GMT) to store data. Consequently, the dates associated with time series returned from the NWSRFS FS5Files database will by default have a time zone of Z. Features of some applications may allow the time zone to be shifted to local time for viewing and analysis.
- Most NWSRFS string data are uppercase.
- The missing data value is `-999` in most cases.

NWSRFS FS5Files Time Series and Standard Time Series Properties

The standard time series identifier for NWSRFS time series read from the operational forecast system (OFS) is one of the forms shown below.

To use *ofs_files Apps_Defaults* variable to find files:

```
Location.NWSRFS.DataType[-PPDB].Interval[.OBS|FUT]~NWSRFS_FS5Files
```

To specify the directory where files are located:

```
Location.NWSRFS.DataType[-PPDB].Interval[.OBS|FUT]~NWSRFS_FS5Files~Directory
```

The NWSRFS Interactive Forecast Program (IFP) copies the operational forecast system FS5Files to a working directory. To specify a time series from the IFP files, use the second form above, where the directory is that of the IFP FS5Files. In the above examples, the scenario, if omitted, indicates that the entire time series is to be read (impacted by run-time start and end date/times). If the scenario is specified as OBS, then only observed data will be read. If the scenario is `FUT`, then only future (forecasted) data will be read. Because the preprocessor database does not contain an indicator for future data, the scenario only applies to processed database time series.

The MAP data type indicates mean areal precipitation and FMAP indicates future mean areal precipitation. All other data types are handled as “merged” data. Special considerations are as follows:

- To retrieve only future mean areal precipitation data, the MAP data type with the FUT scenario can be used or the FMAP data type can be used.
- To retrieve only observed mean areal precipitation, use the MAP data type with scenario OBS.
- To retrieve both observed and future mean areal precipitation, use the MAP data type and no scenario.
- For all other data types, use the normal time series identifier conventions described above.

The NWSRFS FS5Files input type is designed to handle both preprocessor and processed database time series, for the following reasons:

- To remove the need for users to know the difference between databases; therefore, they can concentrate on data types, not files.
- To allow future enhancements to NWSRFS, in which a common database is used, to be supported – in the future, this will allow time series identifiers to be specified for the legacy FS5Files and the new system (with a new input type, to be determined); consequently the same time series could be read from both databases, in order to compare results.

In order to allow data to be read from the preprocessor and processed database without distinction, the data types must be unique because software will use the data type to determine which database file needs to be read. The following data types occur in both the preprocessor and processed database and are used by the RRS preprocessor: `DQIN`, `DQME`, `PELV`, `QIN`, `QME`, `RQIM`, `RQIN`, `RQME`, `RQOT`, `RSTO`, `SNOG`, `SNWE`, `TWEL`, `ZELV`. To differentiate the preprocessor and processed database time series, preprocessor database time series have a sub-datatype of `PPDB`. To have consistency, all preprocessor database time series have the data-subtype of `PPDB`.

Specific conventions for NWSRFS FS5Files are as follows:

- The location is set to the station identifier (preprocessor database) or output time series identifier (processed database). To avoid confusion, it is recommended that NWSRFS systems be defined such that the time series identifier for the segment/operation is the same as the identifier on the processed database.
- The data source is set to `NWSRFS`, indicating that the time series are processed, created, or otherwise managed by NWSRFS. At this time, it is not possible to assign the data source to the originating provider (e.g., USGS, NOAA), in particular for the preprocessor database.
- The data type is set to the data type defined for the time series in the preprocessor or processed databases. Processed database time series are listed in the NWSRFS *DATATYPE* system file, for data types available in the `FDB` (forecast database).

  Preprocessor database data types are as follows (see NWSRFS documentation IX.4.2B-PDBINDEX note 1 and IX.3.4B-RPDDLY-2; RRS data types are apparently not documented). In the following table, a sub-datatype of PPDB is added to identify the data as being in the preprocessor database, simply to avoid confusion with the same data type that may be defined elsewhere in the system.

Preprocessor Database Data Type	Available Interval(s)	Description
`APIG-PPDB`	24Hour, Day?	Grid point API values.
`DQIN-PPDB`	?	Diversion instantaneous flow.
`DQME-PPDB`	?	Diversion mean flow.
`EA24-PPDB`	24Hour, Day?	Potential evaporation.
`MDR6-PPDB`	6Hour	6-hour manually digitized radar.
`PELV-PPDB`	?	Reservoir pool.
`PG24-PPDB`	24Hour, Day?	Grid point 24-hour precipitation.
`PP01-PPDB`	1Hour?	1-hour precipitation accumulations?
`PP03-PPDB`	3Hour?	3-hour precipitation accumulations?
`PP06-PPDB`	6Hour?	6-hour precipitation accumulations?
`PP24-PPDB`	24Hour	24-hour precipitation accumulations.
`PPSR-PPDB`	24Hour? Day?	Stranger precipitation reports.
`PPST-PPDB`	24Hour? Day?	Satellite precipitation estimates.
`PPVR-PPDB`	24Hour? Day?	Less than 24-hour precipitation, one value per day.
`QIN-PPDB`	?	River discharge.
`QME-PPDB`	?	River discharge, mean.
`RC24-PPDB`, `RP24-PPDB`, `RI24-PPDB`	?	Reservoir release, capacity, pool, inflow?
`RQIM-PPDB`	?	Reservoir inflow, mean.
`RQIN-PPDB`	?	Reservoir inflow.
`RQME-PPDB`	?	Reservoir outflow, mean.
`RQOT-PPDB`	?	Reservoir outflow.
`RSTO-PPDB`	?	Reservoir storage.

Preprocessor Database Data Type	Available Interval(s)	Description
SNOG-PPDB	?	Snow cover depth.
SNWE-PPDB	?	Observed snow water equivalent.
STG-PPDB	?	River stage.
TA01-PPDB	1Hour?	1-hour instantaneous air temperature?
TA03-PPDB	3Hour?	3-hour instantaneous air temperature?
TA06-PPDB	6Hour?	6-hour instantaneous air temperature?
TA24-PPDB	24Hour? Day?	Instantaneous air temperature (when recorded?).
TAVR-PPDB	24Hour? Day?	Less than 24-hour instantaneous temperature.
TD24-PPDB	?	?
TF24-PPDB	24Hour? Day?	Forecast maximum/minimum temperature.
TFMN-PPDB	24Hour? Day?	Forecast minimum temperature?
TFMX-PPDB	24Hour? Day?	Forecast maximum temperature?
TM24-PPDB	24Hour? Day?	24-hour maximum/minimum temperature.
TN24-PPDB	24Hour? Day?	Minimum temperature in the previous 24-hour period?
TWEL-PPDB	?	Tailwater stage.
TX24-PPDB	24Hour? Day?	Maximum temperature in the previous 24-hour period?
US24-PPDB	?	?
ZELV-PPDB	?	Freezing level.

- The interval is set to a multiple of the hour interval defined in database files (e.g., `6Hour`). Most time series will be regular, even those in the preprocessor database. The only irregular data will be in SHEF files (see the **SHEF Input Type Appendix**), and preprocessor database time series that have variable duration. Capabilities to read preprocessor database time series are being enabled over time and additional information about available intervals will be documented as features are enabled.
- The scenario is blank if reading observed and future data, `OBS` if only observed data are requested, and `FUT` if only future data are requested.
- The input type is `NWSRFS_FS5Files`.
- The input name is blank to use the current Apps_Defaults settings to locate the files, or a directory name, indicating the location of FS5Files. The latter allows FS5Files to be read directly by software outside of a full NWSRFS implementation (including on PCs if necessary).
- The data units are determined from the database files.

Limitations

The NWSRFS input type has the following limitations:

- The NWSRFS software is typically only installed at National Weather Service River Forecast Centers or large entities that have staff to support the system. Outside of these systems, the NWS Card, SHEF, and ESP trace ensemble formats may be distributed as data products.
- Useful information like the NWSRFS carryover group, forecast group, segment, and operation are not currently passed back with the time series data, other than in time series comments. This is due to the fact that the time series are read directly from the processed database rather than determine which system components use the time series. Consequently, applications like TSTool cannot display this information when time series are listed (other than in comments). Some information is indicated in the time series comments and may eventually be made available as a property list.
- Currently, software has been enabled to read only time series from the NWSRFS *processed* database, and the RRS data types from the preprocessor database. Discussion in this appendix also includes the remaining preprocessor database data types, in order to define conventions for time series identifiers, to allow for future enhancements.
- Full support for the `-998` missing data value (accumulated value) has not been added.

This page is intentionally blank.

Appendix: RCC ACIS Datastore

2013-06-16

Overview

The RCC ACIS datastore corresponds to the Regional Climate Center (RCC) Applied Climate Information System (ACIS), which stores real-time and historical climate data (see overview: http://data.rcc-acis.org/doc/VariableTable.html). ACIS data values are the "best available" and may include values merged from multiple sources. Data are accessed via a REST web service (see http://en.wikipedia.org/wiki/Representational_State_Transfer) application programmer interface (API). In addition to providing access to observations, the ACIS web services provide access to computed statistics such as historical maximum, minimum, and mean. TSTool does not provide commands to access all of the ACIS features; however, additional data types and features will be enabled over time.

The ACIS web services are described on the Documentation page at the following website:

http://data.rcc-acis.org

Support of ACIS in TSTool is being enhanced over time. The basic functionality is in place; however, additional effort is needed due to the large number of data types and data reductions that are offered. The initial effort has focused on basic data access for daily data and integration with TSTool; however, technical issues remain, as described below.

RCC ACIS and Standard Time Series Properties

The standard time series identifier for RCC ACIS time series is of the form:

```
LacationType:Location.DataSource.DataType.Interval~DataStoreName
```

More specifically, the identifier follows the following convention:

```
IDType:StaID.ACIS.Elem.Interval~DataStoreName
```

where identifier parts are described as follows:

- The location is indicated by station ID type and the station identifier. Because ACIS cross-references identifiers used in multiple data systems, the identifier cannot be guaranteed to be unique without knowing the identifier type. For the above identifier, the identifier type is selected in the following order (these abbreviations are taken from the ACIS REST API documentation): COOP, ICAO, NWSLI, FAA, WMO, WBAN, ThreadEx, AWDN, GHCN, CoCoRaHS, ACIS. This order may in the future be configurable as an RCC ACIS datastore configuration property, or command parameter.
- The data source is set to `ACIS`.
- The data type is set to the element name (`elem` in ACIS documentation).
- Interval is determined from the ACIS variable list (e.g., "daily" is translated to "Day") for consistency with TSTool conventions. Intervals that are not explicitly specified (e.g., "sub-hrly") or are not directly supported by TSTool (e.g., "weekly") are converted to "Irregular". TSTool currently only supports daily interval.
- `DataStoreName` is the user-defined datastore name from the configuration information.

- Data units are assigned based on the element name (version 1 assigns using the variable list).
- Data flags (see: *http://www.ncdc.noaa.gov/oa/climate/research/gdcn/GDCN_V1_0.doc*) are handled as follows:
  - Missing numerical values from ACIS are indicated by a data value of `M` and are converted to `NaN` internally, with an `M` flag on the data value.
  - Trace numerical values for precipitation and snow from ACIS are indicated by a data value of `T` and are converted to `0.0` internally, with a `T` flag on the data value.
  - No value with a flag is set to missing and the flag is set.
  - Value with flag is set to the value and the flag is set.
- Time series retrieved by the `MultiStn` and `StnMeta` requests sometime come back with empty `valid_daterange` – these time series are omitted from the time series list when displayed in TSTool.
- Time series with `valid_daterange` having dates with year `9999` have the year replaced with the current year.
- Data values with unexpected values and/or flags (e.g., `12.2A`, `S`) are handled as follows: if a numerical value is at the start of data, it is used to set the data value. Remaining text is used to set the data flag for the value.

Limitations

RCC ACIS data store limitations relative to TSTool standard features are as follows. Some of these issues may have been resolved with ACIS version 2 and will be reviewed as version 2 documentation is made available.

- Limited testing of combinations of query parameters has been implemented. Some technical issues have been identified, as documented below, and if resolved, may clarify other issues. Some of the variable types are only supported in limited fashion and may be eliminated or changed
- The ACIS API currently does not provide the variable list but may do so in the future – currently the variable list is hard-coded to core data types. A complete list of element types has not yet been published as of TSTool ACIS software feature development and consequently not all variables are available in TSTool.
- The ACIS general web services provide area information that can be presented as choices. However, choices have been implemented only for criteria where lists have been published and are reasonably short.
- Although a variety of query parameters can be specified (e.g., for location), not all of the information is returned as metadata in results. Consequently, when displaying lists of station-time series in TSTool, some attributes are omitted or would need to be determined from the query criteria (but this is not as robust). A requested design change is therefore to allow returned metadata for all query parameters (such as `climdiv`, `cwa`).
- Area parameters can include multiple values (e.g, `postal=CO,WY`); however, this is not documented in version 1 text (only in an example). Some clarification may be appropriate, in particular if this behavior only applies to postal.
- It appears that multiple location parameters can be specified to limit the query (e.g., county that intersects a climate division)? However, this feature is not documented. TSTool will allow multiple location criteria to be specified.
- The interpretation of start and end years equal to 9999 needs to be documented.
- Requesting a data type may return a full list of stations, even if that data type is not collected at the stations. The start and end dates for the data can be consulted to determine whether data

actually are available. A cleaner way to omit missing time series needs to be implemented. Perhaps this is addressed in the version 2 `StnMeta` request.

- No attempt has been made to use reduced (calculated) data available using JSON input. For example, reduced data include monthly values computed from daily values. Some anticipated issues are:
    - Would it be possible to implement the reductions via a URI rather than having to provide data behind the scenes? This would make the API more transparent for troubleshooting.
    - Need to understand how to map the reduced data time series identifiers to TSTool time series identifiers?
    - Does the reduction result in a time series or a statistic (table of 1+ numbers)? TSTool has some conventions for dealing with time series and statistics and additional evaluation is needed.
    - Should the API for reduced data fundamentally be the same as getting other time series (e.g., specify an interval of monthly instead of daily) or should the user need to specify additional input parameters to control the reduction. This depends on how acceptable the defaults are. It would seem that some standard reductions could simply result in more time series choices… but how does the major/variable name change?

Datastore Configuration File

A datastore is configured by enabling RCC ACIS data stores in the main *TSTool.cfg* configuration file, and creating a datastore configuration file for each datastore connection. Configurations are processed at software startup. An example of the TSTool configuration file is shown below. Multiple datastores can be defined using the `[DataStore:DataStoreName]` syntax. For ACIS, this would allow, for example, accessing systems at different RCC or different versions of the web services.

```
# Configuration file for TSTool

[TSTool]

RCCACISEnabled = true

# Startup datastores (note that data store name in config file takes precedence)

[DataStore:RCC-ACIS]

ConfigFile = "RCC-ACIS.cfg"
```

TSTool Configuration File with RCC ACIS Datastore Properties

Properties for each datastore are specified in an accompanying configuration file described below.

The following illustrates the RCC ACIS datastore configuration file format, which in this example is located in the same folder as the TSTool configuration file and configures a datastore named "RCC-ACIS".

```
# Configuration information for "RCC-ACIS" datastore.
# Properties are:
#
# The user will see the following when interacting with the datastore:
#
# Type - RccAcisDataStore (required as indicated)
# Name - datastore name used in applications, for example as the
#     input type information for time series identifiers (usually a short string)
# Description - datastore description for reports and user interfaces (short phrase)
# Enabled - whether the datastore is enabled (default=True)
#
# The following are specific to the RCC ACIS datastore:
#
# ServiceRootURI - web service root URI, including the server name and root path
#                  additional information will be appended to make specific requests

Type = "RccAcisDataStore"
Name = "RCC-ACIS"
Description = "RCC ACIS Web Services"
ServiceRootURI = http://data.rcc-acis.org
```

RCC ACIS Datastore Configuration File

Appendix: ReclamationHDB Datastore

2013-04-13

Overview

The ReclamationHDB datastore corresponds to Reclamation's HDB database (version 2), which stores real-time and historical data related to Reclamation operations. Time series in HDB are organized into "real" tables (observations) and "model" tables (from simulations or other external processes).

ReclamationHDB and Standard Time Series Properties

The standard time series identifier for ReclamationHDB time series is of the form:

```
Location.DataSource.DataType.Interval.Scenario~DataStoreName
```

More specifically, the identifier follows the following convention for "real" data:

```
Real:SiteCommonName.HDB.DataTypeCommonName.Interval~DataStoreName
```

and the following for "model" data:

```
      Model:SiteCommonName.HDB.DataTypeCommonName.Interval.ModelName-
ModelRunName-HydrologicIndicator-ModelRunDate~DataStoreName
```

where identifier parts are described as follows:

- A location prefix of `Real:` indicates that data from the `R_` data tables will be read. A location prefix of `Model:` indicates that data from the `M_` data tables will be read. The model identifier is stored in the `Scenario` time series identifier field, described below.
- `SiteCommonName` is the identifier for a location and is taken from the `HDB_SITE.SITE_COMMON_NAME` database column.
- A data source of `HDB` is used in all cases. The HDB design does include agency in several tables, but the information is not integrated enough to be used as part of the time series identifier:
    - The `HDB_AGEN` table lists agencies that are known in the database. `AGEN_NAME` is populated; however, `AGEN_ABBREV` often is null and cannot be relied upon for use in time series identifiers.
    - The `HDB_DATATYPE` table includes `AGEN_ID`, which relates to the `HDB_AGEN` table. However, many values are null and the relationship is not enforced.
    - The `R_BASE.AGEN_ID` value indicates the agency for which values are measured. The general guideline for HDB is that the last data in the table overwrites records. The final data record stored for a site that receives data from multiple agencies is controlled by the Reclamation data load processes. TSTool allows the agency to be specified when writing records, but there is currently no way to request records that were reported by an agency. The agency in this design is simply an annotation to explain the source of the data measurements.
- `DataTypeCommonName` is the time series data type and is taken from the `HDB_DATATYPE.DATATYPE_COMMON_NAME` column. The site common name and the data

type common name are used to look up the HDB_SITE_DATATYPE.SITE_DATATYPE_ID value, which is used internally to access real and model time series data.

- Interval is the data interval using TSTool standards (e.g., Hour, Day, Month, Year, Irregular). HDB does not include metadata for the time series interval. Instead, the data tables (e.g., R_DAY) must be queried to determine if a time series for a specific interval exists. This join occurs when interactively querying time series lists in TSTool to assist the user in forming valid time series identifiers (however, this does result in a slight performance penalty when listing time series).
- The time zone for hourly time series is set to the TIME_ZONE property from the HDB REF_DB_PARAMETER table.
- The scenario part of the time series identifier is omitted for "real" time series and for "model" time series contains dash-separated human-readable model metadata necessary to form a unique identifier, including:
    - ModelName (from HDB_MODEL.MODEL_NAME)
    - ModelRunName (from REF_MODEL_RUN.MODEL_RUN_NAME)
    - HydrologicIndicator (from REF_MODEL_RUN.HYDROLOGIC_INDICATOR)
    - ModelRunDate (from REF_MODEL_RUN.RUN_DATE), to the full second.

  This information is used to look up the REF_MODEL_RUN.MODEL_RUN_ID, which is used internally to access model time series data.
- DataStoreName is the user-defined datastore name from the configuration information.
- Missing numerical values are internally indicated with NaN.

Limitations

ReclamationHDB datastore limitations relative to TSTool standard features are as follows:

- Some database string values that are used in time series identifiers may contain periods, which conflict with the TSID conventions. These characters are converted to "?" in the TSID representation and are converted back to periods when querying the database. An alternative is to use the ReadReclamationHDB() command and assign an alias to time series, rather than relying on the TSID in commands.
- Although data flags are available with data records, they currently are not set in time series data during reads. A future enhancement may transfer the flags.
- HDB time series data tables have a START_DATE_TIME and END_DATE_TIME date/time for each data value, indicating the time span over which data are collected/averaged/summed. However, TSTool uses only a single date/time for each value, which is the same value shown in tables and used for graphing. The TSTool convention is that instantaneous values have a date/time corresponding to the date/time of the observation, and mean and accumulated values have a date/time corresponding to the interval end. For example, a 1-hour accumulated value recorded at YYYY-MM-DD HH would be for the hour ending HH. The following table illustrates the date/time conversions that are applied, using example data for site ADATUNCO flow volume (HOUR, DAY) and diversion volume (MONTH, YEAR) . In summary, for HOUR interval, the END_DATE_TIME is used and for others the START_DATE_TIME is used.

Data Interval	HDB START_DATE_TIME	HDB END_DATE_TIME	HDB POET date/time	TSTool date/time
HOUR	2007-04-04 13:00:00.0	2007-04-04 14:00:00.0	4/4/2007 2:00 PM	2007-04-04 14
DAY	2007-04-05 00:00:00.0	2007-04-06 00:00:00.0	4/5/2007	2007-04-05
MONTH	1978-01-01 00:00:00.0	1978-02-01 00:00:00.0	1/1/1978	1978-01

YEAR	1978-01-01 00:00:00.0	1979-01-01 00:00:00.0	1/1/1978	1978
IRREGULAR	1978-01-01 01:15:00.0	1978-01-01 01:15:00.0		1978-01-01 01:15

- TSTool supports 6Hour and other multiples of each base interval. However, HDB only stored 1Hour data. Therefore, 6Hour data may be represented in HDB as a value with 5 missing values for the other hours (or no hourly records for those hours). TSTool can be used to convert the 1Hour data to 6Hour or values can be filled with interpolation, etc.
- TSTool currently does not support reading HDB water year time series.
- Performance for reading hourly time series is slow. This appears to be in the conversion of Oracle date/times to internal representations and is being evaluated.

Datastore Configuration File

A datastore is configured by enabling ReclamationHDB datastores in the main *TSTool.cfg* configuration file, and creating a datastore configuration file for each datastore connection. Configurations are processed at software startup. An example of the TSTool configuration file is shown below. Multiple datastores can be defined using the `[DataStore:DataStoreName]` syntax. Properties for each datastore are specified in an accompanying configuration file described below.

```
# Configuration file for TSTool

[TSTool]

ReclamationHDBEnabled = true

# Startup datastores (note that datastore name in config file takes precedence)

[DataStore:HDB]

ConfigFile = "HDB.cfg"
```

TSTool Configuration File with ReclamationHDB Datastore Properties

The following illustrates the ReclamationHDB datastore configuration file format, which in this example is located in the same folder as the TSTool configuration file and configures the “HDB” datastore. Authentication for writing data to the datagbase is checked based on the account login and password.

```
# Configuration information for "HDB" datastore (connection).
#
# The user will see the following when interacting with the datastore:
#
# Type - ReclamationHDBDataStore (required as indicated)
# Name - database identifier for use in applications, for example as the
#     input type/name information for time series identifiers (usually a short string)
# Description - database description for reports and user interfaces (a sentence)
# Enabled - whether the datastore is enabled (default=True)
#
# The following are needed to make the low-level data connection:
#
# Type - the datastore type (must be specified exactly as shown)
# DatabaseEngine - the database software (default to Oracle since not specified)
# DatabaseServer - IP or string address for database server
# DatabaseName - database name used by the server
# Enabled - indicates whether datastore is enabled (True or False, default=True)
# KeepAliveSQL - SQL string to periodically run to keep database connection open
#   (useful when remotely accessing the system, default=no keep alive query)
# KeepAliveFrequency - number of seconds between execution of KeepAliveSQL
# SystemLogin - service account login
# SystemPassword - service account password
#
# Use the syntax Env:EnvVarName to retrieve values from the environment.
# Use the syntax SysProp:SysPropName to retrieve values from the JRE system environment.

Type = "ReclamationHDBDataStore"
Name = "HDB"
Description = "Reclamation Test Database"
DatabaseServer = "xxx"
DatabaseName = "xxx"
SystemLogin = "xxx" (for example app_user)
SystemPassword = "xxx"
```

ReclamationHDB Datastore Configuration File

Appendix: RiverWare Input Type

2008-09-03

Overview

RiverWare is a river and reservoir model developed by the Center for Advanced Decision Support for Water and Environmental Systems (CADSWES) at the University of Colorado. RiverWare uses data management interfaces (DMIs) to read time series data from various formats at run-time. The format described in this appendix is a standard time series format that is imported into the RiverWare data sets and can be output during runs. The example below shows the format of a file. Refer to the **RiverWare Data Management Interface** documentation for more information. Important comments about the file format are:

- The file is divided into a header section (top) and data section (bottom). Comments can occur anywhere in the file and are lines starting with `#`.
- The data period is defined by the `start_date` and `end_date` keywords. Date/times must include hours and minutes regardless of the date/time precision (the more precise information is ignored if not needed). For day, month, and year interval data, specify `24:00` at the end of the line.
- The data interval is defined by the `timestep` keyword and consists of an integer multiplier and a base interval string, separated by a space. Recognized intervals are `HOUR`, `DAY`, `WEEK`, `MONTH`, and `YEAR`.
- The data units are specified using the `units` keyword and are the units after the scale is applied. The `scale` keyword indicates a value that should be applied to the data values to result in the specified units. For example, a data value of `1.5` with units of `cfs` and a scale of `1000` will result of a value of `1500 cfs` in memory.
- Optional `set_units` and `set_scale` keywords may be used similar to `units` and `scale` to indicate the units and scale to be converted to when data are read. These properties can be written by TSTool's `writeRiverWare()` command but currently are not evaluated by TSTool when reading data.

The following example illustrates the format of a RiverWare file.

```
# Comments
start_date: 1903-01-01 06:00
end_date: 2001-12-31 24:00
timestep: 6 HOUR
scale: 1
set_scale: 1
units: ft
set_units: ft
1356.00
1356.00
1356.00
1356.00
1356.00
NaN
NaN
…
```

RiverWare Files and Standard Time Series Properties

The standard time series identifier for RiverWare time series files is of the form:

```
Location..DataType.Interval~RiverWare~PathToFile
```

RiverWare time series files contain limited information to assign to standard time series properties. The following assignments are made:

- The location part of the identifier is taken from the first part of the file name. It is assumed that the file name is of the form *ObjectName.SlotName*.
- The data source part of the time identifier is left blank.
- The data type is taken from the second part of the file name. It is assumed that the file name is of the form *ObjectName.SlotName*.
- The data interval is determined from the timestep property in the file.
- The data units are determined from the `units` property in the file. Currently the `set_units` property is not evaluated when reading data.
- The missing data value is assigned to `NaN` (not a number).
- The description is set to the location, a comma, followed by the data type.

Limitations

RiverWare files have the following limitations:

- RiverWare time series files require that units be spelled exactly as required by RiverWare, including upper/lower case. TSTool currently does not know about RiverWare units and therefore commands like `writeRiverWare()` must be used to verify that the units are correct for RiverWare.
- Only one time series can be saved in a file (other RiverWare files support multiple time series and may be supported in the future).
- RiverWare files do not store the data type or location information for the time series. These values are assigned from the file name, as described above. Relying on a file name convention may cause errors if the convention is not followed.
- Data lines do not contain the date. Therefore, it is difficult to use the files in other applications without first assigning dates for all the values.

Appendix: StateCU Input Type

2004-05-27, Acrobat Distiller

Overview

The StateCU time series input type corresponds to the file formats used by the State of Colorado's StateCU consumptive use model, including:

- Crop pattern time series file, yearly (**.cds*)
- Irrigation water requirement, formatted for StateMod (**.ddc*)
- Historical direct diversions, monthly (**.ddh*) and daily (**.ddd*) (in StateMod format but treated as StateCU input when used with a StateCU data set)
- Irrigation practice time series, yearly (**.ipy*)
- Irrigation water requirement (IWR, **.iwr*) and water supply limited (WSL, **.wsl*) output report files, including monthly and yearly values
- Frost dates, yearly
- Precipitation, monthly (in StateMod format)
- Temperature, monthly (in StateMod format)

See also the StateMod input type, which corresponds to StateMod input files, and the StateModB input type, which corresponds to StateMod binary output files.

See the StateCU Documentation for a complete description of StateCU input files. Refer to the **StateMod Documentation** for files that use the StateMod file format.

Important comments about the StateCU file formats are:

- Input files are divided into a header section (top) and data section (bottom). Comments can occur only at the top and are lines that begin with #.
- One or more time series can be stored in a file.
- Consistency in the order and number of the stations is required for each year of data, within the file.
- Other than comments, input files are fixed-format, compatible with FORTRAN applications. See the **StateCU Documentation** and **StateMod Documentation** for field specifications.
- Input file formats are optimized to allow a full year of data to be read for the entire data set. Reading a time series for a single location for the full period requires reading through the entire file.
- The precision of data values may be controlled by software, resulting in more or fewer fractional digits. This may lead to round-off differences when comparing raw data values output by the software.

StateCU Files and Standard Time Series Properties

The standard time series identifier for StateCU files is of the form:

`Location.StateCU..Month~StateCU~PathToDDCFile` (for *DDC* file)
`Location.StateCU.CropArea-AllCrops.Year~StateCU~PathToIWRReport` (for IWR report acreage)
`Location.StateCU.IWR.Month~StateCU~PathToIWRReport` (for IWR report monthly IWR)
`Location.StateCU.IWR.Year~StateCU~PathToIWRReport` (for IWR report yearly IWR)
`Location.StateCU.IWR_Depth.Year~StateCU~PathToIWRReport` (for IWR report IWR depth)
`Location.StateCU.CropArea-AllCrops.Year~StateCU~PathToWSLReport` (for WSL report acreage)
`Location.StateCU.WSL.Month~StateCU~PathToWSLReport` (for WSL report monthly WSL)
`Location.StateCU.WSL.Year~StateCU~PathToWSLReport` (for WSL report yearly WSL)
`Location.StateCU.WSL_Depth.Year~StateCU~PathToWSLReport` (for WSL report WSL depth)
`Location.StateCU.DataType.Interval~StateCU~PathToFile` (historical time series data files)

StateCU files contain limited header information (e.g., period of record but no data type). Time series properties are set using the following guidelines:

- For input files, the location part of the time series identifier is taken from the identifier field in the data records (from the first year of data). A change in the year indicates that all time series have been identified. For output files, the location is identified by lines that start with an underscore (this allows the StateCU interface to search for identifiers in output).
- The source part of the time series identifier is set to `StateCU` or blank.
- The data type may not be assigned because it is not defined in the file (e.g., temperature and precipitation time series). Currently no interpretation of the file name extension occurs. Some specific applications may set the data type, based on reading a StateCU data set response file (and therefore knowing the specific contents of the file).
- The data interval is assigned as `Day`, `Month`, or `Year` based on the file format (determined automatically).
- The scenario is typically not assigned.
- The input type part of the time series identifier is set to `StateCU`, indicating the file format. Software will use the interval and/or examine the file contents to verify whether the data are in daily or monthly format.
- The input name part of the time series identifier is set to the file name, either as the full path or a relative path to the working directory.
- The units are assigned to those indicated in the file header or based on the file type.
- The missing data value is assigned to `-999.0`.
- The description is set to the same value as the location. A verbose description can typically be determined by cross-referencing the identifier with another StateCU data file (e.g., CU Locations, **.str*).
- The period is set based on the header information.

Limitations

StateCU files have the following limitations:

- The formats of the files do not facilitate extracting one time series from the file. Software has been optimized to perform this within current constraints.
- Some time series properties are not explicitly included in StateCU files (e.g., data type). Therefore, general software like TSTool may not be able to provide default information. For example, a graph may show multiple time series with nearly the same legend text because more detailed information cannot be defaulted. The following has not been implemented but may be in the future: DDC file data type = IWR.
- Although the complete output report files contain all values needed to evaluate water balance, these values are not available in files that can be easily read as time series. Currently the verbose reports are not available for reading as part of the StateCU input type.

This page is intentionally blank.

Appendix: StateCUB Input Type

(StateCU Binary Output Files)

2008-08-27

Overview

The StateCUB time series input type corresponds to the file format used by the State of Colorado's StateCU consumptive use model, in particular the binary output file. These files contain important water balance information for every location in the model. The following table summarizes the contents of the binary files and corresponding text report files (all files can be large for large data sets).

Node Type	Monthly Binary File	Monthly Report File
CU locations	**.bd1*	**.dwb*

The following documentation describes the format of the BD1 binary file. See the **StateCU Documentation** for a complete description of StateCU output files. The following is a summary to explain how TSTool handles the format.

Unlike the StateMod binary files, the StateCU BD1 file does not have a fixed length record throughout the file. Different sections of the file have fixed length segments, depending on the contents of the section. The main sections and their format are described below, using terminology consistent with StateCU.

Header Records (File Metadata)

Field	Data	Type	Description
1	NumStr	integer	Number of structures (CU locations).
2	NumTS	integer	Number of time steps. The data period for the file is determined from first time series record plus (NumTS – 1).
3	NumStrVar	integer	Number of variables associated with each structure.
4	NumTSVar	integer	Number of time series variables (parameters) associated with each structure.
5	NumTSA	integer	Number of time steps in a year (1, 12, 365), although 12 (monthly data) is currently the only supported value.

Structure Header Records (Structure Metadata)

Repeat the following for NumStrVar:

Field	Data	Type	Description
1	StructureVarType	char(1)	The type of the structure variable: R for floating point number, I for integer, C for character string.
2	StructureVarLen	integer	Length of structure variable in bytes.
3	StructureVarName	char(24)	Structure variable name.
4	StructureVarInReport	integer	Whether the structure is in the DWB report file.
5	StructureVarReportHeader	char(60)	The report header to use for the structure variable.

Time Series Header Records (Time Series Metadata)

Repeat the following for NumTSVar:

Field	Data	Type	Description
1	TimeSeriesVarType	char(1)	The type of the time series variable: R for floating point number, I for integer, C for character string.
2	TimeSeriesVarLen	integer	Length of time series variable in bytes.
3	TimeSeriesVarName	char(24)	Time series variable name.
4	TimeSeriesVarInReport	integer	Whether the structure is in the DWB report file.
5	TimeSeriesVarUnits	char(10)	The units for the time series.

Structure Variable Data Records

Repeat the following for NumStr, and for each NumStrVar. The order of the variables is not fixed; however, the "Structure Index" variable contains a numeric identifier that is used to sort the structures to lookup structures when reading the time series.

Field	Data	Type	Description
1	StructureVarValue	As per metadata.	The values of the structure variables, including "Structure Index" (1+), "Structure ID", "Structure Name". Additional variables may be added later. The index is used to create a sorted list of structure identifiers and names for applications like TSTool.

Time Series Data Records

Repeat the following looping on NumStr, then NumTimeSteps, and then NumTimeSeriesVar. The order of time series variables is the same for all structures and throughout the entire file (variable "X" will always be in the same position in the inner loop). The order of the structures may not agree with the order of the metadata from above. The "Structure Index" variable in the time series records is used to map the time series to the structure identified in the metadata above.

Field	Data	Type	Description
1	TimeSeriesVarValue	As per metadata.	The values of the time series variables, including "Structure Index" (matching "Structure Index" from the structure data), "Year" (Calendar 4-digit), "Month Index" (1-12), and variable names for values at each time step. Additional variables may be added later.

A visual representation of data is as follows (note that Structure 1 is an internal looping representation and the actual structure is identified by the "Structure Index" variable for the time series):

```
Structure 1
     Timestep 1
          Variable 1
          …
          Variable NumStrVar
     Timestep 2
          Variable 1
          …
```

```
            Variable NumStrVar
        …
        TimeStep NumTS
    Structure 2
        Timestep 1
            Variable 1
            …
            Variable NumStrVar
        Timestep 2
            Variable 1
            …
            Variable NumStrVar
        …
        TimeStep NumTS
    …
    Structure NumStr
```

The order of the structures in the time series data block may not be the same as that in the header metadata due to the constraints of the StateCU model and how it writes each section during different phases of execution. Therefore, at initialization, the “Structure Index” variable value for each time series is read for the first timestep of each structure to determine the mapping of the structure in the time series data block with that in the main header.

Some time series variables are integers (e.g., the year and month) and some are characters (e.g., the month name and model flags). The integer variable “Year” has the same value for 12 monthly time steps and then increases by one. The variable “Month Index” repeats the values 1 – 12 through the period of the time series. Only floating point parameters are read by default. In the future, integer and character time series may be allowed or the character values may be translated to a lookup table of numbers.

StateCU BD1 Files and Standard Time Series Properties

The standard time series identifier for StateCUB binary time series is of the form:

```
Location.StateCU.DataType.Interval~StateCUB~PathToFile
```

Time series properties are set using the following guidelines:

- The location part of the time series identifier is taken from the structure identifier field in the data.
- The data source part of the time series identifier is set to `StateCU`, because StateCU has created the output time series.
- The data type is assigned as the variable (parameter) name described above – See the StateCU documentation for more information.
- The data interval is assigned as `Month`.
- The scenario is set to blank (not used).
- The input type is set to `StateCUB`.
- The input name is set to the name of the file.
- The units are determined from the time series variable metadata.
- The missing data value is assigned to `-999.0`.
- The description is set to the structure name.
- The period is set to the information in the first time series record incremented by the number of timesteps in the file (minus one). Current the file only contains calendar year data (January to December).

Limitations

StateCU binary files have the following limitations:

- The file does not contain a format version; therefore, it is difficult for software to handle changes in the file format. However, the current format is designed to allow for changing structure and time series parameters without changing the file format.
- The file does not contain header information indicating the source of the file (e.g., the creation date, user, directory, StateCU response file, command line). Therefore, it is difficult to know with certainty how a file was created.
- Leap years are not explicitly handled with 29 days during model calculations. Therefore there may be some loss of precision as data are processed through the model. Refer to the StateCU documentation for more information on how values are calculated.

Appendix: StateMod Input Type

2004-07-27, Acrobat Distiller

Overview

The StateMod time series input type corresponds to the file format used by the State of Colorado's StateMod model, including standard daily, monthly, average monthly (referred to as annual in the StateMod documentation) file formats. See also the StateModB input type, which corresponds to StateMod binary output files and the StateCU input type, which corresponds to the State of Colorado's StateCU consumptive use model.

The following example illustrates the format of the three main file formats. See the **StateMod Documentation** for a complete description of StateMod input files. Important comments about the file format are:

- The file is divided into a header section (top) and data section (bottom). Comments can occur only at the top and are lines that begin with #.
- One or more time series can be stored in a file.
- Consistency in the order and number of the stations is required for each year of data, within the file.
- Other than comments, the file is fixed-format, compatible with FORTRAN applications. See the **StateMod Documentation** for field specifications.
- The format is optimized to allow a full year of data to be read for the entire data set. Reading a time series for a single location for the full period requires reading through the entire file.
- In addition to the required values, a total/average value is accepted as the far-right value on each data line. This value may be ignored by applications (it can be computed from the data values on the line if necessary).
- The precision of data values may be controlled by software, resulting in more or fewer fractional digits. This may lead to round-off differences when comparing raw data values with the total/average in the optional end column.

```
# StateMod time series files can have 3 main forms (monthly, average monthly, daily) as
# described below.  The order of time series is important for
# some files (e.g., order of diversion time series should match order of
# diversion stations in .dds file); however, StateMod is being updated over
# time to remove this requirement).  Different StateMod input files have
# slight variations on the general format (e.g., the reservoir target file
# has two time series for each reservoir for minimum and maximum targets).
# Missing data are typically indicated by -999.
# The generic extension for StateMod time series files is .stm, although specific
# extensions are used in a StateMod data set.
#
# 1) This is an example of a StateMod monthly time series for water year data:
#
# Comments are lines at the top of the file starting with the # character.
# The header may contain software-generated comments about the time series.
# The remainder of the file is fixed format, with the first non-comment
# line being a header with the following elements (i5,1x,i4,5x,i5,1x,i4,a5,a5):
#
# Beginning month (1=Jan)
# Beginning year (4-digit)
# Ending month
# Ending year
# Data units (AF/M, ACFT, CFS or ""), where rates are for diversions and
#   flow, and volume is for reservoir contents.  Units are not used for
#   dimensionless data (like weight or percent).
```

```
# Year type (CYR=calendar, WYR=water, IYR=irrigation)
#
# Data lines then follow with:
# Year Station 12-monthly-values year-total/average (i4, 1x, a12, 12f8, f10)
# The year value is optional and is generally not read as input but is
# computed for output.  The year in data lines corresponds to the calendar type.
# An example follows:
   10/1926  -      9/1998 ACFT  WYR
1927 08236000      1229.8   892.6   922.3   737.9   555.4   922.3  7049.4 32263.6
31000.1 14541.0  5662.9  8326.7  104104.0
1927 08235250      -999.0  -999.0  -999.0  -999.0  -999.0  -999.0  -999.0  -999.0  -
999.0  -999.0  -999.0  -999.0       0.0
1927 08235700      -999.0  -999.0  -999.0  -999.0  -999.0  -999.0  -999.0  -999.0  -
999.0  -999.0  -999.0  -999.0       0.0
1927 08236500      1047.3   595.1   614.9   614.9   555.4  1900.2  6769.7 31226.2
20338.8 14777.1  9465.3  4476.8   92381.5
...

#
# 2) This is an example of a StateMod average monthly time series for water year data:
#
# The average monthly time series is a pattern of twelve monthly values
# that are applied for each year in the period.
# The format is exactly the same as a monthly time series; however, the
# years in the header should be set to zero and year and month are ignored in data rows
# and can therefore be blank.
#
# An example follows:
   10/   0  -      9/   0 ACFT  WYR
     08236000      1229.8   892.6   922.3   737.9   555.4   922.3  7049.4 32263.6
31000.1 14541.0  5662.9  8326.7  104104.0
     08235250      -999.0  -999.0  -999.0  -999.0  -999.0  -999.0  -999.0  -999.0  -
999.0  -999.0  -999.0  -999.0       0.0
     08235700      -999.0  -999.0  -999.0  -999.0  -999.0  -999.0  -999.0  -999.0  -
999.0  -999.0  -999.0  -999.0       0.0
     08236500      1047.3   595.1   614.9   614.9   555.4  1900.2  6769.7 31226.2
20338.8 14777.1  9465.3  4476.8   92381.5
...

#
# 3) This is an example of a StateMod daily time series for water year data:
#
# The daily time series is similar to the monthly time series except that
# a year and month are included on the data lines and 28, 30, or 31 daily
# data values can occur on each line (end values ignored, depending on month).
# The data format is (i4, i4, 1x, a12, 31f8, f8).  The month total/average
# is optional and is generally read as input but is computed for output.
# Regardless of the calendar type in the header, the year and month in data records use
# calendar year (month 1 = January).
#
# An example follows:
   10/1926  -      9/1998 ACFT  WYR
1926  10 08236000     -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -
999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -
999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00
0.00      0.00
...
1927   4 08236000       38.00   42.00   42.00   67.00   90.00   90.00  100.00  118.00
93.00   80.00   93.00   80.00   80.00   80.00   80.00   80.00   68.00   80.00   68.00
68.00   80.00   80.00  106.00  136.00  170.00  229.00  250.00  296.00  322.00  348.00
0.00    114.65
1927   4 08235250     -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -
999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -
999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00 -999.00
0.00      0.00
...
```

StateMod Files and Standard Time Series Properties

The standard time series identifier for StateMod files is of the format:

```
Location...Interval~StateMod~PathToFile
```

StateMod files contain limited header information. Time series properties are set using the following guidelines:

- The location part of the time series identifier is taken from the identifier field in the data records (from the first year of data). A change in the year indicates that all time series have been identified.
- The data source part of the time series identifier is set to `StateMod` or blank. In the past this information was used to indicate the input type (file format) in the time series identifier; however, the new input type notation has a specific field for the input type and therefore data source can be used more appropriately. In the future, it may be possible to pass along the original input source but this information cannot currently be saved in the StateMod file format.
- The data type is often not assigned because it is not defined in the file. Currently no interpretation of the file name extension occurs. Some specific applications (e.g., the StateMod GUI) may set the data type, based on reading a StateMod data set response file (and therefore knowing the specific contents of the file).
- The data interval is assigned as `Day` or `Month` based on the file format (determined automatically).
- The scenario is typically not assigned. Older software may use the scenario to store the file name; however, the new time series identifier notation stores the file name as the input name field (see below).
- The input type part of the time series identifier is set to `StateMod`, indicating the file format. Software will use the interval and/or examine the file contents to verify whether the data are in daily or monthly format.
- The input name part of the time series identifier is set to the file name, either as the full path or a relative path to the working directory.
- The units are assigned to those indicated in the file header.
- The missing data value is assigned to `-999.0`.
- The description is set to the same value as the location. A verbose description can typically be determined by cross-referencing the identifier with another StateMod data file (e.g., diversion stations).
- The period is set based on the header information.

Limitations

StateMod files have the following limitations:

- The format of the does not facilitate extracting one time series from the file. Software has been optimized to perform this within current constraints.
- Some time series properties are not explicitly included in StateMod files (e.g., data type). Therefore, general software like TSTool may not be able to provide default information. For example, a graph may show multiple time series with nearly the same legend text because more detailed information cannot be defaulted.
- If two time series for the same station are stored in the same file (e.g., reservoir maximum and minimum targets), there is no way to uniquely identify the two time series. The application or user must understand the file type and data organization. Some specific software (e.g., StateMod GUI) may be able to recognize the specific format.

This page is intentionally blank.

Appendix: StateModB Input Type

(StateMod Binary Output Files)

2004-07-27, Acrobat Distiller

Overview

The StateModB time series input type corresponds to the file format used by the State of Colorado's StateMod model, in particular the binary FORTRAN direct access output files. These files contain important water balance information for every node in the model network. The following table summarizes the contents of the binary files and corresponding text report files (all files can be large for large data sets):

Node Type	Monthly Binary File	Monthly Report File	Daily Binary File	Daily Report File
Diversion	**.b43*	**.xdd*	**.b49*	**.xdy*
Instream flow	**.b43*	**.xdd*	**.b49*	**.xdy*
Reservoir	**.b44*	**.xre*	**.b50*	**.xry*
Stream gage and Stream estimate	**.b43*	**.xdd*	**.b49*	**.xdy*
Well	**.b42*	**.xwe*	**.b65*	**.xwy*

The following documentation describes the format of the B43 binary file. Other files are similar. See the **StateMod Documentation** for a complete description of StateMod output files. Important comments about the file format are:

- The file is generated by StateMod as a direct access binary file with fixed-length records. The record length is 140 bytes.
- The file is divided into a header section (top) and data section (bottom).
- The format is optimized to allow a full year of data to be read for the entire data set. Efficiently reading a time series for a single location for the full period requires reading appropriate lines of the file using direct access. Because the file is binary and consistent for a given data set, file reads can be optimized.
- The data period and the calendar year type are consistent with the StateMod control file.
- All character strings are left justified and are padded with spaces. Therefore, software that reads the file should trim trailing spaces after reading the strings.
- River node identifiers in record 5 are included for all nodes in the network and data records (record 11) follow this order. Subsequent lists for various node types are a subset of the list in record 5 and have data items to reference the position in the river node list. Time series are queried using the identifiers in records 6+. However, the river node position is actually used to retrieve data in the file.

The B43 binary file contains the following records:

Record	Field	StateMod Variable	Type	Description
1	1	iystr0	integer	Beginning year of simulation, for year type in StateMod control file.
	2	iyend0	integer	Ending year of simulation, for year type in StateMod control file.
2	1	numsta	integer	Number of river nodes.
	2	numdiv	integer	Number of direct diversions.
	3	numifr	integer	Number of instream flows.
	4	numres	integer	Number of reservoirs.
	5	numown	integer	Number of reservoir owners.
	6	nrsact	integer	Number of active reservoirs.
	7	numrun	integer	Number of base flow nodes.
	8	numdivw	integer	Number of diversion structures with wells.
	9	numdxw	integer	Number of well only structures.
3	1	xmonam(14)	Each is char(4).	Month names corresponding to the calendar type for the simulation. This information is provided as a convenience for data processing. For example, if the year type is WYR (water year), xmonam(1) is 'OCT'. The 13th value is 'TOT' and the 14th value is 'AVE'.
4	1	mthday(12)	Each is integer.	Number of days per month, corresponding to the calendar type for the simulation. This information is provided as a convenience for data processing and to convert daily data values to monthly. For example, if the year type is WYR (water year), mthday(1) is 31 for October. The number of days in February is typically 28 and is used for all data processing, regardless of whether a year is a leap year.
5 **Repeat record for `numsta`**	1	j	integer	Counter for record type 5.
	2	cstaid(j)	char(12)	River node identifiers.
	3	stanam(j)	real(6)	River node names.
6 **Repeat record for `numdiv`**	1	j	integer	Counter for record type 6.
	2	cdivid(j)	char(12)	Diversion identifier.
	3	divnam(j)	real(6)	Diversion name.
	4	idvsta(j)	integer	River node position (1+) to allow cross-reference with river nodes.
7 **Repeat record for `numifr`**	1	j	integer	Counter for record type 7.
	2	cifrid(j)	char(12)	Instream flow identifier.
	3	xfrnam(j)	real(6)	Instream flow name.
	4	ifrsta(j)	integer	River node position (1+) to allow cross-reference with river nodes.
8 **Repeat record for `numres`**	1	j	integer	Counter for record type 8.
	2	cresid(j)	char(12)	Reservoir identifier.
	3	resnam(j)	real(6)	Reservoir name.
	4	irssta	integer	River node position (1+) to allow cross-reference with river nodes.

Record	Field	StateMod Variable	Type	Description	
9 **Repeat record for `numrun`**	1	j	integer	Counter for record type 9.	
	2	crunid(j)	char(12)	Base flow node identifier.	
	3	runnam(j)	real(6)	Base flow node name.	
	4	irusta(j)	integer	River node position (1+) to allow cross-reference with river nodes.	
10 **Repeat record for `numdivw`**	1	j	integer	Counter for record type 10.	
	2	cdividw(j)	char(12)	Well identifier.	
	3	divnamw(j)	real(6)	Well name.	
	4	idvstw(j)	integer	River node position (1+) to allow cross-reference with river nodes.	
11 **Repeat record for every river node `numsta`, for every month of the simulation.** See the StateMod documentation for a full description of parameters. Parameters are grouped as shown in the **.xdd* file.	1	dat(1)	real	Demand	Total_Demand
	2	dat(2)	real	Demand	CU_Demand
	3	dat(3)	real	Water Supply	From_River_By_Priority
	4	dat(4)	real	Water Supply	From_River_By_Storage
	5	dat(5)	real	Water Supply	From_River_By_Exchange
	6	dat(6)	real	Water Supply	From_Well
	7	dat(7)	real	Water Supply	From_Carrier_By_Priority
	8	dat(8)	real	Water Supply	From_Carrier_By_Storage
	9	dat(9)	real	Water Supply	Carried_Water
	10	dat(10)	real	Water Supply	From_Soil
	11	dat(11)	real	Water Supply	Total_Supply
	12	dat(12)	real	Shortage	Total_Short
	13	dat(13)	real	Shortage	CU_Short
	14	dat(14)	real	Water Use	Consumptive_Use
	15	dat(15)	real	Water Use	To_Soil
	16	dat(16)	real	Water Use	Total_Return
	17	dat(17)	real	Water Use	Loss
	18	dat(18)	real	Station In/Out	Upstream_Inflow
	19	dat(19)	real	Station In/Out	Reach_Gain
	20	dat(20)	real	Station In/Out	Return_Flow
	21	dat(21)	real	Station In/Out	Well_Depletion
	22	dat(22)	real	Station In/Out	To_From_GW_Storage
	23	dat(23)	real	Station Balance	River_Inflow
	24	dat(24)	real	Station Balance	River_Divert
	25	dat(25)	real	Station Balance	River_By_Well
	26	dat(26)	real	Station Balance	River_Outflow
	27	dat(27)	real	Available Flow	Available_Flow
	28	dat(28)	real	Structure type (Na): • < 0 = Baseflow node (e.g., -10001 indicates a diversion that is a baseflow node). • 0 = Well only. • 1-5000 = Diversion • 5001 – 7500 = Instream flow • 7501 – 10000 = Reservoir	
	29	dat(29)	real	Number of structures at this node (typically 1).	

StateMod B43 Files and Standard Time Series Properties

The standard time series identifier for StateMod binary time series is of the form:

```
Location.StateMod.DataType.Interval~StateModB~PathToFile
```

Time series properties are set using the following guidelines:

- The location part of the time series identifier is taken from the identifier field in the data. The identifier for the specific node type (e.g., diversion) is used, not the river node identifier. The river node identifier is often the same as for the specific node type, but this is not a requirement within StateMod.
- The data source part of the time series identifier is set to `StateMod`, because StateMod has created the output time series.
- The data type is assigned as the parameter name (see record 11 above, without using the group).
- The data interval is assigned as `Month` or `Day`, depending on the file extension.
- The scenario is set to `blank`.
- The input type is set to `StateModB`.
- The input name is set to the name of the file.
- The units for daily data are assigned as `CFS`. The units for monthly data in the files are average CFS for the month and are converted to `ACFT`, assuming a constant number of days per month, as read from record 4. February normally has 28 days per month in the header and therefore leap years have one fewer days than actual.
- The missing data value is assigned to `-999.0`.
- The description is set to the node name.
- The period is set based on the header information in record 1 (for the year) and record 3 (to determine the start and end months, based on the calendar type).

Limitations

StateMod binary files have the following limitations:

- The file does not contain a format version; therefore, it is difficult for software to handle changes in the file format.
- The file does not contain header information indicating the source of the file (e.g., the creation date, user, directory, StateMod response file, command line). Therefore, it is difficult to know with certainty how a file was created.
- Leap years are not explicitly handled with 29 days.
- Baseflow nodes in record 9 may have the same identifier as other nodes because any node can be a baseflow node. This can be confusing since software may list the node in more than one list. The software that reads the file filters out duplicate time series identifiers to try to resolve this problem.
- This documentation is limited in that it presents the file format only for the **.b43* file. Additional documentation may be added in the future.

Appendix: UsgsNwisDaily – USGS NWIS Daily Value Web Service Data Store

2012-03-22

Overview

The UsgsNwisDaily data store corresponds to the United States Geological Survey (USGS) National Water Information System (NWIS) Daily Values web service, as described on the following page:

http://waterservices.usgs.gov/rest/DV-Service.html

The following online query page is available for interactive queries:

http://waterservices.usgs.gov/rest/DV-Test-Tool.html

Although the USGS NWIS services are largely compatible with TSTool conventions, there are a number of limitations, which are discussed below.

The NWIS web service allows data to be retrieved as JavaScript Object Notation (JSON), tab-delimited (RDB), or WaterML 1.1. Each of these formats has features and limitations that are discussed in more detail in other documentation. The UsgsNwisDaily data store is used with the `ReadUsgsNwisDaily()` command and the files saved by the command are used with the `ReadUsgsNwisRdb()` and `ReadWaterML()` commands.

USGS NWIS and Standard Time Series Properties

The standard time series identifier for USGS NWIS time series in TSTool is of the form:

```
Location.DataSource.DataType.Interval~DataStoreName
```

More specifically, the identifier adheres to the following convention:

```
SiteNum.AgencyCode.ParameterCode-StatisticCode.Day~DataStoreName
```

where identifier parts are described as follows:

- The `SiteNum` corresponds to a USGS site. The NWIS Mapper (http://wdr.water.usgs.gov/nwisgmap/), USGS Site Inventory (http://waterdata.usgs.gov/nwis/inventory) and other published information from the USGS can be used to determine site numbers.
- `AgencyCode` can be blank for USGS sites or is specified from the following list (for example, use `USGS` for the code): http://nwis.waterdata.usgs.gov/nwis/help/?read_file=nwis_agency_codes&format=table
- `ParameterCode` is taken from the parameter list available from the following list: http://nwis.waterdata.usgs.gov/usa/nwis/pmcodes
  Ideally a parameter name could be used; however, the list of parameters is extensive, descriptions may change, and special characters like the period are used in descriptions and would interfere with the TSID convention. Consequently, a concise unique parameter name is not readily

apparent, and the initial implementation uses the numerical parameter code. In the future, the text name may be allowed and a prefix may be used to indicate whether a code or name is used.

- `StatisticCode` and `StatisticName` are taken from the list of supported statistics: http://waterservices.usgs.gov/rest/USGS-DV-Service.html
  Currently TSTool uses `StatisticCode` rather than `StatisticName`. In the future, the text name may be allowed and a prefix may be used to indicate whether a code or name is used
- Interval defaults to `Day`.
- `DataStoreName` is the user-defined data store name from the configuration information.
- Data units are taken from the following:
  - WaterML `unitCode` in `variable` element
- Missing numerical values are internally represented as `NaN` and are assigned to any date/times in the period that do not have values.
  - WaterML `noDataValue` in `variable` element is checked and matching data values are handled as missing
- Data value flags, if encountered, are retained in the time series. However, because the USGS uses "A" for approved, it may be necessary to ignore this flag so that other flags stand out more when visualized.
- Data value qualifiers definitions are saved with time series and are available to use in time series visualization as flag definitions

Limitations

USGS NWIS data store limitations relative to TSTool standard features are as follows:

- Interpretation of USGS data is limited by WaterML limitations, as follows:
  - WaterML files from NWIS do not indicate the interval of the data. Day can be assumed for the daily values web service; however, trying to read the WaterML file later will require that the interval is specified.
  - Some of the descriptions contain units, which may lead to confusion if time series are processed into different units.
- The USGS web services does not allow for all historical data to be returned. Specifying no period returns only the most recent value. Start and end dates must be specified to retrieve a longer period; however, there is no way to request the entire available period. Consequently, users must request a period of interest for their analysis and the browsing features of TSTool cannot list the available period (because doing so would require querying all data, which would be very slow).

Data Store Configuration File

A data store is configured by enabling UsgsNwisDaily data stores in the main *TSTool.cfg* configuration file, and creating a data store configuration file for each data store. Configurations are processed at software startup. An example of the TSTool configuration file is shown below. Multiple data stores can be defined using the `[DataStore:DataStoreName]` syntax. For NWIS, this would allow, for example, accessing different versions of the web services.

```
# Configuration file for TSTool

[TSTool]

UsgsNwisDailyEnabled = true

# Startup data stores (note that data store name in config file takes precedence)
```

```
[DataStore:UsgsNwisDaily]

ConfigFile = "UsgsNwisDaily.cfg"
```

TSTool Configuration File with UsgsNwisDaily Data Store Properties

Properties for each data store are specified in an accompanying data store configuration file (see below), which in the following example is located in the same folder as the TSTool configuration file and configures a data store named "UsgsNwisDaily".

```
# Configuration information for "UsgsNwisDaily" data store.
# Properties are:
#
# The user will see the following when interacting with the data store:
#
# Type – UsgsNwisDailyDataStore (required as indicated)
# Name - data store identifier used in applications, for example as the
#     input type information for time series identifiers (usually a short string)
# Description - data store description for reports and user interfaces (short phrase)
# Enabled – whether the data store is enabled (default=True)
#
# The following are specific to the USGS NWIS daily data store:
#
# ServiceRootURI - web service root URI, including the server name and root path
# ServiceAPIDocumentationURI - web service API documentation URI, describing
#     the syntax, input, and output
# ServiceOnlineURI - web service interactive page to query data, typically
#     "drill down" or form based

Type = "UsgsNwisDailyDataStore"
Name = "UsgsNwisDaily"
Description = "USGS NWIS Daily Value Web Service"
Enabled = True
ServiceRootURI = "http://waterservices.usgs.gov/nwis/dv"
ServiceAPIDocumentationURI = "http://waterservices.usgs.gov/rest/DV-Service.html"
ServiceOnlineURI = "http://waterservices.usgs.gov/rest/DV-Test-Tool.html"
```

UsgsNwisDaily Data Store Configuration File

This page is intentionally blank.

Appendix: UsgsNwisGroundwater – USGS NWIS Groundwater Web Service Datastore

2012-10-22

Overview

The UsgsNwisGroundwater datastore corresponds to the United States Geological Survey (USGS) National Water Information System (NWIS) groundwater web service, as described on the following page:

http://waterservices.usgs.gov/rest/GW-Levels-Service.html

The following online query page is available for interactive queries:

http://waterservices.usgs.gov/rest/GW-Levels-Test-Tool.html

Although the USGS NWIS services are largely compatible with TSTool conventions, there are a number of limitations, which are discussed below.

The NWIS groundwater web service allows data to be retrieved as JavaScript Object Notation (JSON), tab-delimited (RDB), or WaterML 1.1 (with USGS extenstions, called WaterML 1.2). Each of these formats has features and limitations that are discussed in more detail in other documentation. The UsgsNwisGroundwater datastore is used with the `ReadUsgsNwisGroundwater()` command and the data files saved by the command are used with the `ReadUsgsNwisRdb()` and `ReadWaterML()` commands.

USGS NWIS and Standard Time Series Properties

The standard time series identifier for USGS NWIS time series in TSTool is of the form:

```
Location.DataSource.DataType.Interval~DataStoreName
```

More specifically, the identifier adheres to the following convention:

```
SiteNum.AgencyCode.ParameterCode.Day~DataStoreName
```

where identifier parts are described as follows:

- The `SiteNum` corresponds to a USGS site. The NWIS Mapper (http://wdr.water.usgs.gov/nwisgmap/), USGS Site Inventory (http://waterdata.usgs.gov/nwis/inventory) and other published information from the USGS can be used to determine site numbers.
- `AgencyCode` can be blank for USGS sites or is specified from the following list (for example, use `USGS` for the code): http://nwis.waterdata.usgs.gov/nwis/help/?read_file=nwis_agency_codes&format=table
- `ParameterCode` is taken from the parameter list available from the following list: http://nwis.waterdata.usgs.gov/usa/nwis/pmcodes

Ideally a parameter name could be used; however, the list of parameters is extensive, descriptions may change, and special characters like the period are used in descriptions and would interfere with the TSID convention. Consequently, a concise unique parameter name is not readily apparent, and the initial implementation uses the numerical parameter code. In the future, the text name may be allowed and a prefix may be used to indicate whether a code or name is used. The groundwater web service is constrained to the following parameter codes:

- 72019 - Depth to water level, feet below land surface
- 72020 - Elevation above NGVD 1929, feet,
- 62610 - Groundwater level above NGVD 1929, feet
- 62611 - Groundwater level above NAVD 1988, feet

- Interval defaults to `Day`. Finer resolution date/times are truncated to the day. In the future the datastore may be enhanced to allow the option to return time series in other intervals, including irregular.
- `DataStoreName` is the user-defined data store name from the configuration information.
- Data units are taken from the following:
  - WaterML `unitCode` in `variable` element
- Missing numerical values are internally represented as `NaN` and are assigned to any date/times in the period that do not have values.
  - WaterML `noDataValue` in `variable` element is checked and matching data values are handled as missing
- Data value flags, if encountered, are retained in the time series. However, because the USGS uses “A” for approved, it may be necessary to ignore this flag so that other flags stand out more when visualized.
- Data value `qualifiers` definitions are saved with time series and are available to use in time series visualization as flag definitions.
- Properties are extracted from the XML and saved as time series properties. For example, the time series property `siteName` is set from the following XML:

```
<ns1:sourceInfo xsi:type="ns1:SiteInfoType"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <ns1:siteName>CACHE LA POUDRE RIV AT MO OF CN, NR FT COLLINS,
CO</ns1:siteName>
```

Limitations

USGS NWIS data store limitations relative to TSTool standard features are as follows:

- Interpretation of USGS data is limited by WaterML limitations, as follows:
  - WaterML files from NWIS do not indicate the interval of the data. Day is assumed for the groundwater values web service; however, trying to read the WaterML file later will require that the interval is specified.
  - Some of the descriptions contain units, which may lead to confusion if time series are processed into different units.
- The USGS web service does not allow for all historical data to be returned. Specifying no period returns only the most recent value. Start and end dates must be specified to retrieve a longer period; however, there is no way to request the entire available period. Consequently, users must request a period of interest for their analysis and the browsing features of TSTool cannot list the available period (because doing so would require querying all data, which would be very slow). The USGS NWIS site web service may be utilized in the future to improve data browsing.

Datastore Configuration File

A data store is configured by enabling UsgsNwisGroundwater data stores in the main *TSTool.cfg* configuration file, and creating a datastore configuration file for each datastore. Configurations are processed at software startup. An example of the TSTool configuration file is shown below. Multiple data stores can be defined using the `[DataStore:DataStoreName]` syntax. For NWIS, this would allow, for example, accessing different versions of the web services.

```
# Configuration file for TSTool

[TSTool]

UsgsNwisGroundwaterEnabled = true

# Startup datastores (note that datastore name in config file takes precedence)

[DataStore:UsgsNwisGroundwater]

ConfigFile = "UsgsNwisGroundwater.cfg"
```

TSTool Configuration File with UsgsNwisGroundwater Data Store Properties

Properties for each datastore are specified in an accompanying data store configuration file (see below), which in the following example is located in the same folder as the TSTool configuration file and configures a data store named “UsgsNwisGroundwater”.

```
# Configuration information for "UsgsNwisGroundwater" data store,
# for historical, manually recorded groundwater levels.
#
# Properties are:
#
# The user will see the following when interacting with the data store:
#
# Type - UsgsNwisGroundwaterDataStore (required as indicated)
# Name - data store identifier used in applications, for example as the
#     input type information for time series identifiers (usually a short string)
# Description - data store description for reports and user interfaces (short phrase)
# Enabled - whether the data store is enabled (default=True)
#
# The following are specific to the USGS NWIS groundwater data store:
#
# ServiceRootURI - web service root URI, including the server name and root path
# ServiceAPIDocumentationURI - web service API documentation URI, describing
#     the syntax, input, and output
# ServiceOnlineURI - web service interactive page to query data, typically
#     "drill down" or form based

Type = "UsgsNwisGroundwaterDataStore"
Name = "UsgsNwisGroundwater"
Description = "USGS NWIS Groundwater Values (historical manually recorded values) Web
Service"
Enabled = True
ServiceRootURI = "http://waterservices.usgs.gov/nwis/gwlevels"
ServiceAPIDocumentationURI = "http://waterservices.usgs.gov/rest/GW-Levels-Service.html"
ServiceOnlineURI = "http://waterservices.usgs.gov/rest/GW-Levels-Test-Tool.html"
```

UsgsNwisGroundwater Datastore Configuration File

This page is intentionally blank.

Appendix: UsgsNwisInstantaneous – USGS NWIS Instantaneous Values Web Service Datastore

2012-10-30

Overview

The UsgsNwisInstantaneous datastore corresponds to the United States Geological Survey (USGS) National Water Information System (NWIS) instantaneous values web service, as described on the following page:

http://waterservices.usgs.gov/rest/IV-Service.html

The following online query page is available for interactive queries:

http://waterservices.usgs.gov/rest/IV-Test-Tool.html

Although the USGS NWIS services are largely compatible with TSTool conventions, there are a number of limitations, which are discussed below.

The NWIS instantaneous web service allows data to be retrieved as JavaScript Object Notation (JSON), tab-delimited (RDB), or WaterML 1.1 (with USGS extensions, called WaterML 1.2). Each of these formats has features and limitations that are discussed in more detail in other documentation. The UsgsNwisInstantaneous datastore is used with the `ReadUsgsNwisInstantaneous()` command and the data files saved by the command are used with the `ReadUsgsNwisRdb()` and `ReadWaterML()` commands.

USGS NWIS Instantaneous and Standard Time Series Properties

The standard time series identifier for USGS NWIS instantaneous time series in TSTool is of the form:

```
Location.DataSource.DataType.Interval~DataStoreName
```

More specifically, the identifier adheres to the following convention:

```
SiteNum.AgencyCode.ParameterCode.15Min~DataStoreName
```

where identifier parts are described as follows:

- The `SiteNum` corresponds to a USGS site. The NWIS Mapper (http://wdr.water.usgs.gov/nwisgmap/), USGS Site Inventory (http://waterdata.usgs.gov/nwis/inventory) and other published information from the USGS can be used to determine site numbers.
- `AgencyCode` can be blank for USGS sites or is specified from the following list (for example, use `USGS` for the code): http://nwis.waterdata.usgs.gov/nwis/help/?read_file=nwis_agency_codes&format=table

- `ParameterCode` is taken from the parameter list available from the following list: http://nwis.waterdata.usgs.gov/usa/nwis/pmcodes
  Ideally a parameter name could be used; however, the list of parameters is extensive, descriptions may change, and special characters like the period are used in descriptions and would interfere with the TSID convention. Consequently, a concise unique parameter name is not readily apparent, and the initial implementation uses the numerical parameter code. In the future, the text name may be allowed and a prefix may be used to indicate whether a code or name is used.
- Interval defaults to `15Min`. Date/times that do not align with 15-minute offsets (0, 15, 30, 45) are truncated to the nearest 15 minutes and warnings will be generated. In the future the datastore may be enhanced to allow the option to return time series in other intervals, including irregular.
- `DataStoreName` is the user-defined data store name from the configuration information.
- Data units are taken from the following:
  - WaterML `unitCode` in `variable` element
- Missing numerical values are internally represented as `NaN` and are assigned to any date/times in the period that do not have values.
  - WaterML `noDataValue` in `variable` element is checked and matching data values are handled as missing
- Data value flags, if encountered, are retained in the time series. However, because the USGS uses “A” for approved, it may be necessary to ignore this flag so that other flags stand out more when visualized.
- Data value `qualifiers` definitions are saved with time series and are available to use in time series visualization as flag definitions.
- Properties are extracted from the XML and saved as time series properties. For example, the time series property `siteName` is set from the following XML:

```
<ns1:sourceInfo xsi:type="ns1:SiteInfoType"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <ns1:siteName>CACHE LA POUDRE RIV AT MO OF CN, NR FT COLLINS,
CO</ns1:siteName>
```

Limitations

USGS NWIS data store limitations relative to TSTool standard features are as follows:

- Interpretation of USGS data is limited by WaterML limitations, as follows:
  - WaterML files from NWIS do not indicate the interval of the data. A 15-minute interval is assumed for the instantaneous values web service; however, trying to read the WaterML file later with `ReadWaterML()` will require that the interval is specified.
  - Some of the descriptions contain units, which may lead to confusion if time series are processed into different units.
- The USGS web service does not allow for all archived instantaneous values data to be returned. Specifying no period returns only the most recent value. Start and end dates must be specified to retrieve a longer period; however, there is no way to request the entire available period. Consequently, users must request a period of interest for their analysis and the browsing features of TSTool cannot list the available period (because doing so would require querying all data, which would be very slow). The USGS NWIS site web service may be utilized in the future to improve data browsing.

Datastore Configuration File

A data store is configured by enabling UsgsNwisInstantaneous data stores in the main *TSTool.cfg* configuration file, and creating a datastore configuration file for each datastore. Configurations are processed at software startup. An example of the TSTool configuration file is shown below. Multiple data stores can be defined using the `[DataStore:DataStoreName]` syntax. For NWIS, this would allow, for example, accessing different versions of the web services.

```
# Configuration file for TSTool

[TSTool]

UsgsNwisInstantaneousEnabled = true

# Startup datastores (note that datastore name in config file takes precedence)

[DataStore:UsgsNwisInstantaneous]

ConfigFile = "UsgsNwisInstantaneous.cfg"
```

TSTool Configuration File with UsgsNwisInstantaneous Datastore Properties

Properties for each datastore are specified in an accompanying datastore configuration file (see below), which in the following example is located in the same folder as the TSTool configuration file and configures a data store named “UsgsNwisInstantaneous”.

```
# Configuration information for "UsgsNwisInstantaneous" data store,
# for instantaneous values.
#
# Properties are:
#
# The user will see the following when interacting with the data store:
#
# Type - UsgsNwisInstantaneousDataStore (required as indicated)
# Name - data store identifier used in applications, for example as the
#     input type information for time series identifiers (usually a short string)
# Description - data store description for reports and user interfaces (short phrase)
# Enabled - whether the data store is enabled (default=True)
#
# The following are specific to the USGS NWIS instantaneous value data store:
#
# ServiceRootURI - web service root URI, including the server name and root path
# ServiceAPIDocumentationURI - web service API documentation URI, describing
#     the syntax, input, and output
# ServiceOnlineURI - web service interactive page to query data, typically
#     "drill down" or form based

Type = "UsgsNwisInstantaneousDataStore"
Name = "UsgsNwisInstantaneous"
Description = "USGS NWIS Instantaneous Values Web Service"
Enabled = True
ServiceRootURI = "http://waterservices.usgs.gov/nwis/iv"
ServiceAPIDocumentationURI = "http://waterservices.usgs.gov/rest/IV-Service.html"
ServiceOnlineURI = "http://waterservices.usgs.gov/rest/IV-Test-Tool.html"
```

UsgsNwisInstantaneous Datastore Configuration File

This page is intentionally blank.

Appendix: UsgsNwisRdb Input Type

2012-02-27

Overview

The UsgsNwisRdb time series input type corresponds to the United States Geological Survey (USGS) National Water Information System (NWIS) RDB format (see: http://pubs.usgs.gov/of/2003/ofr03123/6.4rdb_format.pdf). A number of RDB variations are available but currently only the surface water daily format is supported. Data files can be created by saving USGS NWIS website data to a text file or use the TSTool `WebGet()` command. The example below shows the format of a daily surface water file. Important comments about the file format are:

- The file is divided into a header section (top) and data section (bottom). Comments can occur only at the top and are lines that begin with #.
- Optional data flags are saved with the data values, if available (e.g., `e` indicates estimated data).
- HTML remnants may be present at the end of the file. These lines are stripped out as the file is processed.

```
#
# U.S. Geological Survey
# National Water Information System
# Retrieved: 2002-01-28 13:35:25 EST
#
# This file contains published daily mean streamflow data.
#
# This information includes the following fields:
#
#  agency_cd   Agency Code
#  site_no     USGS station number
#  dv_dt       date of daily mean streamflow
#  dv_va       daily mean streamflow value, in cubic-feet per-second
#  dv_cd       daily mean streamflow value qualification code
#
# Sites in this file include:
#  USGS 03451500 FRENCH BROAD RIVER AT ASHEVILLE, NC
#
#
agency_cd      site_no dv_dt   dv_va   dv_cd
5s      15s     10d     12n     3s
USGS    03451500       1895-10-01     740
USGS    03451500       1895-10-02     740
...
USGS    03451500       1985-01-20     1100    e
USGS    03451500       1985-01-21     1100    e
USGS    03451500       1985-01-22     1100    e
...
USGS    03451500       2000-09-28     675
USGS    03451500       2000-09-29     597
USGS    03451500       2000-09-30     550
 <font face="Arial" size=2>
<p>Microsoft VBScript runtime </font> <font face="Arial" size=2>error '800a01a8'</font>
<p>
<font face="Arial" size=2>Object required: 'db'</font>
<p>
<font face="Arial" size=2>/ctp_workgroup/cgi-bin/includes/Inc_htm_utils.asp</font>
<font face="Arial" size=2>, line 217</font>  <font face="Arial" size=2>
<p>Microsoft VBScript runtime </font> <font face="Arial" size=2>error '800a01a8'</font>
<p><font face="Arial" size=2>Object
```

USGS NWIS RDB Files and Standard Time Series Properties

The standard time series identifier for USGS NWIS RDB time series is of the form:

```
Location.DataSource.DataType.Interval~UsgsNwisRdb~PathToFile
```

The limited support of this file format assumes the following:

- The location part of the time series identifier is taken from the second field (`site_no`) in the data records.
- The data source part of the time series identifier is taken from the first field (`agency_cd`) in the data records.
- The data type is assigned as `Streamflow` (interpretation of the verbose `dv_va` field in the header is not implemented).
- The data interval is assigned as `1Day` (interpretation of the verbose `dv_va` field in the header is not implemented).
- The input type is set to `UsgsNwisRdb` (`USGSNWIS` was used in the past) indicating the format of input.
- The input name is set to the absolute or relative path to the file.
- The units are assigned as `CFS`.
- The missing data value is assigned to `-999.0` (gaps in data records will result in this value).
- The description is set to the information after the `Sites in this file include:` line. It is assumed that only one time series per file is used.

Limitations

USGS NWIS RDB files have the following limitations:

- Currently only the daily surface water format has been tested. Additional support will be added in the future.
- Although data flags are read, no standard flag values are enforced (the software user will need to know the meaning of the flags to use them properly).

Appendix: WaterML – WaterML XML Time Series File Input Type

2012-03-05

Overview

The WaterML input type corresponds to the WaterML time series file format developed by the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI), as described on the following page:

> http://his.cuahsi.org/wofws.html#waterml

Currently, only WaterML 1.1 is supported in TSTool; however, version 2.0 is being reviewed as an Open Geospatial Consortium (OGC) standard and will be supported as soon as reference implementations are available for development and testing.

WaterML files can be created by the `ReadUsgsNwisDaily()`, `ReadWaterOneFlow()`, and `WriteWaterML()` TSTool commands. The `WebGet()` command also can be used to retrieve WaterML files from web services such as the USGS NWIS site (see the **UsgsNwisDaily Data Store** appendix).

WaterML and Standard Time Series Properties

The standard time series identifier for WaterML time series in TSTool is of the form:

```
Location.DataSource.DataType.Interval~InputType~InputName
```

More specifically, the identifier adheres to the following convention:

```
SiteCode.AgencyCode.VariableCode-StatisticCode.Interval~WaterML~InputFile
```

where identifier parts are described as follows:

- The `SiteCode` corresponds to the `siteCode` XML element in the `sourceInfo` element (e.g., this is the USGS stream gage station identifier).
- `AgencyCode` corresponds to the `agencyCode` attribute of the `siteCode` XML element (e.g., `USGS` for USGS NWIS stream gage stations).
- `VariableCode` corresponds to the `variableCode` XML element in the `variable` element (e.g., this is `00060` for streamflow for USGS stream gages). In the future, a more humanly-readable text value may be used, perhaps with a string prefix; however, such handling must guarantee that the variable is unique.
- `StatisticCode` corresponds to the `optionCode` attribute of the `option` XML element in the `options` element, in the `variable` element (e.g., `00006` for USGS NWIS sum statistic). In the future, a more humanly-readable text value may be used, perhaps with a string prefix; however, such handling must guarantee that the variable is unique.
- According to the WaterML specification, `Interval` should correspond to the `timeScale` or `timeSpacing` XML elements. However, current USGS NWIS WaterML does not include this information and the interval must be specified externally when reading the WaterML. A more

thorough handling of the interval will be explored as support for WaterOneFlow WaterML web services is added.

- `InputType` is WaterML and `InputName` is the name of the WaterML file.
- Data units correspond to the `unitCode` XML element in the `unit` element of the `variable` element
- Missing numerical values are internally represented as `NaN` and are assigned to any date/times in the period that do not have values. The `noDataValue` element in the `variable` element is checked and matching data values are handled as missing. Often, however, missing values are simply not included as data.
- Data value flags, if encountered, are retained in the time series. However, because the USGS uses "A" for approved, it may be necessary to ignore this flag so that other flags stand out more when visualized.
- Data value qualifiers definitions are saved with time series and are available to use in time series visualization as flag definitions

Limitations

WaterML limitations relative to TSTool standard features are as follows:

- WaterML files from NWIS do not indicate the interval of the data. Day can be assumed for the daily values web service; however, trying to read the WaterML file later requires that the interval is specified.
- Some of the descriptions contain units, which may lead to confusion if time series are processed into different units.

TSTool Configuration File

WaterML support in TSTool is enabled/disabled using a configuration property as shown below.

```
# Configuration file for TSTool

[TSTool]

WaterMLEnabled = true
```

TSTool Configuration File with WaterML Properties

Appendix: WaterOneFlow – WaterOneFlow Web Service Data Store

2012-03-29

Support for WaterOneFlow is under development.

Overview

The WaterOneFlow data store corresponds to the web service specification developed by the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) as part of the Hydrologic Information System (HIS). Access to web service documentation, registry, and other information are available at the following link:

http://his.cuahsi.org/wofws.html

The WaterOneFlow web services are implemented using SOAP technologies. A Java API has been generated using the Web Services Description Language (WSDL) file:

http://river.sdsc.edu/waterOneFlow/NWIS/DailyValues.asmx?WSDL

This Java API is used for all interactions with version 1.0 WaterOneFlow web services. WaterOneFlow/HIS 1.0 uses WaterML 1.0 to return time series data in the SOAP response. The Java API reads the data elements from the WaterML content in order to allow transfer of the data to internal time series properties and data values. Other web services supported by TSTool, such as the UsgsNwisDaily data store (see the **WaterML Data Store Appendix)**, also use WaterML as the transport format. See the **WaterML Input Type Appendix** for more information about the WaterML format and the `ReadWaterML()` command for information about reading WaterML files.

The version of WaterOneFlow and the corresponding WaterML are important because the internal API used to read the web services must be compatible. At this time, TSTool only supports WaterOneFlow 1.0 (and corresponding WaterML 1.0) because the Java API was generated using a version 1.0 WSDL file. However, TSTool's `ReadWaterML()` command provides additional support for WaterML versions because the files are read independent of the WaterOneFlow WSDL API code.

A WaterOneFlow data store is configured by specifying the location of a web service root URL (see section on configuration below). A registry of WaterOneFlow WaterML services can be found at the following location:

http://hiscentral.cuahsi.org/pub_services.aspx

To select a version 1.0 WSDL, review the URL for the WSDL (e.g., http://river.sdsc.edu/wateroneflow/EPA/cuahsi_1_0.asmx?WSDL) and look for "cuahsi_1_0" in the URL. Alternatively, view the WSDL file and look for "his/1.0" in the targetNamespace, as in the following (this information ultimately is checked by the software to determine the version):

```
<wsdl:definitions targetNamespace="http://www.cuahsi.org/his/1.0/ws/">
```

WaterOneFlow/WaterML 1.1 appears to have been a development version and introduces changes that are not supported in TSTool. WaterML 2.0 is being reviewed as an Open Geospatial Consortium (OGC) standard, but is not yet supported in TSTool:

http://www.opengeospatial.org/projects/groups/waterml2.0swg

It is expected that organizations that currently publish data in WaterML will update their services to version 2.0 and support for version 2.0 will be added in TSTool when sites become available for testing and production use.

WaterOneFlow Standard Time Series Properties

This section will be updated to apply to WaterOneFlow. Currently it focuses on UsgsNwisDaily.

The standard time series identifier for WaterOneFlow time series in TSTool is of the form, and are consistent with the WaterML conventions (other than the data store being configured appropriately for a WaterOneFlow data store):

```
Location.DataSource.DataType.Interval~DataStoreName
```

More specifically, the identifier adheres to the following convention:

```
SiteNum.AgencyCode.ParameterCode-StatisticCode.Day~DataStoreName
```

where identifier parts are described as follows:

- The `SiteNum` corresponds to a USGS site. The NWIS Mapper (http://wdr.water.usgs.gov/nwisgmap/), USGS Site Inventory (http://waterdata.usgs.gov/nwis/inventory) and other published information from the USGS can be used to determine site numbers.
- `AgencyCode` can be blank for USGS sites or is specified from the following list (for example, use `USGS` for the code): http://nwis.waterdata.usgs.gov/nwis/help/?read_file=nwis_agency_codes&format=table
- `ParameterCode` is taken from the parameter list available from the following list: http://nwis.waterdata.usgs.gov/usa/nwis/pmcodes
  Ideally a parameter name could be used; however, the list of parameters is extensive, descriptions may change, and special characters like the period are used in descriptions and would interfere with the TSID convention. Consequently, a concise unique parameter name is not readily apparent, and the initial implementation uses the numerical parameter code. In the future, the text name may be allowed and a prefix may be used to indicate whether a code or name is used.
- `StatisticCode` and `StatisticName` are taken from the list of supported statistics: http://waterservices.usgs.gov/rest/USGS-DV-Service.html
  Currently TSTool uses `StatisticCode` rather than `StatisticName`. In the future, the text name may be allowed and a prefix may be used to indicate whether a code or name is used
- Interval defaults to `Day`.
- `DataStoreName` is the user-defined data store name from the configuration information.
- Data units are taken from the following:
  - WaterML `unitCode` in `variable` element
- Missing numerical values are internally represented as `NaN` and are assigned to any date/times in the period that do not have values.

    - WaterML `noDataValue` in `variable` element is checked and matching data values are handled as missing
- Data value flags, if encountered, are retained in the time series. However, because the USGS uses “A” for approved, it may be necessary to ignore this flag so that other flags stand out more when visualized.
- Data value qualifiers definitions are saved with time series and are available to use in time series visualization as flag definitions

Limitations

This section will be updated to apply to WaterOneFlow. Currently it focuses on UsgsNwisDaily.

USGS NWIS data store limitations relative to TSTool standard features are as follows:

- Interpretation of USGS data is limited by WaterML limitations, as follows:
    - WaterML files from NWIS do not indicate the interval of the data. Day can be assumed for the daily values web service; however, trying to read the WaterML file later will require that the interval is specified.
    - Some of the descriptions contain units, which may lead to confusion if time series are processed into different units.
- The USGS web services does not allow for all historical data to be returned. Specifying no period returns only the most recent value. Start and end dates must be specified to retrieve a longer period; however, there is no way to request the entire available period. Consequently, users must request a period of interest for their analysis and the browsing features of TSTool cannot list the available period (because doing so would require querying all data, which would be very slow).

Data Store Configuration File

A data store is configured by enabling WaterOneFlow data stores in the main *TSTool.cfg* configuration file and creating a data store configuration file for each data store. Configurations are processed at software startup. An example of the TSTool configuration file is shown below. Multiple data stores can be defined using the `[DataStore:DataStoreName]` syntax. This allows, for example, accessing web services for different organizations that publish WaterOneFlow web services.

```
# Configuration file for TSTool

[TSTool]

WaterOneFlowEnabled = true

# Startup data stores (note that data store name in config file takes precedence)

[DataStore:WaterOneFlow-NWISDV]

ConfigFile = "WaterOneFlow-NWISDV.cfg"
```

TSTool Configuration File with WaterOneFlow Data Store Properties

Properties for each data store are specified in an accompanying data store configuration file (see below), which in the following example is located in the same folder as the TSTool configuration file and configures a data store named “WaterOneFlow-NWISDV”.

```
# Configuration information for "WaterOneFlow-NWISDV" data store,
# which corresponds to the NWISDV registered network in WaterOneFlow.
# Properties are:
#
# The user will see the following when interacting with the data store:
#
# Type - WaterOneFlowDataStore (required as indicated)
# Name - data store identifier used in applications, for example as the
#     input type information for time series identifiers (usually a short string)
# Description - data store description for reports and user interfaces (short phrase)
# Enabled - whether the data store is enabled (default=True)
#
# The following are specific to the USGS NWIS daily data store:
#
# ServiceRootURI - web service root URI for WSDL
# Version - WaterOneFlow/WaterML version (may be removed in the future)

Type = "WaterOneFlowDataStore"
Name = "WaterOneFlow-NWISDV"
Description = "WaterOneFlow USGS NWIS Daily Value Web Service (1.0)"
ServiceRootURI = "http://river.sdsc.edu/waterOneFlow/NWIS/DailyValues.asmx?WSDL"
Version = "1.0"
```

WaterOneFlow Data Store Configuration File

Documentation Binder Spine Labels

This page, when printed, can be used for a spine in a binder.

Colorado's Decision Support Systems (CDSS)

TSTool – Time Series Tool – Datastore Reference