Appendix: ReclamationHDB Datastore

2014-06-12

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:

```
LocationType:LocationID.DataSource.DataType.Interval.Scenario~
DataStoreName
```

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

```
ObjectTypeName:SiteDataTypeID.HDB.DataTypeCommonName.Interval.
SiteCommonName~DataStoreName
```

and the following for "model" data:

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

Because only the site data type identifier (SDI) and model run identifier (MRI) are guaranteed to be unique, they are used as the primary data in the TSID. Other information is used by convention to improve usability for users, but is not required.

Prior to TSTool 10.24.00 time series identifiers followed the convention shown below. This convention is still handled as much as possible; however, because `SiteCommonName` is not guaranteed to be unique and because model identifier parts contain periods, dashes, and other characters, such identifiers could not be guaranteed to uniquely and reliably identify time series.

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

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

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

and the following for "model" data:

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

Identifier parts and other time series metadata are described below:

- The TSID location type is set to the object type name, for example `diversion` or `reservoir`.
- The TSID location identifier is set to the site data type identifier (SDI) for real time series and is additionally appended with the model run identifier (MRI) for model time series. These integer identifiers are guaranteed to be unique and not change over time. Although these values are enough to allow time series to be uniquely identified in the database, additional metadata are by convention provided in the data type and scenario TSID parts, as described below.
- The TSID data source is set to `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 TSID:
    - 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.
- The TSID data type is set to `DataTypeCommonName` and is taken from the `HDB_DATATYPE.DATATYPE_COMMON_NAME` column.
- `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 TSID scenario contains dash-separated model metadata to help users interpret data:
    - `SiteCommonName` (from `HDB_SITE.SITE_COMMON_NAME`).
    - `ModelName` (from `HDB_MODEL.MODEL_NAME`), only for model time series.
    - `ModelRunName` (from `REF_MODEL_RUN.MODEL_RUN_NAME`), only for model time series.
    - `HydrologicIndicator` (from `REF_MODEL_RUN.HYDROLOGIC_INDICATOR`), only for model time series.
    - `ModelRunDate` (from `REF_MODEL_RUN.RUN_DATE`), to the full second, only for model time series.

  This latter four values are necessary to fully match a model run.
- `DataStoreName` is the user-defined datastore name from the configuration information.
- The time zone for hourly time series is set to the `TIME_ZONE` property from the HDB `REF_DB_PARAMETER` table.
- 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 a space in the TSID

representation. Because the current TSID convention relies on SDI and MDI, text parts of the TSID are not used to look up information in the database and the issue of special characters is lessened. An alternative to relying on TSID commands is to use the `ReadReclamationHDB()` command and assign an alias to time series.

- 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 date/times for instantaneous values correspond 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 HDB `END_DATE_TIME` corresponds to the TSTool date/time and for others the HDB `START_DATE_TIME` corresponds to the TSTool date/time.

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 stores 1Hour data (although this limitation may be removed). When 6Hour data are stored in HDB, the user must specify the interval because there is no way to determine it from HDB. For example, if a 6Hour time series is written to HDB and is then read as 1Hour, TSTool will show a value every 6 hours, with intervening missing values.
- 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. Writing data also is slow. These issues may be improved in the future.

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
# DatabasePort - database connection port (will default to 1521 if not specified
# 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"
DatabasePort = 1521
SystemLogin = "xxx" (for example app_user)
SystemPassword = "xxx"
# How TSTool should create TSIDs
# CommonName - old style that used common name for identifiers – prone to issues because
# of non-unique values in HDB and special characters including periods and dashes
# SDI_MRI - new style that relies on SDI and MRI integer values
# (default if property is not specified)
TSIDStyle = SDI_MRI
# The WRITE_TO_HDB stored procedure should allow an end date to be offset from the start date
by NHour.
# However, it appears to always offset by 1Hour.  Therefore the end date cannot be relied
upon for the TSTool date/time when reading
# Use the following property to adjust the handling of the end date
# ReadNHourEndDateTime = EndDateTime  (use this if WRITE_TO_HDB does write the end date
offset as expected, default)
# ReadNHourEndDateTime = StartDateTimePlusInterval  (use this to compensate for WRITE_TO_HDB
# not writing the end_date_time with offset)
ReadNHourEndDateTime = StartDateTimePlusInterval
# The following properties control the timeone for the initial datastore connection
# and for each stored procedure read, in seconds.
ConnectTimeout = 120
ReadTimeout = 3600
```

ReclamationHDB Datastore Configuration File