**Summary of TSTool/HDB Integration Design Issues**

**Steve Malers, Last update: 2013-09-28**

This document describes TSTool/HDB integration issues. These issues have been identified during the implementation of HDB features in TSTool. A fundamental consideration in any design evaluation is how HDB data are integrated with business processes that are supported by TSTool. The issues are not presented in a particular order, other than to clarify discussion.

1. There is no time series metadata table/view in HDB that joins site and time series metadata, which complicates creating a list of time series for user review and processing by TSTool commands.
   1. For example, there is no table/view that directly provides the information in the TSTool main window time series list. See also the TSTool GenericDatabaseDataStore appendix for an example of such a metadata view. The database query effort needed to create such a data list using the current HDB design is considerable, as discussed further below.
   2. The HDB design does not require that time series be “defined”. Instead, a time series “appears” when records are entered into the R\_ or M\_ tables. In other words, a time series is not fully known until records are written into the R\_ or M\_ tables.
   3. Related to the above point, it is not possible to determine a unique list of time series without doing an SQL DISTINCT query on each of the R\_ and M\_ tables , for site\_datatype\_id and/or model\_run\_id. The TSTool interface asks the user to pick a timestep so that a query can be limited to one of the data tables. However, requiring a distinct query of a full data table is slow and would be unnecessary if time series metadata were stored in a table/view. Querying the data tables puts a significant load on the database and will only get worse as the database grows.
   4. Recommendation: add one or more time series metadata tables that materializes the various joined tables into a static table that can be quickly queried to determine time series metadata. For example, create a HDB\_TIMESERIES\_META table that contains a join of the HDB\_SITE, HDB\_DATATYPE, and other data for each time series that is “defined” (not just possible) in the database. Such a table provides important time series metadata without needing to join to related tables. The table can be updated when metadata changes (via a trigger) or recreate the table on a schedule. The development HDB has approximately 1800 real and model hourly time series and this number will grow if model time series are saved with run dates (as opposed to saving model time series without run date). Separate time series metadata tables for real, model, and ensemble data may be needed, although a join of the ref\_ensemble and ref\_ensemble\_trace tables somewhat functions in this capacity.
2. The database design as currently implemented relies heavily on the use of site\_datatype\_id (SDI) and model\_run\_id (MRI) in human-facing information.
   1. This may be appropriate in cases where other information can be provided to guide the user (e.g., show the site common name and data type name with site\_datatype\_id choices); however, for terse data identifiers such as TSTool time series identifiers, the numeric information results in usability issues because users don’t have the SDI and MRI memorized.
   2. Initial TSTool features, in particular use of the TSID to uniquely identify time series in HDB, relied on what were thought to be unique values (e.g., site\_common\_name). However, Reclamation staff feedback indicates that such database columns do not utilize unique constraints and consequently the SDI and MRI should be used directly to guarantee uniqueness. The following table illustrates columns that are currently used in the TSTool to identify time series and should be evaluated to identify time series. The development HDB database was used to evaluate database contents (yellow indicates currently used in TSID, with red indicting issues that result in time series not having unique identifiers or corruption of the TSID string). Of particular concern is the use of dashes in data, and TSTool’s current practice of using dashes as a delimiter within the TSID scenario to identify model run ID from human-readable parts.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HDB column | Values unique in HDB? | Count of . character | Count of - character | Comment |
| HDB\_SITE.SITE\_NAME | No – 723/726 | 31 | 43 |  |
| HDB\_SITE.SITE\_COMMON\_NAME | No – 714/726 | 0 | 1 | Used in TSID location. |
| HDB\_DATATYPE.DATATYPE\_NAME | Yes – 499/499 | 0 | 17 |  |
| HDB\_DATATYPE.DATATYPE\_COMMON\_NAME | No – 486/499 | 0 | 12 | Used in TSID data type. |
| HDB\_MODEL.MODEL\_NAME | YES – 13/13 | 0 | 0 | Used in TSID scenario. |
| REF\_MODEL\_RUN.MODEL\_RUN\_NAME | No – uniqueness requires model run name, hydrologic indicator, and run date | 0 | 40 | Used in TSID scenario. |
| REF\_MODEL\_RUN.HYDROLOGIC\_INDICATOR |  | 0 | 0 | Used in TSID scenario. |
| REF\_MODEL\_RUN.RUN\_DATE |  | 0 | All | Used in TSID scenario (to minute precision). |
| REF\_ENSEMBLE.ENSEMBLE\_NAME |  |  |  | Uniquely identifies an ensemble and all associated trace data. |
| REF\_ENSEMBLE\_TRACE.TRACE\_NUMERIC |  |  |  | Single time esries that is grouped as a unique ensemble by ENSEMBLE\_ID. |

* 1. One option is to switch to the following convention for TSID, with SDI and MRI providing unique identifier information and other TSID parts providing human-readable information to interpret the TSID (bold indicates values that would be used by software to find time series in HDB and ObjectTypeName is recommended for readability):

ObjectTypeName:**SDI-MRI**-SiteCommonName.HDB.DataTypeCommonName.**6Hour**.ModelInfoInScenario

ObjectTypeName:**SDI**-SiteCommonName.HDB.DataTypeCommonName.**6Hour**

The above TSID would be subject to the issues noted above, namely that TSID special characters (period and dash) cannot be used in TSID parts without translation or enhancing TSTool software to “escape” the characters, such as surrounding TSID parts by single quotes as needed (but this requires that metadata would not include single quotes. In the above examples, model time series are implied if MRI is specified and otherwise real time series are implied. This convention takes advantage of the TSTool LocationType: part at the start of TSIDs, which became available in TSTool version 10.21.00. The above convention has been implemented in TSTool 10.24.00, with backward compatibility for the old TSID – need Reclamation feedback.

* 1. Recommendation: Define unique constraints on SITE\_NAME, SITE\_COMMON\_NAME, DATAYPE\_NAME, DATATYPE\_COMMON\_NAME, MODEL\_NAME, MODEL\_RUN\_NAME, and ENSEMBLE\_NAME. To avoid issues with TSTool, prevent the values from using periods or enhance TSTool to escape the values using single quote in TSID, if the values continue to be used in the TSID. One reason that the “common name” columns are useful for identifying time series in TSTool is that the values are generally shorter and avoid punctuation.

1. If SDI and MRI are unique only in specific HDB databases, then it will not be possible to use the identifiers globally across Reclamation.
   1. In TSTool the work-around is that the time series identifiers will use the datastore name when reading and writing time series, and the datastore name will guarantee a unique TSID.
   2. Operationally, it is possible to define datastores in TSTool for each HDB database, using a datastore name that matches the HDB name (e.g., ecohdb).
   3. Recommendation: Consider using human-readable unique identifiers, in particular if such identifiers are unique across HDB instances.
2. NHour (or NDay, etc.) time series are not cleanly handled in the database design.
   1. The WRITE\_TO\_HDB stored procedure does provide start and end date; however, no matter what offset is specified for the end date, it always writes the end date as 1 hour later than the start.
   2. There is no time series metadata to indicate whether a time series is stored as 1-Hour, 6-hour, etc. Consequently, the user must indicate in TSTool what interval is desired and appropriate R\_HOUR or M\_HOUR table records are processed. However, if the user specifies a different interval when reading than what was used when writing the time series, data integrity is compromised.
   3. Recommendation: Explicitly include in the HDB time series metadata the interval that is used for a time series.
3. The current GET\_TSTOOL\_ENSEMBLE\_MRI stored procedure, when creating an ensemble, does not provide parameters for all ensemble metadata (e.g., agency, ensemble key/value properties).
   1. The GET\_TSTOOL\_ENSEMBLE\_MRI procedure, when creating a new ensemble, does not assign the TRACE\_NAME. This value should be provided by the stored procedure or the default of TRACE\_NUMERIC should be assigned, as was intended as a default.
   2. REF\_ENSEMBLE\_TRACE.TRACE\_NUMERIC currently is used as the human-readable unique identifier for a trace. NWS ensemble files use historical integer year to identify the ensemble trace. However, TSTool has been updated in version 10.24.00 to use string identifiers and internally converts the NWS integer to a string. It is not clear what a trace identifier should be when reading RiverWare MRM RDF files because the input often are not simply historical years of input (e.g., may be index sequential or other resequenced data); currently TSTool uses a sequential integer (1+) to identify traces but a user-supplied identifier may be added to the ReadRiverWare() command.
   3. Recommendation: The current GET\_TSTOOL\_ENSEMBLE\_MRI procedure should be evaluated by Reclamation in operation. Additional parameters should be added to the procedure or an alternate approach should be implemented to set ensemble and trace metadata. TSTool uses stored procedures to write to HDB to ensure compatibility.
4. The REF\_MODEL\_RUN.MODEL\_RUN\_NAME value created by the GET\_TSTOOL\_ENSEMBLE\_MRI procedure is a concatenation of the REF\_ENSEMBLE.ENSEMBLE\_NAME and the REF\_ENSEMBLE\_TRACE.TRACE\_NUMERIC (e.g., “ESSCN SQIN 6**1958**”, where the bold is the trace number). This result would be more readable if a space, comma, underscore, or other delimiter is inserted between the values. Another option is to provide the value as part of the GET\_TSTOOL\_ENSEMBLE\_MRI or another procedure.
5. The WRITE\_TO\_HDB procedure, which inserts single values into real or model tables, is very slow. Loading an ensemble of 50 6Hour interval, year-long time series took over an hour. This is likely unacceptable for production systems and needs to be addressed.