Appendix: Generic Database Data Store

2012-06-12

Overview

The generic database data store can be used to provide general access to database tables and views, for example with the `ReadTableFromDataStore()` command. The trade-off is that tables and views can be accessed; however, 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 during the query. Consequently, subsequent processing must operate on the returned tables or such tables must be converted from a table to time series to use time series processing commands.

The data store 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 uses the `OdbcName` configuration property described below.
- Provide connection information via multiple 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 data store:

- Database permissions control which tables and views are accessible.
- 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.
- 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.

Data Store Configuration Files

A data store is configured by enabling the data store 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 is shown below. Multiple data stores can be defined using the `[DataStore:DataStoreName]` syntax. Properties for each data store are specified in an accompanying configuration file described after the following example.

```
# Configuration file for TSTool

...properties omitted…

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

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

TSTool Configuration File with Generic Database Data Store Properties

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

```
# Configuration information for "SomeDatabaseDataStore" data store (connection).
#
# The user will see the following when interacting with the data store:
#
# 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 data store 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 Data Store Configuration File