
Command Reference: ReadTableFromDataStore()

Read a table from a datastore

Version 11.02.00, 2015-04-09

The `ReadTableFromDataStore()` command executes a database query for a datastore that is associated with a database, and places the result in a TSTool table, which can subsequently be processed with other TSTool commands. This command cannot be used with web service datastores because the underlying software relies on a database to perform the query. If database datastore support is not specifically provided by TSTool, a generic database datastore can be used (see the **Generic Database DataStore** appendix). For example, use a generic database datastore to read data from a Microsoft Access database. This command is useful when the database can provide results with a simple query and tight integration with TSTool is not required or has not been implemented. The query can be specified in the following ways:

- Specify a single table/view to query:
 - the list of tables is filtered to remove internal database tables; however, this capability varies by database product and in some cases internal tables will be listed
 - the query is constructed from the provided database table/view name and column names
 - the output can be sorted by specifying column names
 - “where” clauses currently are not supported but may be added in the future
 - the top N rows of the result can be returned to allow “peeking” at tables (may not be available for all database software)
- Specify a SQL select statement:
 - SQL must be valid for the database (syntax may vary based on database software)
 - Use `${Property}` notation to insert processor property values set with `SetProperty()`.
 - SQL syntax is not checked for validity and therefore error messages from the database may be more difficult to interpret.
 - Comments can be specified using `/* */` notation or `- -` (double dash) for end of line comments. TSTool removes comments if using Microsoft Access because Access does not support comments in SQL.
- Specify an SQL select statement in a file:
 - Similar to the above option; however, the SQL statement is read from a file
 - Useful if the SQL statement is also used by other tools
- Specify a procedure to run:
 - Available procedures are listed and can be selected
 - Currently, only procedures that do not require parameters can be run

General constraints on the query are as follows:

- the table, views, and procedures being queried must be readable (some databases restrict direct access to data and require using stored procedures)
- the resulting table in TSTool will have columns with names that match the database query results
- data types for columns will closely match the database results:
 - data will be treated as strings if unable to match the database column type
 - the precision of floating point numbers for displays is defaulted to 6 digits
 - null values in the database will transfer to null values in the TSTool table and will display as blank table cells

- date/time columns in the database will be represented as such in the TSTool table; however, it may not be possible to limit the precision of the date/time (i.e., hours, minutes, and seconds may be shown with default zero values in output)

Future enhancements will add additional features to intelligently map database results to TSTool tables.

The following dialog is used to edit the command and illustrates the syntax for the command, in this case reading a small table from the State of Colorado's HydroBase.

This command reads a table from a database datastore table, view, or procedure.

The query can be specified in one of four ways:

- 1) Specify a single table or view, related columns, and sort order.
- 2) Specify a free form SQL select statement (allows joins and other SQL constructs supported by the database software).
- 3) Similar to 2; however, the SQL statement is read from a file, which can be specified relative to the working directory. The working directory is: C:\lowf-gitrepos\cdss-app-tstool-test\test\regression\commands\general\ReadTableFromDataStore
- 4) Specify a database procedure to run (under development).

The resulting table columns will have data types based on the query results.

SQL specified with \$(property) notation will be updated to use processor property values before executing the query.

Datastore: **HydroBase** Required - data store containing data to read.

Specify how to query the datastore

Table and columns: **SQL string** | SQL file | Procedure

Specify the catalog and schema as non-blank only if the datastore connection is defined at a higher level and requires additional information to locate the table. Database catalog and schema choices currently do not cascade because database driver metadata features are limited or may be disabled and not provide information.

Datastore catalog (database): Optional - specify if needed for [Database].[Schema].[Table].

Datastore schema: Optional - specify if needed for [Database].[Schema].[Table].

Datastore table: **vw_CDSS_Cropchar** Required - database table/view to read.

Datastore columns: Optional - database table/view columns, separated by commas (default=all).

Order by: Optional - columns to sort by, separated by commas (default=no sort).

Top N rows: Optional - return top N rows (default=return all).

Table ID: **CropChar** Required - unique identifier for the output table.

Row count property: Optional - processor property to set as output table row count.

Command: `ReadTableFromDataStore (DataStore="HydroBase", DataStoreTable="vw_CDSS_Cropchar", TableID="CropChar")`

Buttons: Remove Working Directory, Cancel, OK

ReadTableFromDataStore_Table

ReadTableFromDataStore() Command Editor When Querying a Single Table

The corresponding output table is as shown below:

CROPNUM	METHOD_DESC	CROPNAME	TEMPEARLYMOISTURE	TEMPLATEMOISTURE	INITIALROOT	MAXROOT
1	BLANEY-CRIDDLE_TR-21	ALFALFA.TR21	50.000000	28.000000	4.900000	
8	BLANEY-CRIDDLE_TR-21	GRASS_PASTURE.TR21	45.000000	45.000000	3.300000	
9	BLANEY-CRIDDLE_TR-21	ORCHARD_YMTH_COVER.TR21	50.000000	45.000000	5.000000	
10	BLANEY-CRIDDLE_TR-21	ORCHARD_YVO_COVER.TR21	50.000000	45.000000	5.000000	
7	BLANEY-CRIDDLE_TR-21	GRAPES.TR21	55.000000	50.000000	4.100000	
6	BLANEY-CRIDDLE_TR-21	DRY_BEANS.TR21	60.000000	32.000000	2.500000	

Displaying 39 rows, 20 columns. Ready

ReadTableFromDataStore_Results

Example ReadTableFromDataStore() Command Output Table

The following example illustrates using an SQL query string, in this case to read diversion records for a specific structure in HydroBase:

Edit ReadTableFromDataStore() Command

This command reads a table from a database datastore table, view, or procedure.
 The query can be specified in one of four ways:

- 1) Specify a single table or view, related columns, and sort order.
- 2) Specify a free form SQL select statement (allows joins and other SQL constructs supported by the database software).
- 3) Similar to 2; however, the SQL statement is read from a file, which can be specified relative to the working directory.
 The working directory is: C:\owl-gitrepos\cdss-app-tstool-test\test\regression\commands\general\ReadTableFromDataStore
- 4) Specify a database procedure to run (under development).

The resulting table columns will have data types based on the query results.
 SQL specified with \$(property) notation will be updated to use processor property values before executing the query.

Datastore: **HydroBase** Required - data store containing data to read.

Specify how to query the datastore

Table and columns | **SQL string** | **SQL file** | **Procedure**

SQL String:

```
/* Test reading diversion records */ select meas_num, structure_num, quality, irr_year, fdu, ldu, dwo, maxq, maxq_date,
nobs, amt_nov, amt_dec from vw_CDSS_AnnualAmt where meas_num = (select meas_num from vw_CDSS_StructureStructMeasType
where wd = 1 and id = 501 and meas_type = 'DivTotal' and time_step='Annual') and irr_year >= 1970 and irr_year < 1980
order by irr_year
```

Table ID: **DivTotal** Required - unique identifier for the output table.

Row count property: **omDataStore_HydroBase_Sql_DivTotal** Optional - processor property to set as output table row count.

Command:

```
ReadTableFromDataStore(DataStore="HydroBase",Sql="/* Test reading diversion records */ select meas_num,
structure_num, quality, irr_year, fdu, ldu, dwo, maxq, maxq_date, nobs, amt_nov, amt_dec from vw_CDSS_AnnualAmt where
meas_num=(select meas_num from vw_CDSS_StructureStructMeasType where wd1 and id501 and meas_typeDivTotal and
time_stepAnnual) and irr_year >1970 and irr_year < 1980 order by
irr_year",TableID="DivTotal",RowCountProperty="Test_ReadTableFromDataStore_HydroBase_Sql_DivTotal")
```

Remove Working Directory Cancel OK

ReadTableFromDataStore_SQL

ReadTableFromDataStore() Command Editor When Specifying a SQL Query String

The command syntax is as follows:

```
ReadTableFromDataStore (Parameter=Value, ...)
```

Command Parameters

Parameter	Description	Default
DataStore	The name of a database datastore to read.	None – must be specified.
DataStoreTable	The name of the database table or view to read when querying a single table or view. If specified, do not specify <code>Sql</code> or <code>SqlFile</code> .	None.
DataStoreColumns	When reading a single table/view, the names of the columns to read, separated by commas.	All columns from <code>DataStoreTable</code> are read.
OrderBy	When reading a single table/view, a list of column names separated by commas to control the order of output.	Default database sort order will be used.
Top	Indicate that <code>Top</code> rows should be returned. This functionality may not be implemented for all databases (SQL is not fully standardized for this feature). This parameter is useful to determine the columns for a table prior to using the <code>Sql</code> or <code>SqlFile</code> parameters.	Return all rows.
Sql	The SQL string that will be used to query the database, optionally using <code>\${Property}</code> notation to insert processor property values. If specified, do not specify <code>DataStoreTable</code> or <code>SqlFile</code> .	None.
SqlFile	The name of the file containing an SQL string to execute, optionally using <code>\${Property}</code> notation in the SQL file contents to insert processor property values. If specified, do not specify <code>DataStoreTable</code> or <code>Sql</code> .	None.
DataStoreProcedure	The name of the database procedure to run. Currently, only procedures that do not require parameters can be run.	None.
TableID	Identifier to assign to the output table in TSTool, which allows the table data to be used with other commands. A new table will be created.	None – must be specified.
RowCountProperty	The name of the processor property that will be set to the row count, optionally using <code>\${Property}</code> notation to specify the name.	Property is not set.