Requirements for Teradata

Contents

	Teradata	
Capture	**Hub**	**Integrate**
❌	✅	✅

This section describes the requirements, access privileges, and other features of HVR when using Teradata for replication. For information about compatibility and supported versions of Teradata with HVR platforms, see Platform Compatibility Matrix.

For the Capabilities supported by HVR on Teradata, see Capabilities for Teradata.

For information about the supported data types and mapping of data types in source DBMS to the corresponding data types in target DBMS or file format, see Data Type Mapping.

For instructions to quickly setup replication using Teradata, see Quick Start for HVR - Teradata.

ODBC Connection

HVR uses ODBC connection to the Teradata server for which it requires the Teradata ODBC driver installed on the machine from which it connects to the Teradata server. HVR also requires Teradata Parallel Transporter(TPT) packages to use HVR Refresh in bulk mode.

For information about the supported ODBC driver version, refer to the HVR release notes (**hvr.rel**) available in **hvr_home** directory or the download page.
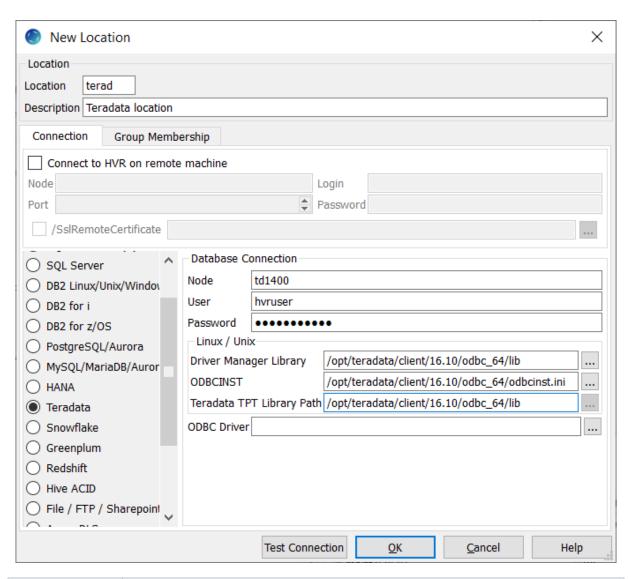
Teradata ODBC driver and TPT packages can be installed using Teradata Tools and Utilities (TTU) package. TTU 16.10 is available for Linux and Windows on Teradata download page, and TTU 15.00 is available for download from Teradata Support Portal (requires user authentication).

The following action definitions are required for TTU to find the correct message files:

Group	Table	Action	Annotation
Teradata	*	**Environment** **/Name**=*NLSPATH*	
Teradata	*	**Environment** **/Value="/opt/teradata/client/***terad ata_version***/odbc_64/msg/%N:/opt/teradata /client/***teradata_version***/tbuild/msg/%"**	For Teradata 15.00
Teradata	*	**Environment** **/Value="/opt/teradata/client/***terad ata_version***/odbc_64/msg/%N:/opt/teradata /client/***teradata_version***/msg/%N"**	For Teradata 16.10 and higher

Location Connection

This section lists and describes the connection details required for creating Teradata location in HVR.

Field	Description
Database Connection	
Node	The hostname or ip-address of the machine on which the Teradata server is running. **Example:** td1400
User	The username to connect HVR to the Teradata **Node** . **Example:** hvruser
Password	The password of the **User** to connect HVR to the Teradata **Node** .
Linux / Unix	
Driver Manager Library	The directory path where the ODBC Driver Manager Library is installed. This field is applicable only for Linux/Unix operating system.
ODBCINST	The directory path where the **odbcinst.ini** file is located. This field is applicable only for Linux/Unix operating system.
Teradata TPT Library Path	The directory path where the Teradata TPT Library is installed. This field is applicable only for Linux/Unix operating system.
ODBC Driver	The user defined (installed) ODBC driver to connect HVR to the Teradata server.

Hub

HVR allows you to create hub database in Teradata. The hub database is a small database which HVR uses to control its replication activities. This database stores HVR catalog tables that hold all specifications of replication such as the names of the replicated databases, the list of replicated tables, and the replication direction.

Grants for Hub Database

The following grants are required for hub database:

1. To perform bulk load (required for **Hvrstats**), the hub **User** must be granted **create table** privilege on the hub database.

   ```
   grant create table on hubdb to hubuser;
   ```

2. To create, update or delete HVR catalog tables, the hub **User** must be granted **select**, **insert**, **update** and **delete** privileges on the hub database.

   ```
   grant select, insert, update, delete on hubdb to hubuser;
   ```

3. To perform bulk load (required for **Hvrstats**), the hub **User** must be granted **create macro** privilege on the hub database.

   ```
   grant create macro on hubdb to hubuser;
   ```

4. To drop tables and macros, the hub **User** must be granted **drop macro, table** privilege on the hub database.

   ```
   grant drop macro, table on hubdb to hubuser;
   ```

Starting HVR Scheduler on Hub

Following is the Linux command line syntax to start HVR scheduler on a Hub with Teradata:

```
$ hvrscheduler -EODBCINST=ODBCINST_directory_path -EHVR_TPT_LIB_PATH=Tera
data_TPT_Library_Path -EHVR_ODBC_DM_LIB=Driver_Manager_Library -h
teradata 'Node~User/Password'
```

Examples,

```
$ hvrscheduler -EODBCINST=/opt/teradata/client/16.10/odbc_64/odbcinst.ini
-EHVR_TPT_LIB_PATH=/opt/teradata/client/16.10/odbc_64/lib -
EHVR_ODBC_DM_LIB=/opt/teradata/client/16.10/odbc_64/lib -h teradata
'td1400~hvruser/hvruser'
```

Integrate and Compare

HVR supports integrating changes into Teradata location. This section describes the configuration requirements for integrating changes (using **Integrate** or **HVR Refresh**) into Teradata location. For the list of supported Teradata versions, into which HVR can integrate changes, see Integrate changes into location in Capabilities. HVR also allows you to perform **HVR Compare** for Teradata database in source location.

HVR uses the Teradata ODBC driver to write data to Teradata during continuous **Integrate** and row-wise **Refresh**. However, the preferred methods for writing data to Teradata are **Integrate** with **/Burst** and Bulk **Refresh** as they provide better performance. HVR uses the following interfaces for this:

- TPT Load, used for copying data to Teradata tables during bulk **Refresh**
- TPT Stream/Upload, used load data into burst table during **Integrate** with **/Burst**.

Grants for Integrate and Compare

By default, the HVR **User** has all required permissions to Integrate changes into default user database. Following are the grants required for integrating changes and perform compare in Teradata:

1. To change/replicate into target tables which are not owned by HVR **User** (using **TableProperties** **/Schema**), the **User** must be granted **select**, **insert**, **update**, and **delete** privileges.

   ```
   grant select, insert, update, delete on targetdb to hvruser;
   ```

2. To create target tables using **Hvrrefresh**, the **User** must be granted **create table** and **drop any table** privileges.

   ```
   grant create table on targetdb to hvruser;

   grant drop any table on targetdb to hvruser;
   ```

3. To read from tables which are not owned by HVR **User** (using **TableProperties** **/Schema**) during **Hvrcompare** or **Hvrrefresh**, the **User** must be granted **select** privilege.

   ```
   grant select on targetdb to hvruser;
   ```

4. To perform bulk load, the **User** must be granted **create macro** privilege.

   ```
   grant create macro on targetdb to hvruser;
   ```