Quick Start for HVR - Teradata

Contents

This quick start guide helps you to get started with HVR for replicating data into Teradata database. To proceed with this replication you must have basic understanding about HVR's architecture and terminologies like Hub, Location, Channel, Location Groups, Actions etc.

The example here demonstrates how to replicate tables from one source location (in Oracle) into a target location (in Teradata).

In real-life scenarios, the source location(s) and the target location(s) reside on different machines and the HVR hub can reside on source or target or a separate machine. However, in this example, for simplicity we have the source, target, and HVR hub on the same machine.

Before proceeding with this example ensure that the requirements for using HVR with Oracle and Teradata are met.

For information about access privileges and advanced configuration changes required for performing replication using Oracle and Teradata, see:

- Requirements for Oracle
- Requirements for Teradata

Create Test Databases and Tables

The initial step of this demonstration is to create:

- a schema in the source location (Oracle)
- tables in the source schema (and insert values into these tables)
- a database and user in the target location (Teradata)

Source Location

Skip this section if you already have a database with tables which you plan to use for this replication.

For this demonstration, in the source database (Oracle), create a schema (e.g. **sourcedb**) with two tables (e.g. **dm51_product** and **dm51_order**), and insert values into these tables.

Following are sample SQL statements to create schema and tables in source location, and also to insert values in the tables.

Create Source Schema

```
create user sourcedb identified by hvr default tablespace users temporary
tablespace temp quota unlimited on users;
```

Create Tables in Source Schema

```
create table sourcedb.dm51_product
(
 prod_id number(10) not null,
 prod_price number(10,2) not null,
 prod_descrip varchar2(100) not null,
 primary key (prod_id)
);
```

```
create table sourcedb.dm51_order
(
 prod_id number(10) not null,
 ord_id number(10) not null,
 cust_name varchar2(100) not null,
 cust_addr varchar2(100),
 primary key (prod_id, ord_id)
);
```

Insert Values in Source Tables

```
insert into sourcedb.dm51_product values (100, 90, 'Book');
```

```
insert into sourcedb.dm51_order values (100, 123, 'Customer1', 'P.O. Box
122, Anytown, Anycountry');
```

Target Location

Create a database and user in the target database (Teradata).

1. Create a database (e.g. **targetdb**) in Teradata.

   ```
   create database targetdb as perm=20000000;
   ```

2. Create a user (e.g. **targetuser**) for the target database in Teradata.

   ```
   create user targetuser as perm=20000000, password=hvr, default
   database=targetdb;
   ```

   If a default database is not assigned, you must define action **TableProperties** **/Schema=***targetdbname* to specify the schema where the tables are to be created.

HVR automatically creates tables in target location during **HVR Refresh** (initial loading) and it is the recommended method for creating tables in the target location. However, if you want to manually create tables in target location, the same can be achieved by executing the required SQL statements.

Sample SQL statements to create schema and tables in the target location,

```
create table dm51_product (
prod_id integer not null,
prod_price decimal(10,2) not null,
prod_descrip varchar(20) not null,
primary key (prod_id)
);
```

```
create table dm51_order (
prod_id integer not null,
ord_id integer not null,
cust_name varchar(100) not null,
cust_addr varchar(100),
primary key (prod_id, ord_id)
);
```

Create Hub Database

This section describes how to create a hub database (user). The hub database is a repository database that HVR uses to control its replication activities. It contains HVR catalog tables that hold all specifications of replication such as the names of the replicated databases, the replication direction and the list of tables to be replicated. For more information about HVR hub server and database, see section Hub Server in System Requirements.

HVR supports the creation of a hub database only on certain databases (location classes). For the list of supported location classes, see section Hub Database in Capabilities.

For this demonstration,

- Create a user (e.g. **hvrhubuser**) for the hub database in Teradata

  ```
  create user hvrhubuser as perm=20000000, password=hvr;
  ```

It is not required to create a separate database for HVR hub.

Grants/Access Privileges

This section describes the grants/access privileges required for the source schema, target database user, and hub database user.

1. Configure the privileges for source schema (**sourcedb**). For more information, see section Grants for Log-Based Capture in Requirements for Oracle.

   ```
   grant create session to sourcedb;
   grant create table to sourcedb;
   grant alter any table to sourcedb;
   grant select any dictionary to sourcedb;
   grant select any transaction to sourcedb;
   ```

2. Configure the privileges for target database user (**targetuser**). For more information, see section Grants for Integrate and Compare in Requirements for Teradata.

   ```
   grant select, insert, update, delete on targetdb to targetuser;
   grant create macro, table on targetdb to targetuser;
   grant drop table on targetdb to targetuser;
   ```

3. Configure the privileges for hub database user (**hvrhubuser**). For more information, see section Grants for Hub Database in Requirements for Teradata.

```
grant create table on hvrhubuser to hvrhubuser;
grant select, insert, update, delete on hvrhubuser to hvrhubuser;
grant create macro on hvrhubuser to hvrhubuser;
grant drop macro, table on hvrhubuser to hvrhubuser;
```

Download and Install HVR

An HVR distribution is available for download at https://www.hvr-software.com/account/. To request a trial version, visit https://www.hvr-software.com/free-trial/.

Install HVR on a hub machine. For details on installing HVR, see the respective operating system sections:

- Installing HVR on UNIX or Linux
- Installing HVR on Windows
- Installing HVR on macOS

The HVR distribution requires a license key in order for the software to operate. Please see the HVR licensing page for more details on how to install the HVR license.

After the installation, you can control HVR using the HVR graphical user interface (**HVR GUI**).

- If the hub machine is Windows, then **HVR GUI** can be executed directly on the hub machine.
  - To control HVR remotely from your PC, connect to the hub machine using Windows Remote Desktop Connection and launch **HVR GUI** on the hub machine.
- If the hub machine is Linux, then **HVR GUI** can be executed directly on the hub machine. However, an application like X Server or VNC viewer must be installed to run **HVR GUI** directly on Linux.
  - To control HVR remotely from your PC, install HVR on the PC (with Windows or macOS) and configure the **HVR Remote Listener** on the hub machine.
- If the hub machine is Unix, then **HVR GUI** should typically be run remotely from a PC to control HVR installed on the hub machine. To do this, install HVR on the PC (with Windows or macOS) and configure the **HVR Remote Listener** on the hub machine.

The **HVR Remote Listener** allows you to connect **HVR GUI** available on your PC to the remote HVR hub machine. For more information about connecting to remote HVR installation, see Configuring Remote Installation of HVR on Unix or Linux and Configuring Remote Installation of HVR on Windows.

Launch HVR GUI

This section describes how to launch **HVR GUI** on various operating systems.

- On Windows and macOS, double-click the HVR shortcut icon available on the desktop or execute command **hvrgui** in the CLI.
- On Linux, double-click the hvrgui file available in the HVR_extracted_path/bin directory or execute command **hvrgui** in the CLI.

  Linux requires applications like X server or VNC viewer to execute **HVR GUI**.
- On Unix, **HVR GUI** is not supported. So, **HVR GUI** should be run on a remote PC (with Windows, Linux, or macOS) to control HVR installed on the Unix machine.

Register Hub

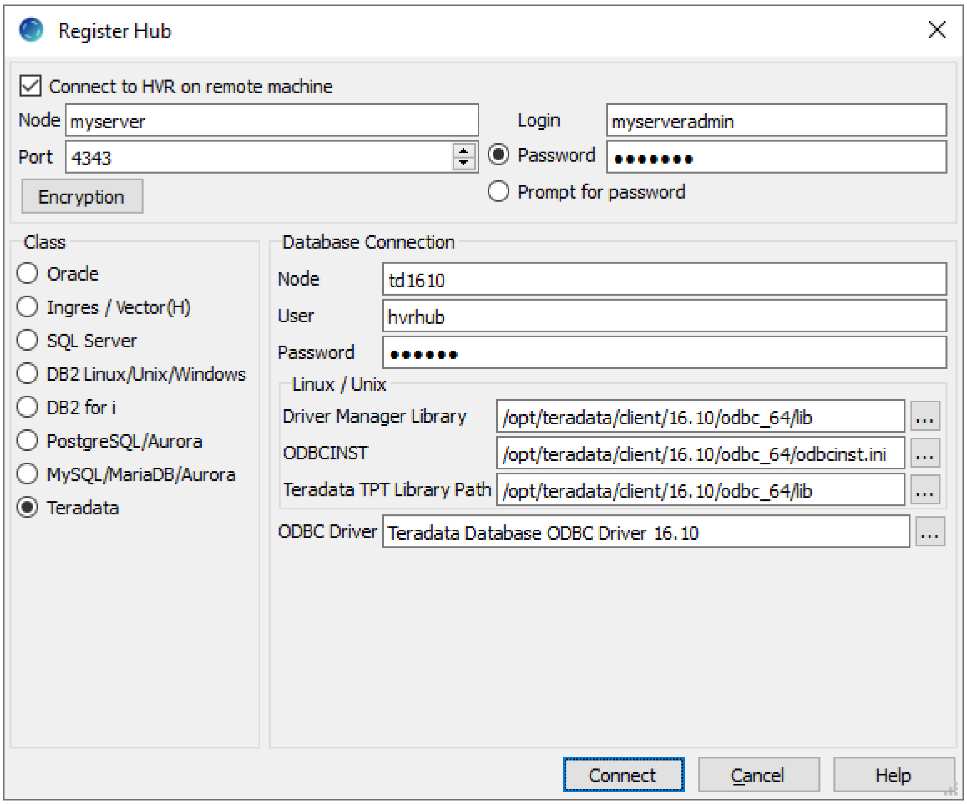
This section describes how to connect **HVR GUI** to the hub database in Teradata.

When you launch HVR GUI for the first time, the **Register Hub** dialog is displayed automatically. The **Register Hub** dialog can also be accessed from menu **File** by selecting **Register Hub**. Skip steps 1 to 4 if you want to run **HVR GUI** directly on the hub machine.

1. Click **Connect to HVR on remote machine**.

To connect **HVR GUI** on a PC to a remote HVR hub machine, the **HVR Remote Listener** must be configured and running on the HVR hub machine.

2. Enter the name or IP address of the hub machine in the **Node** field (e.g. **myserver**).
3. Enter the port number (defined in the **HVR Remote Listener** of the hub machine) in the **Port** field (e.g. **4343**).
4. Enter the **Login** (e.g. **myserveradmin**) and **Password** for the hub machine. By default, this is the operating system login credentials of the hub machine.
5. Select **Teradata** in the **Class** pane.
6. Specify **Database Connection** details. For more information on **Database Connection** fields, see section Location Connection.
    a. Enter the Teradata server hostname or IP-address in **Node**.
    b. Enter the user name of the hub database in **User** (e.g. **hvrhubuser**).
    c. Enter the password for the hub database in **Password** (e.g. **hvr**).
    d. If HVR hub is installed on Linux,
        i. Enter the directory path where the ODBC Driver Manager Library in **Driver Manager Library**.
        ii. Enter the directory path where the odbcinst.ini file is located in **ODBCINST**.
        iii. Enter the directory path where the Teradata TPT Library is installed in **Teradata TPT Library Path**.
    e. Browse and select the user installed driver for Teradata in **ODBC Driver**, if any.
7. Click **Connect**.



Register Hub

☑ Connect to HVR on remote machine

Node myserver

Port 4343

Login myserveradmin

● Password ●●●●●●●

○ Prompt for password

Class

○ Oracle

○ Ingres / Vector(H)

○ SQL Server

○ DB2 Linux/Unix/Windows

○ DB2 for i

○ PostgreSQL/Aurora

○ MySQL/MariaDB/Aurora

● Teradata

Database Connection

Node td1610

User hvrhub

Password ●●●●●●

Linux / Unix

Driver Manager Library /opt/teradata/client/16.10/odbc_64/lib

ODBCINST /opt/teradata/client/16.10/odbc_64/odbcinst.ini

Teradata TPT Library Path /opt/teradata/client/16.10/odbc_64/lib

ODBC Driver Teradata Database ODBC Driver 16.10

8. Click **Yes** in the prompt dialog asking to create catalog tables in the hub database.

    HVR displays this prompt when connecting to a hub database for the first time.

    On connecting successfully to the hub database, the navigation tree pane displays the hub machine and the hub database. **Location Configuration**, **Channel Definitions**, and **Scheduler** are displaye d under the hub database.

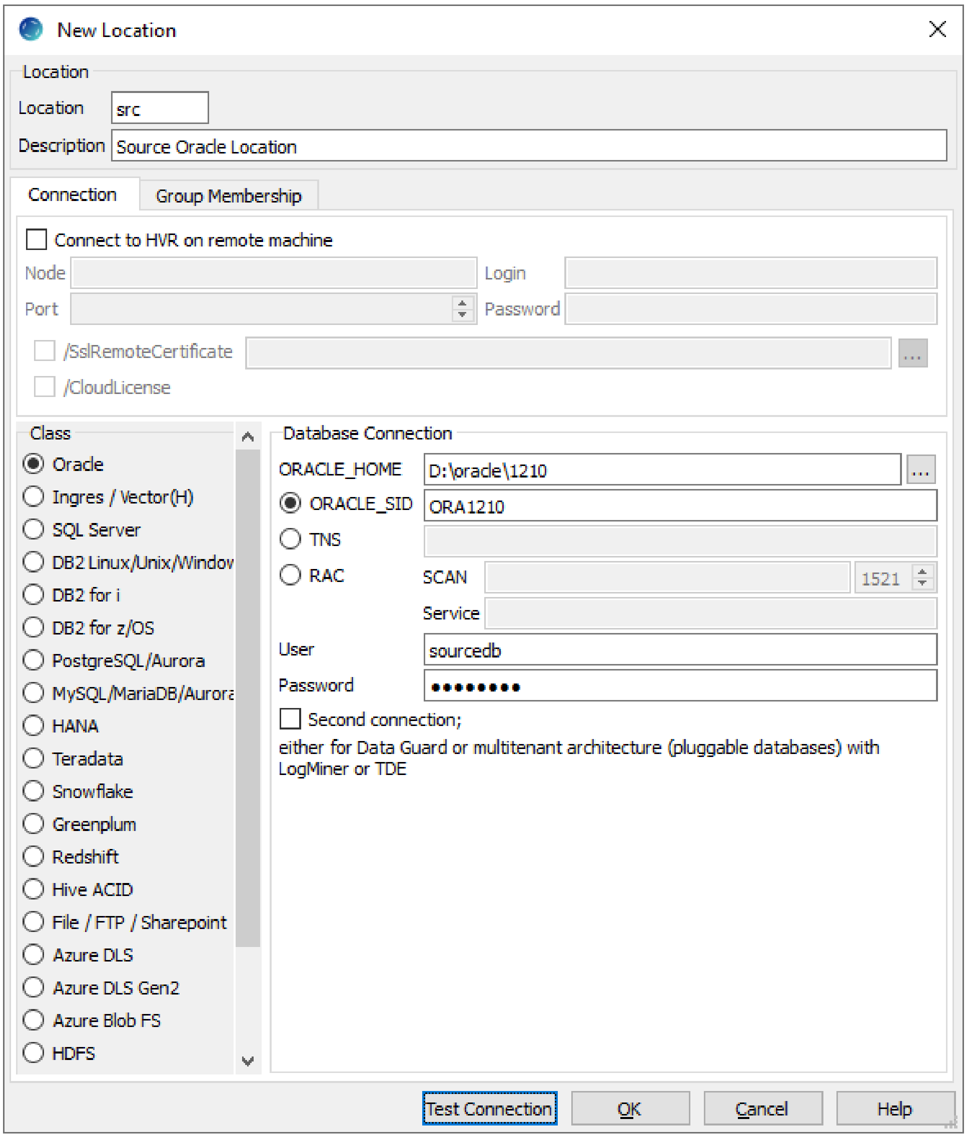
Create Locations

This section describes how to create locations in HVR GUI. Location is a storage place (for example, database or file storage) from where HVR captures (source location) or integrates (target location) changes.

Create one source location (**src**) connected to source schema (**sourcedb**) and a target location (**tgt**) connected to the target database (**targetdb**).

- Create source location (**src**)
    1. In navigation tree pane, right-click **Location Configuration New Location**.
    2. Enter **Location** name and **Description** for the location.
    3. Select **Oracle** in **Class**.
    4. Provide **Database Connection** details. For more information on **Database Connection** fields , see section Location Connection.
        a. Enter directory path for **ORACLE_HOME**. You can also click browse to select directory path.
        b. Enter Oracle System ID in **ORACLE_SID** or **TNS** credential or **RAC** credential.

           For **RAC** connectivity, ensure to provide remote machine connection details under **Connection** tab.
        c. Enter user name of schema in **User**. For example, **sourcedb**.
        d. Enter password for schema in **Password**. For example, **hvr**.
    5. Click **Test Connection** to verify the connection to location database.
    6. Click **OK**.



New Location

Location

Location: src

Description: Source Oracle Location

Connection | Group Membership

□ Connect to HVR on remote machine

Node | Login | Port | Password

□ /SslRemoteCertificate

□ /CloudLicense

Class

● Oracle
○ Ingres / Vector(H)
○ SQL Server
○ DB2 Linux/Unix/Windov
○ DB2 for i
○ DB2 for z/OS
○ PostgreSQL/Aurora
○ MySQL/MariaDB/Aurora
○ HANA
○ Teradata
○ Snowflake
○ Greenplum
○ Redshift
○ Hive ACID
○ File / FTP / Sharepoint
○ Azure DLS
○ Azure DLS Gen2
○ Azure Blob FS
○ HDFS

Database Connection

ORACLE_HOME: D:\oracle\1210

● ORACLE_SID: ORA1210

○ TNS

○ RAC SCAN 1521

Service

User: sourcedb

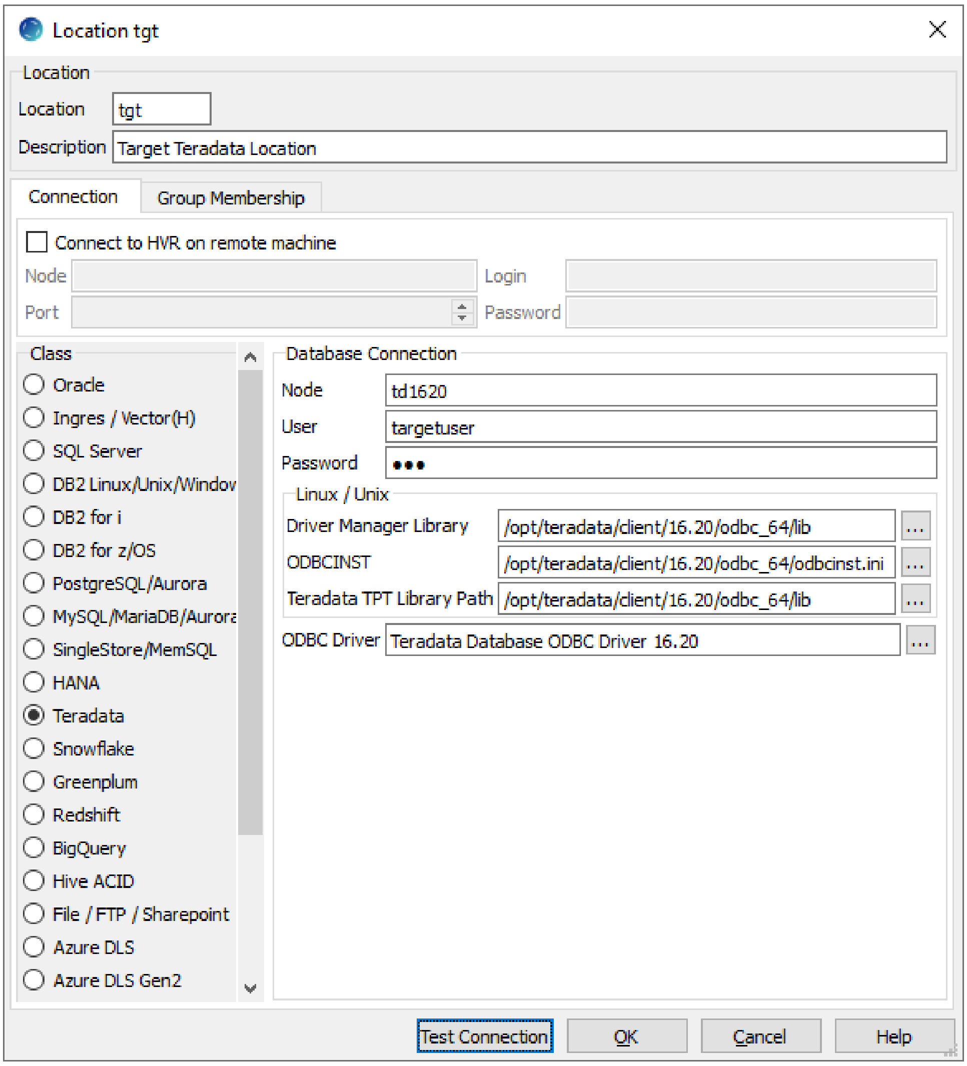
Password: ••••••••

□ Second connection;

either for Data Guard or multitenant architecture (pluggable databases) with LogMiner or TDE

- Create target location (**tgt**)
    1. In navigation tree pane, right-click **Location Configuration New Location**.
    2. Enter **Location** name and **Description** for the location.
    3. Select **Teradata** in **Class**.
    4. Provide **Database Connection** details. For more information on **Database Connection** fields , see section Location Connection.
        a. Enter the hostname or IP address for the Teradata **Node**.
        b. Enter the user name of the target database in **User**. For example, **targetuser**.
        c. Enter password for the target database in **Password**. For example, **hvr**.
        d. If HVR is installed on Linux/Unix,

        i. Enter the directory path where the ODBC Driver Manager Library in **Driver Manager Library**.
        ii. Enter the directory path where the odbcinst.ini file is located in **ODBCINST**.
        iii. Enter the directory path where the Teradata TPT Library is installed in **Teradata TPT Library Path**.
    e. Browse and select the user installed driver for Teradata in **ODBC Driver**, if any.
5. Click **Test Connection** to verify the connection to the target database.
6. Click **OK**.



Location

Location: tgt

Description: Target Teradata Location

□ Connect to HVR on remote machine

Node | Login

Port | Password

Class

○ Oracle
○ Ingres / Vector(H)
○ SQL Server
○ DB2 Linux/Unix/Windov
○ DB2 for i
○ DB2 for z/OS
○ PostgreSQL/Aurora
○ MySQL/MariaDB/Aurora
○ SingleStore/MemSQL
○ HANA
● Teradata
○ Snowflake
○ Greenplum
○ Redshift
○ BigQuery
○ Hive ACID
○ File / FTP / Sharepoint
○ Azure DLS
○ Azure DLS Gen2

Node: td1620

User: targetuser

Password: •••

Driver Manager Library: /opt/teradata/client/16.20/odbc_64/lib

ODBCINST: /opt/teradata/client/16.20/odbc_64/odbcinst.ini

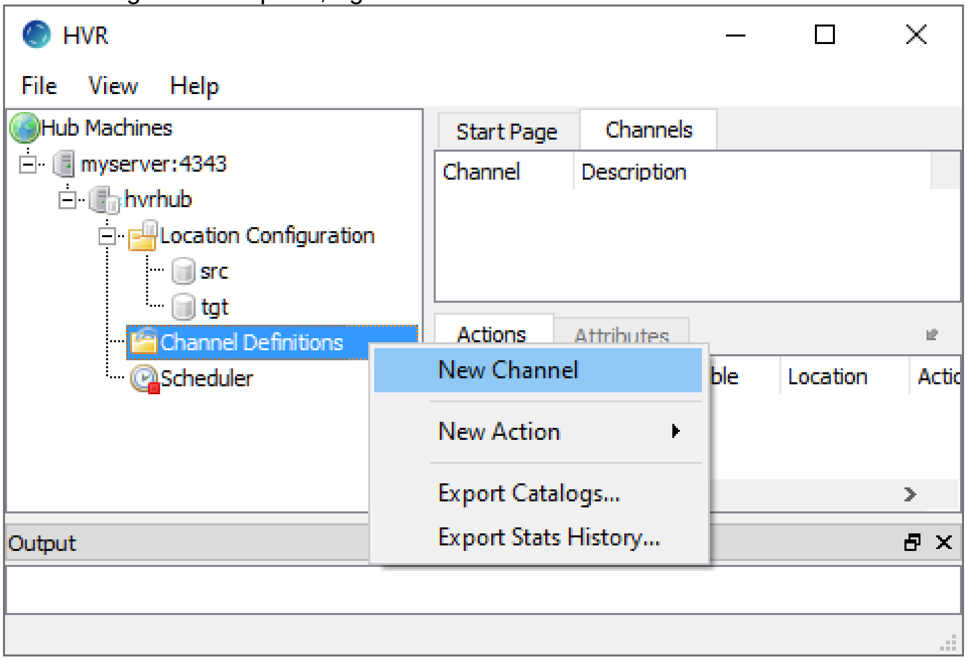
Teradata TPT Library Path: /opt/teradata/client/16.20/odbc_64/lib

ODBC Driver: Teradata Database ODBC Driver 16.20

Create Channel

This section describes how to create a channel (**hvrdemo**) in HVR. A channel groups together the locations and tables that are involved in replication. It also contain actions that control the replication.

1. In the navigation tree pane, right-click **Channel Definitions New Channel**.



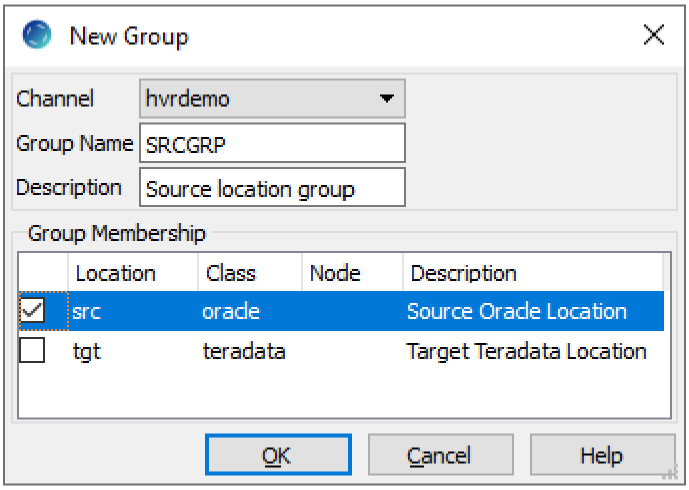
2. In the **New Channel** dialog, enter **Channel** name and **Description** for the channel.
3. Click **OK**.

Create Location Groups

This section describes how to create location groups in a channel. The location groups are used for defining actions on the location. Typically a channel contains two location groups - one for the source location and one for the target location. Each location group can contain multiple locations.

In this example, create one source location group (**SRCGRP**) and one target location group (**TGTGRP**).

1. In navigation tree pane, click **+** next to the channel (**hvrdemo**).
2. Create source location group (**SRCGRP**):
    a. Right-click **Location Groups New Group**.
    b. Enter **Group Name** and **Description** for the location group.
    c. Select source location (**src**) from **Group Membership**.
    d. Click **OK**.



Channel: hvrdemo

Group Name: SRCGRP

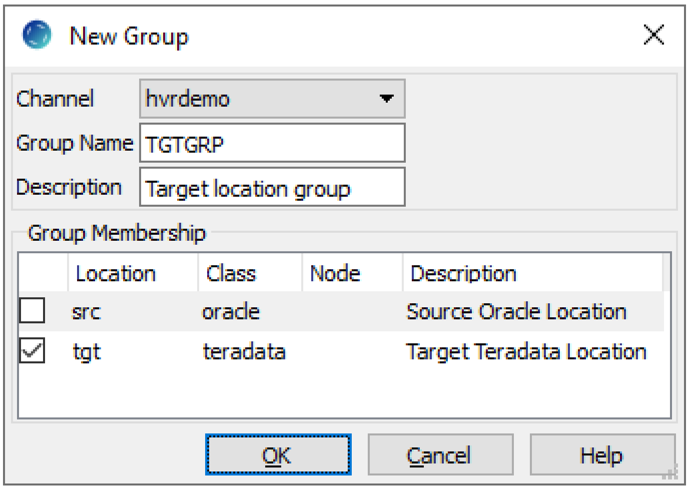
Description: Source location group

Group Membership

	Location	Class	Node	Description
☑	src	oracle		Source Oracle Location
□	tgt	teradata		Target Teradata Location

3. Create target location group (**TGTGRP**):
    a. Right-click **Location Groups New Group**.
    b. Enter **Group Name** and **Description** for the location group.
    c. Select target location (**tgt**) from **Group Membership**.

d. Click **OK**.



New Group

Channel: hvrdemo

Group Name: TGTGRP

Description: Target location group

Group Membership

	Location	Class	Node	Description
□	src	oracle		Source Oracle Location
☑	tgt	teradata		Target Teradata Location

Select Table(s)

This section describes how to select the tables (**dm51_product** and **dm51_order**) from source location for replication. **Table Explore** allows you to select schema(s) and/or table(s) for replication.

1. Right-click **Tables** **Table Explore**.
2. Select source location (**src**) from the list.

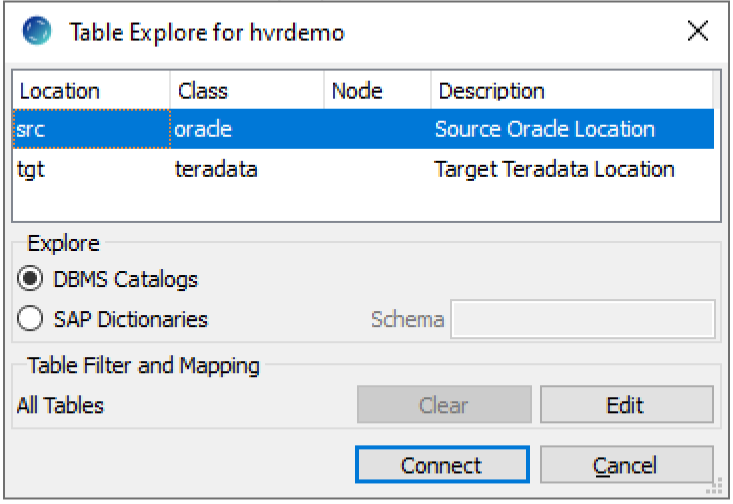


Table Explore for hvrdemo

Location	Class	Node	Description
src	oracle		Source Oracle Location
tgt	teradata		Target Teradata Location

Explore

● DBMS Catalogs

○ SAP Dictionaries

Schema

Table Filter and Mapping

All Tables

3. Click **Connect**.
4. Select tables from **Table Explore** dialog. Press **Shift** key to select multiple tables or **Ctrl+A** to select all tables.
5. Click **Add** to add the selected tables.
6. Click **OK** in **HVR Table Name** dialog.
7. Click **Close** in **Table Explore** dialog.

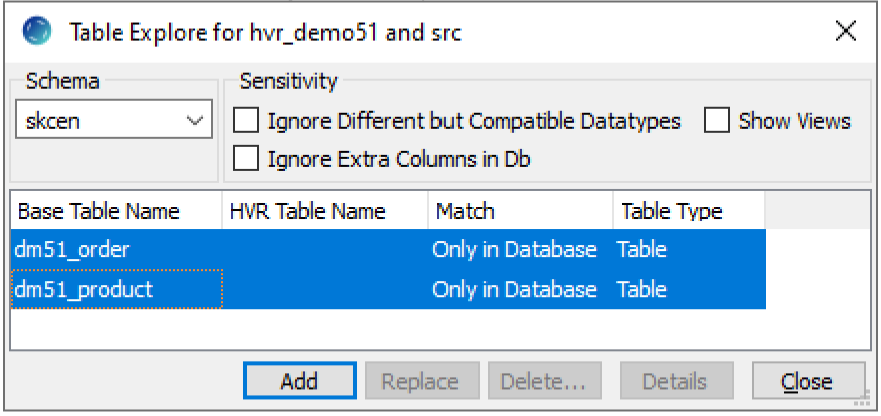


Table Explore for hvr_demo51 and src

Schema: skcen

Sensitivity

□ Ignore Different but Compatible Datatypes □ Show Views

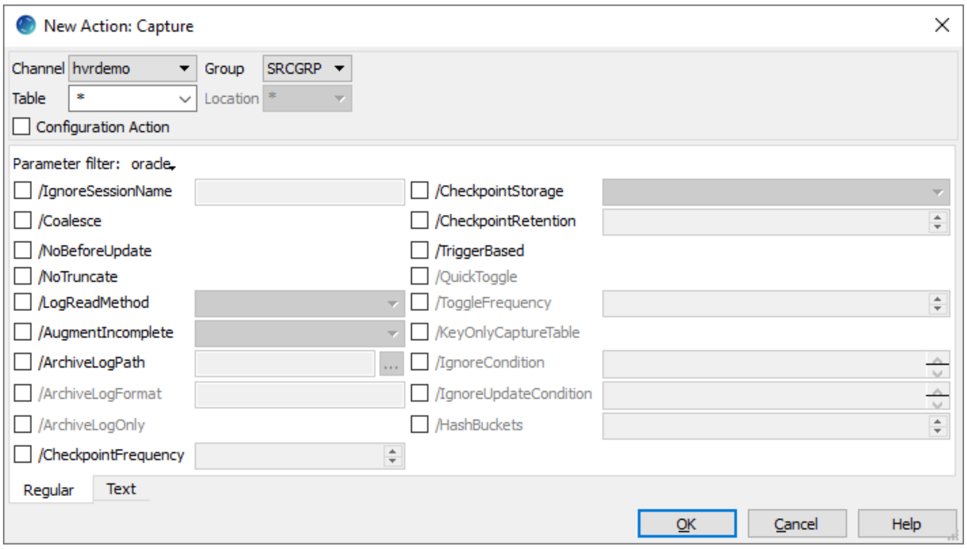
□ Ignore Extra Columns in Db

Base Table Name	HVR Table Name	Match	Table Type
dm51_order		Only in Database	Table
dm51_product		Only in Database	Table

Define Actions

This section describes how to define **Actions** on the location groups (**SRCGRP** and **TGTGRP**). Actions define the behavior of a replication activity.

1. Define action **Capture** to capture changes from all tables in the source location group.
   a. Right-click source location group **SRCGRP New Action Capture**.
   b. Click **OK**.

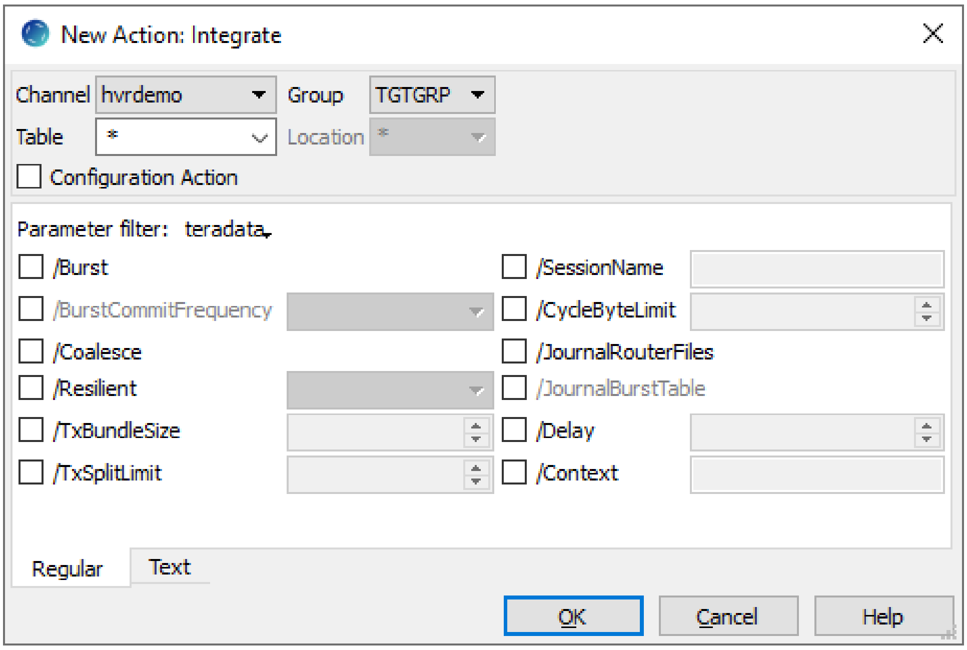


New Action: Capture

Channel hvrdemo
Group SRCGRP
Table *
Location *
□ Configuration Action

Parameter filter: oracle

2. Define action **Integrate** to integrate changes into the target location group.
   a. Right-click target location group **TGTGRP New Action Integrate**.
   b. Click **OK**.



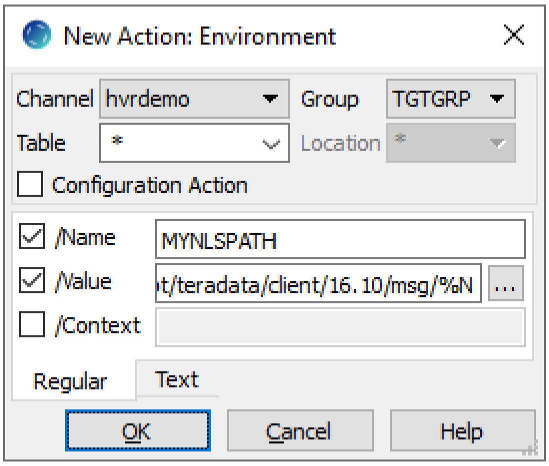
New Action: Integrate

Channel hvrdemo
Group TGTGRP
Table *
Location *
□ Configuration Action

Parameter filter: teradata

3. Define action **Environment** for TTU to find the correct message files. For more information, refer to section ODBC Connection in Requirements for Teradata.
   a. Right-click target location group **TGTGRP New Action Environment**.
   b. Enter a name for the **NLSPATH** in **/Name**.
   c. Enter the *NLSPATH* in **/Value**.

d. Click **OK**.



New Action: Environment

Channel hvrdemo Group TGTGRP

Table * Location *

□ Configuration Action

☑ /Name MYNLSPATH

☑ /Value t/teradata/client/16.10/msg/%N

□ /Context

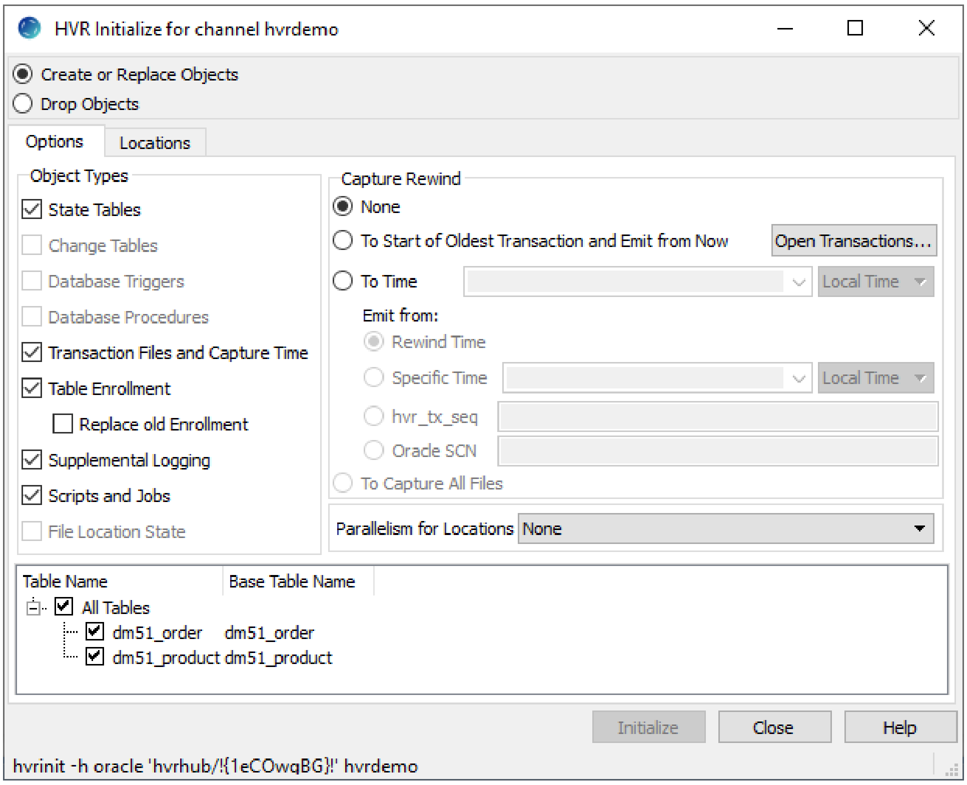
The **Actions** pane only displays actions related to the object selected in the navigation tree pane. Click on the channel name (**hvrdemo**) to view actions defined for all location groups in the selected channel.

Initialize

This section describes how to initialize the replication. **HVR Initialize** first checks the channel and creates replication jobs in the **HVR Scheduler**.

In this example, **HVR Initialize** creates one capture job (**hvr_demo-cap-src**) and one integrate job (**hvr_demo-integ-tgt**).

1. Right-click channel **hvrdemo** **HVR Initialize**.
2. Select **Create or Replace Objects** in **HVR Initialize** dialog.
3. Click **Initialize**.



HVR Initialize for channel hvrdemo

● Create or Replace Objects
○ Drop Objects

Object Types
☑ State Tables
□ Change Tables
□ Database Triggers
□ Database Procedures
☑ Transaction Files and Capture Time
☑ Table Enrollment
□ Replace old Enrollment
☑ Supplemental Logging
☑ Scripts and Jobs
□ File Location State

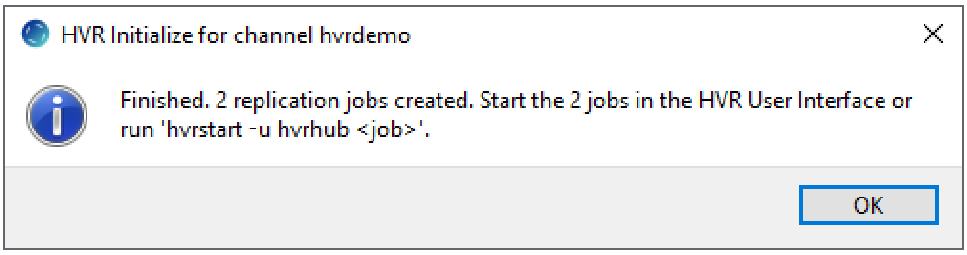
Capture Rewind
● None
○ To Start of Oldest Transaction and Emit from Now
○ To Time Local Time
Emit from:
● Rewind Time
○ Specific Time Local Time
○ hvr_tx_seq
○ Oracle SCN
○ To Capture All Files

Parallelism for Locations None

Table Name	Base Table Name
☑ All Tables	
☑ dm51_order	dm51_order
☑ dm51_product	dm51_product

```
hvrinit -h oracle 'hvrhub/!{1eCOwqBG}!' hvrdemo
```

4. Click **OK**.



Finished. 2 replication jobs created. Start the 2 jobs in the HVR User Interface or run 'hvrstart -u hvrhub <job>'.

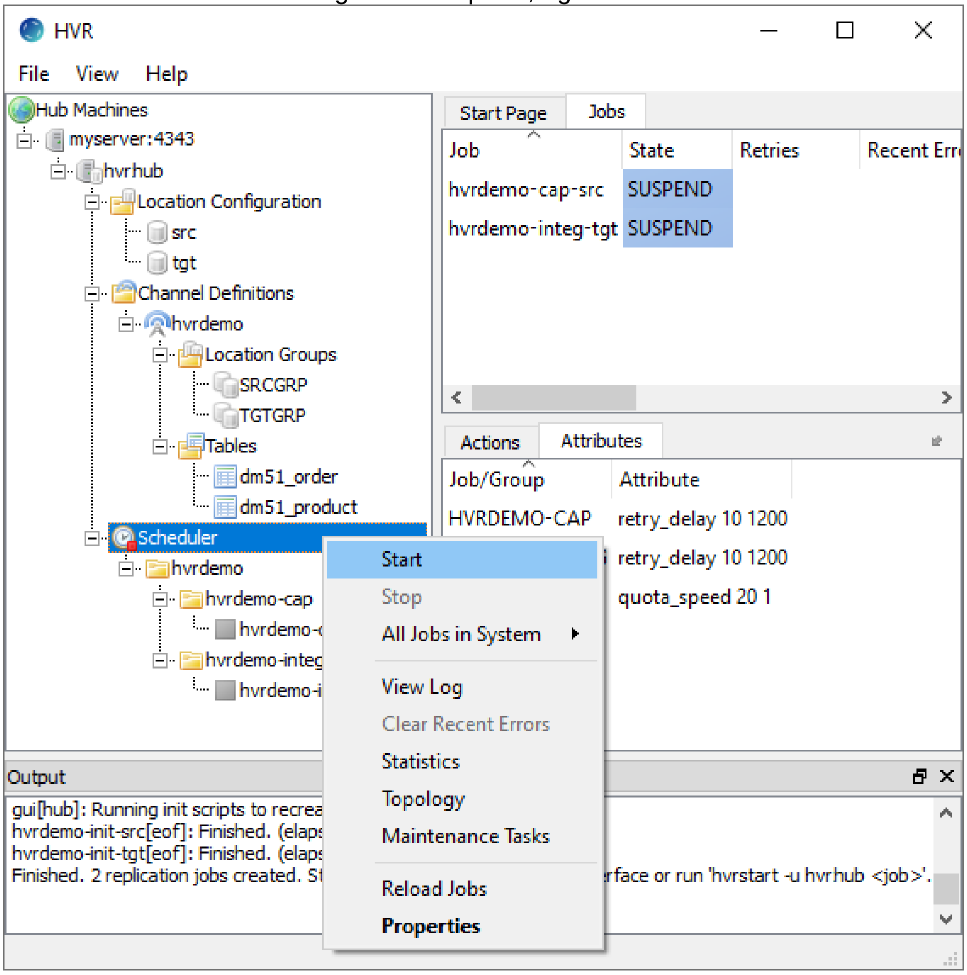
5. Click **Close**.
   Click **Scheduler** node in navigation tree pane to view the capture and integrate jobs in **Jobs** tab.

For more information about initiating replication in HVR, see Replication Overview.

Start Scheduler

This section describes how to start the **HVR Scheduler**. The **HVR Scheduler** is a process which runs jobs defined in the catalog table **HVR_JOB**. This catalog table can be found in the hub database. On Unix or Linux, the **HVR Scheduler** runs as a daemon. On Windows, the **HVR Scheduler** runs as a system service.

1. Start **Scheduler**. In the navigation tree pane, right-click **Scheduler** **Start**.



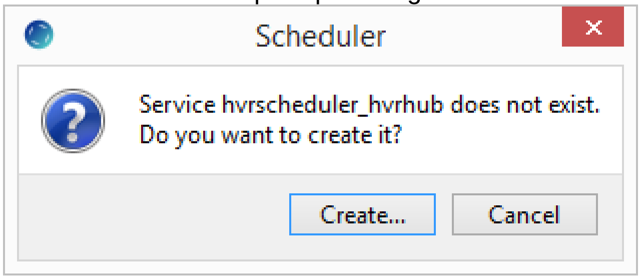
Job	State	Retries	Recent Err
hvrdemo-cap-src	SUSPEND		
hvrdemo-integ-tgt	SUSPEND		

Job/Group	Attribute
HVRDEMO-CAP	retry_delay 10 1200
	retry_delay 10 1200
	quota_speed 20 1

```
gui[hub]: Running init scripts to recrea
hvrdemo-init-src[eof]: Finished. (elaps
hvrdemo-init-tgt[eof]: Finished. (elaps
Finished. 2 replication jobs created. St                rface or run 'hvrstart -u hvrhub <job>'.
```

2. On Windows, the following steps are required to create the **HVR Scheduler** system service.

a. Click **Create...** in the prompt asking to create the service **hvrscheduler_hvrhub**.



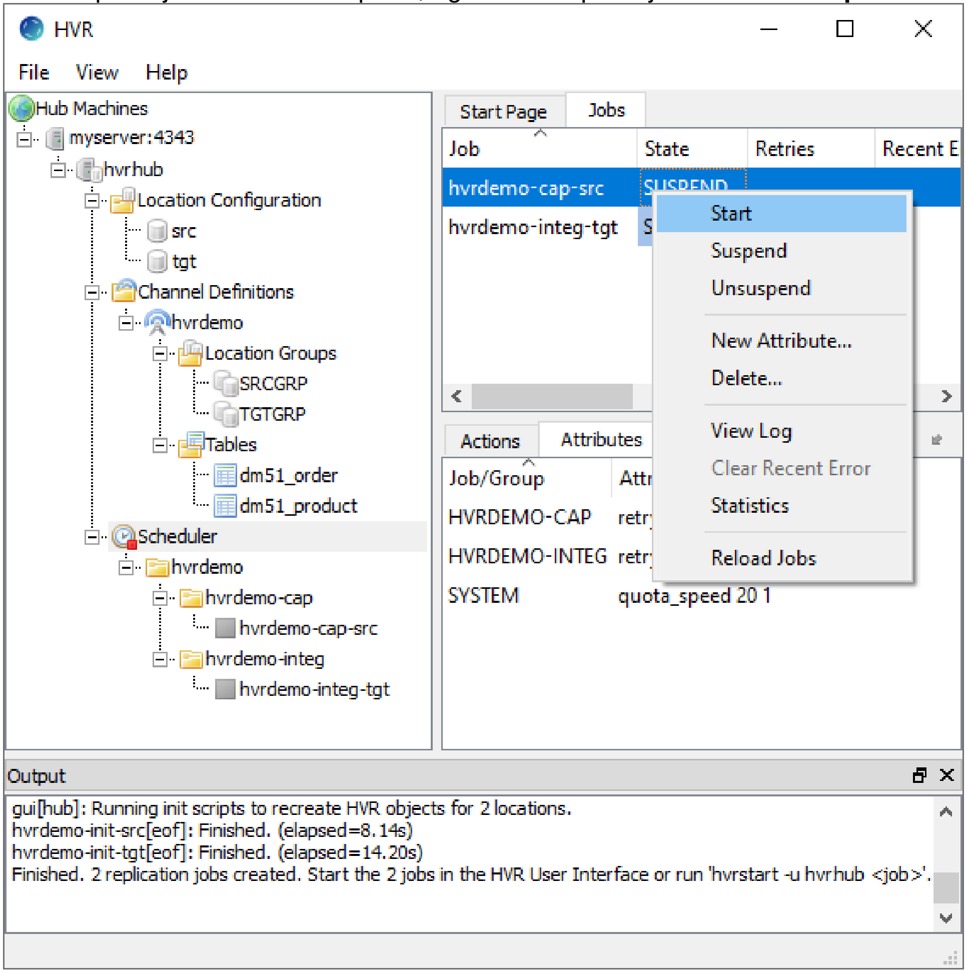
Service hvrscheduler_hvrhub does not exist.
Do you want to create it?

b. In the **Create Windows Service** dialog, select **Local System Account ('SYSTEM')** and click **Create**.

Start Capture Job

This section describes how to start the job for capturing changes from source location (**src**). By starting the **Capture** job in **HVR Scheduler**, HVR begins capturing all changes since the moment **HVR Initialize** was executed. This 'capture begin moment' can be modified using the option **Capture Rewind** available in the **Advanced Options** tab of **HVR Initialize** dialog.

1. In the navigation tree pane, click **Scheduler**.
2. Start capture job. In the **Jobs** pane, right-click capture job **hvrdemo-cap-src Start**.



Job	State	Retries	Recent E
hvrdemo-cap-src	SUSPEND		
hvrdemo-integ-tgt	S		

Job/Group	Attr
HVRDEMO-CAP	retr
HVRDEMO-INTEG	retr
SYSTEM	quota_speed 20 1

```
gui[hub]: Running init scripts to recreate HVR objects for 2 locations.
hvrdemo-init-src[eof]: Finished. (elapsed=8.14s)
hvrdemo-init-tgt[eof]: Finished. (elapsed=14.20s)
Finished. 2 replication jobs created. Start the 2 jobs in the HVR User Interface or run 'hvrstart -u hvrhub <job>'.
```

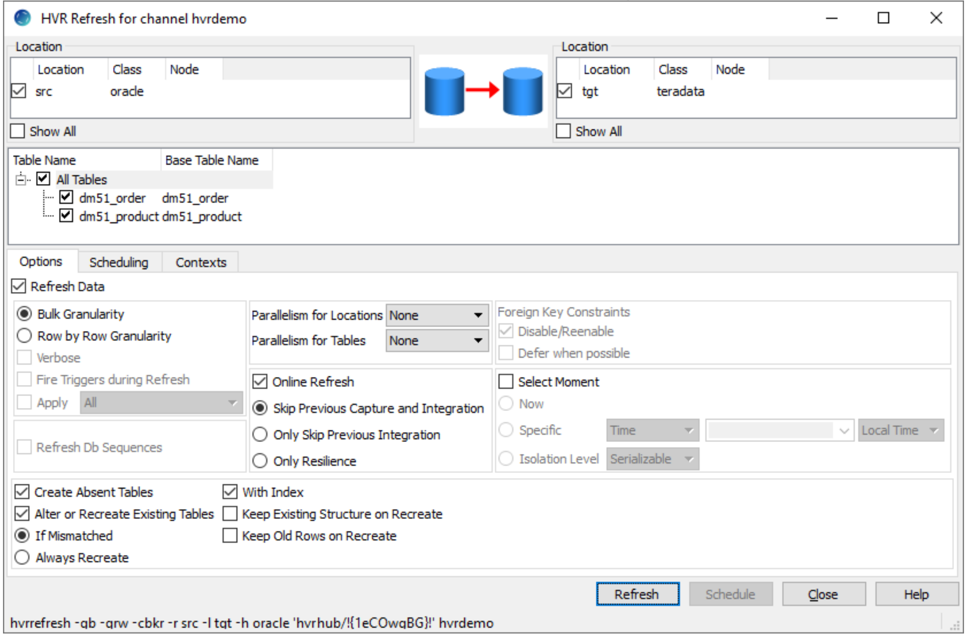
3. Click **Yes** in **Start** dialog.

   On starting the capture job (**hvrdemo-cap-src**) successfully, the status of the job changes from **SUSPEND** to **RUNNING**.

Refresh

This section describes how to perform initial load into the target database. HVR Refresh copies all existing data from source location (**src**) to the target location (**tgt**) and optionally creates new tables and keys in target location.

1. In the navigation tree pane, right-click channel **hvrdemo** **HVR Refresh**.
2. Select **Create Absent Tables** in **HVR Refresh** dialog.



HVR Refresh for channel hvrdemo

Location

	Location	Class	Node
☑	src	oracle	

☐ Show All

Location

	Location	Class	Node
☑	tgt	teradata	

☐ Show All

Table Name	Base Table Name
☑ All Tables	
☑ dm51_order	dm51_order
☑ dm51_product	dm51_product

Options | Scheduling | Contexts

☑ Refresh Data

● Bulk Granularity
○ Row by Row Granularity
☐ Verbose
☐ Fire Triggers during Refresh
☐ Apply: All
☐ Refresh Db Sequences

Parallelism for Locations: None
Parallelism for Tables: None

☑ Online Refresh
● Skip Previous Capture and Integration
○ Only Skip Previous Integration
○ Only Resilience

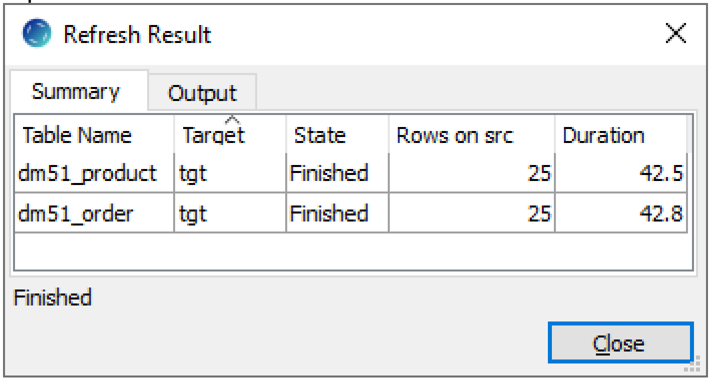
Foreign Key Constraints
☑ Disable/Reenable
☐ Defer when possible

☐ Select Moment
○ Now
○ Specific: Time, Local Time
○ Isolation Level: Serializable

☑ Create Absent Tables
☑ Alter or Recreate Existing Tables
● If Mismatched
○ Always Recreate
☑ With Index
☐ Keep Existing Structure on Recreate
☐ Keep Old Rows on Recreate

```
hvrrefresh -qb -qrw -cbkr -r src -l tgt -h oracle 'hvrhub/!{1eCOwqBG}!' hvrdemo
```

3. Click **Refresh**.
4. Click **Yes** to begin **HVR Refresh**.
   When the refresh is completed, the **Refresh Result** dialog displays the total number of rows replicated from the selected tables.



Refresh Result

Summary | Output

Table Name	Target	State	Rows on src	Duration
dm51_product	tgt	Finished	25	42.5
dm51_order	tgt	Finished	25	42.8

Finished

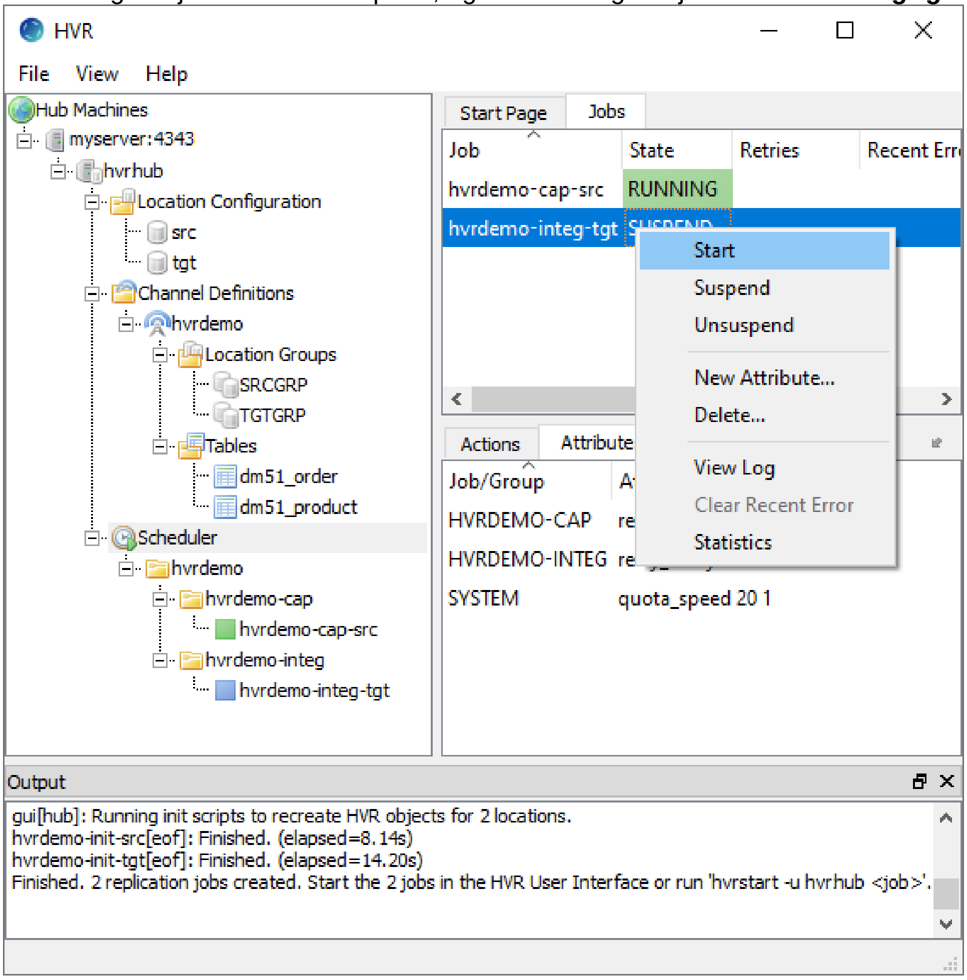
5. Click **Close** in **Refresh Result** dialog.
6. Click **Close** in **HVR Refresh** dialog.

Start Integrate Job

This section describes how to start the job to integrate changes into the target location (**tgt**).

1. In the navigation tree pane, click **Scheduler**.

2. Start integrate job. In the **Jobs** pane, right-click integrate job **hvrdemo-integ-tgt Start**.



Job	State	Retries	Recent Err
hvrdemo-cap-src	RUNNING		
hvrdemo-integ-tgt	SUSPEND		

Job/Group	A
HVRDEMO-CAP	re
HVRDEMO-INTEG	re
SYSTEM	quota_speed 20 1

```
gui[hub]: Running init scripts to recreate HVR objects for 2 locations.
hvrdemo-init-src[eof]: Finished. (elapsed=8.14s)
hvrdemo-init-tgt[eof]: Finished. (elapsed=14.20s)
Finished. 2 replication jobs created. Start the 2 jobs in the HVR User Interface or run 'hvrstart -u hvrhub <job>'.
```

3. Click **Yes** in **Start** dialog.

   On starting the integrate job (**hvr_demo-integ-tgt**) successfully, the status of the job changes from **SUSPEND** to **RUNNING**.

Verify Replication

This section describes the two methods for verifying HVR's replication activity.

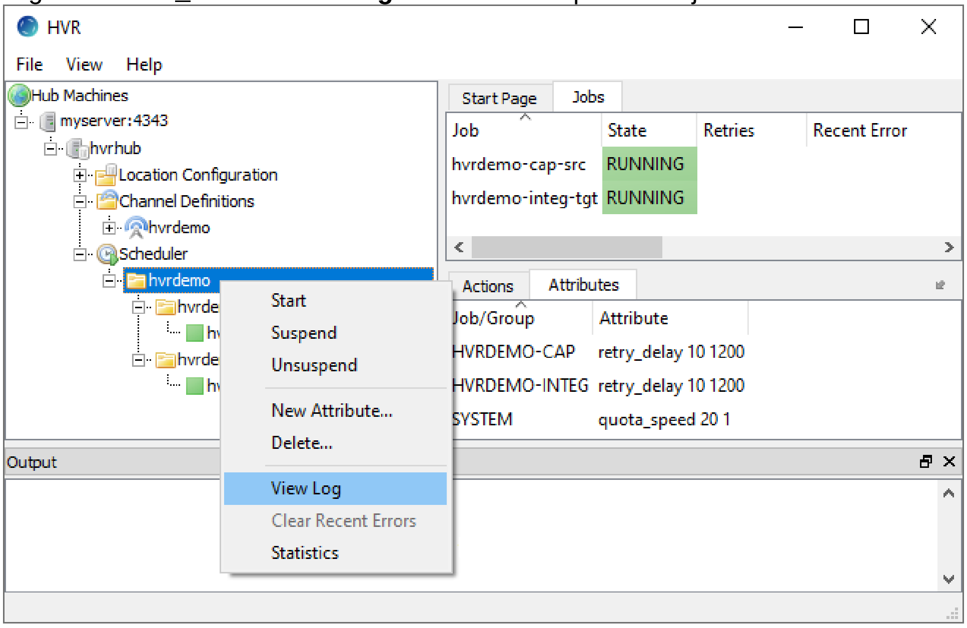
- Viewing Log File
- Using HVR Compare

Viewing Log File

HVR creates separate log file for the hub, channel (**hvrdemo**), and for each replication jobs (**hvrdemo-cap-src** and **hvrdemo-integ-tgt**). This log file contains the details of the changes captured and integrated. To view the replication activity log,

1. In navigation tree pane, click **+** next to the **Scheduler**.

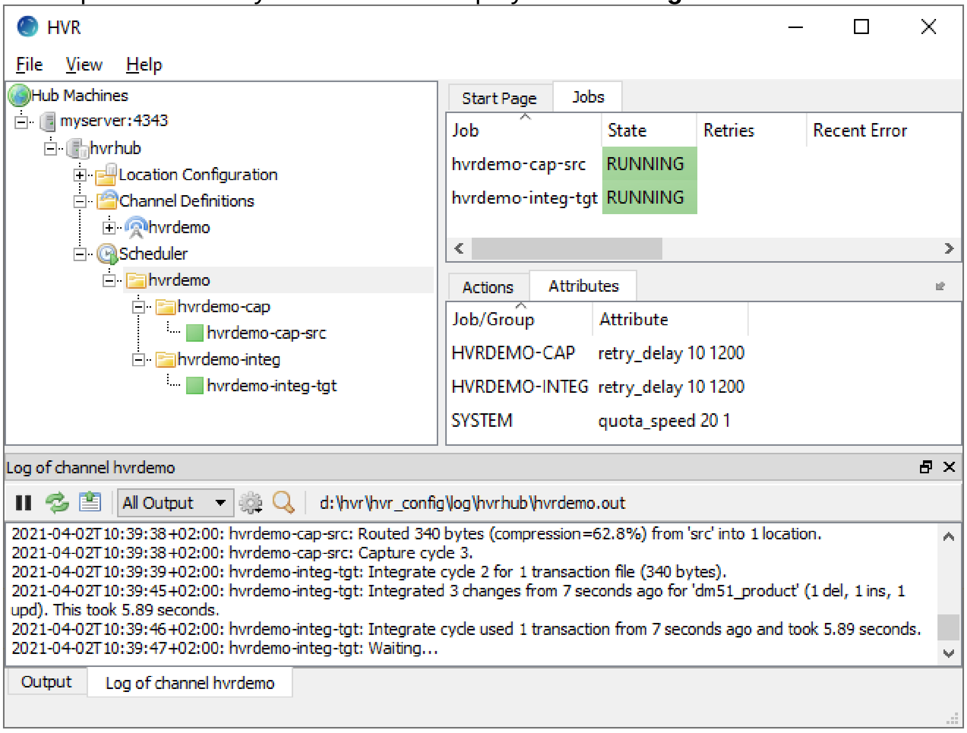
2. Right-click **hvr_demo** **View Log** to view the output of the jobs.



Job	State	Retries	Recent Error
hvrdemo-cap-src	RUNNING		
hvrdemo-integ-tgt	RUNNING		

Job/Group	Attribute
HVRDEMO-CAP	retry_delay 10 1200
HVRDEMO-INTEG	retry_delay 10 1200
SYSTEM	quota_speed 20 1

3. Update the value(s) in source location database.
4. The replication activity details will be displayed in the **Log of channel** tab.



Job	State	Retries	Recent Error
hvrdemo-cap-src	RUNNING		
hvrdemo-integ-tgt	RUNNING		

Job/Group	Attribute
HVRDEMO-CAP	retry_delay 10 1200
HVRDEMO-INTEG	retry_delay 10 1200
SYSTEM	quota_speed 20 1

d:\hvr\hvr_config\log\hvrhub\hvrdemo.out

```
2021-04-02T10:39:38+02:00: hvrdemo-cap-src: Routed 340 bytes (compression=62.8%) from 'src' into 1 location.
2021-04-02T10:39:38+02:00: hvrdemo-cap-src: Capture cycle 3.
2021-04-02T10:39:39+02:00: hvrdemo-integ-tgt: Integrate cycle 2 for 1 transaction file (340 bytes).
2021-04-02T10:39:45+02:00: hvrdemo-integ-tgt: Integrated 3 changes from 7 seconds ago for 'dm51_product' (1 del, 1 ins, 1
upd). This took 5.89 seconds.
2021-04-02T10:39:46+02:00: hvrdemo-integ-tgt: Integrate cycle used 1 transaction from 7 seconds ago and took 5.89 seconds.
2021-04-02T10:39:47+02:00: hvrdemo-integ-tgt: Waiting...
```

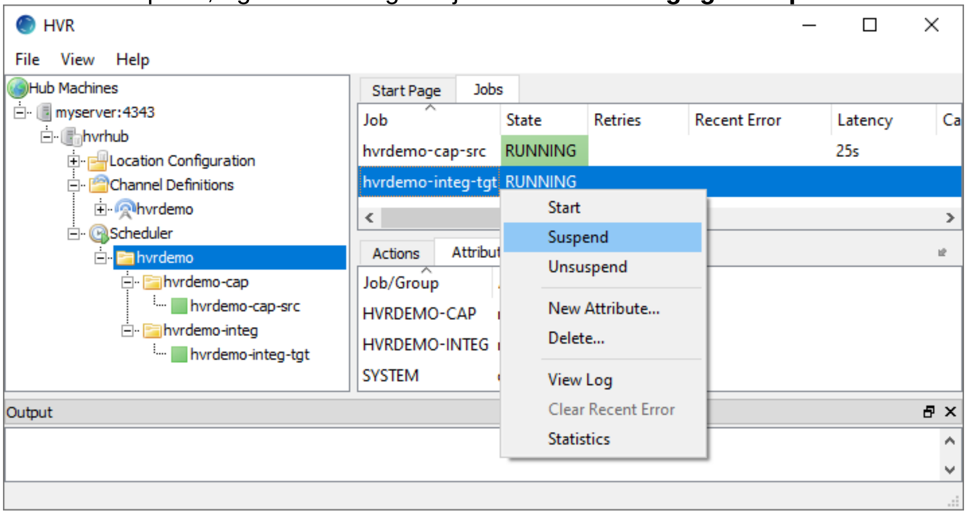
The directory path for HVR log files is displayed in the **Log of channel** tab.

Using HVR Compare

HVR Compare allows you to verify the replication activity by comparing the data in source and target locations. To compare the source and target locations,

1. Stop the Integrate job (**hvrdemo-integ-tgt**),
    a. In the navigation tree pane, click **Scheduler**.

b. In the **Jobs** pane, right-click integrate job **hvrdemo-integ-tgt Suspend**.



Job	State	Retries	Recent Error	Latency	Ca
hvrdemo-cap-src	RUNNING			25s	
hvrdemo-integ-tgt	RUNNING				

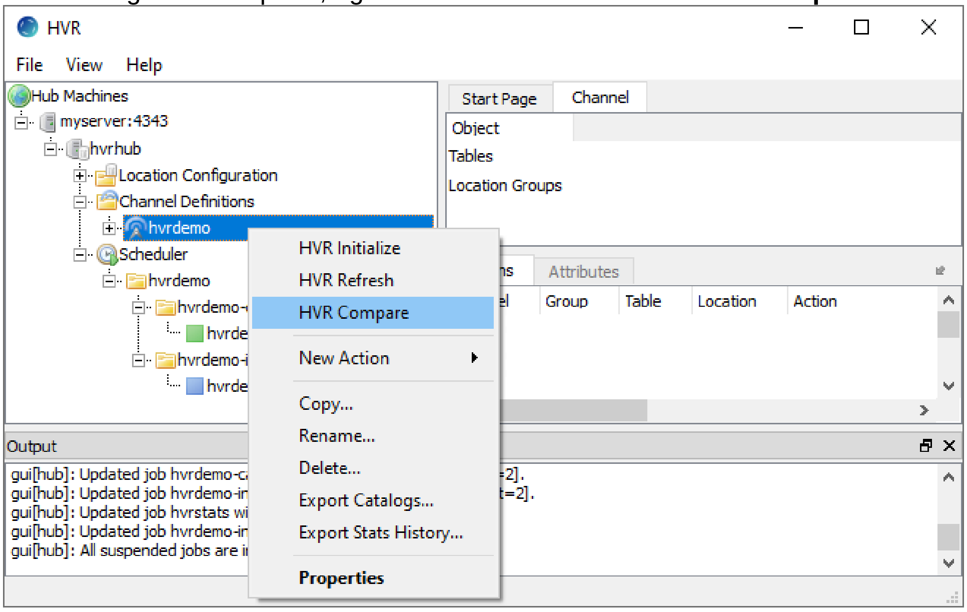
Job/Group
HVRDEMO-CAP
HVRDEMO-INTEG
SYSTEM

c. Click **Yes** in **Start** dialog.

2. Update the value(s) in source location database.

3. Execute **HVR Compare**,

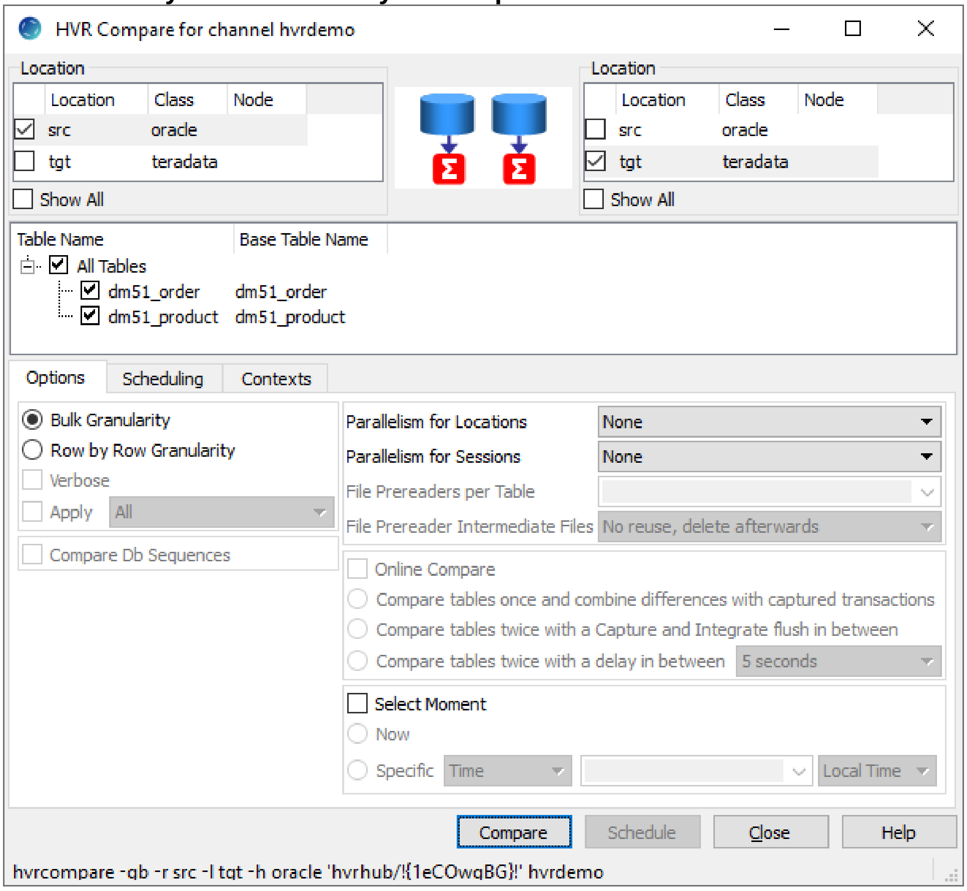
a. In the navigation tree pane, right–click channel **hvrdemo HVR Compare**.



```
gui[hub]: Updated job hvrdemo-c                               =2].
gui[hub]: Updated job hvrdemo-ir                              t=2].
gui[hub]: Updated job hvrstats wi
gui[hub]: Updated job hvrdemo-ir
gui[hub]: All suspended jobs are i
```

b. Select source location (**src**) on the left side and target location (**tgt**) on the right side.

c. Select **Row by Row Granularity** in the **Options** tab.



HVR Compare for channel hvrdemo

Location

	Location	Class	Node
☑	src	oracle	
□	tgt	teradata	

□ Show All

Location

	Location	Class	Node
□	src	oracle	
☑	tgt	teradata	

□ Show All

Table Name	Base Table Name
☑ All Tables	
☑ dm51_order	dm51_order
☑ dm51_product	dm51_product

Options | Scheduling | Contexts

● Bulk Granularity
○ Row by Row Granularity
□ Verbose
□ Apply All
□ Compare Db Sequences

Parallelism for Locations: None
Parallelism for Sessions: None
File Prereaders per Table:
File Prereader Intermediate Files: No reuse, delete afterwards

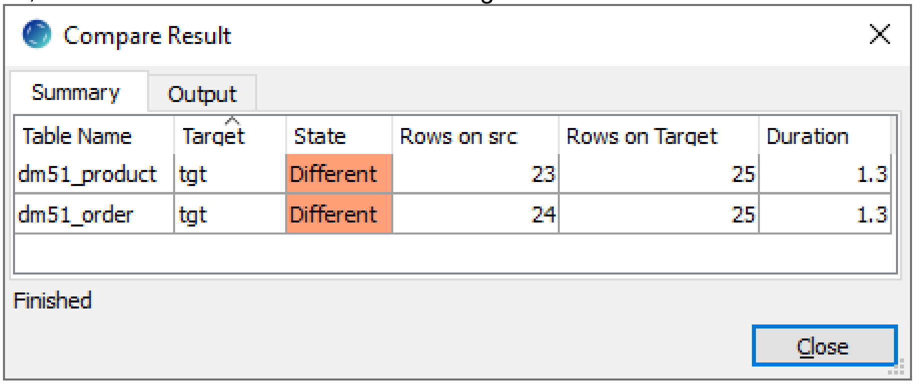
□ Online Compare
○ Compare tables once and combine differences with captured transactions
○ Compare tables twice with a Capture and Integrate flush in between
○ Compare tables twice with a delay in between: 5 seconds

□ Select Moment
○ Now
○ Specific: Time | | Local Time

```
hvrcompare -gb -r src -l tgt -h oracle 'hvrhub/!{1eCOwqBG}!' hvrdemo
```

d. Click **Compare**.

e. On completion, the **Compare Result** dialog is displayed. If the **State** column displays **Different**, it indicates the data in source and target locations are not identical.



Compare Result

Summary | Output

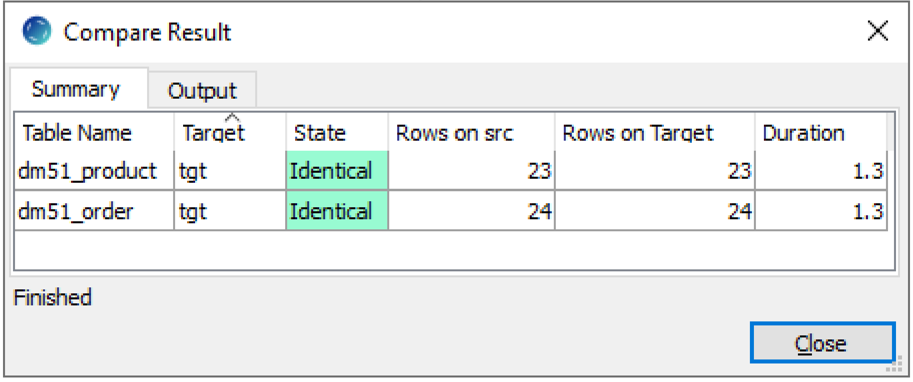
Table Name	Target	State	Rows on src	Rows on Target	Duration
dm51_product	tgt	Different	23	25	1.3
dm51_order	tgt	Different	24	25	1.3

Finished

f. Click **Close** in **Compare Result** dialog and **HVR Compare** dialog.

4. Start Integrate Job (**hvrdemo-integ-tgt**).

5. Execute **HVR Compare** again (step 3). In **Compare Result** dialog, if the **State** column displays **Identical**, it indicates all changes are replicated successfully.



Compare Result

Summary | Output

Table Name	Target	State	Rows on src	Rows on Target	Duration
dm51_product	tgt	Identical	23	23	1.3
dm51_order	tgt	Identical	24	24	1.3

Finished