Quick Start for HVR - Greenplum

Contents

This quick start guide helps you to get started with HVR for replicating data into Greenplum 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 Oracle schema (source location) to a Greenplum database (target location).

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

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

- Requirements for Oracle
- Requirements for Greenplum

Create Demo Schemas and Tables

This section describes the procedure for creating Oracle schemas and tables that are to be replicated.

Skip this section if you already have a source schema with tables which you plan to use for this replication and/or you do not want to create tables manually in the target schema.

For this demonstration, create one schema (**sourcedb**) with two tables (**dm51_product** and **dm51_order**) in source location and one schema (**targetdb**) in target location. Also, ensure to insert values into these tables in source location.

Sample SQL statements to create schema and tables in source location, and also to insert values into these 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');
```

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

Sample SQL statements to create schema and tables in target location:

Create Target Schema

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

Create Tables in Target 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)
);
```

Create Hub Database

This section describes how to create a hub database (schema). 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 on certain databases (location classes) only. For the list of supported location classes, see section Hub Database in Capabilities.

For this demonstration, the hub database (e.g. **hvrhub**) is created in Oracle.

- Create the hub database (**hvrhub**) with password (**hvr**).

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

Grants/Access Privileges

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

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 table to sourcedb;
grant select any dictionary to sourcedb;
grant select any transaction to sourcedb;
```

2. Configure the privileges for target schema (**targetdb**). For more information, see section Grants for Compare, Refresh and Integrate in Requirements for Greenplum.
3. Configure the privileges for hub schema (**hvrhub**). For more information, see section Grants for Hub Schema in Requirements for Oracle.

```
grant create session to hvrhub;
grant create table to hvrhub;
grant create procedure to hvrhub;
grant create trigger to hvrhub;
grant execute on dbms_alert to hvrhub;
```

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.

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 **Oracle** in the **Class** pane.
6. Specify **Database Connection** details.
    a. Enter the directory path in **ORACLE_HOME**. You can also click the browse button to select the directory path.
    b. Enter the Oracle System ID in **ORACLE_SID** or **TNS** credentials.
    c. Enter the user name of the hub database in **User** (e.g. **hvrhub**).
    d. Enter the password for the hub database in **Password** (e.g. **hvr**).
7. Click **Connect**.

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 source location (**src**) connected to source schema (**sourcedb**) in Oracle.
    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 in Requirements for Oracle.

    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 username of the source schema in **User**. For example, **sourcedb**.

    d. Enter password for source schema in **Password**.

5. Click **Test Connection** to verify the connection to location database.

6. Click **OK**.

- Create target location (**tgt**) connected to target schema (**targetdb**) in Greenplum.

  1. In navigation tree pane, right–click **Location Configuration ▸ New Location**.

  2. Enter **Location** name and **Description** for the location.

  3. Select **Greenplum** in **Class**.

  4. Provide **Database Connection** details. For more information on **Database Connection** fields, see section Location Connection in Requirements for Greenplum.

     a. Enter the hostname or ip-address of the Greenplum server in **Node**.

     b. Enter the **Port** number on which the Greenplum server is expecting connections.

     c. Enter the name of the Greenplum **Database**.

     d. Enter username of the target database in **User**. For example, **targetdb**.

     e. Enter password for target database in **Password**.

  5. Click **Test Connection** to verify the connection to location database.

6. Click **OK**.
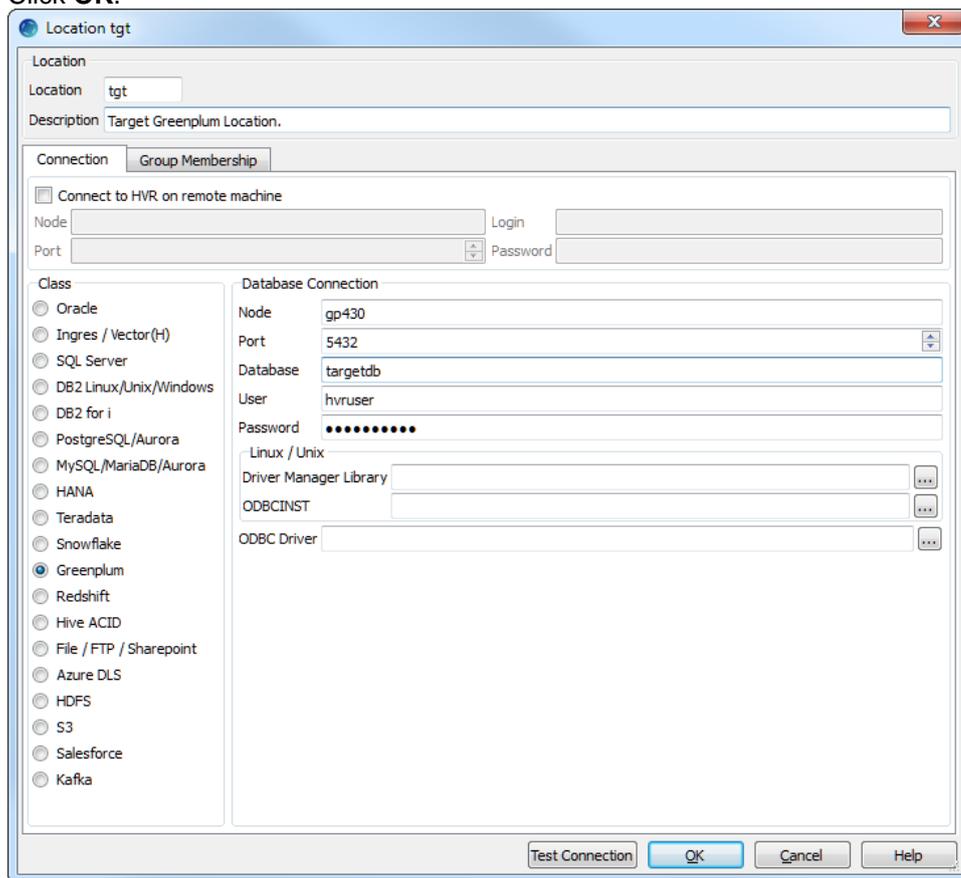
Create Channel

This section describes how to create a channel (hvrdemo) in HVR. A channel is a replication flow which captures changes from source location(s) and integrates them into target location(s).

1. In navigation tree pane, right-click **Channel Definitions** ▸ **New Channel**.
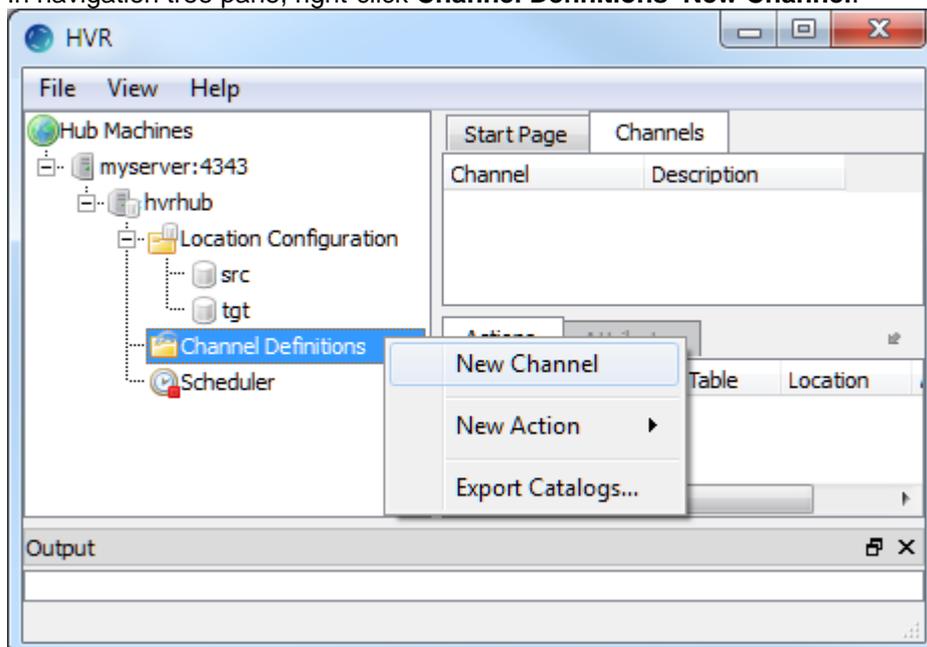
2. Enter **Channel** name and **Description** for the channel in **New Channel** dialog.
3. Click **OK**.

Create Location Groups

This section describes how to create location groups in a channel. The location groups are used for defining action 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**.
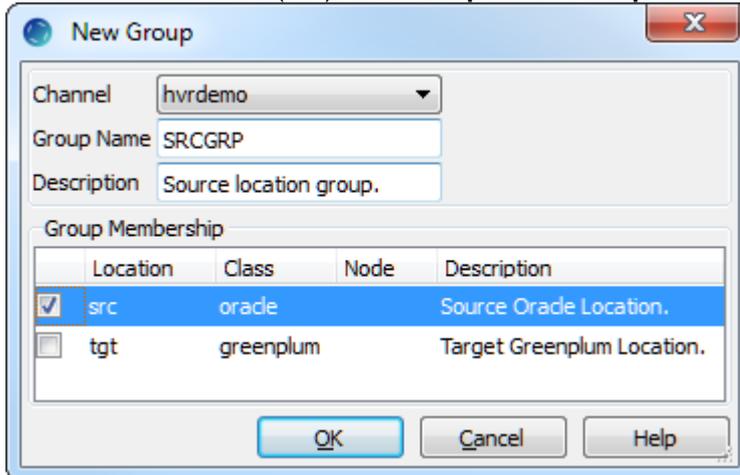
   d. Click **OK**.
3. Create target location group (**TGTGRP**):
   a. Right–click **Location Groups New Group**.
   b. Enter **Group Name** and **Description** for the location group.
   c. Select target location (**tgt**) from **Group Membership**.
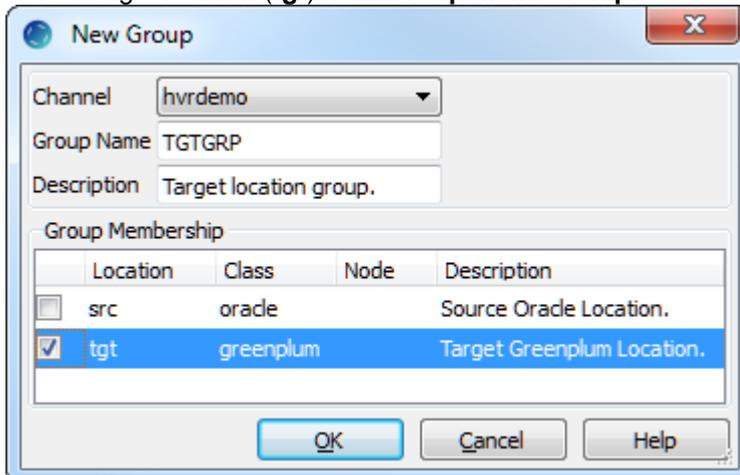

   d. Click **OK**.

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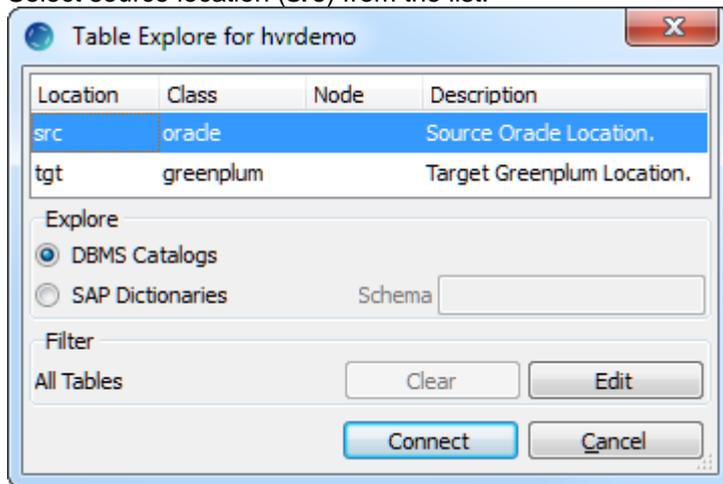
3. Click **Connect**.
4. Select tables from **Table Explore** dialog. Press **Shift** key to select multiple tables or **Ctrl+A** to select all tables.
5. Click **Add** to add the selected tables.
6. Click **OK** in **HVR Table Name** dialog.
7. Click **Close** in **Table Explore** dialog.
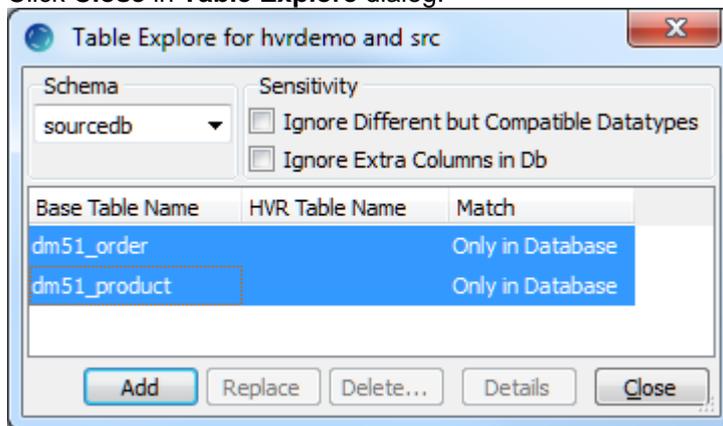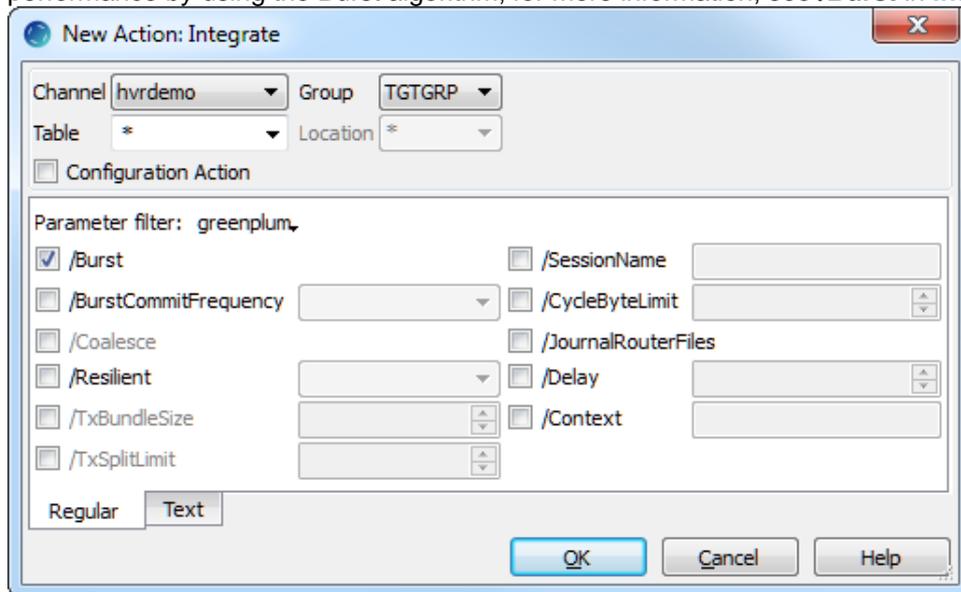

Define Actions

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**.
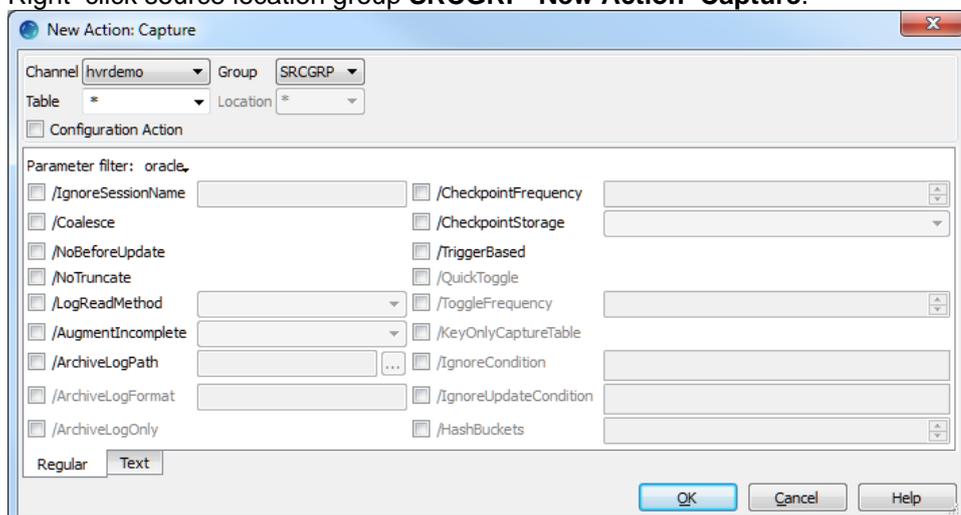
    b. Click **OK**.

2. Define action **Integrate** to integrate changes into all tables in the target location group.
   a. Right–click target location group **TGTGRP ▶ New Action ▶ Integrate**.
   b. Select parameter **/Burst**. HVR's **Integrate** with Burst improves the data replication performance by using the Burst algorithm, for more information, see **/Burst** in **Integrate**.

   c. Click **OK**.
3. Define action **LocationProperties** to use the Greenplum Parallel File Distribution (**gpfdist**) server for bulk loading operations (Bulk Refresh and Integrate with Burst) into Greenplum for maximum performance. Using Greenplum's External Tables with **gpfdist** data loading is the fastest way to load large tables into Greenplum . For more information, see section Burst Integrate and Bulk Refresh in Requirements for Greenplum.

   Before proceeding with this step, ensure to set values for **/StagingDirectoryHvr** and **/StagingDirectoryDb** as mentioned in section Burst Integrate and Bulk Refresh.
   a. Right–click target location group **TGTGRP ▶ New Action ▶ LocationProperties**.
   b. Select parameter **/StagingDirectoryHvr**.
   c. Browse and select the directory for bulk load staging files. This directory should be on the machine where HVR connects to the source database
   d. Select parameter **/StagingDirectoryDb**.
   e. Enter the local directory on the Greenplum head-node or a URL pointing to **/StagingDirectoryHvr**.
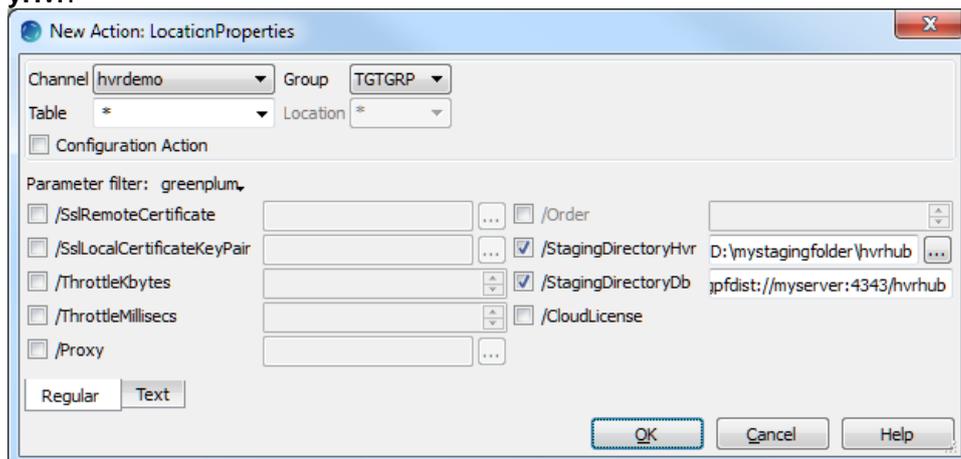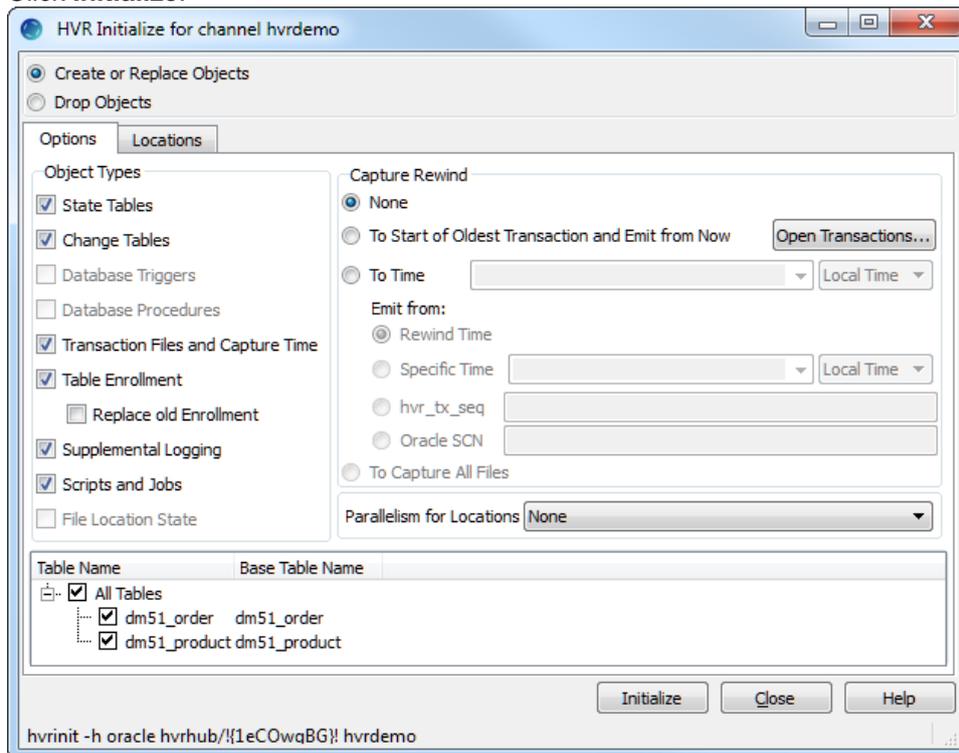

   f. Click **OK**.

   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**.

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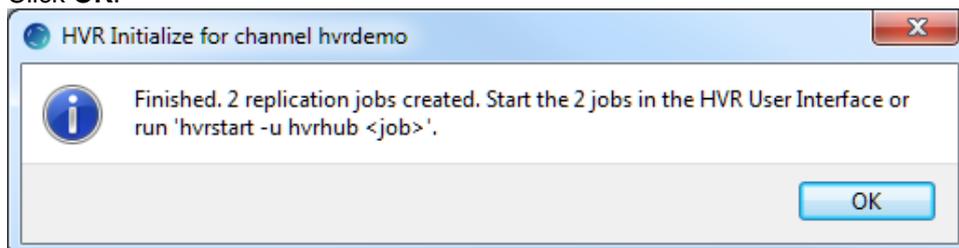
5. Click **Close**.
   Click **Scheduler** node in navigation tree pane to view the capture and integrate jobs in **Jobs** tab.

For more information about initiating replication in HVR, see Replication Overview.
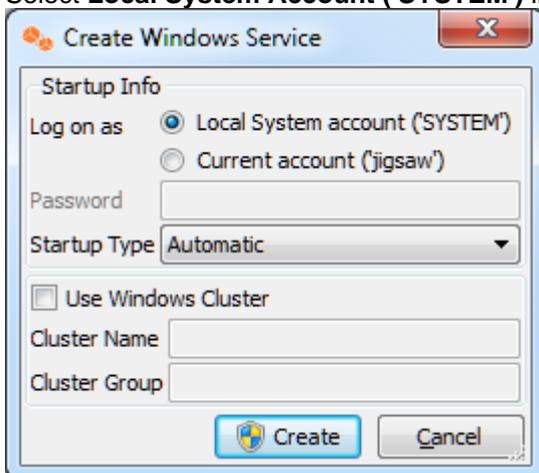
Start Scheduler and Create Windows Service

This section describes how to start the **HVR Scheduler** and create Windows service.

1. Start **Scheduler**. In the navigation tree pane, right-click **Scheduler Start**.
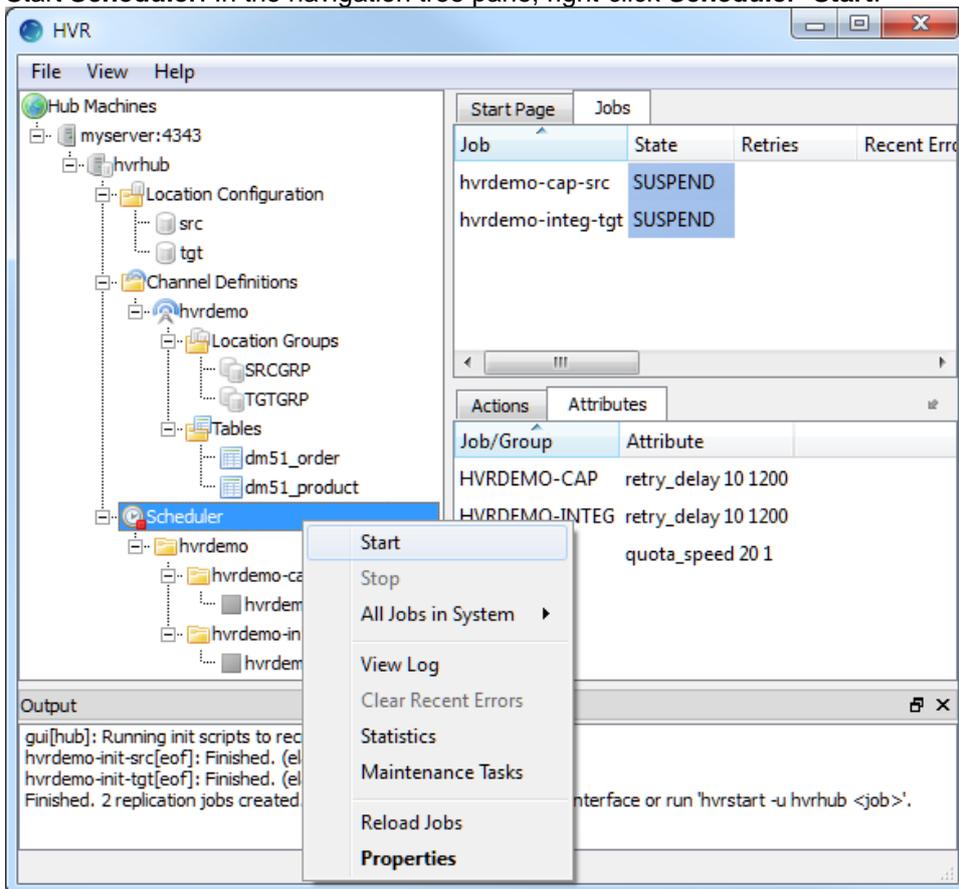
2. Click **Create...** in the prompt asking to create the service **hvrscheduler_hvrhub**.
3. Select **Local System Account ('SYSTEM')** in **Create Windows Service** dialog.

4. 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**.
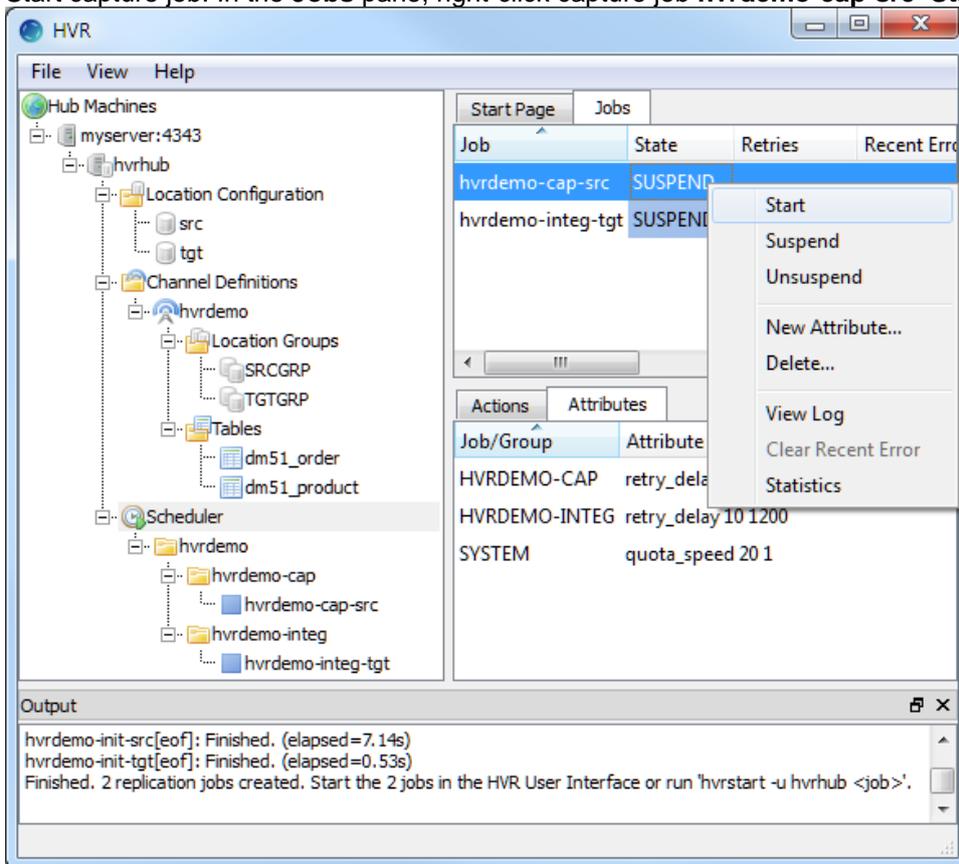
3. Click **Yes** in **Start** dialog.

   On starting the capture job (**hvrdemo-cap-src**) successfully, the status of the job changes from **SUS PEND** 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.
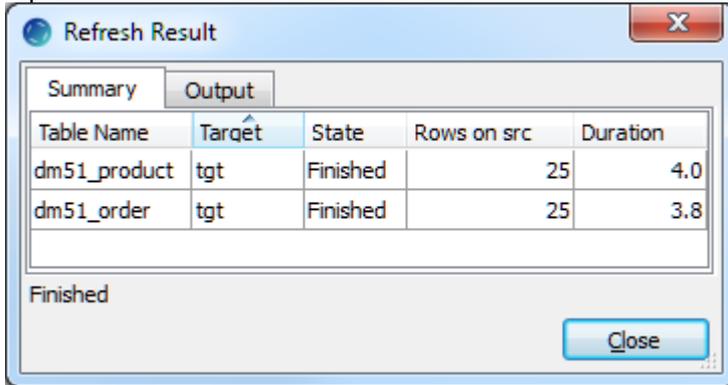
1. In the navigation tree pane, right–click channel **hvrdemo HVR Refresh**.
2. Select **Create Absent Tables**.
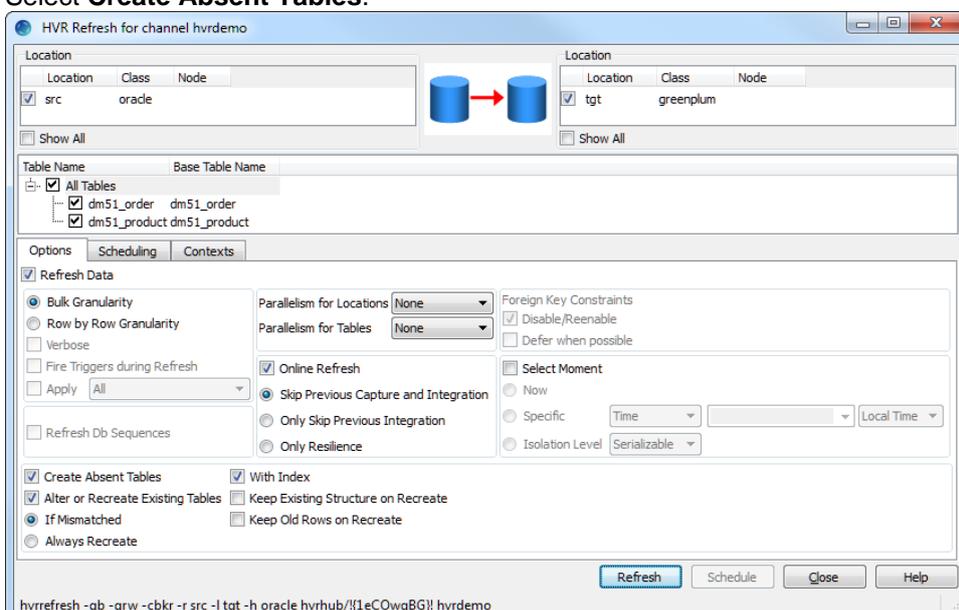
3. Click **Refresh**.

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.
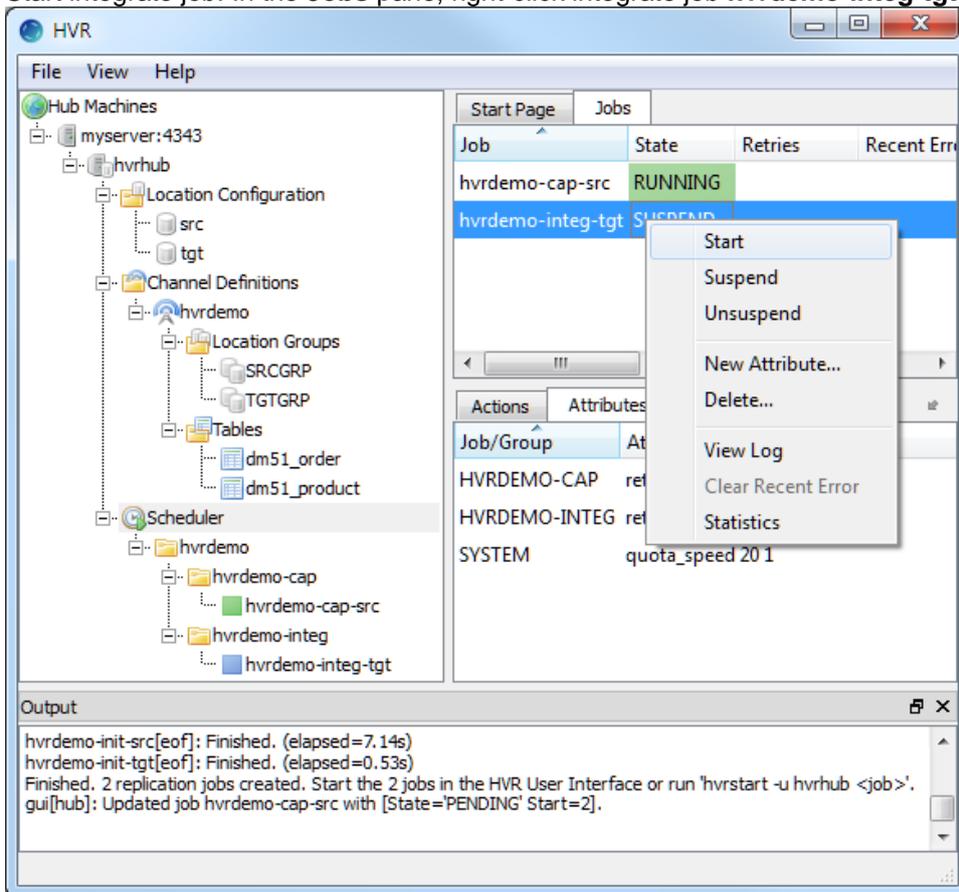

5. Click **Close** in **Refresh Result** dialog.
6. Click **Close** in **HVR Refresh** dialog.

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**.

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

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**.
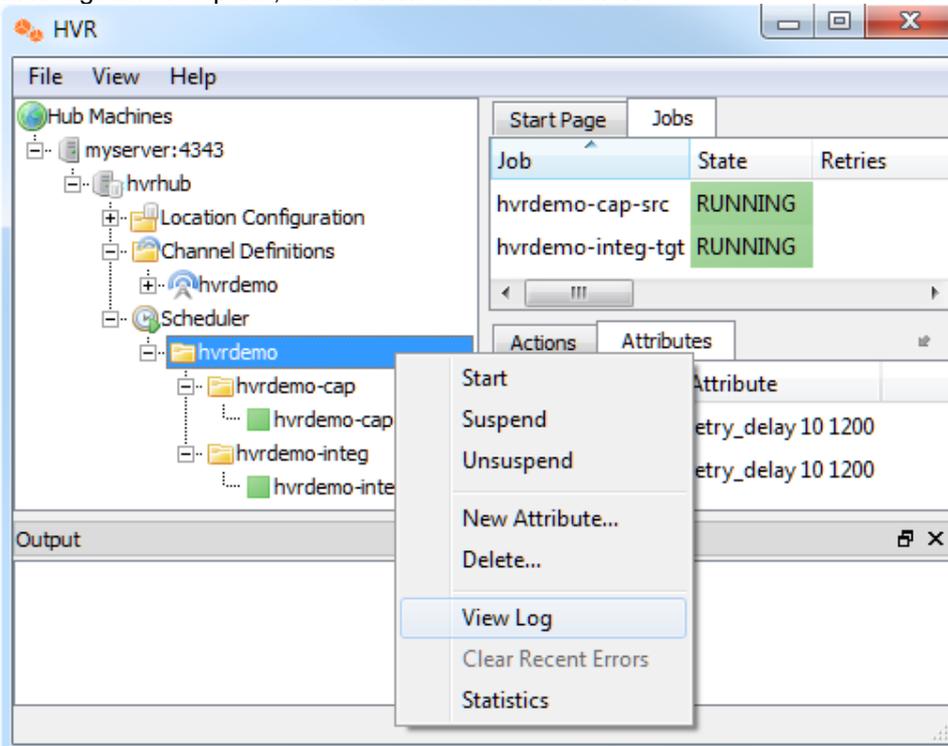
2. Right-click **hvr_demo View Log** to view the output of the jobs in **Log** tab for the selected channel (e.g. **Log of channel hvrdemo**).
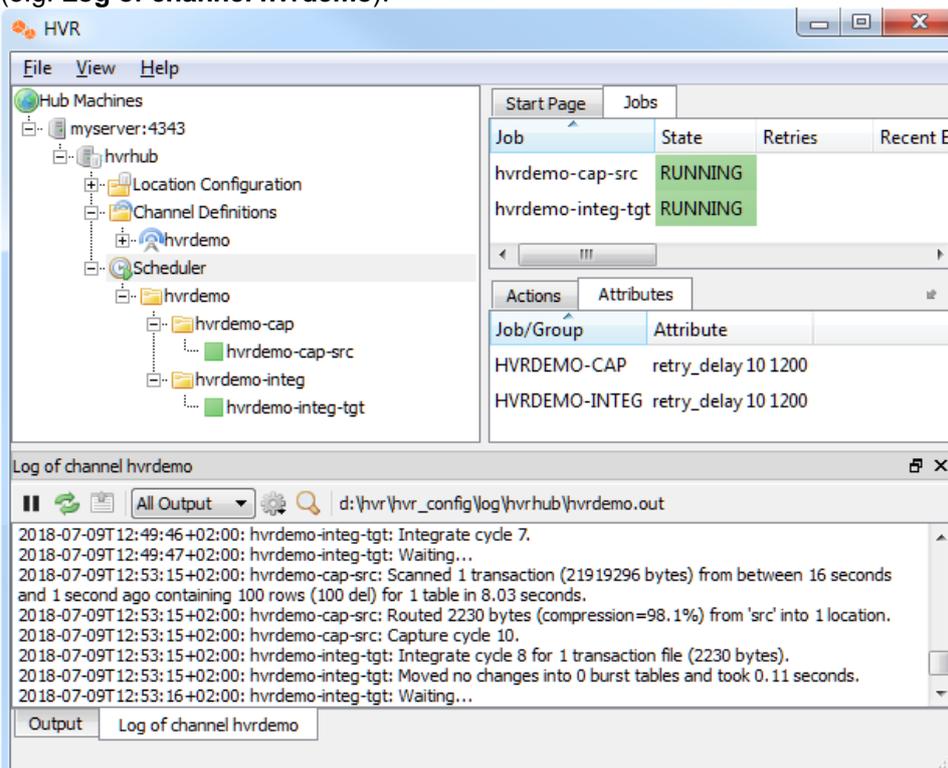
The directory path for HVR log files is displayed in the log tab.

3. Update the value(s) in source location database. The **Log** tab will immediately display the details of capture job (**hvrdemo-cap-src**) and integrate job (**hvrdemo-integ-tgt**) to reflect the capture and integrate of this change.

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**.
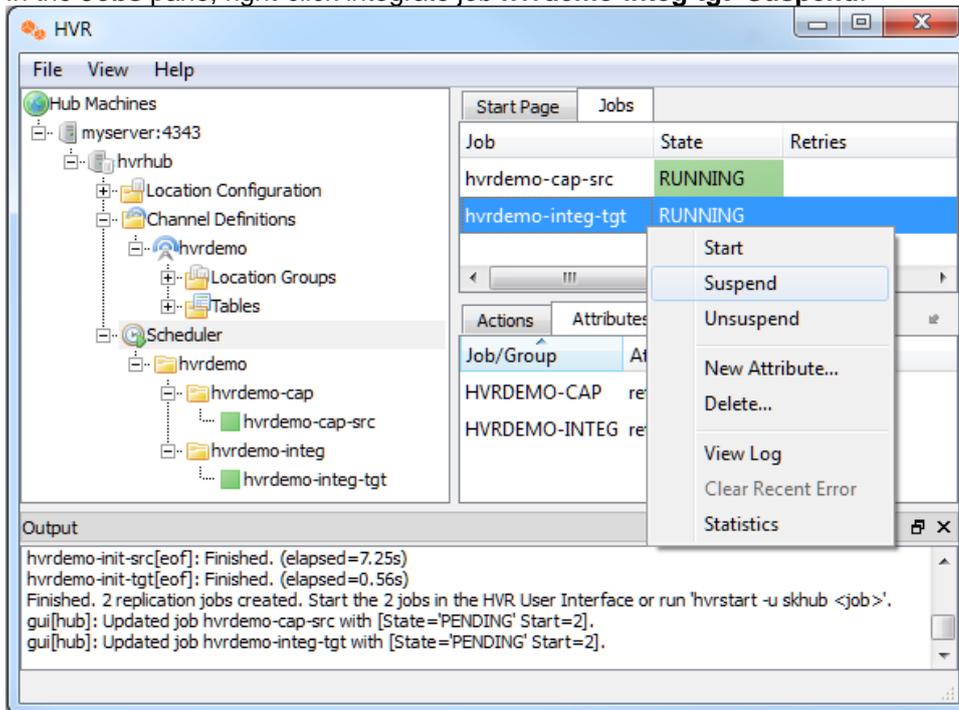
   c. Click **Yes** in **Start** dialog.
2. Update the value(s) in source location database.
3. Execute **HVR Compare**,
   a. In the navigation tree pane, right–click channel **hvrdemo** **HVR Compare**.
   b. Select source location (**src**) on the left side and target location (**tgt**) on the right side.

c. Select **Row by Row Granularity** in the **Options** tab.
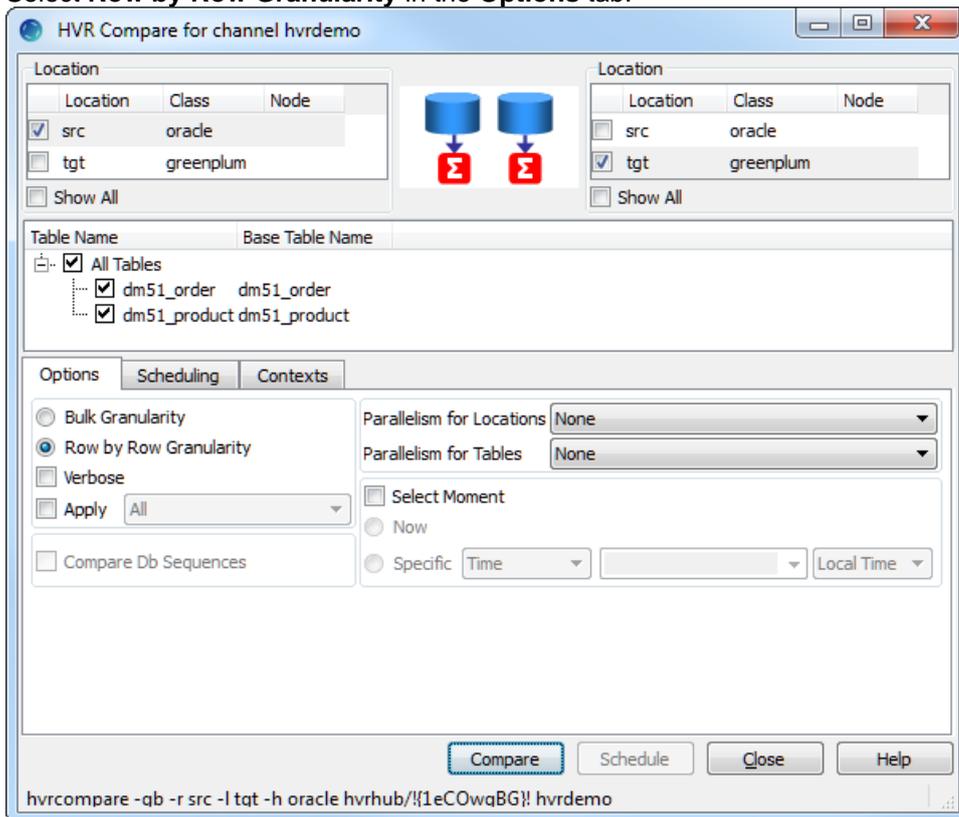
d. Click **Compare**.
e. On completion, **Compare Result** dialog is displayed. If the **State** column displays **Different**, it indicates the data in source and target locations are not identical.
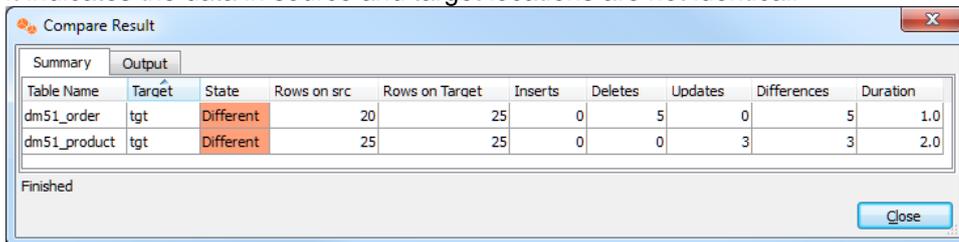
f. Click **Close** in **Compare Result** dialog and **HVR Compare** dialog.
4. Start Integrate Job (**hvrdemo-integ-tgt**).
5. Execute **HVR Compare** again (step 3). In **Compare Result** dialog, if the **State** column displays **Iden tical**, it indicates the changes are replicated successfully.