

Quick Start for HVR - SQL Server

Contents

- [Create Test Databases and Tables](#)
- [Create the Hub Database](#)
- [Download and Install HVR](#)
- [Launch HVR GUI](#)
- [Register Hub](#)
- [Create SQL Server Locations](#)
- [Create Location Groups](#)
- [Define Actions](#)
- [Enable Replication with HVR Initialize](#)
- [Start Scheduling of Replication Jobs](#)
- [Test Replication](#)
- [HVR Compare and Refresh](#)

This quick start guide helps you to get started with HVR for replicating data between SQL Server databases.

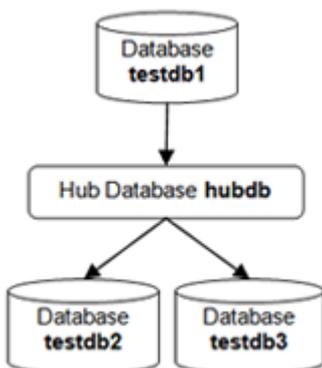
To proceed with this replication you must have basic understanding about HVR's [architecture](#) and [terminologies](#) like Hub, Location, Channel, Location Groups, Actions etc.

The example here demonstrates how to replicate tables from one SQL Server database (source location) to another SQL Server database (target location).

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 SQL Server are met.

For information about access privileges and advanced configuration changes required for performing replication using SQL Server, see [Requirements for SQL Server](#).



Create Test Databases and Tables

Generally when getting started with HVR a source schema with tables and data already exists. If so then this step can be skipped.

This Quickstart uses two empty tables named dm01_product and dm01_order. In an existing SQL server database, create a test schema and create the tables using the following commands.

Make sure the database is setup for log-based capture as described in [Grants for log-based capture](#).

In SQL Server Management Studio, create database **testdb1**. Next, create the test tables:

```
C:\> cd %HVR_HOME%\demo\hvr_demo01\base\sqlserver
C:\> osql -U hvr -P hvr -d testdb1 < hvr_demo01.cre
C:\> osql -U hvr -P hvr -d testdb1 < hvr_demo01.mod
```

For the target, create two test databases, each containing two empty tables named **dm01_product** and **dm01_order**.

In SQL Server Management Studio, create databases **testdb2** and **testdb3**. You can either create the tables using HVRs scripts or let HVR create them during initial loading (**HVR Refresh with Create Absent Tables**).

Create the test tables using HVRs script:

```
C:\> cd %HVR_HOME%\demo\hvr_demo01\base\sqlserver
C:\> osql -U hvr -P hvr -d testdb2 < hvr_demo01.cre
C:\> osql -U hvr -P hvr -d testdb2 < hvr_demo01.mod
C:\> osql -U hvr -P hvr -d testdb3 < hvr_demo01.cre
C:\> osql -U hvr -P hvr -d testdb3 < hvr_demo01.mod
```

Create the Hub Database

This section describes how to create a hub database. 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 SQL Server.

Create the hub database using the SQL Server Management Studio. Alternatively, use the following SQL statement:

```
create database 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 **HVR GUI** is launched for the first time, the **Register Hub** dialog is displayed automatically. The **Register Hub** dialog can also be accessed from the main menu **File Register Hub**.

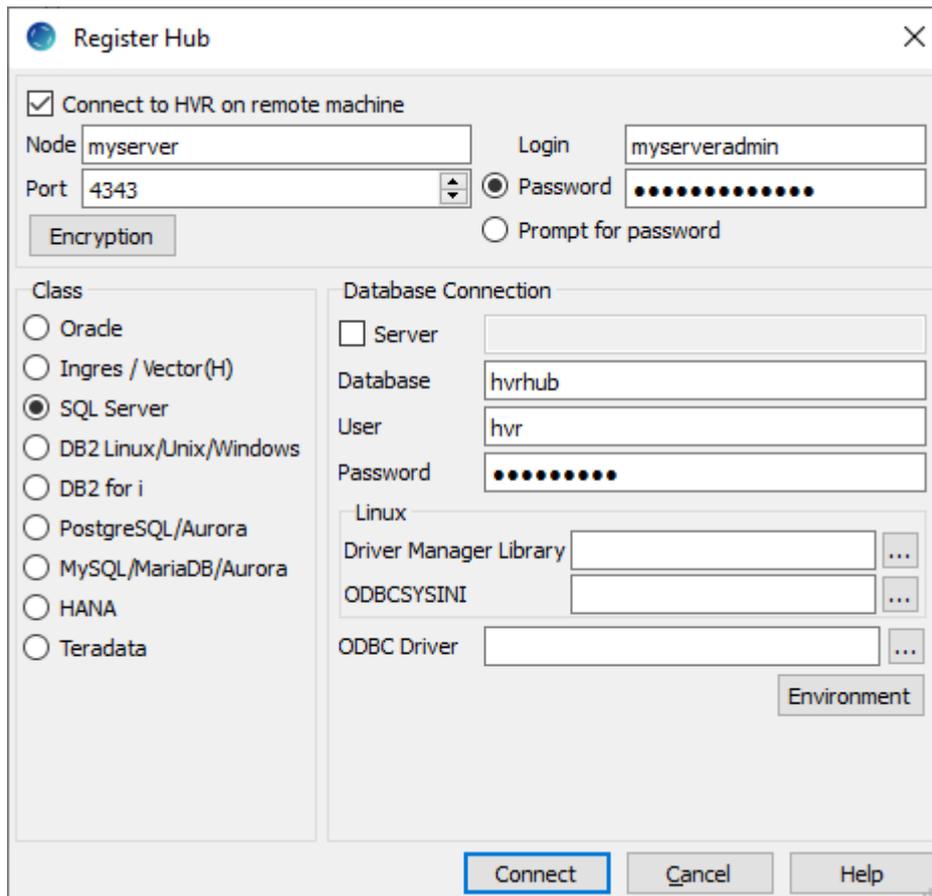
Skip steps 1 to 4, if **HVR GUI** is executed directly on the hub machine or if HVR hub is connected to a remote SQL Server database without using the SQL Server protocol. For more information about connecting HVR hub with remote SQL Server database, see section [Connecting HVR Hub to a Remote SQL Server Database](#) in [Requirements for SQL Server](#).

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 **SQL Server** in the **Class** pane.
6. Specify **Database Connection** details. For more information about the **Database Connection** fields, see section [Location Connection](#).
 - a. Enter the HVR hub database name in the **Database** field. For example, **hvrhub**.
 - b. Enter the SQL Server user name in the **User** field. This username is used to connect HVR to the SQL Server database. For example, **hvr**.
 - c. Enter the password for the SQL Server user in the **Password** field.

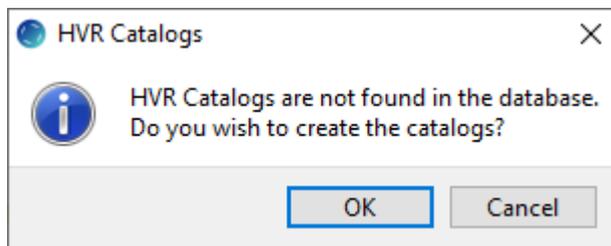
7. Click **Connect**.



The 'Register Hub' dialog box is shown with the following configuration:

- Connect to HVR on remote machine
- Node: myserver
- Login: myserveradmin
- Port: 4343
- Password (masked with dots)
- Prompt for password
- Encryption: (button)
- Class: SQL Server
- Database Connection: Server
- Database: hvrhub
- User: hvr
- Password: (masked with dots)
- Linux: Driver Manager Library ...
- ODBCSYSINI: ...
- ODBC Driver: ...
- Environment: (button)
- Buttons: Connect, Cancel, Help

8. Click **OK** 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.



The 'HVR Catalogs' dialog box displays the following message:

HVR Catalogs are not found in the database. Do you wish to create the catalogs?

Buttons: OK, Cancel

Upon successful connection to the hub database, the navigation tree pane displays the hub machine and the hub database. **Location Configuration**, **Channel Definitions**, and **Scheduler** are displayed under the hub database.

Create SQL Server Locations

Next create three locations (one for each test database) using right-click on **Location Configuration** **New Location**.

In this example there is no need to check **Connect to HVR on remote machine** because **testdb1** is on the same machine as the hub.

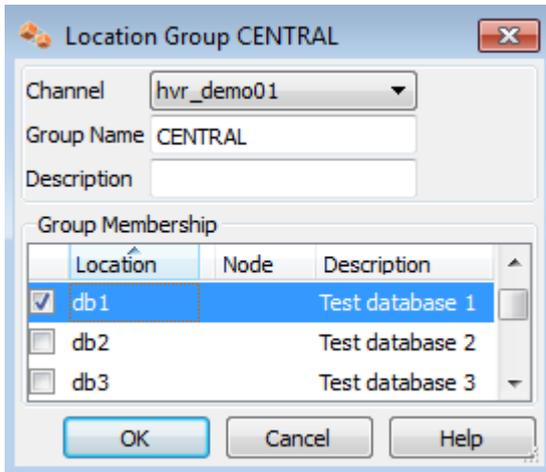
Ignore the **Group Membership** tab for now.

Make locations for **testdb2** and **testdb3** too.

Now define a channel using **Channel Definitions New Channel**.

Create Location Groups

The channel needs two location groups. Under the new channel, right-click **Location Groups New Group**. Enter a group name (for instance **CENTRAL**).



Add location **db1** as a member of this group by checking the box for **db1**.

Then create a second location group, called **DECENTRAL** that has members **db2** and **db3**.

The new channel also needs a list of tables to replicate. This can be done as follows: right-click **Tables Table Explore**.

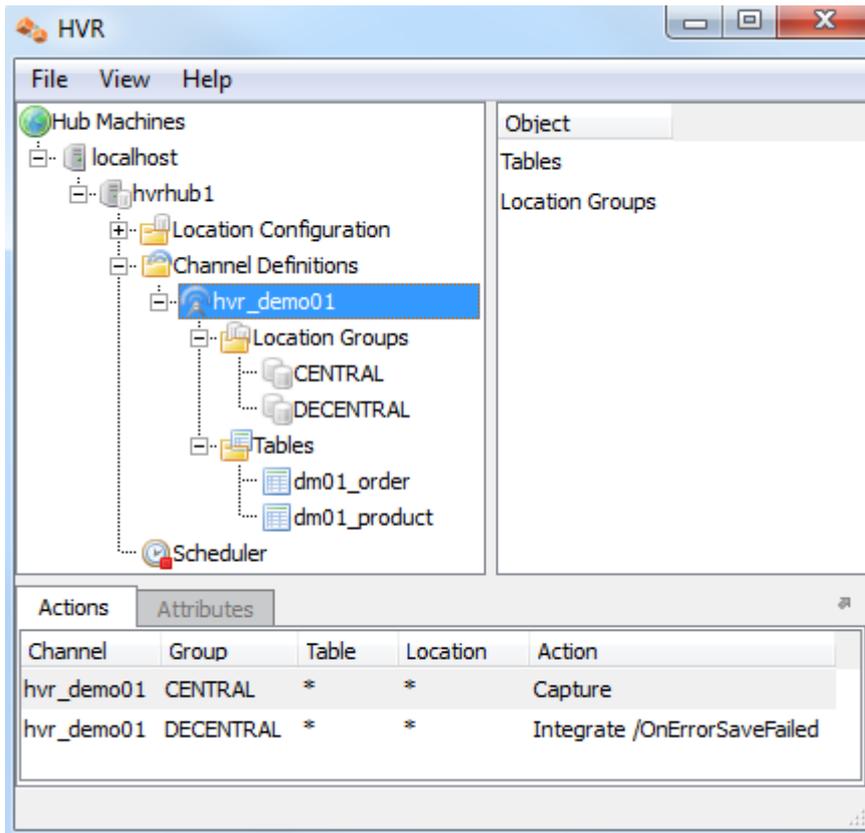
- Choose the first of the three locations **Connect**.
- In the **Table Explore** window, click on both tables and click **Add**.
- In new dialog **HVR Table Name** click **OK**.
- Close the **Table Explore** window.
- Perform table select again on one of the other locations and confirm that all tables to be replicated have value **Same** in column **Match**.

Define Actions

The new channel needs two actions to indicate the direction of replication.

- Right-click group **CENTRAL** **New Action Capture**.

- Right-click **Group DECENTRAL** **New Action Integrate**. Check **/OnErrorSaveFailed**, this affects how replication errors are handled.

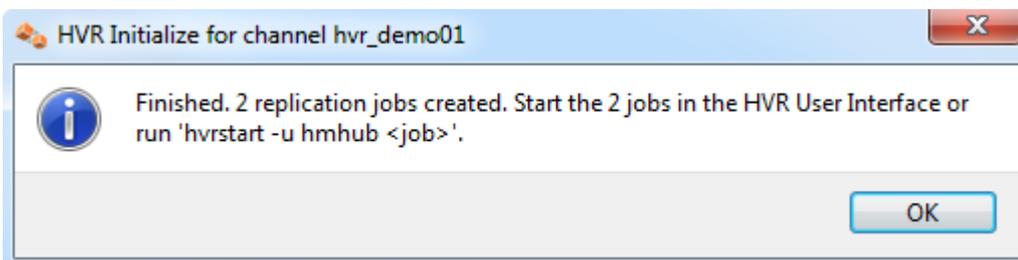


Note that the **Actions** pane only displays actions related to the objects selected in the left-hand pane. So click channel **hvr_demo01** to see both actions.

Enable Replication with HVR Initialize

Now that the channel definition is complete, create the runtime replication system.

1. Right-click channel **hvr_demo01** **HVR Initialize**.
2. Choose **Create or Replace Objects** and click **HVR Initialize**.

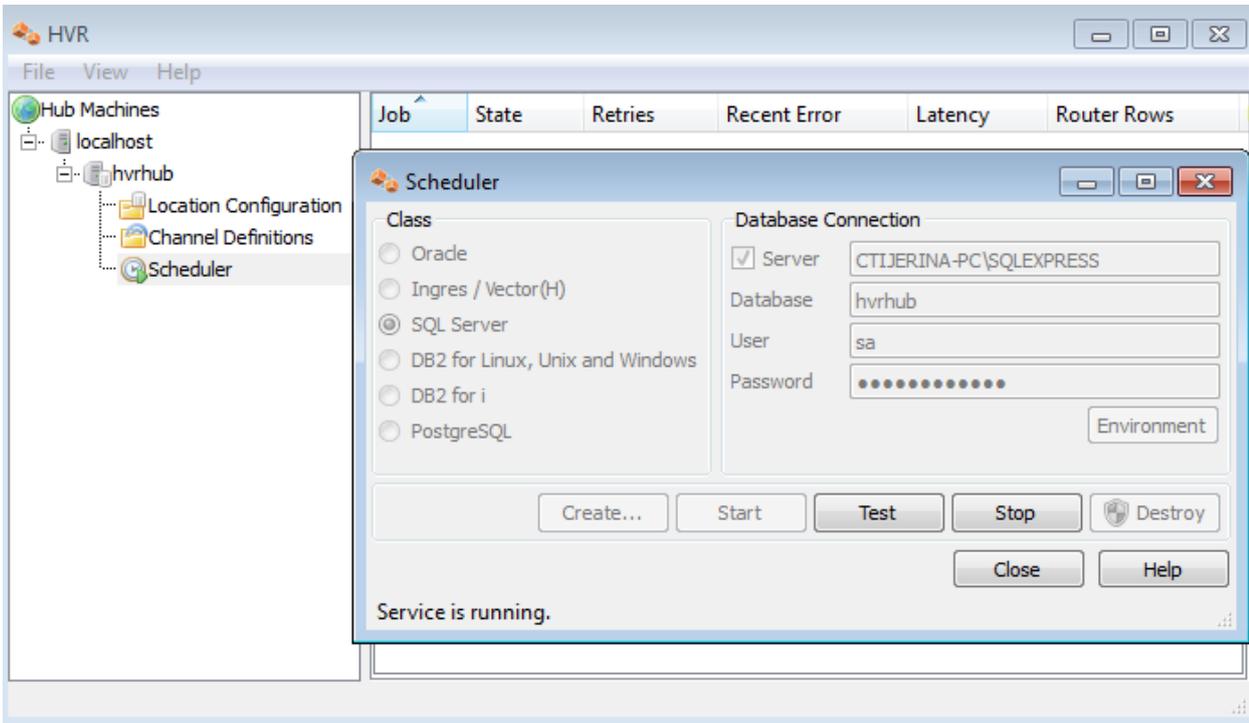


From the moment that HVR Initialize is done, all changes to database **sourcedb** will be captured by HVR when its capture job looks inside the logging.

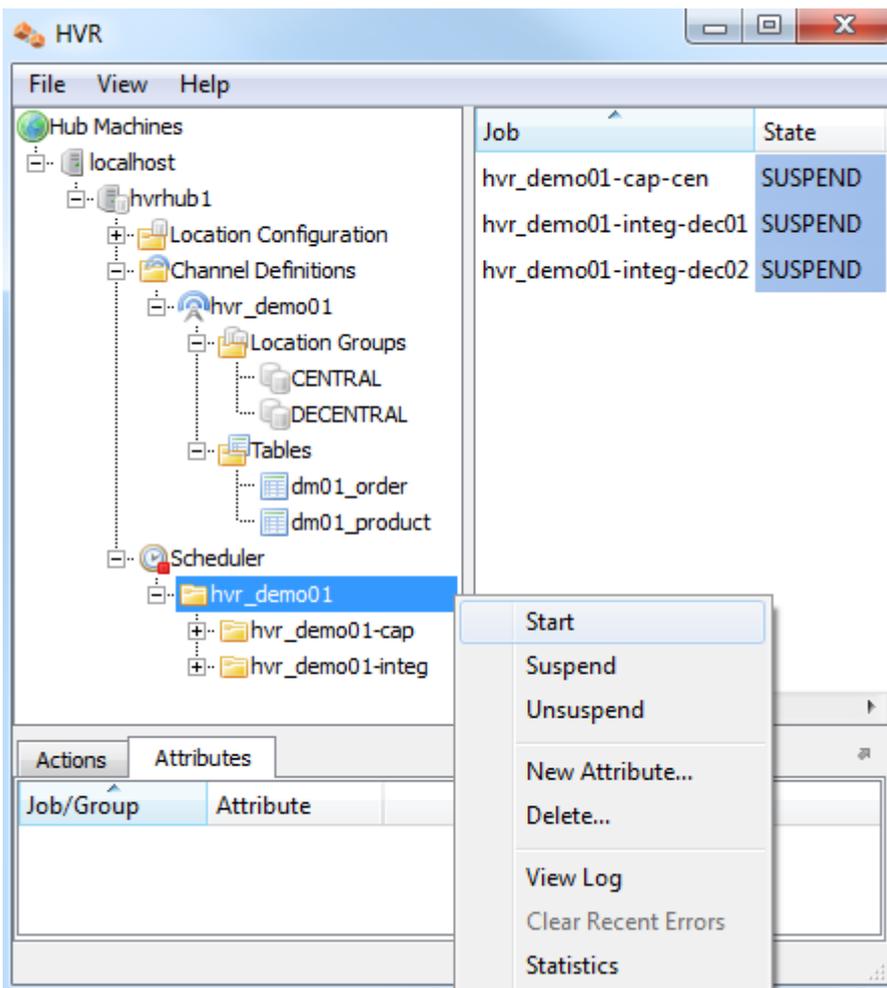
HVR initialize also creates three replication jobs, which can be seen under the **Scheduler** node in the GUI.

Start Scheduling of Replication Jobs

Start the Scheduler on the hub machine by clicking in the HVR GUI on the **Scheduler** node of the hub database.



Next, instruct the HVR Scheduler to trigger the replication jobs.



The replication jobs inside the Scheduler each execute a script under **\$HVR_CONFIG/job/hvrhub/hvr_demo01** that has the same name as the job. So job **hvr_demo01-cap-db1** detects changes on

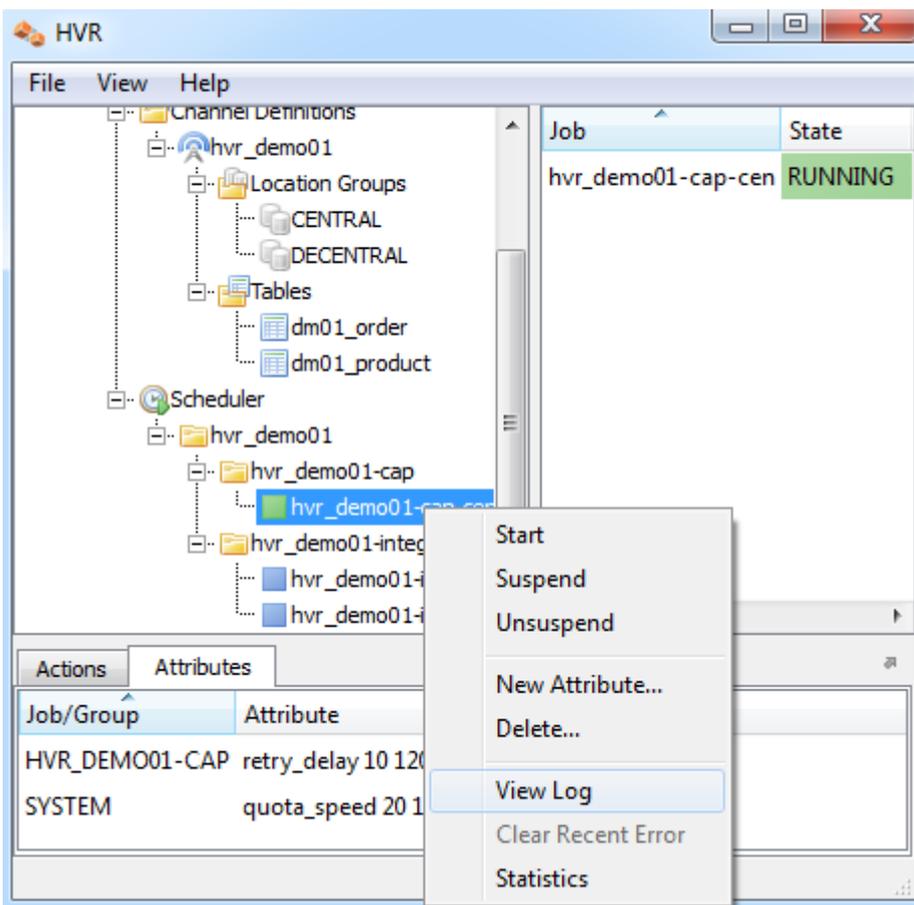
database **testdb1** and stores these as transactions files on the hub machine. The other two jobs (**hvr_demo01-integ-db2** and **hvr_demo01-integ-db3**) pick up these transaction files and perform inserts, updates and deletes on the two target databases.

Test Replication

To test replication, make a change in **testdb1**:

```
SQL> insert into dm01_product values (1, 19.99, 'DVD');
```

In the HVR log file you can see the output of the jobs by clicking on **View Log**. This log file can be found in `*%HVR_CONFIG%\log*hubdb\hvr_demo01-cap-db1`.



The job output looks like this:

```
C:\> notepad %HVR_CONFIG%\log\hvrhub\hvr.out
```

```
hvr_demo01-cap-db1: Capture cycle 1.
hvr_demo01-cap-db1: Selected 1 row from 'dm01_product__c0' (201 wide).
hvr_demo01-cap-db1: Routed 212 bytes (compression=42.7%) from 'db1' into \
2 locations.
hvr_demo01-cap-db1: Finished. (elapsed=1.20s)
hvr_demo01-integ-db2: Integrate cycle 1 for 1 transaction file (212 bytes).
hvr_demo01-integ-db2: Integrated 1 change from 'dm01_product' (1 ins).
hvr_demo01-integ-db2: Integrate used 1 transaction and took 0.017 seconds.
hvr_demo01-integ-db2: Finished. (elapsed=0.14s)
hvr_demo01-integ-db3: Integrate cycle 1 for 1 transaction file (212 bytes).
```

```

hvr_demo01-integ-db3: Integrated 1 change from 'dm01_product' (1 ins).
hvr_demo01-integ-db3: Integrate used 1 transaction and took 0.02 seconds.
hvr_demo01-integ-db3: Finished. (elapsed=0.15s)

```

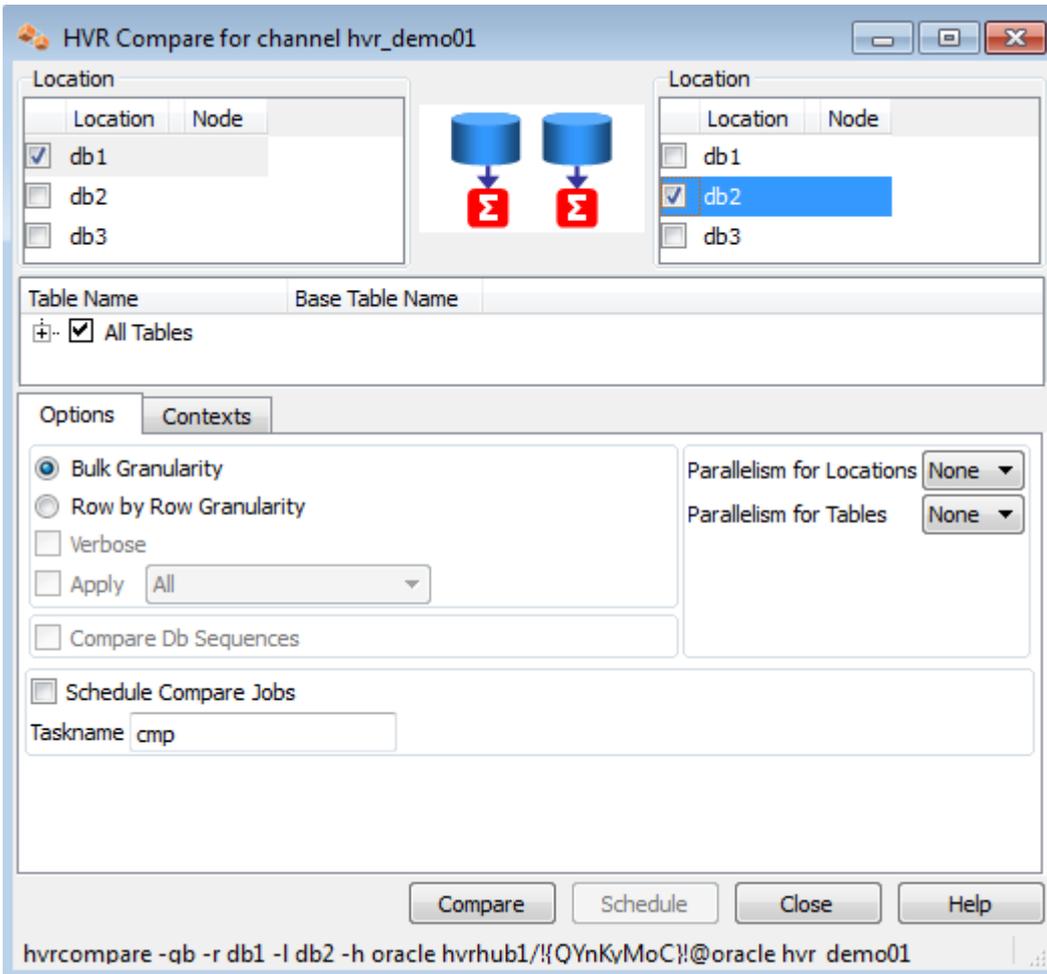
This indicates that the jobs replicated the original change to **testdb2** and **testdb3**. A query on **testdb2** confirms this:

```
SQL> select * from dm01_product;
```

prod_id	prod_price	prod_descrip
1	19.99	DVD

HVR Compare and Refresh

HVR Compare checks whether two locations have identical rows, and HVR Refresh copies the content of one location to the second location. In the HVR GUI, right-click on a channel and select **HVR Compare** (or **HVR Refresh**). Choose source and target locations.



The outcome of the comparison is displayed below;

Compare Result

Summary Output

Table Name	Target	State	Rows on oracl	Rows on Target	Duration
dm01_product	dec01	Identical	0	0	0.0
dm01_order	dec01	Identical	0	0	0.0

Finished

Close