

# Quick Start for HVR - Kafka

## Contents

- [Create Test Schema and Tables](#)
- [Create the Hub Database](#)
- [Download and Install HVR](#)
- [Launch HVR GUI](#)
- [Register Hub](#)
- [Create Locations](#)
- [Create Channel](#)
- [Create Location Groups](#)
- [Create Tables](#)
- [Define Actions](#)
- [Initialize](#)
- [Start Scheduler and Capture Job](#)
- [Refresh](#)
- [Start Integrate Job](#)

This quick start guide helps you to get started with HVR for replicating data into Kafka.

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 two Kafka locations (target locations) - one with JSON format messages and other with Confluent's Schema Registry as its 'micro' AVRO format..

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

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

- [Requirements for Oracle](#)
- [Requirements for Kafka](#)

## Create Test Schema and 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');
```

## Create the 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;
```

## 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**.

The screenshot shows the 'Register Hub' dialog box. At the top, there is a checkbox labeled 'Connect to HVR on remote machine' which is checked. Below this, there are input fields for 'Node' (containing 'myserver'), 'Port' (containing '4343'), 'Login' (containing 'myserveradmin'), and 'Password' (masked with dots). There are radio buttons for 'Password' (selected) and 'Prompt for password'. Below these fields is an 'Encryption' button. The main area is divided into two panes: 'Class' on the left and 'Database Connection' on the right. The 'Class' pane has radio buttons for 'Oracle' (selected), 'Ingres / Vector(H)', 'SQL Server', 'DB2 Linux/Unix/Windows', 'DB2 for i', 'PostgreSQL/Aurora', 'MySQL/MariaDB/Aurora', and 'Teradata'. The 'Database Connection' pane has fields for 'ORACLE\_HOME' (D:\oracle\1210), 'ORACLE\_SID' (ORA1210), 'TNS' (radio button), 'User' (hvrhub), and 'Password' (masked). At the bottom right, there are three buttons: 'Connect', 'Cancel', and 'Help'.

## 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 displayed under the hub database.

## Create Locations

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

### Create Oracle Location

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**.

## Create Kafka Locations

To create a Kafka location, right-click on Location **Configuration** **New Location**.

- HVR creates a Kafka location that sends messages in JSON format by default.

**Location kafjs**

Location: kafjs  
Description: Kafka location with JSON format messages

Connection | Group Membership

Connect to HVR on remote machine

Node: \_\_\_\_\_ Login: \_\_\_\_\_  
Port: \_\_\_\_\_ Password: \_\_\_\_\_

**Class**

- Oracle
- Ingres / Vector(H)
- SQL Server
- DB2 Linux/Unix/Windows
- DB2 for i
- PostgreSQL
- HANA
- Teradata
- Greenplum
- Redshift
- File / FTP / Sharepoint
- HDFS
- S3
- Salesforce
- Kafka

**Kafka**

Broker: 192.168.10.251 Port: 9092

Authentication Mechanism: No Authentication

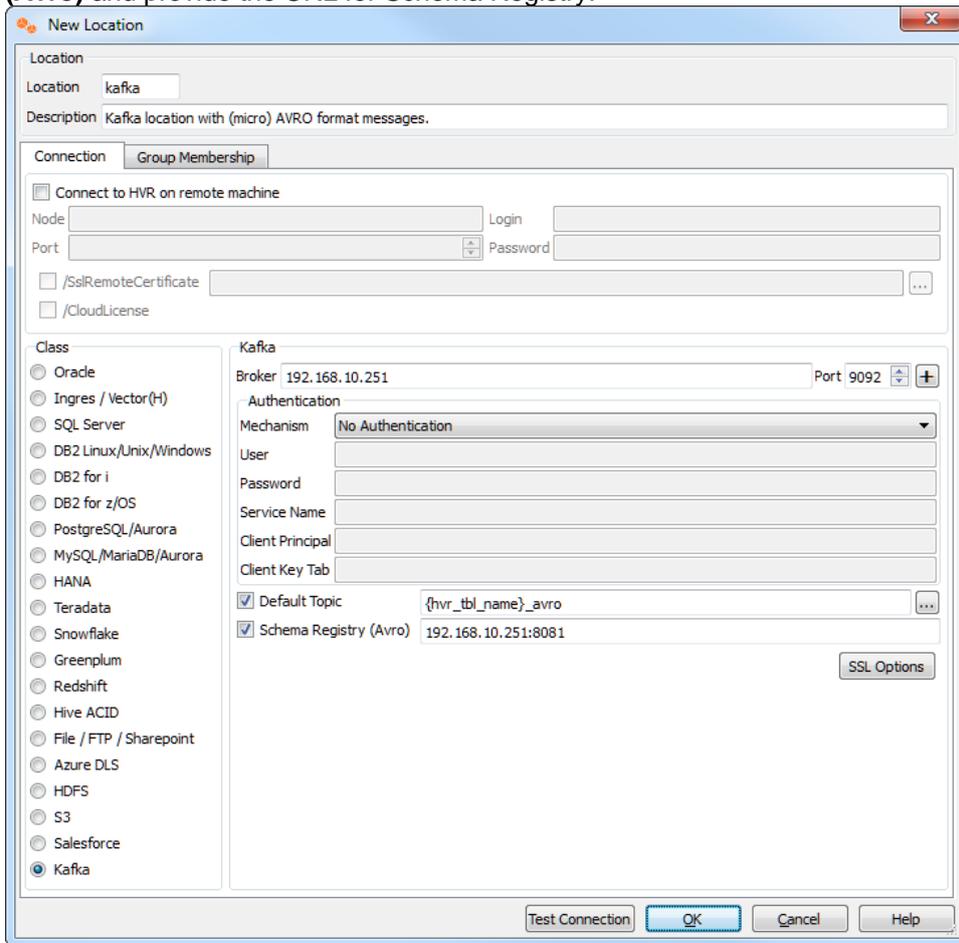
User: \_\_\_\_\_  
Password: \_\_\_\_\_  
Service Name: \_\_\_\_\_  
Client Principal: \_\_\_\_\_  
Client Key Tab: \_\_\_\_\_

Default Topic: {hvr\_tbl\_name} ...  
 Schema Registry (Avro) \_\_\_\_\_

SSL Options

Test Connection | OK | Cancel | Help

- To create a Kafka location that sends messages in (micro) AVRO format, select **Schema Registry (Avro)** and provide the URL for Schema Registry.



## Create Channel

The next step is to create a channel. For a relational database, the channel represents a group of tables that is captured as a unit. To create a channel, perform the following procedure:

1. Right-click on **Channel Definitions** **New Channel**.
2. Enter a channel name (for example **hvr\_demo01**).
3. Click **OK** to create a channel.

## Create Location Groups

The channel must have two location groups (for example **SRC** and **KAFKA**). Perform the following procedure to create a location group:

1. Right-click the **Location Groups** **New Group**.
2. Enter a group name (for example **SRC**).
3. Select the location (for example **src**) to add it as a member of the group.
4. Click **OK** to save.  
To create the second location group:
5. Right-click the **Location Groups** **New Group**.
6. Enter a group name (for example **KAFKA**).
7. Select the location (for example **kafjs** and **kafav**) to add it as a member of the group.
8. Click **OK** to save.

## Create Tables

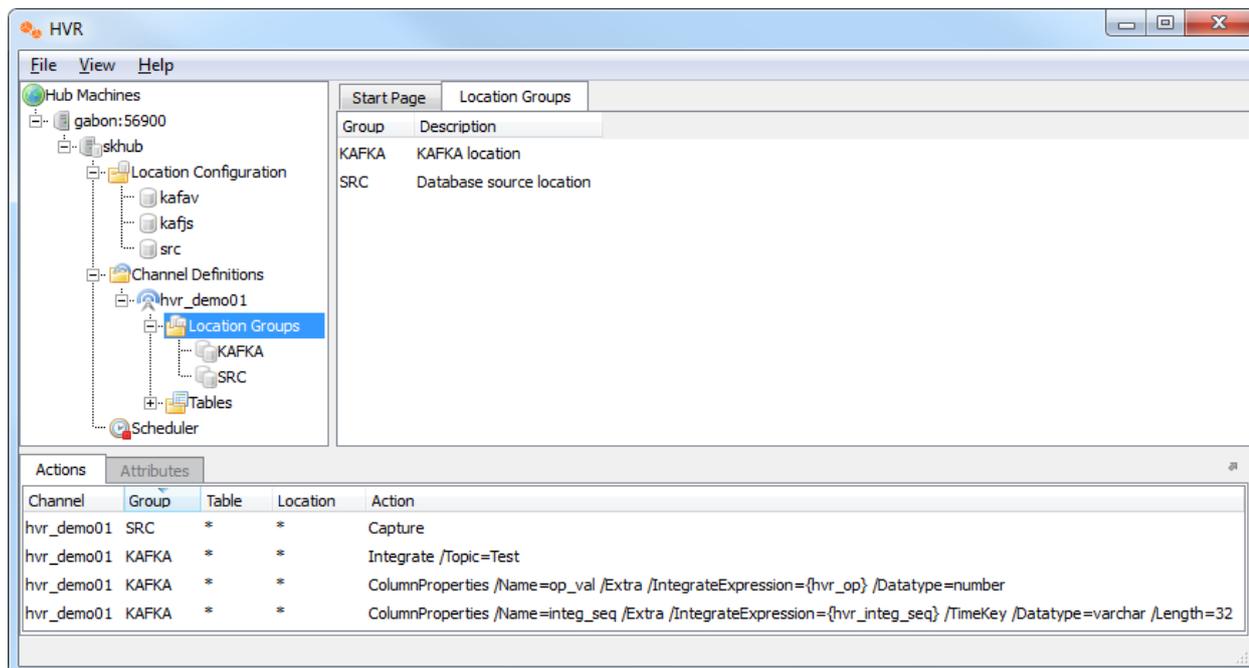
The new channel also needs a list of tables to replicate. Perform the following procedure to create tables:

1. Right-click on **Tables Table Explore**.
2. Choose the database location and click **Connect**.
3. In the **Table Explore** window, select the table(s) and click **Add**.
4. In new dialog **HVR Table Name** click **OK**.
5. Close the **Table Explore** window.

## Define Actions

The new channel needs actions to define the replication. To define actions:

1. Right-click on group **SRC New Action Capture**.
2. Right-click on group **KAFKA New Action Integrate**.
  - a. Select **/Topic=Test**
3. Right-click on group **KAFKA New Action ColumnProperties**.
  - a. Select **/Name=op\_val**.
  - b. Select **/Extra**.
  - c. Select **/IntegrateExpression={hvr\_op}**.
  - d. Select **/Datatype=number**.
  - e. Click **OK** to save and close the New Action window.
4. Right-click on group **KAFKA New Action ColumnProperties**.
  - a. Select **/Name=integ\_seq**.
  - b. Select **/Extra**.
  - c. Select **/IntegrateExpression={hvr\_integ\_seq}**.
  - d. Select **/Timekey**.
  - e. Select **/Datatype=varchar**.
  - f. Select **/Length=32**
  - g. Click **OK** to save and close the **New Action** window.



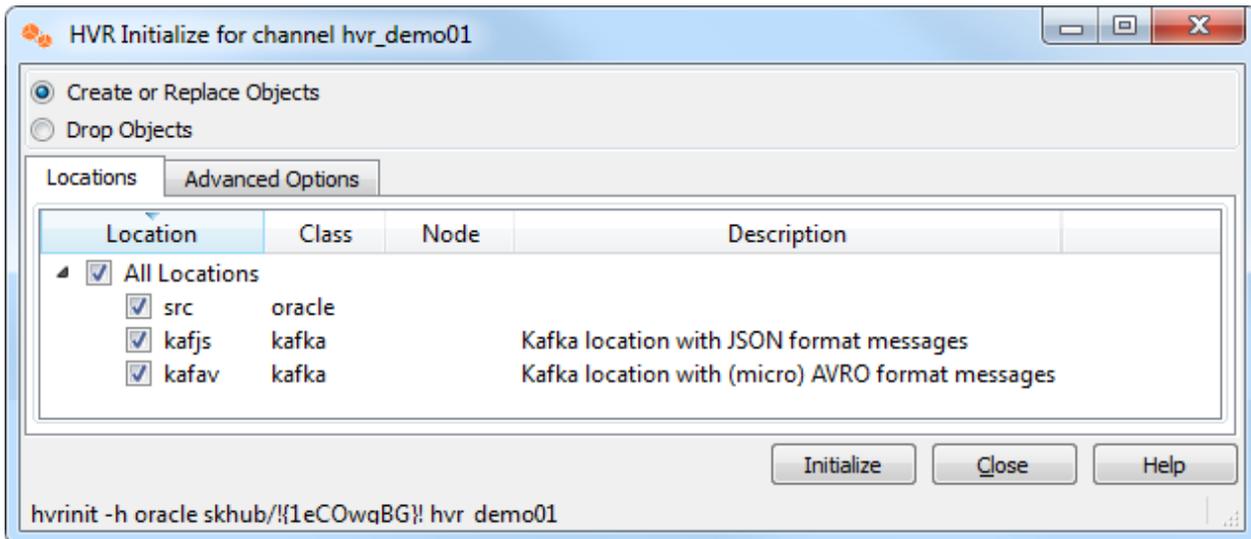
Note that the Actions pane only displays actions related to the object selected in the left-hand pane. So click on channel **hvr\_demo01** to see both actions.

## Initialize

Now that the channel definition is complete, proceed to create the runtime replication system with HVR initialize.

1. Right-click on channel **hvr\_demo01 HVR Initialize**.

2. Select **Create or Replace Objects**.
3. Click **Initialize**.



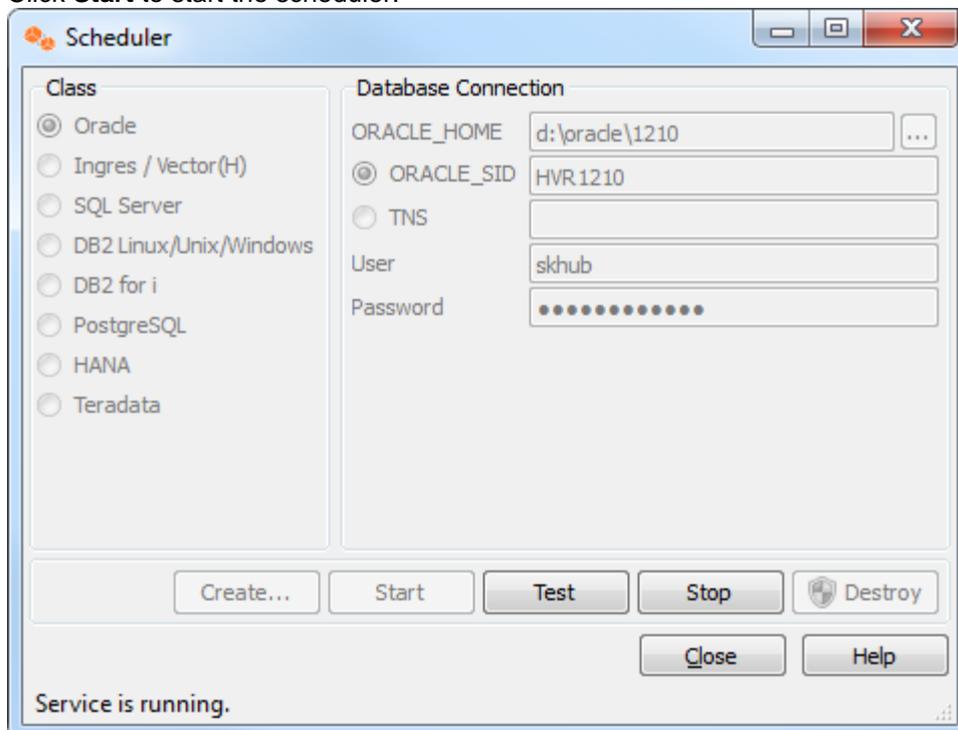
After initializing the channel **hvr\_demo01**, all new transactions that start on the database **testdb1** will be captured by HVR (when its capture job looks inside the transaction logs).

HVR Initialize also creates two replication jobs under the **Scheduler** node.

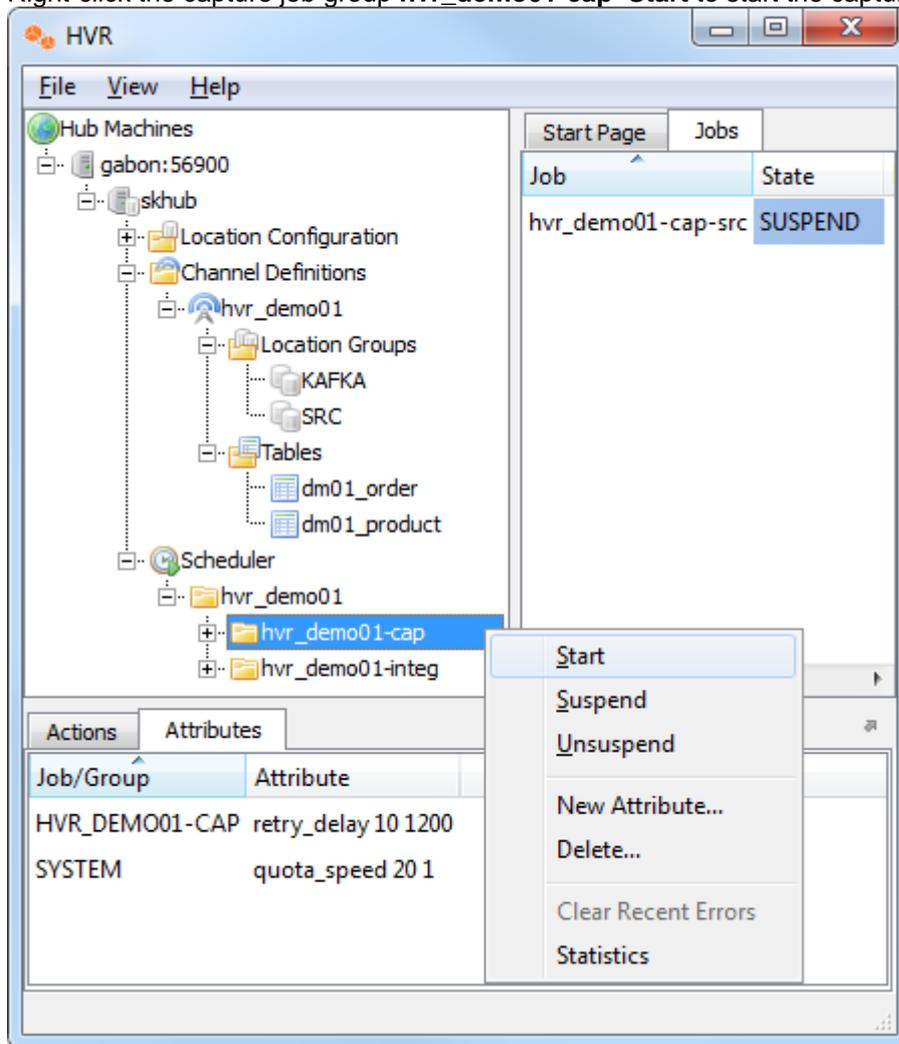
## Start Scheduler and Capture Job

Next, perform the following procedure to start scheduling of the capture job:

1. Right-click the **Scheduler** node of the hub database.
2. Click **Start** to start the scheduler.



- Right-click the capture job group **hvr\_demo01-cap Start** to start the capture job.



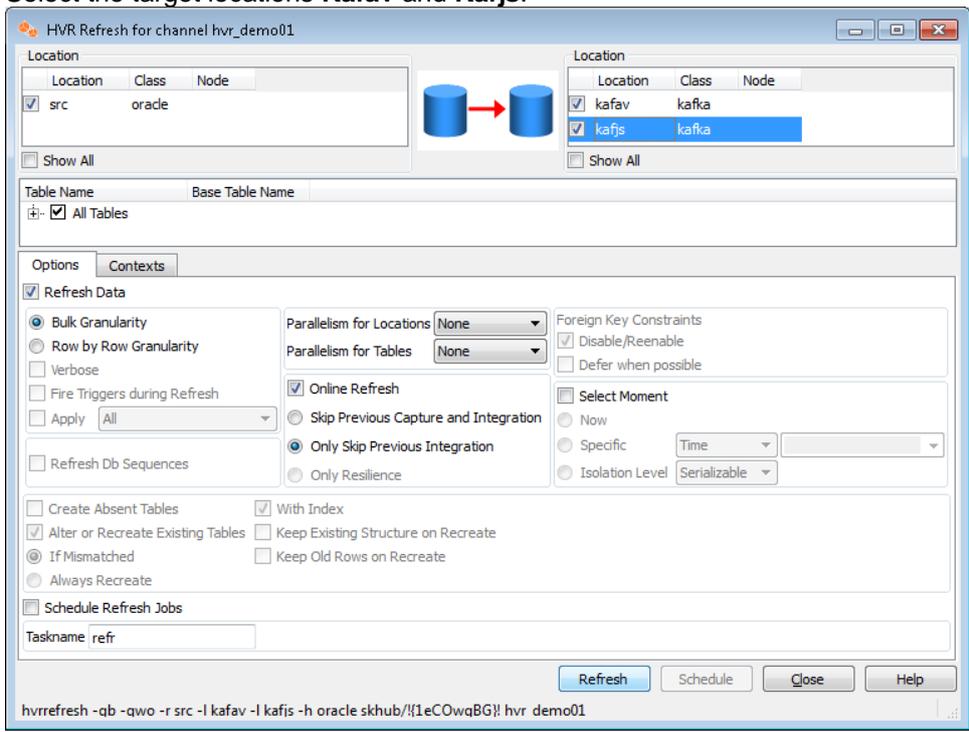
The capture job inside the Scheduler executes a script under **\$HVR\_CONFIG/job/hvrhub/hvr\_demo01** that has the same name as the job. So job **hvr\_demo01-cap-src** detects changes on database **testdb1** and stores these as transactions files on the hub machine.

### Refresh

Perform the following procedure for the initial load from OLTP database into Kafka using **HVR Refresh**.

- Right-click the channel **hvr\_demo01 HVR Refresh**.

2. Select the target locations **Kafav** and **Kafjs**.



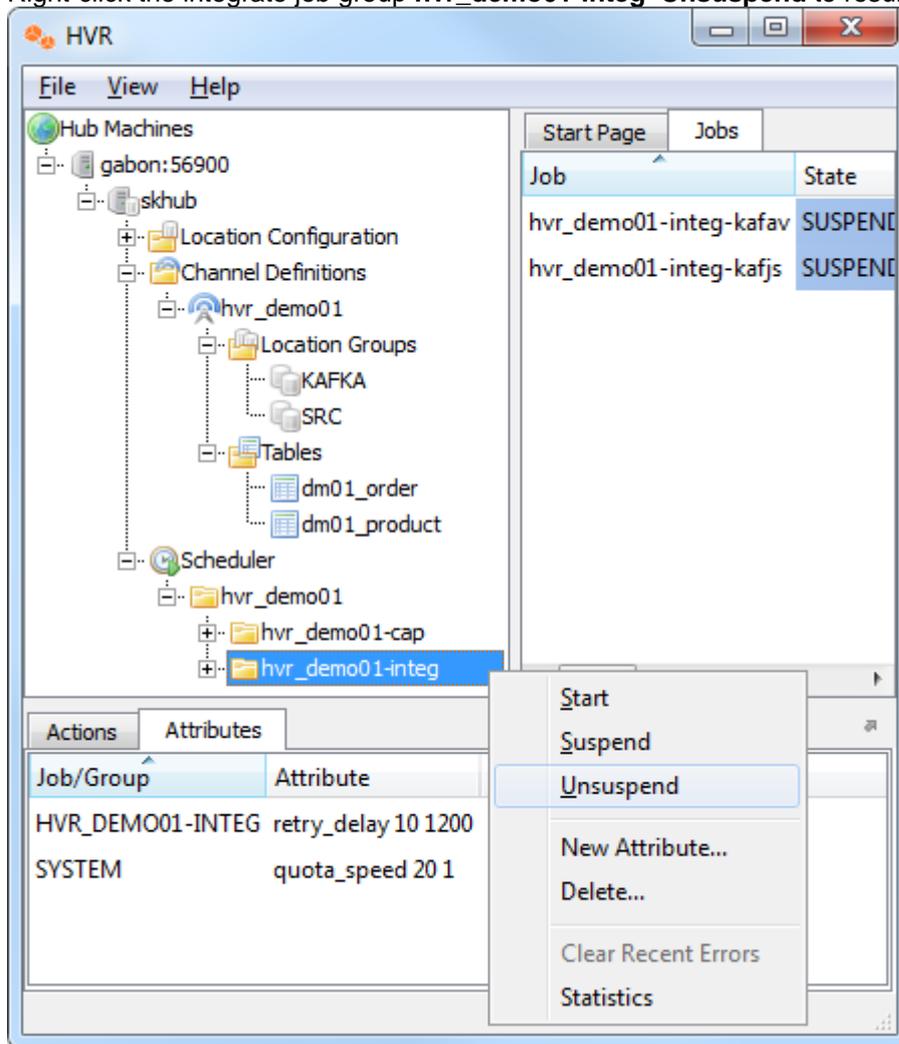
3. You can optionally set **Parallelism for Tables**.

4. Click **Refresh** to begin HVR Referesh.

## Start Integrate Job

Perform the following procedure to unsuspend the integrate job in HVR Scheduler to push changes into Kafka according to the defined schedule.

1. Right-click the integrate job group **hvr\_demo01-integ Unsuspend** to resume the integrate job.



The integrate job inside the Scheduler executes a script under **\$HVR\_CONFIG/job/hvrhub/hvr\_demo01** that has the same name as the job. So the jobs **hvr\_demo01-integ-kafav** and **hvr\_demo01-integ-kafjs** pick up transaction files on the hub on a defined schedule and create files on Kafka containing these changes.

A scheduled job that is not running is in a **PENDING** state.