

Requirements for HANA

Contents

- [Contents](#)
- [ODBC Connection](#)
- [Location Connection](#)
- [Connecting to Remote HANA Location from Hub](#)
- [Capture](#)
 - [Table Types](#)
 - [Grants for Capture](#)
 - [Capture Limitations](#)
 - [Configuring Log Mode and Transaction Archive Retention Requirements](#)
 - [Archive Log Only Method](#)
 - [OS Level Permissions or Requirements](#)
 - [Channel Setup Requirements](#)
- [Integrate and Refresh Target](#)
 - [Integrate Limitations](#)
 - [Grants for Integrate and Refresh Target](#)
 - [Burst Integrate and Bulk Refresh](#)
 - [Grants for Burst Integrate and Bulk Refresh](#)
- [Compare and Refresh Source](#)
 - [Grants for Compare and Refresh Source](#)
- [Limitations](#)
- [Upgrading HANA Database](#)

HANA		
Capture	Hub	Integrate
		

This section describes the requirements, access privileges, and other features of HVR when using SAP HANA for replication. For information about compatibility and supported versions of HANA with HVR platforms, see [Platform Compatibility Matrix](#).

For the [Capabilities](#) supported by HVR on HANA, see [Capabilities for HANA](#).

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

For instructions to quickly setup replication into HANA, see [Quick Start for HVR - HANA](#).

ODBC Connection

HVR requires that the HANA client (which contains the HANA ODBC driver) to be installed on the machine from which HVR connects to HANA. HVR uses HANA ODBC driver to connect, read and write data to HANA.

For information about the supported ODBC driver version, refer to the HVR release notes (**hvr.rel**) available in **hvr_home** directory or the download page.

HVR does not support integrating changes captured from HANA into databases where the distribution key cannot be updated (e.g. Greenplum, Azure Synapse Analytics).

Location Connection

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

New Location
✕

Location

Description

Connection

Group Membership

Connect to HVR on remote machine

Node Login

Port Password

/SslRemoteCertificate

/CloudLicense

MySQL/MariaDB/Aurora

HANA

Teradata

Snowflake

Greenplum

Redshift

Hive ACID

File / FTP / Sharepoint

Azure DLS

Azure Blob FS

HDFS

S3

Salesforce

Kafka

Database Connection

Node

Mode

Instance Number

Port

Database

User

Password

Linux

Driver Manager Library

ODBCSYSINI

ODBC Driver

Field	Description
Database Connection	
Node	The hostname or ip-address of the machine on which the HANA server is running. Example: myhananode
Mode	The mode for connecting HVR to HANA server. Available options: <ul style="list-style-type: none"> • Single-container • Multiple containers - Tenant database • Multiple containers - System database • Manual port selection - This option is used only if database Port needs to be specified manually.
Instance Number	The database instance number. Example: 90
Port	The port on which the HANA server is expecting connections. For more information about TCP/IP ports in HANA, refer to SAP Documentation Example: 39015

Database	The name of the specific database in a multiple-container environment. This field is enabled only if the Mode is either Multiple containers - Tenant database or Manual port selection .
User	The username to connect HVR to the HANA Database . Example: hvruser
Password	The password of the User to connect HVR to the HANA Database .
Linux	
Driver Manager Library	The optional directory path where the ODBC Driver Manager Library is installed. This field is applicable only for Linux/Unix operating system. For a default installation, the ODBC Driver Manager Library is available at /usr/lib64 and does not need to be specified. However, when UnixODBC is installed in for example /opt/unixodbc the value for this field would be /opt/unixodbc/lib .
ODBCSYSINI	The optional directory path where odbc.ini and odbcinst.ini files are located. This field is applicable only for Linux/Unix operating system. For a default installation, these files are available at /etc and do not need to be specified. However, when UnixODBC is installed in for example /opt/unixodbc the value for this field would be /opt/unixodbc/etc . The odbcinst.ini file should contain information about the HANA ODBC Driver under heading [HDBODBC] or [HDBODBC32] .
ODBC Driver	The user defined (installed) ODBC driver to connect HVR to the HANA database. If the user does not define the ODBC driver in the ODBC Driver field, then HVR will automatically load the correct driver for your current platform: HDBODBC (64-bit) or HDBODBC32 (32bit).

Connecting to Remote HANA Location from Hub

HVR allows you to connect from a hub machine to a remote HANA database by using any of the following two methods:

1. Connect to an HVR installation running on the HANA database machine using HVR's protocol on a special TCP port number (e.g. 4343). This option must be used for log-based capture from HANA.
2. Use ODBC to connect directly to a HANA database remotely. In this case no additional software is required to be installed on the HANA database server itself. This option cannot be used for log-based capture from HANA database.

Capture

HVR only supports capture from HANA on Linux.

For the list of supported HANA versions, from which HVR can capture changes, see [Capture changes from location](#) in [Capabilities](#).

Table Types

HVR supports capture from column-storage tables in HANA.

Grants for Capture

The following grants are required for capturing changes from HANA:

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

```
grant select on tbl to hvruser;
```

- The **User** should also be granted select permission from some system table and views. In HANA, however, it is impossible to directly grant permissions on system objects to any user. Instead, special wrapper views should be created, and **User** should be granted read permission on this views. To do this:
 1. Connect to the HANA database as user **SYSTEM**.
 2. Create a schema called **_HVR**.

```
create schema _HVR;
```

3. Grant **SELECT** privilege on this schema to **User**.

```
grant select on schema _HVR to hvruser;
```

4. Execute the **hvrhanaviews.sql** script from the **\$HVR_HOME/sql/hana** directory. This will create a set of views that HVR uses to read from system dictionaries on HANA.

Capture Limitations

This section describes the limitations for capturing changes from HANA using HVR.

- HVR does not support **Capture** from multi-node HANA clusters.
- HVR does not support **Capture** from HANA encrypted logs.

HANA allows encryption of transaction logs and transaction log backups separately. So if only the transaction logs are encrypted and not the transaction log backups, then HVR can capture using **Archive Log Only** method.

- Since HANA does not support supplemental logging, HVR will not be able to process certain **actions/parameters** that requires the value of a column, and that column is not present in the HANA logs.

The actions/parameters affected by this limitation are listed below.

- The following action definitions will not function due to this limitation:
 - **CollisionDetect**
 - **ColumnProperties /TimeKey**
 - **Integrate /DbProc**
 - **Integrate /Resilient** (will not function only if a row is missing on the target)
 - **FileFormat /BeforeUpdateColumns**
 - **FileFormat /BeforeUpdateColumnsWhenChanged**
- The following action definitions will not function if it requires the value of a column, and that column is not present in the HANA logs:
 - **ColumnProperties /CaptureExpression** (however, **CaptureExpressionType=SQL_WHERE_ROW** will function normally)
 - **ColumnProperties /IntegrateExpression**
 - **Integrate /RenameExpression**
 - **Restrict /CaptureCondition**
 - **Restrict /IntegrateCondition**
 - **Restrict /HorizColumn**
 - **Restrict /AddressTo**

The above mentioned **Restrict** parameters for **compare** and **refresh** will function normally.

Configuring Log Mode and Transaction Archive Retention Requirements

For HVR to capture changes from HANA, the automatic backup of transaction logs must be enabled in HANA. Normally HVR reads changes from the 'online' transaction log file, but if HVR is interrupted (say for 2 hours) then it must be able to read from transaction log backup files to capture the older changes. HVR is not interested in full backups; it only reads transaction log backup file.

To enable automatic log backup in HANA, the log mode must be set to **normal**. Once the log mode is changed from **overwrite** to **normal**, a full data backup must be created. For more information, search for [Log Modes](#) in [SAP HANA Documentation](#).

The log mode can be changed using HANA Studio. For detailed steps, search for [Change Log Modes](#) in [SAP HANA Documentation](#). Alternatively, you can execute the following SQL statement:

```
alter system alter configuration ('global.ini', 'SYSTEM') set ('persistence', 'log_mode') = 'normal' with reconfigure;
```

Transaction log (archive) retention: If a backup process has already moved these files to tape and deleted them, then HVR's [capture](#) will give an error and a [refresh](#) will have to be performed before replication can be restarted. The amount of 'retention' needed (in hours or days) depends on organization factors (how real-time must it be?) and practical issues (does a refresh take 1 hour or 24 hours?).

When performing log-shipping (**Capture /ArchiveLogOnly**), file names must not be changed in the process because begin-sequence and timestamp are encoded in the file name and capture uses them.

Archive Log Only Method

The **Archive Log Only** method can be used for capturing changes from HANA. This allows the HVR process to reside on machine other than that on which HANA DBMS resides and read changes from backup transaction log files that may be sent to it by some file transfer mechanism. HVR must be configured to find these files by defining action [Capture](#) with parameters **/ArchiveLogOnly**, **/ArchiveLogPath**, and **/ArchiveLogFormat** (optional).

The **Archive Log Only** method will generally expose higher latency than **non-Archive Log Only** method because changes can only be captured when the transaction log backup file is created. The **Archive Log Only** method enables high-performance log-based capture with minimal OS privileges, at the cost of higher capture latency.

OS Level Permissions or Requirements

To capture from HANA database, HVR should be installed on the HANA database server itself, and HVR remote listener should be configured to accept remote connections. The Operating System (OS) user the HVR is running under should have READ permission on the HANA database files. This can typically be achieved by adding this user to the **sapsys** user group.

Channel Setup Requirements

It is not possible to enable 'supplemental logging' on HANA. This means that the real key values are not generally available to HVR during capture. A workaround for this limitation is capturing the Row ID values and use them as a surrogate replication key.

The following two additional actions should be defined to the channel, prior to using **Table Explore** (to add tables to the channel), to instruct HVR to capture Row ID values and to use them as surrogate replication keys.

Location	Action	Annotation
Source	ColumnProperties /Name=hvr_row id /CaptureFromRowId	This action should be defined for capture locations only.

*	ColumnProperties /Name=hvr_row id /SurrogateKey	This action should be defined for both capture and integrate locations
---	--	--

Integrate and Refresh Target

HVR supports integrating changes into HANA location. This section describes the configuration requirements for integrating changes (using **Integrate** and **refresh**) into HANA location. For the list of supported HANA versions, into which HVR can integrate changes, see [Integrate changes into location](#) in [Capabilities](#).

HVR uses HANA ODBC driver to write data to HANA during continuous **Integrate** and row-wise **Refresh**. For the method used during **Integrate** with **/Burst** and Bulk **Refresh**, see section [Burst Integrate and Bulk Refresh](#) below.

Integrate Limitations

This section describes the limitations for integrating changes into HANA using HVR.

- HVR does not support **Integrate** into multi-node HANA clusters.

Grants for Integrate and Refresh Target

- The **User** should have permission to read and change replicated tables

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

- The **User** should have permission to create and alter tables in the target schema

```
grant create any, alter on schema schema to hvruser;
```

- The **User** should have permission to create and drop HVR state tables

Burst Integrate and Bulk Refresh

While **HVR Integrate** is running with parameter **/Burst** and Bulk **Refresh**, HVR can stream data into a target database straight over the network into a bulk loading interface specific for each DBMS (e.g. direct-path-load in Oracle), or else HVR puts data into a temporary directory ('staging file') before loading data into a target database.

For best performance, HVR performs **Integrate** with **/Burst** and Bulk **Refresh** on HANA location using staging files.

HVR implements **Integrate** with **/Burst** and Bulk **Refresh** (with file staging) into HANA as follows:

1. HVR first stages data into a local staging file (defined in **/StagingDirectoryHvr**)
2. HVR then uses SAP HANA SQL command 'import' to ingest the data into SAP HANA target tables from the staging directory (defined in **/StagingDirectoryDb**).

To perform **Integrate** with parameter **/Burst** and Bulk **Refresh**, define action **LocationProperties** on HANA location with the following parameters:

- **/StagingDirectoryHvr**: the location where HVR will create the temporary staging files.
- **/StagingDirectoryDb**: the location from where HANA will access the temporary staging files. This staging-path should be configured in HANA for importing data using the following command,

```
alter system alter configuration ('indexserver.ini', 'SYSTEM') set
('import_export', 'csv_import_path_filter') = 'STAGING-PATH' with
reconfigure;
```

Grants for Burst Integrate and Bulk Refresh

The **User** should have permission to import data:

```
grant import to hvruser;
```

Compare and Refresh Source

HVR supports **compare** and **refresh** in HANA location. This section describes the configuration requirements for performing **compare** and **refresh** in HANA (source) location.

Grants for Compare and Refresh Source

The **User** should have permission to read replicated tables

```
grant select on tbl to hvruser;
```

Limitations

This section describes general limitations for capturing or integrating data into HANA using HVR.

- HVR does not support replication to/from multi-node HANA clusters.

For information about limitations related to capturing from HANA, see section [Capture Limitations](#).

Upgrading HANA Database

When upgrading a HANA database (e.g. from HANA 2.0 SPS 04 to HANA 2.0 SPS 05), HVR may fail with the 'invalid column name' DBMS error.

The solution is to recreate all the views in the **_HVR** schema by running the **\$HVR_HOME/sql/hana/hvrhanaviews.sql** script after the upgrade. The script must be run by the **SYSTEM** user.