



Requirements for Greenplum

Contents
<ul style="list-style-type: none">• ODBC Connection• Location Connection• Integrate and Refresh Target<ul style="list-style-type: none">• Burst Integrate and Bulk Refresh• Grants for Integrate, Refresh, and Compare• Compare Limitation

Greenplum		
Capture	Hub	Integrate
		

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

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

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 using Greenplum, see [Quick Start for HVR - Greenplum](#).

ODBC Connection

It is not required to install HVR on any of the nodes of the Greenplum cluster. HVR can be installed on a standalone machine from which it connects to the Greenplum cluster. HVR requires the DataDirect Connect XE ODBC driver for Greenplum installed (on the machine from which HVR connects to a Greenplum server).

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.

Location Connection

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

New Location
✕

Location

Location

Description

Connection

Group Membership

Connect to HVR on remote machine

Node Login

Port Password

/SslRemoteCertificate ...

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

Port

Database

User

Password

Linux / Unix

Driver Manager Library ...

ODBCINST ...

ODBC Driver ...

Field	Description
Database Connection	
Node	The hostname or ip-address of the machine on which the Greenplum server is running. Example: gp430
Port	The port on which the Greenplum server is expecting connections. Example: 5432
Database	The name of the Greenplum database. Example: sfdec02
User	The username to connect HVR to the Greenplum Database . Example: hvruser
Password	The password of the User to connect HVR to the Greenplum Database .
Linux / Unix	
Driver Manager Library	The directory path where the ODBC Driver Manager Library is installed. This field is applicable only for Linux/Unix operating system.

ODBCINST	The directory path where odbcinst.ini file is located. This field is applicable only for Linux/Unix operating system.
ODBC Driver	The user defined (installed) ODBC driver to connect HVR to the Greenplum server.

Integrate and Refresh Target

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

HVR uses DataDirect Connect XE ODBC driver to write data to Greenplum during continuous [Integrate](#) and row-wise [Refresh](#). However, the preferred methods for writing data to Greenplum is [Integrate](#) with [/Burst](#) and Bulk [Refresh](#) using staging as they provide better performance.

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](#) into Greenplum using staging files and the Greenplum Parallel File Distribution (**gpfdist**) server. To use the **gpfdist** server for bulk loading operations, ensure that **gpfdist** is configured on the machine from which HVR will connect to Greenplum.

HVR implements [Integrate](#) with [/Burst](#) and Bulk [Refresh](#) (with file staging) into Greenplum as follows:

1. HVR first writes data into a temporary file in a staging directory on the machine where HVR connects to Greenplum. This directory does not have to be on the Greenplum database machine. The temporary file is written in the **.csv** format and is compressed.
2. HVR then uses Greenplum SQL **'copy'** command to pull the compressed data from **gpfdist://** or **gpfdists://** directory into a target table. This requires that a special Greenplum 'external table' exists for each target table that HVR loads data into. HVR will create these tables with names having the following patterns **'__x'** or **'__bx'**.

To perform [Integrate](#) with [/Burst](#) and Bulk [Refresh](#), define action [LocationProperties](#) on a Greenplum location with the following parameters:

- **/StagingDirectoryHvr**: the location where HVR will create the temporary staging files. This should be the **-d** (directory) option of the **gpfdist** server command.
- **/StagingDirectoryDb**: the location from where Greenplum will access the temporary staging files. This should be set to **gpfdist: //<hostname>:<port>** where *hostname* is the name of the machine used to connect to Greenplum and *port* is the **-p** (http port) option of the **gpfdist** server command.

Example of a **gpfdist** command line in Linux & Unix:

```
/opt/gpfdist-4.3.0/gpfdist -p 33333 -d /home/hvr/tsuite_staging -l /home/hvr/staging/gpfdist.log -m 10485760
```

On Windows, **gpfdist** is a service and the values can be retrieved from the "Path to Executable" in the properties dialog of the service.

Grants for Integrate, Refresh, and Compare

The **User** should have the following privileges:

```
grant connect and create table on the database
grant select, insert, update, delete on replicated tables
```

If **User** needs to change tables which are in another schema (using action **TableProperties /Schema=*myschema***) then the following grants are needed:

```
grant usage on myschema to hvruser ;
```

When **HVR Refresh** is used to create the target tables, the following privilege is also needed:

```
grant create on myschema to hvruser ;
```

HVR's internal tables, like burst and state-tables, will be created in schema **public**.

Compare Limitation

HVR does not support comparing tables with bytea data type in Greenplum version 6.