Requirements for Ingres and Vector

Contents

Ingres

Capture	Hub	Integrate
✓	✓	✓

Vector

Capture	Hub	Integrate
✗	✓	✓

This section describes the requirements, access privileges, and other features of HVR when using Ingres or Vector for replication. For information about compatibility and supported versions of Ingres or Vector with HVR platforms, see Platform Compatibility Matrix.

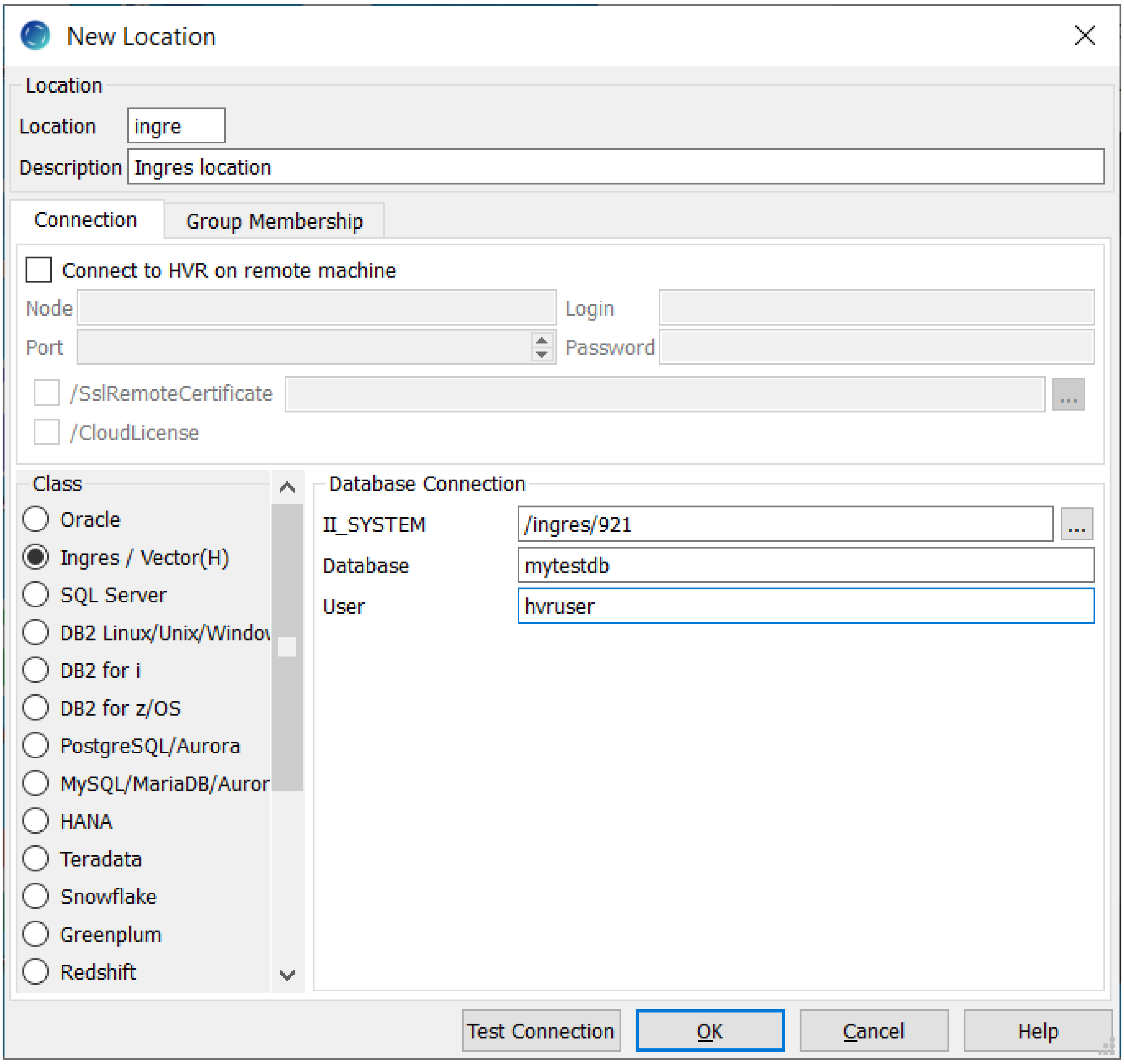
For the Capabilities supported by HVR on Ingres and Vector, see Capabilities for Ingres and Capabilities for Vector respectively.

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 Ingres, see Quick Start for HVR - Ingres.

Location Connection

This section lists and describes the connection details required for creating an Ingres/Vector location in HVR. HVR uses Ingres OpenAPI interface to connect to an Ingres/Vector location.



Location: ingre

Description: Ingres location

□ Connect to HVR on remote machine

Class: ● Ingres / Vector(H)

Database Connection

II_SYSTEM: /ingres/921

Database: mytestdb

User: hvruser

Field	Description
Database Connection	
II_SYSTEM	The directory path where the Ingres/Vector database is installed. **Example:** /ingres/921
Database	The name of the Ingres/Vector database. **Example:** mytestdb
User	The username to connect HVR to Ingres/Vector **Database**. **Example:** hvruser

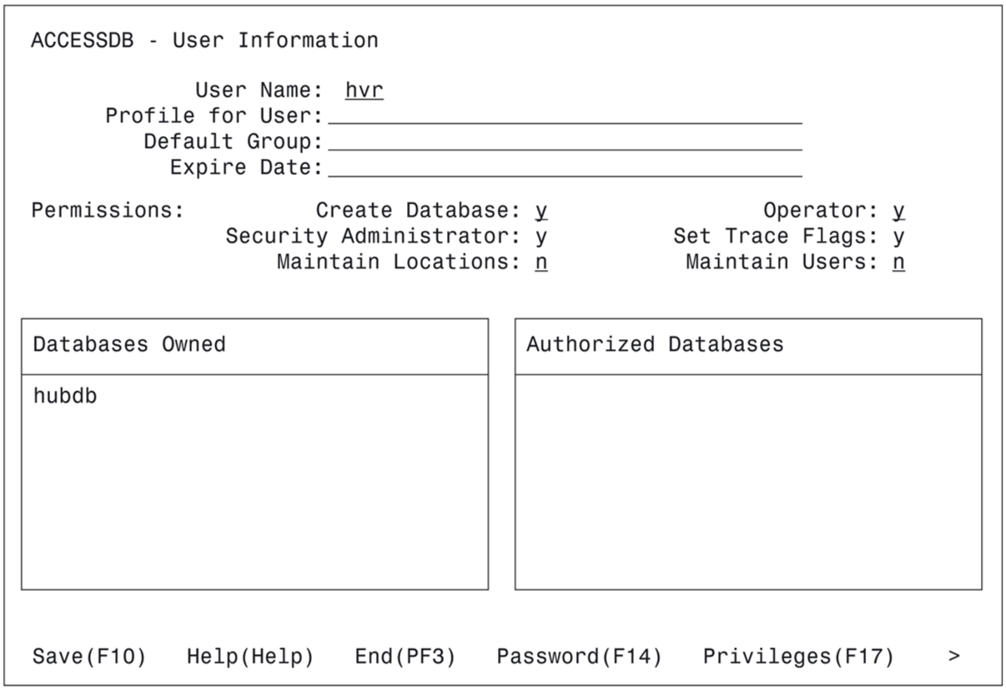
Access Privileges

For an Ingres or Vector hub database or database location, each account used by HVR must have permission to use Ingres.

The HVR user should be the owner of the hub database.

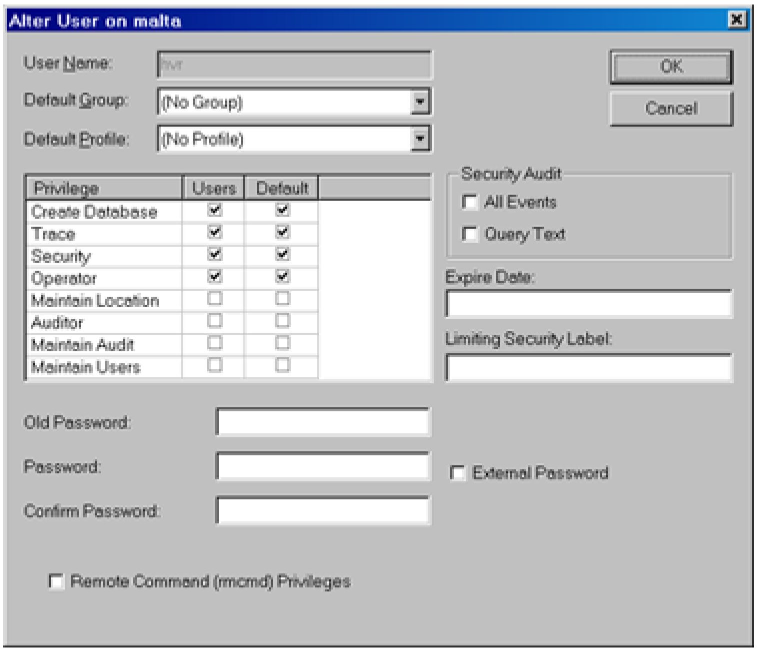
Typically, HVR connects to database locations as the owner of that database. This means that either HVR is already running as the owner of the database, or it is running as a user with Ingres **Security Privilege**. HVR can also connect to a database location as a user who is not the database's owner, although the row-wise refresh into such a database is not supported if database rules are defined on the target tables.

Accessdb permission screen:



```
ACCESSDB - User Information

             User Name:  hvr
      Profile for User: ____________________________________
         Default Group: ____________________________________
           Expire Date: ____________________________________

Permissions:          Create Database: y                 Operator: y
               Security Administrator: y          Set Trace Flags: y
                   Maintain Locations: n           Maintain Users: n

 Databases Owned                      Authorized Databases

 hubdb

Save(F10)   Help(Help)   End(PF3)   Password(F14)   Privileges(F17)    >
```

DBA permission screen:



Alter User on malta

User Name: hvr

Default Group: (No Group)

Default Profile: (No Profile)

Privilege	Users	Default
Create Database	☑	☑
Trace	☑	☑
Security	☑	☑
Operator	☑	☑
Maintain Location	□	□
Auditor	□	□
Maintain Audit	□	□
Maintain Users	□	□

Security Audit

□ All Events

□ Query Text

Expire Date:

Limiting Security Label:

Old Password:

Password:

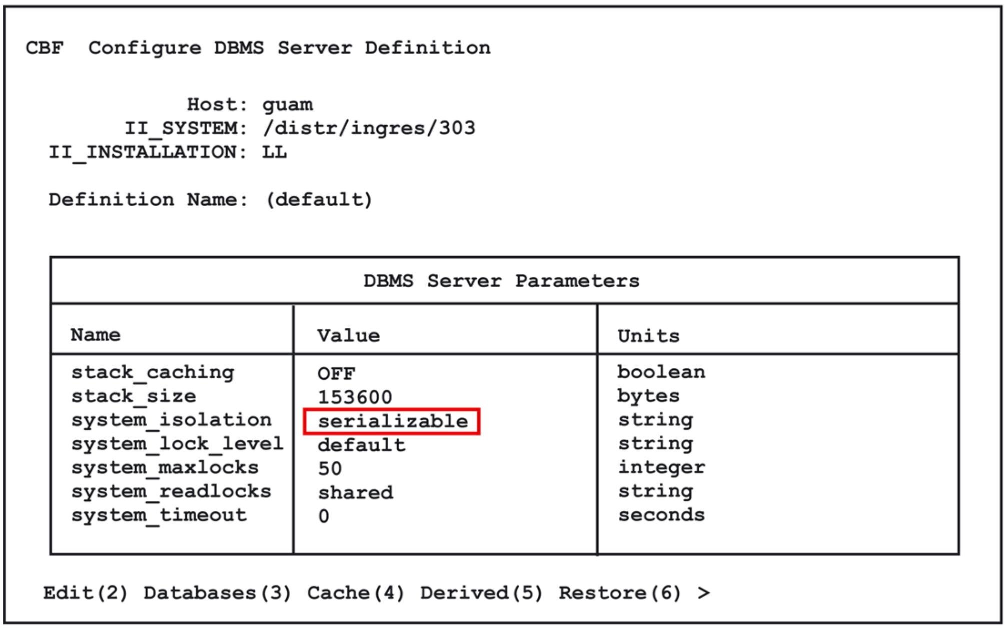
Confirm Password:

□ External Password

□ Remote Command (rmcmd) Privileges

CBF screen:

For trigger based capture from Ingres databases, the isolation level (parameter **system_isolation**) must be set to **serializable**. Other parameters (e.g. **system_readlocks**) can be anything.



```
CBF  Configure DBMS Server Definition

             Host: guam
        II_SYSTEM: /distr/ingres/303
  II_INSTALLATION: LL

  Definition Name: (default)

 +-----------------------------------------------------------------------+
 |                        DBMS Server Parameters                         |
 +-------------------+-----------------------+---------------------------+
 | Name              | Value                 | Units                     |
 +-------------------+-----------------------+---------------------------+
 | stack_caching     | OFF                   | boolean                   |
 | stack_size        | 153600                | bytes                     |
 | system_isolation  | serializable          | string                    |
 | system_lock_level | default               | string                    |
 | system_maxlocks   | 50                    | integer                   |
 | system_readlocks  | shared                | string                    |
 | system_timeout    | 0                     | seconds                   |
 +-------------------+-----------------------+---------------------------+

 Edit(2) Databases(3) Cache(4) Derived(5) Restore(6) >
```

Creating Trusted Executable

In UNIX & Linux, to perform log-based capture from Ingres a trusted executable must be created so that HVR can read from the internal DBMS logging files.

Execute the following commands while logged in as the DBMS owner (**ingres**):

```
$ cd /usr/hvr/hvr_home
$ cp bin/hvr sbin/hvr_ingres
$ chmod 4755 sbin/hvr_ingres
```

It is not required to create a trusted executable when either of the following are true:

- capture is trigger-based
- capture will be from another machine
- HVR is running as the DBMS owner (**ingres**)

Additionally, on Linux, the trusted executable should be patched using the following command:

```
$ /usr/hvr/hvr_home/lib/patchelf --set-rpath /usr/hvr/hvr_home/lib --
force-rpath /usr/hvr/hvr_home/sbin/hvr_ingres
```

If HVR and **ingres** share the same Unix group, then the permissions can be tightened from 4755 to 4750. Permissions on directories **$HVR_HOME** and **$HVR_CONFIG** may need to be loosened so that user **Ingres** can access them;

```
$ chmod g+rX $HVR_HOME
$ chmod -R g+rwX $HVR_CONFIG
```

Capture

HVR supports capturing changes from an Ingres location. HVR uses Ingres OpenAPI interface to capture changes from the Ingres location. This section describes the configuration requirements for **capturing** changes from Ingres location. For the list of supported Ingres versions, from which HVR can capture changes, see Capture changes from location in Capabilities.

Table Types

HVR supports capture from the following table types in Ingres:

- Regular tables (HEAP, HASH, ISAM, BTREE)
- Partitioned tables
- Compressed tables

Trigger-Based Capture

If trigger–based capture is defined for an Ingres database, HVR uses SQL DDL statement **modify to truncated** to empty the capture tables. The locks taken by this statement conflicts with locks taken by an on–line checkpoint. This can lead to HVR jobs hanging or deadlocking. These problems can be solved by creating file **$HVR_CONFIG/files/***dbname***.avoidddl** just before checkpointing database *dbname* and deleting it afterwards. HVR will check for this file, and will avoid DDL when it sees it.

In Unix, do this as follows:

```
$ touch $HVR_CONFIG/files/mydb.avoidddl
$ sleep 5
$ ckpdb mydb
$ rm $HVR_CONFIG/files/mydb.avoidddl
```

Log-Based Capture

If log–based capture is defined for an Ingres database (action **Capture**) then HVR may need to go back to reading the Ingres journal files. But each site has an existing backup/recovery regime that periodically deletes these Ingres checkpoint and journal files. Command **Hvrlogrelease** can make cloned copies of these files so that HVR's capture is not affected when these files are purged by the site's backup/recovery regime. When the capture job no longer needs these cloned files, then **Hvrlogrelease** will delete them again.

Integrate and Refresh

HVR supports integrating changes into Ingres and Vector locations. HVR uses Ingres OpenAPI interface to write data to Ingres/Vector during **Integrate** and **Refresh**. For **Integrate** with **/Burst** and Bulk **Refresh**, HVR uses Ingres SQL "**copy table ... () from program**" command. This section describes the configuration requirements for integrating changes (using **Integrate**) into Ingres and Vector locations. For the list of supported Ingres and Vector versions, into which HVR can integrate changes, see Integrate changes into location in Capabilities.

For HVR to integrate changes into an Ingres installation on a remote machine, special database roles (**hvr_integrate**, **hvr_refresh** and **hvr_scheduler**) must be created in that Ingres installation. Execute the following script to create these roles:

In UNIX & Linux,

```
$ sql iidbdb < $HVR_HOME/sql/ingres/hvrrolecreate.sql
```

In Windows,

```
C:\>sql iidbdb < %HVR_HOME%\sql\ingres\hvrrolecreate.sql
```