Hvradapt

Contents

Name

hvradapt - Explore base table definitions in the database(s) and adapt them into channel information

Synopsis

hvradapt [**-***options*] **-l***loc hubdb chn*

Description

Command **hvradapt** compares the base tables in the database with the table information for a channel *chn*. It will then either add, replace or delete table information in the catalog tables (**hvr_table** and **hvr_column**) so this information matches.

- If the location (**-l***loc*) from where the **hvradapt** explores the base table definitions contains a table which is not present in the channel but is matched by the table filter statement then it is added to the channel.
- If a table is in the channel but is not matched by the table filter statement then it is deleted from the channel.
- If a table is both matched by the table filter statement and included in the channel, but has the wrong column information in the channel, then this column information is updated.
- If table filter statement is not supplied (no **-n** or **-N** option), then tables are not added or deleted; only existing column information is updated where necessary.

The argument *hubdb* specifies the connection to the hub database. For more information about supported hub databases and the syntax for using this argument, see Calling HVR on the Command Line.

Hvradapt is equivalent to the "**Table Explore**" dialog along with the **Table Filter** dialog in HVR GUI.

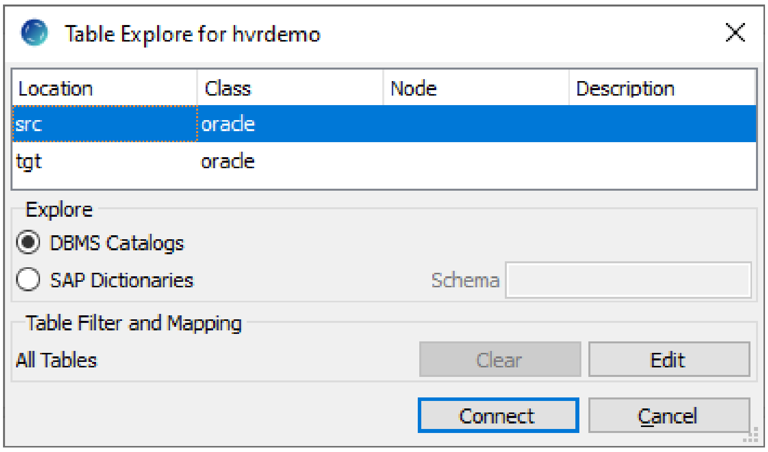


Table Explore for hvrdemo

Location	Class	Node	Description
src	oracle		
tgt	oracle		

Explore

● DBMS Catalogs

○ SAP Dictionaries

Schema

Table Filter and Mapping

All Tables

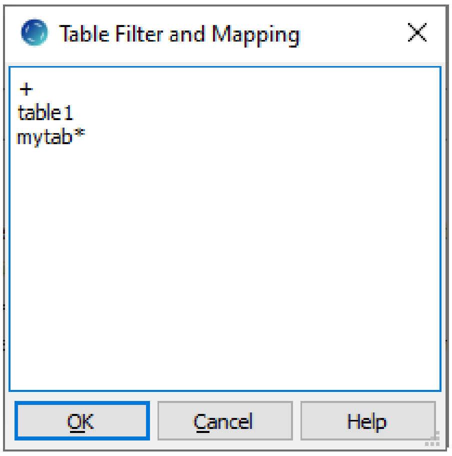


Table Filter and Mapping

```
+
table1
mytab*
```

Options

This section describes the options available for command **hvradapt**.

Parameter	Description
-d*tblname...*	Delete specified table from channel. No other tables are compared or changed.
-f*fname*	Write (append) list of modified or added HVR table names to file *fname*. This can be useful in a script which calls **hvradapt** and then does extra steps (e.g. **hvrrefresh**) for tables which were affected (see example below).
-h*class*	Location *class* of the hub database. Valid values for *class* are **db2**, **db2i**, **ingres**, **mysql**, **oracle**, **postgresql**, **sqlserver**, or **teradata**. For more information, see Calling HVR on the Command Line.
-i*letters*	Ignore certain differences. Value *letters* can contain: • **c** - Column was dropped. • **C** - Column was added. • **d** - Data type changed. • **D** - Data type family changed. • **f** - Column range became smaller. • **F** - Column range became bigger. • **h** - Distribution key removed. • **H** - Distribution key added. • **k** - Unique index removed. • **K** - Unique index added. • **n** - Nullability removed. • **N** - Nullability added. • **r** - Column was renamed. • **s** - Encoding changed.
-I	Controls whether HVR should convert catalog data types into data types that could be created in the DBMS. If not supplied then the data types are converted before they are compared. Otherwise the actual catalog data type is compared without any conversion.
-l*loc*	Specifies the adapt location *loc*, typically the channel's capture location.

-n*tablefilterfile*	Specifies a table filter file *tablefilterfile* for the channel. This file can contain 'table filter' statement(s) to define which base tables in the database should be included (or excluded) in the channel. The *tablefilterfile* can contain names of the schema, table, column, and/or a pattern (such as **mytbl***). Multiple table filter statements can be supplied in HVR GUI and CLI. For more information, see section Table Filter. In HVR GUI, the contents in table filter file can only be copy pasted into the **Table Filter** dialog (click **Edit** in the **Table Explore** dialog).
-N*tablefilterstmt*	Specifies a table filter statement (pattern) *tablefilterstmt*. This statement defines which base tables in the database should be included (or excluded) in the channel. The *tablefilterstmt* can contain names of the schema, table, column, and/or a pattern (such as **mytbl***). Multiple table filter statements can be supplied in HVR GUI and CLI. For more information, see section Table Filter. In HVR GUI, to specify the table filter statement, click **Edit** in the **Table Explore** dialog. In CLI, the table filter statement can also be supplied in a table filter file using option **-n**.
-R	Do not re-describe tables.
-r*tbls*	Re-describe only specific tables *tbls*.
-s*scope*	Add **TableProperties /Schema** to *scope.* Valid values for *scope* are: - **a** (**default**) - Add action to all (*) location(s). - **g** - Add action onto location's group - **l** - Add action onto the specific location).
-S*scope*	Add **ColumnProperties /Absent** to *scope* instead of updating the column information. The **default** is not to add any **ColumnProperties /Absent** to *scope*, but instead to delete the column information from the channel. This affects how the channel is changed when a column does not exist in the database but exist in the channel. If this option is supplied, a **ColumnProperties /Absent** is created. Valid values for *scope* are: - **g** - Add action onto location's group. - **l** - Add action onto the specific location.
-u*user*[*/pwd*]	Connect to hub database using DBMS account *user*. For some databases (e.g. SQL Server) a password must also be supplied.
-U*dictsch*	SAP dictionary tables are owned by DB schema *dictsch*. If this option is not supplied then HVR will look for the SAP dictionary tables in the HVR location's default schema, and also in schemas indicated in any **TableProperties /Schema** parameter. This option can only be used with option **-X**.
-V **Since** v5.6.0/0	Show views and materialized views. For Oracle, the materialized views are always shown.
-x	Check only mode. Do not apply any changes to the catalogs.

-X	Explore table and column information from SAP dictionaries, instead of DBMS catalogs. For SAP "transparent" tables this can give more correct data type information. It will also show the logical tables inside SAP "cluster" and "pool" tables. These are shown as 'unpacked' tables.

Table Filter

Hvradapt supplied with table filter statement (option **-n** or **-N**) allows you to define which base tables in the adapt location should be included in or excluded from the channel. Only tables matching any of the given statement will be included or excluded.

Syntax for Table Filter

This section describes the syntax for the classic table filter statement that can be used with options **-n** or **-N**.

```
[schema.]tablename [-T target_schema] [-K (col_list)]
```

```
![schema.]tablename
```

```
+
```

- Value *schema* or *tablename* can be a literal (optionally enclosed in double quotes) or a pattern matching can be done (only for tables or columns) using the special symbols *****, **?** or **[***characters***]**.
- Option **-K** marks the listed columns as distribution key columns in the 'column catalog', instead of defining a new **ColumnProperties /DistributionKey** action. In HVR GUI, marking a column as distribution key can be done from the table **Properties** dialog. For more information, see Marking a Column as Distribution Key in HVR GUI.
- Option **-T** defines the target schema into which the table should be replicated. **Hvradapt** will automatically define a new **TableProperties /Schema** action in this case.
- Options **-K** and **-T** only have an effect at the moment a table is added to channel, but are ignored otherwise.
- Special symbol **!** (NOT) is used to define negative patterns. This type of pattern can only be used after a regular/positive pattern and therefore cannot be used as an orphan (without other patterns) or the first pattern in the adapt template. Tables matching the preceding pattern and the negative pattern are excluded from the channel. For example,

  ```
  *                  # Match all tables in default schema.
  !tmp_*             # Exclude all tables whose name begin with 'tmp_' in
  default schema.
  ```

- Special symbol **+** matches all tables already in the channel.

Empty lines and comments (e.g. **#Test**) are ignored.

Example for Table Filter

Hvradapt can filter tables using a table filter file (option **-n**) or on the command line (option **-N**).

- Examples for filtering tables or schemas.
  The following is an example of table filter file (e.g. **/tmp/adapt.tmpl**):

```
+                        # Match all tables already in channel.
tbl1                     # Match table named 'tbl1' in default schema.
schema1.*                # Match all tables in 'schema1'.
schema2.tbl1             # Match table named 'tbl1' in 'schema2'.
history_*                # Match all tables whose name begin with
'history_' in default schema.
"my table"               # Match table named "my table" in default
schema. Use double quotes if there is a space in table name.
schema2."tbl.tab*"       # Match all tables whose name begin with 'tbl.
tab' in 'schema2'. Use double quotes if there is a special character in
the table name.
schema3.xx* -T schema4   # Match all tables whose name begin with 'xx'
in schema3 for replicating them to schema4.
tbl2 -K(col1 col2)       # Match table named 'tbl2' in default schema
and add columns named 'col1' and 'col2' as distribution key columns.
```

This table filter pattern can be supplied in the command line as:

```
hvradapt -N + -N tbl1 -N schema1.* -N schema2.tbl1 -N history_* -N "my
table" -N schema2."tbl.tab*" -N schema3.xx* -T schema4 -N tbl2 -K(col1
col2) -l mylocation hvrhub/hvrhub hvrdemochn
```

A shell script as shown in section Shell Script to Run Hvradapt can be created to run **hvradapt** for checking new or modified tables in a location.

Shell Script to Run Hvradapt

A shell script can be created to run **hvradapt** for checking new or modified tables in a location.

The following example demonstrates the use of a shell script to run **hvradapt** for checking new or modified tables in location **loc1** and if any new or modified tables are found in location **loc1**, the script executes the necessary commands to enroll the tables into the channel **mychn**.

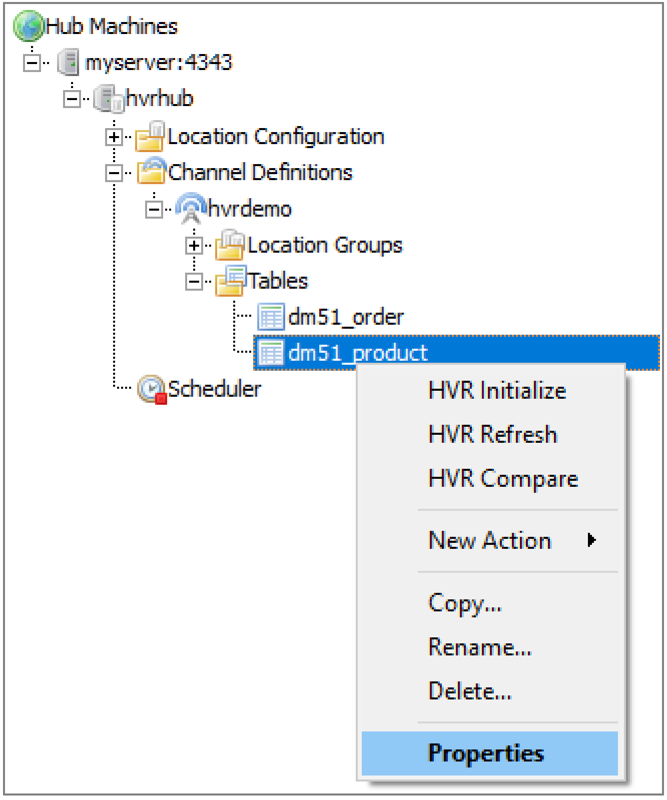
```
#!/bin/sh
hub=myhub/passwd                                    # Hub database and
password (if required)
chn=mychn                                           # Channel
src=loc1                                            # Source location
F=/tmp/adapt_$chn.out                               # File where hvradapt
writes list of new or changed tables
hvradapt -f$F -n/tmp/adapt.tmpl -l$src $hub $chn    # Add new or changed
tables from source to channel based on the patterns defined in the pattern
file.
if test -f $F                                       # If file $F exists then
new or changed tables were detected
then
    hvrsuspend $hub $chn-integ                      # Stop integrate jobs
    hvrinit -oelj $hub $chn                         # Regenerate
supplemental-logging, jobs and enroll info
    hvrrefresh -r$src -t$F -qrw -cbkr $hub $chn     # Re-create and online
refresh tables in file $F
    hvrstart -r -u $hub $chn-integ                  # Re-trigger stopped jobs
    hvradapt -x -l$src $hub $chn                    # Double-check channel
now matches target location (optional)

    rm $F                                           # Remove file with list
of new or changed tables
fi
```

Marking a Column as Distribution Key in HVR GUI

Perform the following steps to mark a column as distribution key in HVR GUI:

1. To view a table's details or properties, right-click on a table in the channel and select **Properties**.



2. By default, the distribution key column is not displayed in the table **Properties** dialog. To display this column in the table **Properties** dialog, right-click on the header and select **Distr. Key**.

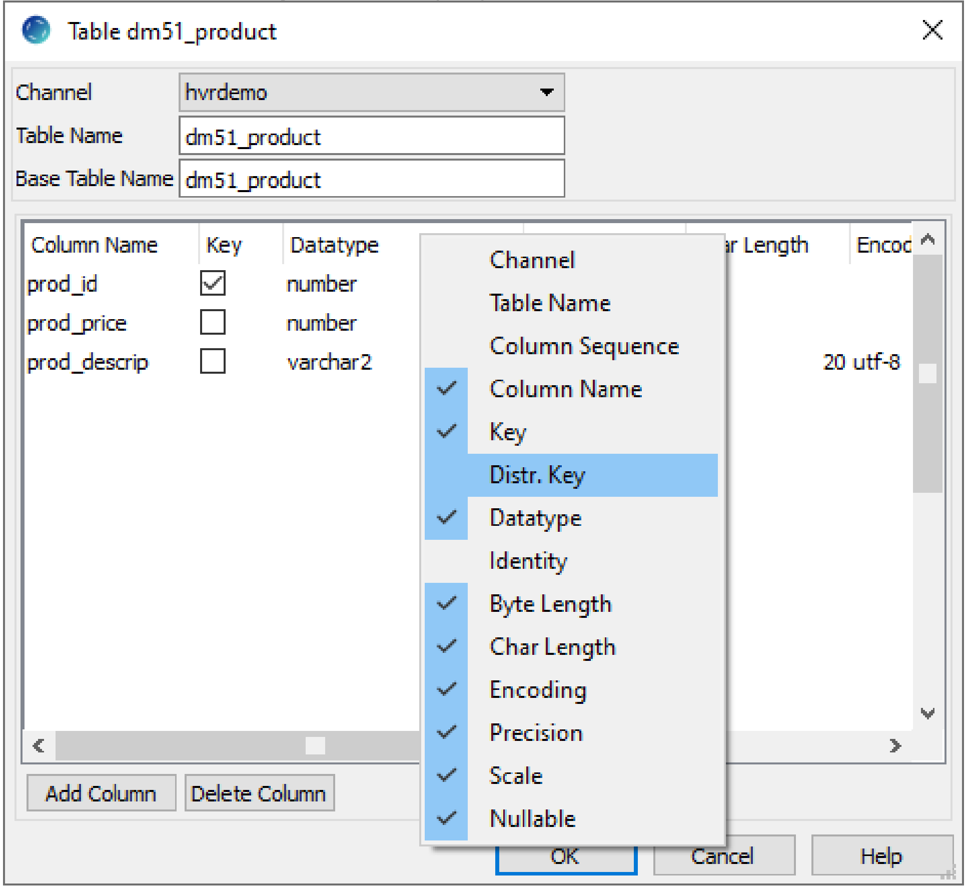


Table dm51_product

Channel: hvrdemo
Table Name: dm51_product
Base Table Name: dm51_product

Column Name	Key	Datatype	r Length	Encod
prod_id	☑	number		
prod_price	□	number		
prod_descrip	□	varchar2	20	utf-8

3. To mark a column as distribution key, select the respective column's checkbox available under **Distr. Key**.

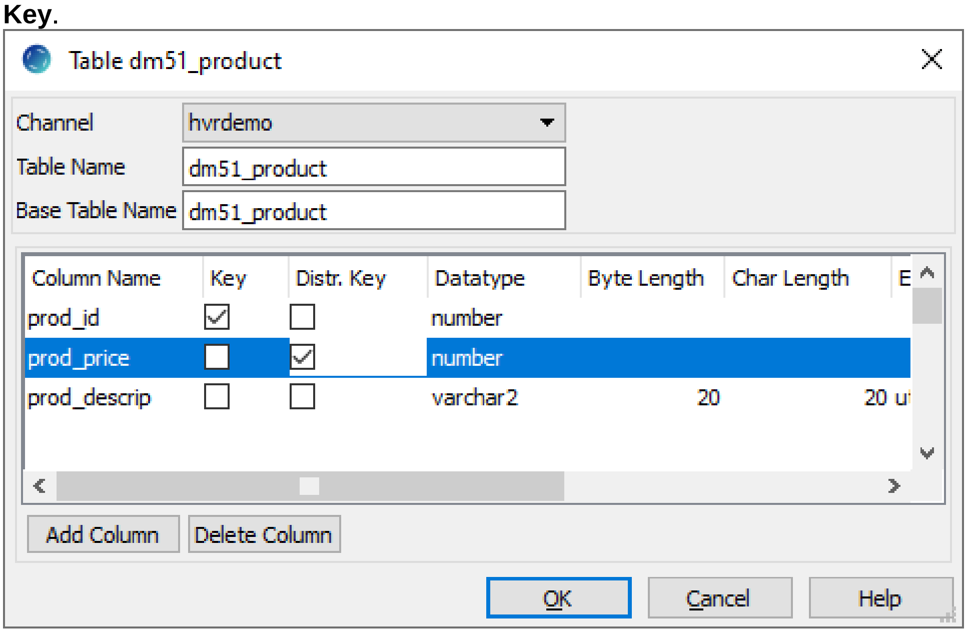


Table dm51_product

Channel: hvrdemo
Table Name: dm51_product
Base Table Name: dm51_product

Column Name	Key	Distr. Key	Datatype	Byte Length	Char Length	E
prod_id	☑	□	number			
prod_price	□	☑	number			
prod_descrip	□	□	varchar2	20	20	u