

Hvrrouterview

Contents

- [Name](#)
- [Synopsis](#)
- [Description](#)
- [General Options](#)
- [Restrict Options](#)
- [XML Options](#)
- [Extract Options](#)
- [Examples](#)
- [Files](#)
- [See Also](#)

Name

hvrrouterview - View or extract contents from internal router files.

Synopsis

```
hvrrouterview [-restrict opts] [-xml opts] [-F] hubdb chn [txfile]...
```

```
hvrrouterview -xtgt [-restrict opts] [-extract opts] [-F] hubdb chn [txfile]...
```

```
hvrrouterview -s [-cloc] [-i/od] [-xml opts] hubdb chn
```

```
hvrrouterview [-xml opts] hubdb chn hvrfile...
```

Description

This command can be used to view or extract data from internal HVR files such as transaction and journal files in the router directory on the hub machine.

The first form (in the Synopsis above) shows the contents of any transaction files currently in the channel's router directory. Options **-b**, **-c**, **-e**, **-f**, **-i**, **-n**, **-t** and **-w** can be used to restrict the changes shown. Option **-j** shows journal files instead of transaction files. The output is shown as XML, which is sent to **stdout**.

The second form (with option **-x**) extracts the data from the transaction files into a target, which should be either a database (for database changes) or a directory (for a blob file channel).

The third form (with option **-s**) shows the contents of control files as XML.

The fourth form can be used to view the contents of many internal HVR files, such as a ***.enroll** or ***.cap_state** file in a **router** directory, any of the files in a file location's **_hvr_state** directory, a control file (in directory **\$HVR_CONFIG/router/*hub/chnl/control***) or the GUI preferences file (**\$HVR_CONFIG/files/*hvr*gui.ini**).

The first argument *hubdb* specifies the connection to the hub database. For more information about supported hub databases, see [Calling HVR on the Command Line](#).

General Options

Parameter	Description
-----------	-------------

-F	Identifies transaction file as captured by a 'blob file' channel (these use a different decompression algorithm). This is seldom necessary, because HVR should deduce the decompression algorithm from the basename of the transaction file.
-s	View contents of control files instead of transaction files. Control files are created by Hvrrefresh with option -q , or by command Hvrcontrol .
-u <i>user[/pwd]</i>	Connect to hub database using DBMS account <i>user</i> . For some databases (e.g. SQL Server) a password <i>pwd</i> must also be supplied.

Restrict Options

Parameter	Description
-b <i>time</i>	Only changes after capture time. Valid formats are <i>YYYY-MM-DD [HH:MM:SS]</i> in local time or <i>YYYY-MM-DDTHH:MM:SS+TZD</i> or <i>YYYY-MM-DDTHH:MM:SSZ</i> or today or now [[±]SECS] or an integer (seconds since 1970-01-01 00:00:00 UTC). For example, -b "2010-07-29 10:30:21" or -b now-3600 (changes in the last hour). This option is equivalent to -whvr_cap_tstamp>=time .
-c <i>loc</i>	Only changes from specific capture location.
-e <i>time</i>	Only changes before capture time. Valid formats are <i>YYYY-MM-DD [HH:MM:SS]</i> in local time or <i>YYYY-MM-DDTHH:MM:SS+TZD</i> or <i>YYYY-MM-DDTHH:MM:SSZ</i> or today or now [[±]SECS] or an integer (seconds since 1970-01-01 00:00:00 UTC). For example, -b "2010-07-29 10:30:21" or -b now-3600 (changes in the last hour). This option is equivalent to -whvr_cap_tstamp>=time .
-f <i>txfile</i>	Contents of a specific transaction file. This option can be specified multiple times. Another way to see the contents of a specific transaction file is to list the file(s) after the channel name (as a third positional parameter). The advantage is that 'globbing' can be used for a list of files (e.g. *.tx_1*) whereas -f only accepts one file (although it can be supplied multiple times).
-i <i>loc</i>	Only changes for a specific integrate location.
-j	Show the contents of HVR journal files in directory \$HVRCONFIG/jnl/hubdblchn . These files are kept if parameter /Journal is defined for Integrate .
-n	Newest. Only most recent transaction file(s).
-t <i>y</i>	Only rows for tables specified by <i>y</i> . Values of <i>y</i> may be one of the following: <ul style="list-style-type: none"> <i>tbl</i>: Only table <i>tbl</i>. <i>t1-t2</i>: All tables that fall alphabetically between <i>t1</i> and <i>t2</i> inclusive. !tbl: All tables except <i>tbl</i>. !t1-t2: All tables except for those that fall alphabetically between <i>t1</i> and <i>t2</i> inclusive. Several -ty instructions can be supplied together.

-w where	<p><i>Where</i> condition which must have form <i>columnname operator value</i>.</p> <p>The <i>operator</i> can be either = != <> > < >= or <=. The <i>value</i> can be a number, 'str', X'hex', or a date. Valid date formats are YYYY-MM-DD [HH:MM:SS] in local time or YYYY-MM-DDTHH:MM:SS+TZD or YYYY-MM-DDTHH:MM:SSZ or today or now [[±]SECS] or an integer (seconds since 1970-01-01 00:00:00 UTC). For some operators (= != <>) the value can be a list separated by ' '.</p> <p>If multiple -w options are supplied then they are AND-ed together.</p>
-----------------	---

XML Options

Parameter	Description
-d	Show column data type information.
-h	Print data values in hexadecimal format.

Extract Options

Parameter	Description
-x tgt	Extract data from transaction files into a target. Value <i>tgt</i> should be either an actual database name (e.g. myuser/pwd) or a directory (e.g. /tmp) and not a HVR location name. By default, this target should be on the hub machine, unless option -R is specified, in which case hvrrouerview will extract the data to a target on a different machine.
-Cpubcert	SSL public certificate of remote location. This must be used with options -x and -R .
-En=v	Set environment variable <i>n</i> to value <i>v</i> for the HVR processes started on the remote node.
-Kpair	SSL public certificate and private key pair of hub location.
-Llogin/pwd	Login and password for remote node. This must be used with options -x and -R .
-Rnode:port	Remote node name and HVR port number so data is extracted to remote target. This must be used with option -x .

Examples

To show the contents of certain transaction files:

```
cd $HVR_CONFIG/router/hubdb/hvr_demo01/loc_cen
```

```
hvrrouerview hubdb chn *.tx_dec01
```

To show the contents of file **4a82a8e8_c31e6.cap_state**:

```
hvrrouerview hubdb chn 4a82a8e8_c31e6.cap_state
```

To show any changes for table **cust** from the journals:

```
hvrrouterview -j -tcust hubdb chn
```

To retrieve all files moved by a blob file channel in the last hour use the following command. The data is read from the channel's journals and the extracted files are written into **/tmp**.

```
hvrrouterview -F -j -bnow-3600 -x/tmp hubdb chn
```

Files

▼  router	Directory containing replication state.
▼  <i>hub</i>	
▼  <i>chn</i>	
▼  catalog	
 <i>timestamp.cache</i>	Cache of HVR catalogs used for routing. This is refreshed if option -or is supplied.
▼  control	
 <i>tstamp.ctrl-jobname-control</i>	Control file containing instructions for a replication job. The contents of the file can be inspected using command hvrrouterview .
▼  loc_caploc	
 <i>timestamp.tx_intloc</i>	Data captured from location <i>caploc</i> that has been routed to location <i>intloc</i> . The contents of this file can be viewed using command hvrrouterview . If the base name of this file (<i>timestamp</i>) is bigger than the base name of any *.cap_state file, then this router data is not yet revealed and is still invisible to integrate locations.
 <i>timestamp.cap_state</i>	Timestamps and capture status of capture job. The contents of this file can be viewed with command hvrrouterview .

See Also

Command [Hvrinit](#).