

Agent Plugin for Cassandra

Contents

- [Name](#)
- [Synopsis](#)
- [Description](#)
- [Options](#)
- [Environment Variables](#)
- [Installing Python Environment](#)
- [Use Case](#)

Name

`hvincassagent.py`

Synopsis

`hvincassagent.py mode chn loc [userargs]`

Description

The [agent plugin Hvincassagent](#) enables HVR to replicate data into Cassandra database. This agent plugin should be defined in the HVR channel using action [AgentPlugin](#). The behaviour of this agent plugin depends on the *options* supplied in **/UserArgument** field of [AgentPlugin](#) screen.

This agent plugin supports only Cassandra data type **text**.

Options

This section describes the parameters that can be used with **Hvincassagent**:

Parameter	Description
-p	Preserves existing row(s) in target during refresh and appends data into table. Not applicable if table structure has been changed. If this option is not defined, truncates existing data from target, then recreates table and insert new rows.
-s	Converts DELETE in source location as UPDATE in target location. To indicate a delete in source, the extra column hvr_is_deleted available only in target is updated as "1". For more information, see ColumnProperties /SoftDelete .
-t <i>timecol</i>	Converts all changes (INSERT, UPDATE, DELETE) in source location as INSERT in target location. For more information, see ColumnProperties /TimeKey .

The column name **hvr_is_deleted** is hardcoded into this plugin, so it is not allowed to change this name.

Environment Variables

The [Environment](#) variables listed in this section should be defined when using this agent plugin:

Environment Variable Name	Description
\$HVR_CASSANDRA_PORT	The port number of the Cassandra server. If this environment variable is not defined, then the default port number 9042 is used.
\$HVR_CASSANDRA_HOST	The IP address or hostname of the Cassandra server. It is mandatory to define this environment variable.
\$HVR_CASSANDRA_KEYSPACE	The name of Cassandra keyspace. It is mandatory to define this environment variable.
\$HVR_CASSANDRA_USER	The username to connect HVR to Cassandra database. The default value is blank (blank password - leave field empty to connect). This environment variable is used only if Cassandra requires authorization.
\$HVR_CASSANDRA_PWD	The password of the \$HVR_CASSANDRA_USER to connect HVR to Cassandra database.

Installing Python Environment

To enable data upload into Cassandra using HVR, perform the following on HVR [Integrate](#) machine:

1. Install Python 2.7.x +/3.x. Skip this step if the mentioned python version is already installed in the machine.
2. Install the following python client modules:

```
pip install cassandra-driver
pip install six
pip install scales
pip install enum
```

Use Case

Use Case 1: Cassandra tables with plain insert/update/delete.

Group	Table	Action
CASS	*	Integrate /Burst
CASS	*	FileFormat /Csv /QuoteCharacter="
CASS	*	AgentPlugIn /Command=hvrcassagent.py /Context=! preserve_during_refr
CASS	*	AgentPlugIn /Command=hvrcassagent.py /UserArgument="-p" /Context=preserve_during_refr
CASS	*	Environment /Name=HVR_CASSANDRA_HOST /Value=<valid host list comma separated>
CASS	*	Environment /Name=HVR_CASSANDRA_KEYSPACE /Value=<valid keyspace>

In this use case, during the execution of mode **refr_write_begin**,

- If option **-p** is not defined, then HVR drops and recreates each Cassandra table.

- If option **-p** is defined, then HVR appends data into the Cassandra table. If the table does not exist in target, then creates table.

During the execution of mode **refr_write_end** and **integ_end**,

- HVR loads data from CSV file into Cassandra table.

Use Case 2: Cassandra tables with soft delete column.

Group	Table	Action
CASS	*	Integrate /Burst
CASS	*	FileFormat /Csv /QuoteCharacter="
CASS	*	ColumnProperties /Name=hvr_is_deleted /Extra /SoftDelete
CASS	*	AgentPlugIn /Command=hvrcassagent.py /UserArgument="-s" /Context=!preserve_during_refr
CASS	*	AgentPlugIn /Command=hvrcassagent.py /UserArgument="-p -s" /Context=preserve_during_refr
CASS	*	Environment /Name=HVR_CASSANDRA_HOST /Value=<valid host list comma separated>
CASS	*	Environment /Name=HVR_CASSANDRA_KEYSPACE /Value=<valid keyspace>

In this use case, during the execution of mode **refr_write_begin**,

- If option **-p** is not defined, then HVR drops and recreates each Cassandra table with an extra column **hvr_is_deleted**.
- Else do create-if-not-exists instead.

During the execution of mode **refr_write_end** and **integ_end**,

- HVR loads data from CSV file into Cassandra table.

Use Case 3: Cassandra tables with timekey column.

Group	Table	Action
CASS	*	Integrate /Burst
CASS	*	FileFormat /Csv /QuoteCharacter="
CASS	*	ColumnProperties /Name=hvr_op_val /Extra /IntegrateExpression={hvr_op} /Datatype=int
CASS	*	ColumnProperties /Name=hvr_integ_key /Extra /IntegrateExpression={hvr_integ_seq} /TimeKey /Key /Datatype=varchar /Length=36
CASS	*	AgentPlugIn /Command=hvrcassagent.py /UserArgument="-t" /Context=!preserve_during_refr
CASS	*	AgentPlugIn /Command=hvrcassagent.py /UserArgument="-t -p" /Context=preserve_during_refr
CASS	*	Environment /Name=HVR_CASSANDRA_HOST /Value=<valid host list comma separated>
CASS	*	Environment /Name=HVR_CASSANDRA_KEYSPACE

	/Value=<valid keyspace>
--	-------------------------

In this use case, during the execution of mode **refr_write_begin**,

- If option **-p** is not defined, then HVR drops and recreates each Cassandra table with two extra columns **hvr_op_val**, **hvr_integ_key**.
- Else do create-if-not-exists instead.

During the execution of mode **refr_write_end** and **integ_end**,

- HVR loads data from CSV file into Cassandra table.