

Agent Plugin for MongoDB

Contents

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

Name

hvrmongodbagent.py

Synopsis

hvrmongodbagent.py *mode chn loc* [*userargs*]

Description

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

This agent plugin supports replication of data in JSON format only and it is mandatory to define action **1=FileFormat /JsonMode=ROW_FRAGMENTS**.

Options

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

Parameter	Description
-r	Truncates existing data from target and then recreates table and insert new rows. If this option is not defined, appends data into table.
-s <i>col_name</i>	Soft deletes the column <i>col_name</i> .

Environment Variables

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

Environment Variable Name	Description
\$HVR_MONGODB_DATABASE	The name of the database on MongoDB server.
\$HVR_MONGODB_HOST	The IP address or hostname of the MongoDB server.

\$HVR_MONGODB_PORT	The port number of the MongoDB server. If this environment variable is not defined, then the default port number 27017 is used.
\$MONGODB_COLLECTION	Support for the special substitutions - hvr_tbl_name , hvr_base_name and hvr_schema . Example: Source database contains a table TEST1 . In HVR catalog this table has following names: TEST1 and TEST1_BASE . Destination schema REMOTE_USER (defined using Environment variable \$HVR_SCHEMA). So, if \$HVR_MONGODB_COLLECTION is defined as {hvr_schema}. {hvr_base_name}_{hvr_tbl_name}_tag , it will be encoded as REMOTE_USER. TEST1_BASE_TEST1_tag .

Installing Python Environment and MongoDB Client

MongoDB client is required for uploading data into MongoDB from local source and convert it into MongoDB collections. To enable data upload into MongoDB 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 pymongo (version > 3.0)
pip install enum
```

Use Case

- **Use Case 1:** MongoDB collections with timekey column.

Group	Table	Action
FILE	*	Integrate /ReorderRows=SORT_COALESCE /RenameExpression="{hvr_integ_tstamp}-{hvr_tbl_name}.json"
FILE	*	FileFormat /Json /JsonMode=ROW_FRAGEMENTS
FILE	*	ColumnProperties /Name=hvr_op_val /Extra /IntegrateExpression="{hvr_op}" /Datatype=integer
FILE	*	ColumnProperties /Name=hvr_integ_seq /Extra /IntegrateExpression="{hvr_integ_seq}" /Datatype=varchar /Length=24 /Key /TimeKey
FILE	*	AgentPlugIn /Command=hvrmongodbagent.py /Context=!preserve
FILE	*	AgentPlugIn /Command=hvrmongodbagent.py /UserArgument="-r" /Context=preserve
FILE	*	Environment /Name=HVR_MONGODB_HOST /Value=<host>
FILE	*	Environment /Name=HVR_MONGODB_PORT /Value=<port>

FILE	*	Environment /Name=HVR_MONGODB_COLLECTION /Value={hvr_tbl_name}
FILE	*	Environment /Name=HVR_MONGODB_DATABASE /Value=<database>

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

- If option **-r** is not defined, then HVR appends new row into MongoDB Collection.
- If option **-r** is defined, then HVR re-creates MongoDB Collection and inserts new rows.

Tables are mapped to MongoDB collection. Each collection contains documents and each document is mapped to one row from file.

- **Use Case 2:** MongoDB collections with timekey column and static collection name.

Group	Table	Action
FILE	*	Integrate /ReorderRows=SORT_COALESCE /RenameExpression="{hvr_integ_tstamp}-{hvr_tbl_name}.json"
FILE	*	FileFormat /Json /JsonMode=ROW_FRAGMENTS
FILE	*	ColumnProperties /Name=hvr_op_val /Extra /IntegrateExpression={hvr_op} /Datatype=integer
FILE	*	ColumnProperties /Name=hvr_integ_seq /Extra /IntegrateExpression={hvr_integ_seq} /Datatype=varchar /Length=24 /Key /TimeKey
FILE	*	ColumnProperties /Name=table_name /Extra /IntegrateExpression={hvr_tbl_name} /Datatype=varchar /Length=1000
FILE	*	AgentPlugIn /Command=hvrmongodbagent.py /Context=!preserve
FILE	*	AgentPlugIn /Command=hvrmongodbagent.py /UserArgument="-r" /Context=preserve
FILE	*	Environment /Name=HVR_MONGODB_HOST /Value=<host>
FILE	*	Environment /Name=HVR_MONGODB_PORT /Value=<port>
FILE	*	Environment /Name=HVR_MONGODB_COLLECTION /Value=collection_name
FILE	*	Environment /Name=HVR_MONGODB_DATABASE /Value=<database>

- **Use Case 3:** MongoDB collection with softdelete column and dynamic collection name.

Group	Table	Action
FILE	*	Integrate /ReorderRows=SORT_COALESCE /RenameExpression="{hvr_integ_tstamp}-{hvr_tbl_name}.json"
FILE	*	FileFormat /Json /JsonMode=ROW_FRAGMENTS

FILE	*	ColumnProperties /Name=hvr_is_deleted /Extra /SoftDelete /Datatype=integer
FILE	*	ColumnProperties /Name=hvr_integ_tstamp /Extra /IntegrateExpression={hvr_integ_tstamp} /Datatype=timestamp
FILE	*	AgentPlugIn /Command=hvrmongodbagent.py /UserArgument="-s hvr_is_deleted" /Context=!preserve
FILE	*	AgentPlugIn /Command=hvrmongodbagent.py /UserArgument="-r -s hvr_is_deleted" /Context=preserve
FILE	*	Environment /Name=HVR_MONGODB_HOST /Value=<host>
FILE	*	Environment /Name=HVR_MONGODB_PORT /Value=<port>
FILE	*	Environment /Name=HVR_MONGODB_COLLECTION /Value={hvr_tbl_name}
FILE	*	Environment /Name=HVR_MONGODB_DATABASE /Value=<database>

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

- If option **-r** is not defined, then HVR appends new row into MongoDB Collection.
- If option **-r** is defined, then HVR re-creates MongoDB Collection and inserts new rows.

_id is a special name for the unique document identifier. The extra column **_id** is built based on key columns in table.

All values are converted to string like {"c1": 100, "c2": "string", "c3": value, "hvr_is_deleted": 1} where **c1** and **c2** are key columns. So **_id** will look like {"_id": "100string"}.

- **Use Case 4:** MongoDB collection with softdelete column and static collection name.

In case of using static collection names for all tables in channel, a new synthetic key column should be added.

Group	Table	Action
FILE	*	Integrate /ReorderRows=SORT_COALESCE /RenameExpression="{hvr_integ_tstamp}-{hvr_tbl_name}.json"
FILE	*	FileFormat /Json /JsonMode=ROW_FRAGMENTS
FILE	*	ColumnProperties /Name=hvr_is_deleted /Extra /SoftDelete /Datatype=integer
FILE	*	ColumnProperties /Name=hvr_integ_tstamp /Extra /IntegrateExpression={hvr_integ_tstamp} /Datatype=timestamp
FILE	*	ColumnProperties /Name=table_name /Extra /IntegrateExpression={hvr_tbl_name} /Key /Datatype=varchar /Length=1000
FILE	*	AgentPlugIn /Command=hvrmongodbagent.py /UserArgument="-s hvr_is_deleted" /Context=!preserve

FILE	*	AgentPlugIn /Command=hvrmongodbagent.py /UserArgument="-r -s hvr_is_deleted" /Context=preserve
FILE	*	Environment /Name=HVR_MONGODB_HOST /Value=<host>
FILE	*	Environment /Name=HVR_MONGODB_PORT /Value=<port>
FILE	*	Environment /Name=HVR_MONGODB_COLLECTION /Value=collection_name
FILE	*	Environment /Name=HVR_MONGODB_DATABASE /Value=<database>