**hawqunload tool**

Contents

[Introduction 1](#_Toc459971381)

[Synopsys 1](#_Toc459971382)

[Pre-Requisites 1](#_Toc459971383)

[Installation 2](#_Toc459971384)

[Description 2](#_Toc459971385)

[Options 2](#_Toc459971386)

[Connection Options 3](#_Toc459971387)

[Control File Format 3](#_Toc459971388)

[Control File Schema Element Descriptions 4](#_Toc459971389)

[Control File Schema Element Descriptions (Windows specific) 7](#_Toc459971390)

[Notes 8](#_Toc459971391)

[Log File Format 8](#_Toc459971392)

[Examples 8](#_Toc459971393)

# Introduction

Runs an extract job as defined in a YAML formatted control file.

# Synopsys

With control file:

hawqunload -f <control\_file> [-l <log\_file>] [-h <hostname>] [-p <port>]

[-U <username>] [-d <database>] [-W] [--gpfdist\_timeout <seconds>]

[[-v | -V] [-q]] [-D]

Without control file:

hawqunload -t <source\_table\_name> [--output\_dir <output\_directory>] [-l <log\_file>] [-h <hostname>]

[-p <port>] [-U <username>] [-d <database>] [-W] [--gpfdist\_timeout <seconds>]

[[-v | -V] [-q]] [-D]

Other commands:

hawqunload -?

hawqunload --version

# Pre-Requisites

The client machine where hawqunload is executed must have the following:

\* Python 2.6.2 or later, pygresql (the Python interface to PostgreSQL), and pyyaml. Note that Python and the required Python libraries are included with the Greenplum Database server installation, so if you have Greenplum Database installed on the machine where hawqunload is running, you do not need a separate Python installation.

Note: Greenplum UnLoaders for Windows supports only Python 2.5

(available from www.python.org).

\* The gpfdist parallel file distribution program installed and in your $PATH. This program is located in $GPHOME/bin of your Greenplum Database server installation.

\* Network access to and from all hosts in your Greenplum Database array (master and segments).

\* Network access to and from the hosts where the data to be extracted resides (ETL servers).

# Installation

On Greenplum Cluster:

1. Put the scripts hawqunload and hawqunload.py in the directory **$GPHOME/bin**
2. Put the help file hawqunload\_help in the directory **$GPHOME/docs/cli\_help**

# Description

hawqunload is a data extracting utility that acts as an interface to Greenplum Databases writable external table parallel extracting feature. Using a unload specification defined in a YAML formatted control file, hawqunload executes an extract by invoking the Greenplum parallel file server (gpfdist), creating a writable external table definition based on the source data defined, and executing an INSERT operation into the external table to extract the source table into the target file.

# Options

-f <control\_file>

Required if the option -t is not used. A YAML file that contains the unload specification details. See following section "Control File Format".

This option is not compatible with the option -t.

-t <source\_table\_name>

Required if the option -f is not used. This option prevents to use a YAML file and simply extracts the source table with no transformation into a csv file.

This option is not compatible with the option -f.

--output\_dir <output\_directory>

Only used with the option -t. Defines the directory in which the extracted file will be written.

If not set, the default output directory is /tmp/

--gpfdist\_timeout <seconds>

Sets the timeout for the gpfdist parallel file distribution program to send a response. Enter a value from 0 to 30 seconds (entering "0" to disables timeouts). Note that you might need to increase this value when operating on high-traffic networks.

-l <log\_file>

Specifies where to write the log file.

Defaults to ~/gpAdminLogs/hawqunload\_YYYYMMDD. See Also: LOG FILE FORMAT section.

-q (no screen output)

Run in quiet mode. Command output is not displayed on the screen, but is still written to the log file.

-D (debug mode)

Check for error conditions, but do not execute the unload.

-v (verbose mode)

Show verbose output of the unload steps as they are executed.

-V (very verbose mode)

Shows very verbose output.

-? (show help)

Show help, then exit.

--version

Show the version of this utility, then exit.

# Connection Options

-d <database>

The database to extract from. If not specified, reads from the unload control file, the environment variable $PGDATABASE or defaults to the current system user name.

-h <hostname>

Specifies the host name of the machine on which the Greenplum master database server is running. If not specified, reads from the unload control file, the environment variable $PGHOST or defaults to localhost.

-p <port>

Specifies the TCP port on which the Greenplum master database server is listening for connections. If not specified, reads from the unload control file, the environment variable $PGPORT or defaults to 5432.

-U <username>

The database role name to connect as. If not specified, reads from the unload control file, the environment variable $PGUSER or defaults to the current system user name.

-W (force password prompt)

Force a password prompt. If not specified, reads the password from the environment variable $PGPASSWORD or from a password file specified by $PGPASSFILE or in ~/.pgpass.

If these are not set, then hawqunload will prompt for a password even if -W is not supplied.

# Control File Format

The hawqunload control file uses the YAML 1.1 document format and then implements its own schema for defining the various steps of a Greenplum Database unload operation. The control file must be a valid YAML document.

The hawqunload program processes the control file document in order and uses indentation (spaces) to determine the document hierarchy and the relationships of the sections to one another. The use of white space is significant. White space should not be used simply for formatting purposes, and tabs should not be used at all.

The basic structure of an extract control file is:

---

VERSION: 1.0.0.1

DATABASE: <db\_name>

USER: <db\_username>

HOST: <master\_hostname>

PORT: <master\_port>

HAWQUNLOAD:

INPUT:

- TABLE: <schema>.<table\_name>

- EXTRACT\_CONDITION: <filter>

OUTPUT:

- TARGET:

LOCAL\_HOSTNAME:

- <hostname\_or\_ip>

PORT: <http\_port>

| PORT\_RANGE: [<start\_port\_range>, <end\_port\_range>]

WINDOWS\_DISK: <disk letter on Windows>

FILE:

- </path/to/output\_file>

SSL: true | false

CERTIFICATES\_PATH: </path/to/certificates>

- COLUMNS:

- <field\_name>: <data\_type>

- MAPPING:

<target\_column\_name>: <source\_column\_name> | '<expression>'

- MAX\_LINE\_LENGTH: <integer>

- FORMAT: text | csv

- DELIMITER: '<delimiter\_character>'

- ESCAPE: '<escape\_character>' | 'OFF'

- NULL\_AS: '<null\_string>'

- FORCE\_QUOTE:

- <field\_name>

- QUOTE: '<csv\_quote\_character>'

- HEADER: true | false

- ENCODING: <database\_encoding>

EXTERNAL:

- SCHEMA: <schema> | '%'

OPTIONS:

- APPEND: true | false

- REUSE\_TABLES: true | false

SQL:

- BEFORE: "<sql\_command>"

- AFTER: "<sql\_command>"

# Control File Schema Element Descriptions

VERSION - Optional.

The version of the hawqunload control file schema. The current version is 1.0.0.1.

DATABASE - Optional.

Specifies which database in Greenplum to connect to. If not specified, defaults to $PGDATABASE if set or the current system user name. You can also specify the database on the command line using the -d option.

USER - Optional.

Specifies which database role to use to connect. If not specified, defaults to the current user or $PGUSER if set.

You can also specify the database role on the command line using the -U option.

If the user running hawqunload is not a Greenplum superuser, then the server configuration parameter gp\_external\_grant\_privileges must be set to on in order for the load to be processed. See the "Greenplum Database Reference Guide" for more information.

HOST - Optional.

Specifies Greenplum master host name. If not specified, defaults to localhost or $PGHOST if set. You can also specify the master host name on the command line using the -h option.

PORT - Optional.

Specifies Greenplum master port. If not specified, defaults to 5432 or $PGPORT if set. You can also specify the master port on the command line using the -p option.

HAWQUNLOAD - Required.

Begins the load specification section. A HAWQUNLOAD specification must have an INPUT and an OUTPUT section defined.

OUTPUT - Required.

Defines the location and the format of the output data to be unloaded. hawqunload will start one or more instances of the gpfdist file distribution program on the current host and create the required external table definition(s) in Greenplum Database that point to the source data.

Note that the host from which you run hawqunload must be accessible over the network by all Greenplum hosts (master and segments).

TARGET - Required.

The TARGET block of an OUTPUT specification defines the location of a target file. An OUTPUT section can have more than one TARGET block defined. Each TARGET block defined corresponds to one instance of the gpfdist file distribution program that will be started on the local machine. Each TARGET block defined must have a FILE specification.

For more information about using the gpfdist parallel file server and single and multiple gpfdist instances, see the "Greenplum Database Database Administrator Guide."

LOCAL\_HOSTNAME - Optional.

Specifies the host name or IP address of the local machine on which hawqunload is running. If this machine is configured with multiple network interface cards (NICs), you can specify the host name or IP of each individual NIC to allow network traffic to use all NICs simultaneously. The default is to use the local machines primary host name or IP only.

PORT - Optional.

Specifies the specific port number that the gpfdist file distribution program should use. You can also supply a PORT\_RANGE to select an available port from the specified range.

If both PORT and PORT\_RANGE are defined, then PORT takes precedence.

If neither PORT or PORT\_RANGE are defined, the default is to select an available port between 8000 and 9000.

If multiple host names are declared in LOCAL\_HOSTNAME, this port number is used for all hosts. This configuration is desired if you want to use all NICs to load the same file or set of files in a given directory location.

PORT\_RANGE - Optional.

Can be used instead of PORT to supply a range of port numbers from which hawqunload can choose an available port for this instance of the gpfdist file distribution program.

FILE - Required.

Specifies the location of a file, named pipe, or directory location on the local file system where the data are extracted. You can declare more than one file so long as the data is of the same format in all files specified.

SSL - Optional.

Specifies usage of SSL encryption. If SSL is set to true, hawqunload starts the gpfdist server with the --ssl option and uses the gpfdists protocol.

CERTIFICATES\_PATH - Required when SSL is true

cannot be specified when SSL is false or unspecified. The location specified in CERTIFICATES\_PATH must contain the following files:

\* The server certificate file, server.crt

\* The server private key file, server.key

\* The trusted certificate authorities, root.crt

The root directory (/) cannot be specified as CERTIFICATES\_PATH.

COLUMNS - Optional.

Specifies the schema of the target data file(s) in the format of <field\_name>: <data\_type>. The DELIMITER character in the target file is what separates two data value fields (columns).

A row is determined by a line feed character (0x0a).

If the output COLUMNS are not specified, then the schema of the input TABLE is implied, meaning that the target file must have the same column order, number of columns, and data format as the source table.

The default source-to-target mapping is based on a match of column names as defined in this section and the column names in the target TABLE.

This default mapping can be overridden using the MAPPING section.

MAX\_LINE\_LENGTH - Optional.

An integer that specifies the maximum length of a line passed to gpfdist.

FORMAT - Optional.

Specifies the format of the target data file(s) - either plain text (TEXT) or comma separated values (CSV) format. Defaults to TEXT if not specified. For more information about the format of the target data, see the "Greenplum Database Database Administrator Guide."

DELIMITER - Optional.

Specifies a single ASCII character that separates columns within each row (line) of data. The default is a tab character in TEXT mode, a comma in CSV mode. You can also specify a non-printable ASCII character or a non-printable unicode character, for example: "\x1B" or "\u001B". The escape string syntax, E'<character-code>', is also supported for non-printable characters. The ASCII or unicode character must be enclosed in single quotes. For example: E'\x1B' or E'\u001B'.

ESCAPE - Optional.

Specifies the single character that is used for C escape sequences (such as \n,\t,\100, and so on) and for escaping data characters that might otherwise be taken as row or column delimiters. Make sure to choose an escape character that is not used anywhere in your actual column data. The default escape character is a \ (backslash) for text-formatted files and a " (double quote) for csv-formatted files, however it is possible to specify another character to represent an escape. It is also possible to disable escaping in text-formatted files by specifying the value 'OFF' as the escape value. This is very useful for data such as text-formatted web log data that has many embedded backslashes that are not intended to be escapes.

NULL\_AS - Optional.

Specifies the string that represents a null value.

The default is \N (backslash-N) in TEXT mode, and an empty value with no quotations in CSV mode. You might prefer an empty string even in TEXT mode for cases where you do not want to distinguish nulls from empty strings. Any source data item that matches this string will be considered a null value.

FORCE\_QUOTE - Optional.

Specifies the columns that has to be enclosed inside quote for CSV mode.

QUOTE - Required when FORMAT is CSV.

Specifies the quotation character for CSV mode. The default is double-quote (").

HEADER - Optional.

Specifies that a header row (contains the names of the columns) has to be added as the first line in the target file(s).

If using multiple data target files, each file will have a header row.

The default is that the output files do not have a header row.

ENCODING - Optional.

Character set encoding of the source data. Specify a string constant (such as 'SQL\_ASCII'), an integer encoding number, or 'DEFAULT' to use the default client encoding. If not specified, the default client encoding is used. For information about supported character sets, see the "Greenplum Database Reference Guide."

EXTERNAL - Optional.

Defines the schema of the external table database objects created by hawqunload. The default is to use the Greenplum Database search\_path.

SCHEMA - Required when EXTERNAL is declared.

The name of the schema of the external table. If the schema does not exist, an error is returned.

If % (percent character) is specified, the schema of the table name specified by TABLE in the INPUT section is used.

If the table name does not specify a schema, the default schema is used.

INPUT - Required.

Define the source table that is to be unloaded from the database.

TABLE - Required.

The name of the source table to unload from.

EXTRACT\_CONDITION - Optional.

Filter the data to extract: it adds a WHERE clause to the SELECT query.

MAPPING - Optional.

If a mapping is specified, it overrides the default source-to-target column mapping. The default source-to-target mapping is based on a match of column names as defined in the target COLUMNS section and the column names of the source TABLE.

A mapping is specified as either:

<target\_column\_name>: <source\_column\_name>

or

<target\_column\_name>: '<expression>'

Where expression is any expression that you would specify in the SELECT list of a query, such as a constant value, a column reference, an operator invocation, a function call, and so on.

OPTIONS - Optional.

Specifies operations to run prior to the unload operation.

APPEND - Optional.

If set to true, hawqunload will add the new extracted rows at the end of the existing file. If the file doesn't already exist, the file is created.

If HEADER = true, the header is added only if the target file doesn't already exist

If set to false, a new target file is created, even if it already exists.

If not set, the behavior is like APPEND=False

REUSE\_TABLES - Optional.

If set to true, hawqunload will not drop the external table objects it creates. These objects will be reused for future unload operations that use the same unload specifications. This improves performance of trickle unloads (ongoing small unloads from the same source table).

SQL - Optional.

Defines SQL commands to run before and/or after the load operation. You can specify multiple BEFORE and/or AFTER commands. List commands in the order of desired execution.

BEFORE - Optional.

An SQL command to run before the load operation starts. Enclose commands in quotes.

AFTER - Optional.

An SQL command to run after the load operation completes. Enclose commands in quotes.

# Control File Schema Element Descriptions (Windows specific)

WINDOWS\_DISK - Optional.

By default, the output files will be placed into the C disk.

If the output files must be stored on a different disk, use this option to specify the windows disk.

For example: windows\_disk: Z

# Notes

If your database object names were created using a double-quoted identifier (delimited identifier), you must specify the delimited name within single quotes in the hawqunload control file.

For example, if you create a table as follows:

CREATE TABLE "MyTable" ("MyColumn" text);

Your YAML-formatted hawqunload control file would refer to the above table and column names as follows:

- COLUMNS:

- '"MyColumn"': text

INPUT:

- TABLE: public.'"MyTable"'

# Log File Format

Log files output by hawqunload have the following format:

<timestamp>|<level>|<message>

Where <timestamp> takes the form: YYYY-MM-DD HH:MM:SS,

<level> is one of DEBUG, LOG, INFO, ERROR,

and <message> is a normal text message.

Some INFO messages that may be of interest in the log files are (where # corresponds to the actual number of seconds, units of data, or failed rows):

INFO|running time: #.## seconds

INFO|rows Extracted = #

INFO|hawqunload succeeded

INFO|hawqunload failed

# Examples

Run an extract job as defined in my\_unload.yml:

hawqunload -f my\_unload.yml

Example extract control file:

---

VERSION: 1.0.0.1

DATABASE: ops

USER: gpadmin

HOST: mdw-1

PORT: 5432

HAWQUNLOAD:

INPUT:

- TABLE: payables.expenses

OUTPUT:

- TARGET:

PORT: 8081

FILE:

- /var/unload/data/payables\_expenses.dat

- COLUMNS:

- name: text

- amount: float4

- category: text

- desc: text

- date: date

- FORMAT: text

- DELIMITER: '|'

SQL:

- BEFORE: "INSERT INTO audit VALUES('start', current\_timestamp)"

- AFTER: "INSERT INTO audit VALUES('end', current\_timestamp)"