

Greenplum Database 4.3.4 Release Notes

Rev: A02

Updated: March, 2015

Welcome to Pivotal Greenplum Database 4.3.4

Greenplum Database is a massively parallel processing (MPP) database server that supports next generation data warehousing and large-scale analytics processing. By automatically partitioning data and running parallel queries, it allows a cluster of servers to operate as a single database supercomputer performing tens or hundreds times faster than a traditional database. It supports SQL, MapReduce parallel processing, and data volumes ranging from hundreds of gigabytes, to hundreds of terabytes.

Note: This document contains pertinent release information about Greenplum Database 4.3.4. For previous versions of the release notes for Greenplum Database, go to [Pivotal Documentation](#) or EMC [Support Zone](#).

About Greenplum Database 4.3.4

Greenplum Database 4.3.3 is a maintenance release that introduces a number of significant new features, as well as performance and stability enhancements. Please refer to the following sections for more information about this release.

- [Product Enhancements](#)
- [Changed Features](#)
- [Supported Platforms](#)
- [Resolved Issues in Greenplum Database 4.3.4](#)
- [Known Issues in Greenplum Database 4.3.4](#)
- [Upgrading to Greenplum Database 4.3.4](#)
- [Greenplum Database Tools Compatibility](#)
- [Greenplum Database Extensions Compatibility](#)
- [Hadoop Distribution Compatibility](#)
- [Greenplum Database 4.3.4 Documentation](#)

Warning: The Pivotal Query Optimizer feature is a beta-level feature that is not supported and that is not permitted to be used in general availability releases. Running Greenplum Database with the Pivotal Query Optimizer enabled is not supported. For information about the Pivotal Query Optimizer in Greenplum Database, contact gpdb@pivotal.io.

Product Enhancements

Greenplum Database 4.3.4 includes enhancements in these areas:

- *Management of Query Execution*
- *Database Table Storage Options*
- *Default Database Table Distribution Policy*
- *Enhancement for Encrypting Data*
- *Control of Client Connections to Greenplum Database*
- *Greenplum Database Utilities*
- *External Table Support for MapR*
- *Estimating the Time to Migrate from 4.2.x to 4.3.x*

Management of Query Execution

Greenplum Database 4.3.4 includes these enhancements to manage query execution:

- With the SQL script `gp_session_state.sql`, you can create the `session_level_memory_consumption` view that provides information about the current memory utilization for sessions that are running queries on Greenplum Database. The view contains session information and information such as the database that the session is connected to, the query that the session is currently running, and memory consumed by the session processes. The script is installed in the directory `$GPHOME/share/postgresql/contrib`.

To create the view, run the script once for each database. For example, to install the view in the database `testdb`, use this command:

```
$ psql -d testdb -f $GPHOME/share/postgresql/contrib/gp_session_state.sql
```

- The Greenplum Database server configuration parameter `runaway_detector_activation_percent` controls the percentage of Greenplum Database vmem memory that is utilized that triggers the termination of queries. If the percentage of vmem memory that is utilized for a Greenplum Database segment exceeds the specified value, Greenplum Database terminates queries based on memory usage, starting with the query consuming the largest amount of memory. Queries are terminated until the percentage of utilized vmem is below the specified percentage.

For information about the `session_level_memory_consumption` view, see "Monitoring a Greenplum System" in the *Greenplum Database Administrator Guide*. For information about the parameter, see `runaway_detector_activation_percent`.

Database Table Storage Options

Greenplum Database 4.3.4 includes these enhancements to database table storage options:

- To simplify the creation of database tables, you can specify the default values for some table storage options with the Greenplum Database server configuration parameter `gp_default_storage_options`.
- The `CHECKSUM` storage option can be specified in the `CREATE TABLE WITH` clause. This option is valid only for append-optimized tables (`APPENDONLY=TRUE`). The value `TRUE` is the default and enables CRC checksum validation for append-optimized tables. The checksum is calculated during block creation and is stored on disk. Checksum validation is performed during block reads. If the checksum calculated during the read does not match the stored checksum, the transaction is aborted. If you set the value to `FALSE` to disable checksum validation, checking the table data for on-disk corruption will not be performed.

For information about the parameter, see *gp_default_storage_options*. For information about the CHECKSUM storage option, see the CREATE TABLE command in the *Greenplum Database Reference Guide*.

Default Database Table Distribution Policy

Greenplum Database 4.3.4 supports the server configuration parameter `gp_create_table_random_default_distribution` that controls table creation when a Greenplum Database table is created with a CREATE TABLE or CREATE TABLE AS command that does not contain a DISTRIBUTED BY clause.

If the value of the parameter is OFF, (the default), and the table creation command does not contain a DISTRIBUTED BY clause, Greenplum Database chooses the distribution key based on the table creation command.

If the value of the parameter is ON, the table is created with a random distribution if the command does not specify PRIMARY KEY or UNIQUE columns.

For information about the parameter, see *gp_create_table_random_default_distribution*. For information about the table distribution policy and the DISTRIBUTED BY clause, see the CREATE TABLE command in the *Greenplum Database Reference Guide*.

Enhancement for Encrypting Data

With the Greenplum Database pgcrypto extension, you can use pgcrypto functions to store columns of data in encrypted form. When the Greenplum Database pgcrypto package version 1.2 is installed, you can enable pgcrypto support for *Federal Information Processing Standard* (FIPS) 140-2. The Greenplum Database server configuration parameter `pgcrypto.fips` controls the pgcrypto support for FIPS 140-2.

For information about the parameter, see "Server Configuration Parameters" in the *Greenplum Database Reference Guide*. For information about FIPS, see <http://www.nist.gov/itl/fips.cfm>.

For information about the parameter, see *pgcrypto.fips*.

Control of Client Connections to Greenplum Database

The Greenplum Database 4.3.4 server configuration parameter `gp_connection_send_timeout` controls the timeout value for sending data to unresponsive Greenplum Database user clients during query processing. When the timeout is reached, the query is cancelled.

For information about the parameter, see *gp_connection_send_timeout*.

Greenplum Database Utilities

- The Greenplum Database utility `gpstop` includes these enhancements:
 - The time required to stop Greenplum Database has been reduced. The `gpstop` utility stops the Greenplum Database segments as a parallel operation instead of a sequential operation.
 - When stopping Greenplum Database, the `gpstop` utility displays the progress as a percentage complete.
- The Greenplum Database utility `gpcrondump` supports backing up a set of tables based on database schema names. To include tables based on schema names, list the database schema names with `-s` option or specify a file containing a list of schema names with the `--schema-file` option. To exclude tables based on schema names list the schemas with the `-S` option or specify a file containing a list of schema names with the option `--exclude-schema-file`.
- The Greenplum Database utility `gpdbrstore` supports the `--truncate` option to table data before restoring the table data.

For information about the Greenplum Database utilities, see the *Greenplum Database Utility Guide*.

External Table Support for MapR

Greenplum Database 4.3.4 supports MapR with Greenplum Database external tables that use the `gp_hdfs` protocol to access HDFS data.

For information about MapR support, see [Hadoop Distribution Compatibility](#). For information about external tables, see "Loading and Unloading Data" in the *Greenplum Database Administrator Guide*.

gpsupport Utility for Troubleshooting and Log Collection

You can use the `gpsupport` utility to collect Greenplum Database information when troubleshooting and diagnosing issues with Greenplum Database installations. The utility works on 4.2.x and 4.3.x Greenplum Database systems that are running on the Linux operating system.

For information about the `gpsupport` utility, see the documentation for the [gpsupport utility](#).

Estimating the Time to Migrate from 4.2.x to 4.3.x

The migration process from Greenplum Database 4.2.x to 4.3.x converts append-only tables that are in Greenplum Database to append-optimized tables. For a database that contains a large number of append-only tables, the conversion to append-optimized tables might take a considerable amount of time. Pivotal supplies a user-defined function that can help estimate the time required to migrate from Greenplum Database 4.2.x to 4.3.x.

Append-optimized tables are introduced in Greenplum Database 4.3.0. For information about append-optimized tables, see the release notes for Greenplum Database 4.3.0.

For information about the user-defined function, see the documentation for the UDF `estimate_42_to_43_migrate_time`. There is a link to the document and UDF on the [Greenplum Database documentation web site](#).

Changed Features

- [Changed Features](#)
- [New Server Configuration Parameters](#)

Changed Features

Greenplum Database 4.3.4, includes these changes:

- The Greenplum Database `CREATE TABLE` command has been enhanced. The `WITH` clause supports the `CHECKSUM` option. See [Database Table Storage Options](#).
- Greenplum Database external tables support MapR. See [External Table Support for MapR](#).
- Greenplum Database `pgcrypto` package has been updated to version 1.2. This version supports FIPS 140-2. See [Enhancement for Encrypting Data](#).
- Greenplum Database PostGIS package version has been updated to version 2.0.1. Enhanced with better error handling for functions that are not supported by the Greenplum Database PostGIS package.
- Greenplum Database PL/Java package version has been updated to version 1.2. This version fixes some PL/Java issues.
- Greenplum Database PL/R package has been updated to version 2.1. This version updates the R version to 8.3.0.15.
- To enhance the performance of the Greenplum Command Center database `gpperfmon`, the distribution key of the `log_alert_*` tables has been changed to the `logtime` column from the `logsegment` column.

The first time you start Greenplum Database after an upgrade, the distribution key is changed and the table data is redistributed. For information about the `gpperfmon` database migration, see *Upgrading from 4.3.x to 4.3.4*.

New Server Configuration Parameters

These server configuration parameters have been introduced in Greenplum Database 4.3.4.

- `gp_connection_send_timeout`
- `gp_create_table_random_default_distribution`
- `gp_default_storage_options`
- `pgcrypto.fips`
- `runaway_detector_activation_percent`

Greenplum Database server configuration parameter information is in "Server Configuration Parameters" of the *Greenplum Database Reference Guide*.

`gp_connection_send_timeout`

Timeout for sending data to unresponsive Greenplum Database user clients during query processing. A value of 0 disables the timeout, Greenplum Database waits indefinitely for a client. When the timeout is reached, the query is cancelled with this message:

```
Could not send data to client: Connection timed out.
```

Value Range	Default	Set Classifications
number of seconds	3600 (1 hour)	master system reload

`gp_create_table_random_default_distribution`

Controls table creation when a Greenplum Database table is created with a `CREATE TABLE` or `CREATE TABLE AS` command that does not contain a `DISTRIBUTED BY` clause.

If the value of the parameter is `OFF`, (the default), and the table creation command does not contain a `DISTRIBUTED BY` clause, Greenplum Database chooses the table distribution key based on the command. If the `LIKE` or `INHERITS` clause is specified in table creation command, the created table uses the same distribution key as the source or parent table.

If the value of the parameter is set to `ON`, Greenplum Database follows these rules to create a table when the `DISTRIBUTED BY` clause is not specified:

- If `PRIMARY KEY` or `UNIQUE` columns are not specified, the distribution of the table is random (`DISTRIBUTED RANDOMLY`). Table distribution is random even if the table creation command contains the `LIKE` or `INHERITS` clause.
- If `PRIMARY KEY` or `UNIQUE` columns are specified, a `DISTRIBUTED BY` clause must also be specified. If a `DISTRIBUTED BY` clause is not specified as part of the table creation command, the command fails.

Value Range	Default	Set Classifications
boolean	off	master system reload

gp_default_storage_options

Set the default values for the following table storage options when creating a table with the `CREATE TABLE` command.

- APPENDONLY
- BLOCKSIZE
- CHECKSUM
- COMPRESSTYPE
- COMPRESSLEVEL
- ORIENTATION

Specify multiple storage option values as a comma separated list.

You can set the storage options with this parameter instead of specifying the table storage options in the `WITH` clause of the `CREATE TABLE` command. The table storage options that are specified with the `CREATE TABLE` command override the values specified by this parameter.

Not all combinations of storage option values are valid. If the specified storage options are not valid, an error is returned. See the `CREATE TABLE` command for information about table storage options.

The defaults can be set for a database and user. If the server configuration parameter is set at different levels, this the order of precedence, from highest to lowest, of the table storage values when a user logs into a database and creates a table:

1. The values specified in a `CREATE TABLE` command with the `WITH` clause or `ENCODING` clause
2. The value of `gp_default_storage_options` that set for the user with the `ALTER ROLE...SET` command
3. The value of `gp_default_storage_options` that is set for the database with the `ALTER DATABASE...SET` command
4. The value of `gp_default_storage_options` that is set for the Greenplum Database system with the `gpconfig` utility

The parameter value is not cumulative. For example, if the parameter specifies the `APPENDONLY` and `COMPRESSTYPE` options for a database and a user logs in and sets the parameter to specify the value for the `ORIENTATION` option, the `APPENDONLY`, and `COMPRESSTYPE` values set at the database level are ignored.

This example `ALTER DATABASE` command sets the default `ORIENTATION` and `COMPRESSTYPE` table storage options for the database `mystest`.

```
ALTER DATABASE mytest SET gp_default_storage_options = 'orientation=column,
  compressstype=rle_type'
```

To create an append-optimized table in the `mystest` database with column-oriented table and RLE compression. The user needs to specify only `APPENDONLY=TRUE` in the `WITH` clause.

This example `gpconfig` utility command sets the default storage option for a Greenplum Database system. If you set the defaults for multiple table storage options, the value must be enclosed in single quotes and then in double quotes.

```
gpconfig -c 'gp_default_storage_options' -v "'appendonly=true,
orientation=column'"
```

This example `gpconfig` utility command shows the value of the parameter. The parameter value must be consistent across the Greenplum Database master and all segments.

```
gpconfig -s 'gp_default_storage_options'
```

Value Range	Default	Set Classifications ¹
APPENDONLY= TRUE FALSE	APPENDONLY=FALSE	master
BLOCKSIZE= integer between 8192 and 2097152	BLOCKSIZE=32768	session
CHECKSUM= TRUE FALSE	CHECKSUM=TRUE	reload
COMPRESSTYPE= ZLIB QUICKLZ RLE_TYPE NONE	COMPRESSTYPE=none	
COMPRESSLEVEL= integer between 0 and 9	COMPRESSLEVEL=0	
ORIENTATION= ROW COLUMN	ORIENTATION=ROW	

Note: ¹The set classification when the parameter is set at the system level with the `gpconfig` utility.

pgcrypto.fips

Enables support for *Federal Information Processing Standard (FIPS) 140-2*. For information about FIPS, see <http://www.nist.gov/itl/fips.cfm>

To enable FIPS 140-2 support for Greenplum Database, the following are required.

- The Greenplum Database `pgcrypto` package version 1.2 or later must be installed.
- If the value of `pgcrypto.fips` is set to `on`, the value of the parameter `custom_variable_classes` must contain `pgcrypto`.

When FIPS 140-2 support is enabled, these `pgcrypto` changes occur:

- FIPS mode is initialized in the OpenSSL library
- The functions `digest()` and `hmac()` allow only the SHA encryption algorithm (MD5 is not allowed)
- The functions for `crypt` and `gen_salt` algorithms are disabled
- PGP encryption and decryption functions support only AES and 3DES encryption algorithms (other algorithms such as blowfish are not allowed)
- RAW encryption and decryption functions support only AES and 3DES (other algorithms such as blowfish are not allowed)

These `gpconfig` commands set the parameters to enable FIPS 140-2 support.

```
$ gpconfig -c custom_variable_classes -v pgcrypto --masteronly
$ gpconfig -c pgcrypto.fips -v on --masteronly
```

The value of the `custom_variable_classes` parameter is a comma separated list of classes. For more than one class, the list is inclosed in single quotes. To check the value of the parameter use `gpconfig` with the `-s` option to show the current value.

```
$ gpconfig -s custom_variable_classes
```

If the parameter is already set with custom classes, you can add `pgcrypto`. For example, if the value of `custom_variable_classes` is `plr`, this command adds `pgcrypto`.

```
$ gpconfig -c custom_variable_classes -v \'plr,pgcrypto\' --masteronly
--skipvalidation
```

In the command, use a backslash (\) to escape the single quotes.

Value Range	Default	Set Classifications
Boolean	off	master system restart

runaway_detector_activation_percent

Sets the percentage of Greenplum Database vmem memory that triggers the termination of queries. If the percentage of vmem memory that is utilized for a Greenplum Database segment exceeds the specified value, Greenplum Database terminates queries based on memory usage, starting with the query consuming the largest amount of memory. Queries are terminated until the percentage of utilized vmem is below the specified percentage.

Specify the maximum vmem value for active Greenplum Database segment instances with the server configuration parameter `gp_vmem_protect_limit`.

For example, if vmem memory is set to 10GB, and the value of `runaway_detector_activation_percent` is 90 (90%), Greenplum Database starts terminating queries when the utilized vmem memory exceeds 9 GB.

A value of 0 disables the automatic termination of queries based on percentage of vmem that is utilized.

Value Range	Default	Set Classifications
percentage (integer)	90	master system reload

Downloading Greenplum Database

The location for downloading Greenplum Database software and documentation has changed.

- Greenplum Database 4.3.x software is available from [Pivotal Network](#).
- Current release Greenplum Database documentation is available from the [Pivotal Documentation](#) site.

Previous release versions of Greenplum Database documentation, as well as other Greenplum Database documents, are available from [Support Zone](#)

Supported Platforms

Greenplum Database 4.3.4 runs on the following platforms:

- Red Hat Enterprise Linux 64-bit 5.x and 6.x
- SuSE Linux Enterprise Server 64-bit 10 SP4, 11 SP1, 11 SP2
- Oracle Unbreakable Linux 64-bit 5.5
- CentOS 64-bit 5.x, and 6.x

Note: Starting with Greenplum Database 4.3.0.0, Solaris is no longer a supported operating system. Please send any questions or comments about the changes to supported platforms to gpdb@pivotal.io.

Greenplum Database 4.3.3 supports Data Domain Boost on Red Hat Enterprise Linux.

This table lists the versions of Data Domain Boost SDK and DDOS supported by Greenplum Database 4.3.x.

Table 1: Data Domain Boost Compatibility

Greenplum Database	Data Domain Boost	DDOS
4.3.4.0	3.0.0.3	5.5.0.7
4.3.3.0	2.6.2.0	5.2, 5.3, and 5.4
4.3.2.0	2.6.2.0	5.2, 5.3, and 5.4
4.3.1.0	2.6.2.0	5.2, 5.3, and 5.4
4.3.0.0	2.4.2.2	5.0.1.0, 5.1, and 5.2

Greenplum Database support on DCA:

- Greenplum Database 4.3.x, all versions, is supported on DCA V2, and requires DCA software version 2.1.0.0 or greater due to known DCA software issues in older DCA software versions.
- Greenplum Database 4.3.x, all versions, is supported on DCA V1, and requires DCA software version 1.2.2.2 or greater due to known DCA software issues in older DCA software versions.

Note: In the next major release of Greenplum Database, connecting to IBM Cognos software with an ODBC driver will not be supported. Greenplum Database supports connecting to IBM Cognos software with a JDBC driver.

Pivotal recommends that user migrate to a version of IBM Cognos software that support connecting Greenplum Database with an JDBC driver.

Resolved Issues in Greenplum Database 4.3.4

The table below lists issues that are now resolved in Greenplum Database 4.3.3.

For issues resolved in prior 4.3 releases, refer to the corresponding release notes available from *Pivotal Network*.

Table 2: Resolved Issues in 4.3.4

Issue Number	Category	Description
24913	Storage Access Methods	In some cases, after a <code>VACUUM</code> operation was performed on an append-optimized table that contains an index, SQL queries that were run against the table returned incorrect data.
24778	Security	Greenplum Database software has been updated to use OpenSSL 0.9.8zc in response to the OpenSSL Security Advisory [15 Oct 2014]. For information about the advisory, see http://www.openssl.org/news/secadv_20141015.txt
24770	Dispatch	In some cases, <code>NULL</code> pointer handling was not performed correctly during query execution. This caused a panic on Greenplum Database master.
24694	Upgrade / Downgrade	The attribute name of a visimap metadata table for append-optimized tables was incorrectly named during the upgrade to 4.3.x from Greenplum Database 4.2.x. This issue has been resolved.
24693		For the visimap metadata table for append-optimized tables, the value of <code>attstorage</code> was set incorrectly in the <code>pg_attribute</code> table during the upgrade to 4.3.x from Greenplum Database 4.2.x. This issue has been resolved.
24692	Upgrade / Downgrade	A unnecessary record for the internal built-in function <code>pg_get_stat_walsenders</code> was added to the <code>pg_depend</code> table during the upgrade to 4.3.x from Greenplum Database 4.2.x. This issue has been resolved.
24684	Locking, Signals, Processes	In some cases, Greenplum Database did not clean up processes correctly during a commit transaction. This caused crash recovery issues after a Greenplum Database restart of both a primary segment and its mirror segment.
24683	Backup and Restore	There were performance issues when backing up specific tables from Greenplum Database using the Greenplum Database utility <code>gpcrondump</code> with the <code>--table-file</code> , <code>-t</code> , or <code>--exclude-table-file</code> , <code>-T</code> options. The back up performance has been improved.
24679	Locking, Signals, Processes	In some cases, Greenplum Database PL/Java did not exit properly and caused crash recovery issues after a Greenplum Database restart of both a primary segment and its mirror segment. See Changed Features.
24677	Backup and Restore	In some cases after a successful back up operation, error messages about lock files were incorrectly displayed. This issue has been resolved.

Issue Number	Category	Description
24667	Build, Test	When creating a temporary table with the <code>ON COMMIT DELETE ROWS</code> clause in a heavy workload environment, the prepared transaction that created the temporary table failed in some cases.
24662	Upgrade/ Downgrade	The privilege of the internal system view <code>pg_stat_walsenders</code> was set to <code>NULL</code> during the upgrade to 4.3.x from Greenplum Database 4.2.x. The upgrade process now sets the privilege correctly.
24630	Management Scripts: <code>gpsd</code>	The Greenplum Database support utility <code>gpsd</code> has been enhanced to be more memory efficient in order to prevent failures when used on a database that contains large schemas.
24606	Storage access method	In some cases, queries that were run against the <code>gp_toolkit</code> view <code>gp_size_of_table_and_indexes_disk</code> returned an error when the queries are run concurrently with DDL statements.
24515	Replication: Segment Mirroring	In some cases under a heavy workload, logging onto the Greenplum Database segment host as a UNIX user was not possible. This was caused by a Greenplum Database <code>filerrep</code> process that was incorrectly sending signals to the user after the process failed to create a sub-process.
24383	GPHDFS	Greenplum Database external tables did not support using the <code>gphdfs</code> protocol and MapR to access HDFS data.
24264	Catalog and Metadata	The commands <code>REINDEX TABLE table_name</code> and <code>REINDEX INDEX index_name</code> did not re-index child partition indexes of a partitioned table.
24216	Management Scripts: <code>gpstart/ gpstop</code>	In some cases, the Greenplum Database utility <code>gpstop</code> failed to shutdown a database.
24052	Query Planner	In some cases, when executing a query that joined tables and the tables have extremely inaccurate statistics, Greenplum Database would return an error that reported being out of memory. This issue has been resolved.
24003	Dispatch, Interconnect	In some cases, executing an SQL query caused a memory leak.
23802	Query Execution	Greenplum Database did not manage temporary workfiles (spill files) properly. In some cases, this caused a query that required workfiles to fail with a message that stated that a Greenplum Database segment had reached the maximum configured workfile usage limit.
23751	Monitoring: <code>gpperfmon</code> server	In some cases, a memory leak caused the <code>gpmon</code> process to consume a large amount of memory and CPU resources.

Issue Number	Category	Description
23395	Transaction Management	If a Greenplum Database segment failed during two phase transaction processing, the transaction remained in a uncompleted state and was cleaned up only during a Greenplum Database restart. In many cases, this caused high disk consumption by the Greenplum Database <code>xlog</code> process.
23130	Resource Management	Some queries were terminated when they were run with a specific <code>statement_mem</code> value due to a rounding error.
23031	Backup and Restore	When restoring a specific table that was not in the default schema (public) with the <code>gpdbrestore</code> utility from a back up, the table was restored in the public schema.
22526	OS Abstraction	In some cases, Greenplum Database would wait indefinitely for a response from a client application. This issue has been resolved. See the server configuration parameter <code>gp_connection_send_timeout</code> .
19612	Replication: Segment Mirroring	In some cases, Greenplum Database did not properly handle the transition when segment mirror becomes unavailable. As a result, Greenplum Database becomes unavailable.
18509	Functions and Languages	In some cases, Greenplum Database did not handle data of type <code>date</code> properly and caused a segmentation fault.

Known Issues in Greenplum Database 4.3.4

This section lists the known issues in Greenplum Database 4.3.4. A workaround is provided where applicable.

For known issues discovered in previous 4.3.x releases, see the release notes at [Pivotal Network](#). For known issues discovered in other previous releases, including patch releases to Greenplum Database 4.2.x, 4.1 or 4.0.x, see the corresponding release notes, available from EMC [Support Zone](#):

Table 3: All Known Issues in 4.3.3

Issue	Category	Description
25004	Upgrade / Downgrade	In Greenplum Database 4.3.3 and later, the <code>GRANT</code> and <code>REVOKE</code> commands support the <code>TRUNCATE</code> privilege on a table. The upgrade process to a 4.3.3 or later release does not grant the <code>TRUNCATE</code> permission to database tables that were created before the upgrade. Workaround: Explicitly grant the <code>TRUNCATE</code> privilege to database tables that were created before the upgrade. <pre>GRANT TRUNCATE on table to table_owner</pre>

Issue	Category	Description
24944	DDL and Utility Statements	<p>The <code>set_config()</code> function changes the server configuration parameter only on the Greenplum Database master. The parameter value is not changed on the Greenplum Database segment instances.</p> <p>Workaround: Set the value of the server configuration parameter with the <code>SET</code> command.</p>
24588	Management Scripts: gpconfig	<p>The Greenplum Database <code>gpconfig</code> utility does not display the correct information for the server configuration parameter <code>gp_enable_gpperfmon</code>. The parameter displays the state of the Greenplum Command Center data collection agents (<code>gpperfmon</code>).</p> <p>Workaround: The SQL command <code>SHOW</code> displays the correct <code>gp_enable_gpperfmon</code> value.</p>
22798	Management Scripts: expansion, Management Scripts: master mirroring	<p>If it is not possible to use SSH to connect from the Greenplum Database master host to 'localhost' a failure occurs when running the Greenplum Database <code>gpactivatestandby</code> or <code>gpexpand</code> utility because of an SSH failure.</p> <p>Workaround: Enable SSH to 'localhost' on the master host to work around this issue.</p>
23646	DML	<p>Running an <code>UPDATE</code> command after a <code>DROP COLUMN</code> and <code>ADD PARTITION</code> command on a partitioned table causes a Greenplum Database segment instance failure.</p>
24031	gphdfs	<p>If a readable external table is created with <code>FORMAT 'CSV'</code> and uses the gphdfs protocol, reading a record fails if the record spans multiple lines and the record is stored in multiple HDFS blocks.</p> <p>Workaround: Remove line separators from within the record so that the record does not span multiple lines.</p>
23924	Backup and Restore	<p>In some cases, performing some operations on an append-optimized table and then performing a full backup with the <code>gpcrondump</code> utility to a Data Domain system with DDBoost fails with this error:</p> <pre>ERROR: relation "file_name" does not exist</pre>
23824	Authentication	<p>In some cases, LDAP client utility tools cannot be used after running the source command:</p> <pre>source \$GPHOME/greenplum_path.sh</pre> <p>because the LDAP libraries included with Greenplum Database are not compatible with the LDAP client utility tools that are installed with operating system.</p> <p>Workaround: The LDAP tools can be used without running the source command in the environment.</p>
23525	Query Planner	<p>Some SQL queries that contain sub-selects fail with this error.</p> <pre>ERROR: Failed to locate datatype for paramid 0</pre>

Issue	Category	Description
22792	Build	Greenplum Database is not certified on Red Hat Enterprise Linux 5.10.
22215	Build	Greenplum Database is not certified with these connectivity drivers: Data Direct v 7.022; PowerExchange for Greenplum 9.5.1 32-bit Microstrategy ODBC for Greenplum Wire Protocol 6.10.01.80 Open source ODBC 9.01.0100 and JDBC 9.1.902 Type 4 SAS/ACCESS 9.3 driver provided with SAS software2
23366	Resource Management	In Greenplum Database 4.2.7.0 and later, the priority of some running queries, cannot be dynamically adjusted with the <code>gp_adjust_priority()</code> function. The attempt to execute this request might silently fail. The return value of the <code>gp_adjust_priority()</code> call indicates success or failure. If 1 is returned, the request was not successfully executed. If a number greater than 1 is returned, the request was successful. If the request fails, the priority of all running queries are unchanged, they remain as they were before the <code>gp_adjust_priority()</code> call.
23492	Backup and Restore,	A backup from a Greenplum Database 4.3.x system that is created with a Greenplum Database back up utility, for example <code>gpcrondump</code> , cannot be restored to a Greenplum Database 4.2.x system with the <code>psql</code> utility or the corresponding restore utility, for example <code>gpdbrestore</code> .
23521	Client Access Methods and Tools	Hadoop YARN based on Hadoop 2.2 or later does not work with Greenplum Database. Workaround: For Hadoop distributions based on Hadoop 2.2 or later that are supported by Greenplum Database, the classpath environment variable and other directory paths defined in <code>\$GPHOME/lib/hadoop/hadoop_env.sh</code> must be to be modified so that the paths point to the appropriate JAR files.
20453	Query Planner	For SQL queries of either of the following forms: <pre>SELECT columns FROM table WHERE table.column NOT IN subquery; SELECT columns FROM table WHERE table.column = ALL subquery;</pre> tuples that satisfy both of the following conditions are not included in the result set: <ul style="list-style-type: none"><code>table.column</code> is NULL.<code>subquery</code> returns the empty result.
21724	Query Planner	Greenplum Database executes an SQL query in two stages if a scalar subquery is involved. The output of the first stage plan is fed into the second stage plan as a external parameter. If the first stage plan generates zero tuples and directly contributes to the output of the second stage plan, incorrect results might be returned.

Issue	Category	Description
21838	Backup and Restore	<p>When restoring sets of tables with the Greenplum Database utility <code>gpdbrstore</code>, the table schemas must be defined in the database. If a table's schema is not defined in the database, the table is not restored. When performing a full restore, the database schemas are created when the tables are restored.</p> <p>Workaround: Before restoring a set of tables, create the schemas for the tables in the database.</p>
21129	DDL and Utility Statements	SSL is only supported on the master host. It is not supported on segment hosts.
20822	Backup and Restore	<p>Special characters such as <code>!</code>, <code>\$</code>, <code>#</code>, and <code>@</code> cannot be used in the password for the Data Domain Boost user when specifying the Data Domain Boost credentials with the <code>gpcrondump</code> options <code>--ddboost-host</code> and <code>--ddboost-user</code>.</p>
18247	DDL and Utility Statements	<p><code>TRUNCATE</code> command does not remove rows from a sub-table of a partitioned table. If you specify a sub-table of a partitioned table with the <code>TRUNCATE</code> command, the command does not remove rows from the sub-table and its child tables.</p> <p>Workaround: Use the <code>ALTER TABLE</code> command with the <code>TRUNCATE PARTITION</code> clause to remove rows from the sub-table and its child tables.</p>
19705	Loaders: gpload	<p><code>gpload</code> fails on Windows XP with Python 2.6.</p> <p>Workaround: Install Python 2.5 on the system where <code>gpload</code> is installed.</p>
19493 19464 19426	Backup and Restore	<p>The <code>gpcrondump</code> and <code>gpdbrstore</code> utilities do not handle errors returned by DD Boost or Data Domain correctly.</p> <p>These are two examples:</p> <ul style="list-style-type: none"> • If invalid Data Domain credentials are specified when setting the Data Domain Boost credentials with the <code>gpcrondump</code> utility, the error message does not indicate that invalid credentials were specified. • Restoring a Greenplum database from a Data Domain system with <code>gpdbrstore</code> and the <code>--ddboost</code> option indicates success even though segment failures occurred during the restore. <p>Workaround: The errors are logged in the master and segment server backup or restore status and report files. Scan the status and report files to check for error messages.</p>

Issue	Category	Description
15692 17192	Backup and Restore	<p>Greenplum Database's implementation of RSA lock box for Data Domain Boost changes backup and restore requirements for customers running SUSE.</p> <p>The current implementation of the RSA lock box for Data Domain Boost login credential encryption only supports customers running on Red Hat Enterprise Linux.</p> <p>Workaround: If you run Greenplum Database on SUSE, use NFS as your backup solution. See the <i>Greenplum Database Administrator Guide</i> for information on setting up a NFS backup.</p>
18850	Backup and Restore	<p>Data Domain Boost credentials cannot be set up in some environments due to the absence of certain libraries (for example, <code>libstdc++</code>) expected to reside on the platform.</p> <p>Workaround: Install the missing libraries manually on the system.</p>
18851	Backup and Restore	<p>When performing a data-only restore of a particular table, it is possible to introduce data into Greenplum Database that contradicts the distribution policy of that table. In such cases, subsequent queries may return unexpected and incorrect results. To avoid this scenario, we suggest you carefully consider the table schema when performing a restore.</p>
18713	Catalog and Metadata	<p>Drop language <code>plpgsql</code> cascade results in a loss of <code>gp_toolkit</code> functionality.</p> <p>Workaround: Reinstall <code>gp_toolkit</code>.</p>
18710	Management Scripts Suite	<p>Greenplum Management utilities cannot parse IPv6 IP addresses.</p> <p>Workaround: Always specify IPv6 hostnames rather than IP addresses</p>
18703	Loaders	<p>The <code>bytenum</code> field (byte offset in the load file where the error occurred) in the error log when using <code>gpfdist</code> with data in text format errors is not populated, making it difficult to find the location of an error in the source file.</p>
12468	Management Scripts Suite	<p><code>gpexpand --rollback</code> fails if an error occurs during expansion such that it leaves the database down</p> <p><code>gpstart</code> also fails as it detects that expansion is in progress and suggests to run <code>gpexpand --rollback</code> which will not work because the database is down.</p> <p>Workaround: Run <code>gpstart -m</code> to start the master and then run <code>rollback</code>.</p>
18785	Loaders	<p>Running <code>gpload</code> with the <code>--ssl</code> option and the relative path of the source file results in an error that states the source file is missing.</p> <p>Workaround: Provide the full path in the yaml file or add the loaded data file to the certificate folder.</p>

Issue	Category	Description
18414	Loaders	Unable to define external tables with fixed width format and empty line delimiter when file size is larger than <code>gpfdist</code> chunk (by default, 32K).
14640	Backup and Restore	<code>gpdbrstore</code> outputting incorrect non-zero error message. When performing single table restore, <code>gpdbrstore</code> gives warning messages about non-zero tables however prints out zero rows.
17285	Backup and Restore	NFS backup with <code>gpcrondump -c</code> can fail. In circumstances where you haven't backed up to a local disk before, backups to NFS using <code>gpcrondump</code> with the <code>-c</code> option can fail. On fresh systems where a backup has not been previously invoked there are no dump files to cleanup and the <code>-c</code> flag will have no effect. Workaround: Do not run <code>gpcrondump</code> with the <code>-c</code> option the first time a backup is invoked from a system.
17837	Upgrade/ Downgrade	Major version upgrades internally depend on the <code>gp_toolkit</code> system schema. The alteration or absence of this schema may cause upgrades to error out during preliminary checks. Workaround: To enable the upgrade process to proceed, you need to reinstall the <code>gp_toolkit</code> schema in all affected databases by applying the SQL file found here: <code>\$GPHOME/share/postgresql/gp_toolkit.sql</code> .
17513	Management Scripts Suite	Running more than one <code>gpfilespace</code> command concurrently with itself to move either temporary files (<code>--movetempfilespace</code>) or transaction files (<code>--movetransfilespace</code>) to a new filespace can in some circumstances cause OID inconsistencies. Workaround: Do not run more than one <code>gpfilespace</code> command concurrently with itself. If an OID inconsistency is introduced <code>gpfilespace --movetempfilespace</code> or <code>gpfilespace --movetransfilespace</code> can be used to revert to the default filespace.
17780	DDL/DML: Partitioning	ALTER TABLE ADD PARTITION inheritance issue When performing an ALTER TABLE ADD PARTITION operation, the resulting parts may not correctly inherit the storage properties of the parent table in cases such as adding a default partition or more complex subpartitioning. This issue can be avoided by explicitly dictating the storage properties during the ADD PARTITION invocation. For leaf partitions that are already afflicted, the issue can be rectified through use of EXCHANGE PARTITION.
17795	Management Scripts Suite	Under some circumstances, <code>gppkg</code> on SUSE is unable to correctly interpret error messages returned by rpm. On SUSE, <code>gppkg</code> is unable to operate correctly under circumstances that require a non-trivial interpretation of underlying rpm commands. This includes scenarios that result from overlapping packages, partial installs, and partial uninstalls.

Issue	Category	Description
17604	Security	<p>A Red Hat Enterprise Linux (RHEL) 6.x security configuration file limits the number of processes that can run on gpadmin.</p> <p>RHEL 6.x contains a security file (/etc/security/limits.d/90-nproc.conf) that limits available processes running on gpadmin to 1064.</p> <p>Workaround: Remove this file or increase the processes to 131072.</p>
17415	Installer	<p>When you run <code>gppkg -q -info<some gppkg></code>, the system shows the GPDB version as main build dev.</p>
17334	Management Scripts Suite	<p>You may see warning messages that interfere with the operation of management scripts when logging in.</p> <p>Greenplum recommends that you edit the <code>/etc/motd</code> file and add the warning message to it. This will send the messages to be redirected to stdout and not stderr. You must encode these warning messages in UTF-8 format.</p>
17221	Resource Management	<p>Resource queue deadlocks may be encountered if a cursor is associated with a query invoking a function within another function.</p>
17113	Management Scripts Suite	<p>Filespaces are inconsistent when the Greenplum database is down.</p> <p>Filespaces become inconsistent in case of a network failure. Greenplum recommends that processes such as moving a filespace be done in an environment with an uninterrupted power supply.</p>
17189	Loaders: gpfdist	<p><code>gpfdist</code> shows the error "Address already in use" after successfully binding to socket IPv6.</p> <p>Greenplum supports IPv4 and IPv6. However, <code>gpfdist</code> fails to bind to socket IPv4, and shows the message "Address already in use", but binds successfully to socket IPv6.</p>

Issue	Category	Description
16519	Backup and Restore	<p>Limited data restore functionality and/or restore performance issues can occur when restoring tables from a full database backup where the default backup directory was not used.</p> <p>In order to restore from backup files not located in the default directory you can use the <code>-R</code> option to point to another host and directory. This is not possible however, if you want to point to a different directory on the same host (NFS for example).</p> <p>Workaround: Define a symbolic link from the default dump directory to the directory used for backup, as shown in the following example:</p> <ol style="list-style-type: none"> 1. Perform a full Database Backup to a specific NFS directory: <pre data-bbox="683 617 1385 701"> \$ gpccrondump -x <db_name> -z -u /backup/ DCA-93 -a </pre> 2. Create a file listing the segment servers: <pre data-bbox="683 758 1175 911"> \$ vi /home/gpadmin/segments sdw1 sdw2 sdw3 ... </pre> 3. Remove the relevant date folder from <code>db_dumps</code> directories on the master and segments: <pre data-bbox="683 1024 1414 1192"> \$ rm -r /data/master/gpseg-1/db_ dumps/20120619 \$ gpssh -f segments 'rm -r /data1/primary/ gpseg*/db_dumps/20120619' \$ gpssh -f segments 'rm -r /data2/primary/ gpseg*/db_dumps/20120619' </pre> 4. Create a symbolic link between the master and segment directories and the directory to which you backed up in step 1. Only the master and <code>sdw1</code> was shown here, write a script for the remaining segments: <pre data-bbox="683 1371 1398 1843"> \$ ln -s /backup/DCA-93/db_dumps/20120619/ data/master/gpseg-1/db_dumps/20120619 \$ gpssh -h sdw1 'ln -s /backup/DCA-93/ db_dumps/20120619/data1/primary/gpseg0/db_ dumps/20120619' \$ gpssh -h sdw1 'ln -s /backup/DCA-93/ db_dumps/20120619/data1/primary/gpseg1/db_ dumps/20120619' \$ gpssh -h sdw1 'ln -s /backup/DCA-93/ db_dumps/20120619/data1/primary/gpseg2/db_ dumps/20120619' \$ gpssh -h sdw1 'ln -s /backup/DCA-93/ db_dumps/20120619/data2/primary/gpseg3/db_ dumps/20120619' \$ gpssh -h sdw1 'ln -s /backup/DCA-93/ db_dumps/20120619/data2/primary/gpseg4/db_ dumps/20120619' </pre>

Issue	Category	Description
		<pre>\$ gpssh -h sdw1 'ln -s /backup/DCA-93/db_dumps/20120619/data2/primary/gpseg5/db_dumps/20120619'</pre> <p>5. Restore from backup files:</p> <pre>\$ gpdbrestore -t 20120619061835 -T <schema.table> -a</pre> <p>6. Remove the symbolic links:</p> <pre>\$ rrm -r /data/master/gpseg-1/db_dumps/20120619 \$ gpssh -f segments 'rm -r /data1/primary/gpseg*/db_dumps/20120619' \$ gpssh -f segments 'rm -r /data2/primary/gpseg*/db_dumps/20120619'</pre>
16064	Backup and Restore	<p>Restoring a compressed dump with the <code>--ddboost</code> option displays incorrect dump parameter information.</p> <p>When using <code>gpdbrestore --ddboost</code> to restore a compressed dump, the restore parameters incorrectly show “Restore compressed dump = Off”. This error occurs even if <code>gpdbrestore</code> passes the <code>--gp-c</code> option to use <code>gunzip</code> for in-line de-compression.</p>
15899	Backup and Restore	<p>When running <code>gpdbrestore</code> with the <code>list (-L)</code> option, external tables do not appear; this has no functional impact on the restore job.</p>

Upgrading to Greenplum Database 4.3.4

The upgrade path supported for this release is Greenplum Database 4.2.x.x to Greenplum Database 4.3.4. The minimum recommended upgrade path for this release is from Greenplum Database version 4.2.x.x. If you have an earlier major version of the database, you must first upgrade to version 4.2.x.x.

Prerequisites

Before starting the upgrade process, Pivotal recommends the following:

- Verify the health of the Greenplum Database host hardware, and that you verify that the hosts meet the requirements for running Greenplum Database. The Greenplum Database `gpcheckperf` utility can assist you in confirming the host requirements.
- Run the `gpcheckcat` utility to check for Greenplum Database catalog inconsistencies. The utility is in `$GPHOME/bin/lib`. Pivotal recommends that Greenplum Database be in restricted mode when you run `gpcheckcat` utility. See the *Greenplum Database Utility Guide* for information about the `gpcheckcat` utility.

If `gpcheckcat` reports catalog inconsistencies, you can run `gpcheckcat` with the `-g` option to generate SQL scripts to fix the inconsistencies.

After you run the SQL scripts, run `gpcheckcat` again. You might need to repeat the process of running `gpcheckcat` and creating SQL scripts to ensure that there are no inconsistencies. Pivotal recommends that the SQL scripts generated by `gpcheckcat` be run on a quiescent system. The utility might report false alerts if there is activity on the system.

Important: If the `gpcheckcat` utility reports errors, but does not generate a SQL script to fix the errors, contact Pivotal support. Information for contacting Pivotal Support is at <https://support.pivotal.io>.

For detailed upgrade procedures and information, see the following sections:

- [Upgrading from 4.3.x to 4.3.4](#)
- [Upgrading from 4.2.x.x to 4.3.4](#)
- [For Users Running Greenplum Database 4.1.x.x](#)
- [For Users Running Greenplum Database 4.0.x.x](#)
- [For Users Running Greenplum Database 3.3.x.x](#)
- [Migrating a Greenplum Database That Contains Append-Only Tables](#)

If you are utilizing Data Domain Boost, you have to re-enter your DD Boost credentials after upgrading from Greenplum Database 4.2.x.x to 4.3 as follows:

```
gpcrondump --ddboost-host ddboost_hostname --ddboost-user ddboost_user
--ddboost-backupdir backup_directory
```

Note: If you do not reenter your login credentials after an upgrade, your backup will never start because the Greenplum Database cannot connect to the Data Domain system. You will receive an error advising you to check your login credentials.

Upgrading from 4.3.x to 4.3.4

An upgrade from 4.3.x to 4.3.4 involves stopping Greenplum Database, updating the Greenplum Database software binaries, and restarting Greenplum Database.

Note: If you are upgrading from Greenplum Database between 4.3.0 and 4.3.2, run the `fix_ao_upgrade.py` utility to check Greenplum Database for the upgrade issue and fix the upgrade issue (See step 9). The utility is in this Greenplum Database 4.3.4 directory: `$GPHOME/share/postgresql/upgrade`

For information about the utility, see [fix_ao_upgrade.py Utility](#). There is a link to the utility documentation on the [Greenplum Database documentation web site](#).

Note: If the Greenplum Command Center database `gpperfmon` is installed in your Greenplum Database system, the migration process changes the distribution key of the Greenplum Database `log_alert_*` tables to the `logtime` column. The redistribution of the table data might take some time the first time you start Greenplum Database after migration. The change occurs only the first time you start Greenplum Database after a migration.

1. Log in to your Greenplum Database master host as the Greenplum administrative user:

```
$ su - gpadmin
```

2. Uninstall the Greenplum Database gNet extension package if it is installed.

The gNet extension package contains the software for the gphdfs protocol. For Greenplum Database 4.3.1 and later releases, the extension is bundled with Greenplum Database. The files for gphdfs are installed in `$GPHOME/lib/hadoop`.

3. Perform a smart shutdown of your current Greenplum Database 4.3.x system (there can be no active connections to the database):

```
$ gpstop
```

4. Run the installer for 4.3.4 on the Greenplum Database master host. When prompted, choose an installation location in the same base directory as your current installation. For example:

```
/usr/local/greenplum-db-4.3.4.0
```

5. Edit the environment of the Greenplum Database superuser (`gpadmin`) and make sure you are sourcing the `greenplum_path.sh` file for the new installation. For example change the following line in `.bashrc` or your chosen profile file:

```
source /usr/local/greenplum-db-4.3.0.0/greenplum_path.sh
```

to:

```
source /usr/local/greenplum-db-4.3.4.0/greenplum_path.sh
```

Or if you are sourcing a symbolic link (`/usr/local/greenplum-db`) in your profile files, update the link to point to the newly installed version. For example:

```
$ rm /usr/local/greenplum-db
$ ln -s /usr/local/greenplum-db-4.3.4.0 /usr/local/greenplum-db
```

6. Source the environment file you just edited. For example:

```
$ source ~/.bashrc
```

7. Run the `gpsegininstall` utility to install the 4.3.4 binaries on all the segment hosts specified in the `hostfile`. For example:

```
$ gpsegininstall -f hostfile
```

8. After all segment hosts have been upgraded, you can log in as the `gpadmin` user and restart your Greenplum Database system:

```
$ su - gpadmin
$ gpstart
```

9. If you are upgrading a version of Greenplum Database between 4.3.0 and 4.3.2, check your Greenplum Database for inconsistencies due to an incorrect conversion of 4.2.x append-only tables to 4.3.x append-optimized tables.

Important: The Greenplum Database system must be started but should not be running any SQL commands while the utility is running.

- a. Run the `fix_ao_upgrade.py` utility with the option `--report`. The following is an example.

```
$ $GPHOME/share/postgresql/upgrade/fix_ao_upgrade.py --host=mdw --port=5432 --report
```

- b. If the utility displays a list of inconsistencies, fix them by running the `fix_ao_upgrade.py` utility without the `--report` option.

```
$ $GPHOME/share/postgresql/upgrade/fix_ao_upgrade.py --host=mdw --port=5432
```

- c. (optional) Run the `fix_ao_upgrade.py` utility with the option `--report` again. No inconsistencies should be reported.

10. If you are utilizing Data Domain Boost, you have to re-enter your DD Boost credentials after upgrading from Greenplum Database 4.3.x to 4.3.3 as follows:

```
gpcrondump --ddboost-host ddboost_hostname --ddboost-user ddboost_user
--ddboost-backupdir backup_directory
```

Note: If you do not reenter your login credentials after an upgrade, your backup will never start because the Greenplum Database cannot connect to the Data Domain system. You will receive an error advising you to check your login credentials.

fix_ao_upgrade.py Utility

The `fix_ao_upgrade.py` utility checks Greenplum Database for an upgrade issue that is caused when upgrading Greenplum Database 4.2.x to a version of Greenplum Database between 4.3.0 and 4.3.2.

The upgrade process incorrectly converted append-only tables that were in the 4.2.x database to append-optimized tables during an upgrade from Greenplum Database 4.2.x to a Greenplum Database 4.3.x release prior to 4.3.2.1. The incorrect conversion causes append-optimized table inconsistencies in the upgraded Greenplum Database system.

Syntax

```
fix_ao_upgrade.py {-h master_host | --host=master_host}
                 {-p master_port | --port=master_port}
                 [-u user | --user=user ]
                 [--report] [-v | --verbose] [--help]
```

Options

-r | --report

Report inconsistencies without making any changes.

-h master_host | --host=master_host

Greenplum Database master hostname or IP address.

-p master_port | --port=master_port

Greenplum Database master port.

-u user | --user=user

User name to connect to Greenplum Database. The user must be a Greenplum Database superuser. Default is `gpadmin`.

v | --verbose

Verbose output that includes table names.

--help

Show the help message and exit.

If you specify the optional `--report` option, the utility displays a report of inconsistencies in the Greenplum Database system. No changes to Greenplum Database system are made. If you specify the `--verbose` option with `--report`, the table names that are affected by the inconsistencies are included in the output.

Upgrading from 4.2.x.x to 4.3.4

This section describes how you can upgrade from Greenplum Database 4.2.x.x or later to Greenplum Database 4.3.4. For users running versions prior to 4.2.x.x of Greenplum Database, see the following:

- *For Users Running Greenplum Database 4.1.x.x*
- *For Users Running Greenplum Database 4.0.x.x*
- *For Users Running Greenplum Database 3.3.x.x*

Planning Your Upgrade

Before you begin your upgrade, make sure the master and all segments (data directories and filespace) have at least 2GB of free space.

Prior to upgrading your database, Pivotal recommends that you run a pre-upgrade check to verify your database is healthy.

You can perform a pre-upgrade check by executing the `gpmigrator (_mirror)` utility with the `--check-only` option.

For example:

```
source $new_gphome/greenplum_path.sh;
gpmigrator_mirror --check-only $old_gphome $new_gphome
```

Note: Performing a pre-upgrade check of your database with the `gpmigrator (_mirror)` utility should be done during a database maintenance period. When the utility checks the database catalog, users cannot access the database.

Migrating a Greenplum Database That Contains Append-Only Tables

The migration process converts append-only tables that are in a Greenplum Database to append-optimized tables. For a database that contains a large number of append-only tables, the conversion to append-optimized tables might take a considerable amount of time. Pivotal supplies a user-defined function that can help estimate the time required to migrate from Greenplum Database 4.2.x to 4.3.x. For information about the user-defined function, [estimate_42_to_43_migrate_time.pdf](#).

Append-optimized tables are introduced in Greenplum Database 4.3.0. For information about append-optimized tables, see the release notes for Greenplum Database 4.3.0.

Upgrade Procedure

This section divides the upgrade into the following phases: pre-upgrade preparation, software installation, upgrade execution, and post-upgrade tasks.

We have also provided you with an [Upgrade Checklist](#) that summarizes this procedure.

Important: Carefully evaluate each section and perform all required and conditional steps. Failing to perform any of these steps can result in an aborted upgrade, placing your system in an unusable or even unrecoverable state.

Pre-Upgrade Preparation (on your 4.2.x system)

Perform these steps on your current 4.2.x Greenplum Database system. This procedure is performed from your Greenplum master host and should be executed by the Greenplum superuser (`gpadmin`).

1. Log in to the Greenplum Database master as the `gpadmin` user:

```
$ su - gpadmin
```

2. *(optional)* Vacuum all databases prior to upgrade. For example:

```
$ vacuumdb database_name
```

3. *(optional)* Clean out old server log files from your master and segment data directories. For example, to remove log files from 2011 from your segment hosts:

```
$ gpssh -f seg_host_file -e 'rm /gpdata/*/gp*/pg_log/gpdb-2011-*.csv'
```

Running Vacuum and cleaning out old logs files is not required, but it will reduce the size of Greenplum Database files to be backed up and migrated.

4. Run `gpstate` to check for failed segments.

```
$ gpstate
```

5. If you have failed segments, you must recover them using `gprecoverseg` before you can upgrade.

```
$ gprecoverseg
```


Note: It might be necessary to restart the database if the preferred role does not match the current role; for example, if a primary segment is acting as a mirror segment or a mirror segment is acting as a primary segment.

6. Copy or preserve any additional folders or files (such as backup folders) that you have added in the Greenplum data directories or `$GPHOME` directory. Only files or folders strictly related to Greenplum Database operations are preserved by the migration utility.

Install the Greenplum Database 4.3 Software Binaries

1. Download or copy the installer file to the Greenplum Database master host.
2. Unzip the installer file. For example:

```
# unzip greenplum-db-4.3.4.0-PLATFORM.zip
```

3. Launch the installer using `bash`. For example:

```
# /bin/bash greenplum-db-4.3.4.0-PLATFORM.bin
```

4. The installer will prompt you to accept the Greenplum Database license agreement. Type `yes` to accept the license agreement.
5. The installer will prompt you to provide an installation path. Press `ENTER` to accept the default install path (for example: `/usr/local/greenplum-db-4.3.4.0`), or enter an absolute path to an install location. You must have write permissions to the location you specify.
6. The installer installs the Greenplum Database software and creates a `greenplum-db` symbolic link one directory level above your version-specific Greenplum installation directory. The symbolic link is used to facilitate patch maintenance and upgrades between versions. The installed location is referred to as `$GPHOME`.
7. Source the path file from your new 4.3.3 installation. For example:

```
$ source /usr/local/greenplum-db-4.3.3.0/greenplum_path.sh
```

8. Run the `gpsegininstall` utility to install the 4.3.3 binaries on all the segment hosts specified in the `hostfile`. For example:

```
$ gpsegininstall -f hostfile
```

Upgrade Execution

During upgrade, all client connections to the master will be locked out. Inform all database users of the upgrade and lockout time frame. From this point onward, users should not be allowed on the system until the upgrade is complete.

1. Source the path file from your old 4.2.x.x installation. For example:

```
$ source /usr/local/greenplum-db-4.2.6.3/greenplum_path.sh
```

2. (*optional but strongly recommended*) Back up all databases in your Greenplum Database system using `gpcrondump`. See the *Greenplum Database Administrator Guide* for more information on how to do backups using `gpcrondump`. Make sure to secure your backup files in a location outside of your Greenplum data directories.
3. If your system has a standby master host configured, remove the standby master from your system configuration. For example:

```
$ gpinitstandby -r
```

4. Perform a clean shutdown of your current Greenplum Database 4.2.x.x system. For example:

```
$ gpstop
```

5. Source the path file from your new 4.3.4.0 installation. For example:

```
$ source /usr/home/greenplum-db-4.3.4.0/greenplum_path.sh
```

6. Update the Greenplum Database environment so it is referencing your new 4.3.3 installation.

- a. For example, update the `greenplum-db` symbolic link on the master and standby master to point to the new 4.3.4 installation directory. For example (as root):

```
# rm -rf /usr/local/greenplum-db
# ln -s /usr/local/greenplum-db-4.3.4.0 /usr/local/greenplum-db
# chown -R gpadmin /usr/local/greenplum-db
```

- b. Using `gpssh`, also update the `greenplum-db` symbolic link on all of your segment hosts. For example (as root):

```
# gpssh -f segment_hosts_file
=> rm -rf /usr/local/greenplum-db
=> ln -s /usr/local/greenplum-db-4.3.4.0 /usr/local/greenplum-db
=> chown -R gpadmin /usr/local/greenplum-db
=> exit
```

7. (*optional but recommended*) Prior to running the migration, perform a pre-upgrade check to verify that your database is healthy by executing the 4.3.4 version of the migration utility with the `--check-only` option. This example uses the `gpmigrator_mirror` utility:

```
$ gpmigrator_mirror --check-only
  /usr/local/greenplum-db-4.2.6.3
  /usr/local/greenplum-db-4.3.4.0
```

8. As `gpadmin`, run the 4.3.4 version of the migration utility specifying your old and new `GPHOME` locations. If your system has mirrors, use `gpmigrator_mirror`. If your system does not have mirrors, use `gpmigrator`. For example on a system with mirrors:

```
$ su - gpadmin
$ gpmigrator_mirror /usr/local/greenplum-db-4.2.6.3
  /usr/local/greenplum-db-4.3.4.0
```

Note: If the migration does not complete successfully, contact Customer Support (see [Troubleshooting a Failed Upgrade](#)).

9. The migration can take a while to complete. After the migration utility has completed successfully, the Greenplum Database 4.3.4 system will be running and accepting connections.

Note: After the migration utility has completed, the resynchronization of the mirror segments with the primary segments continues. Even though the system is running, the mirrors are not active until the resynchronization is complete.

Post-Upgrade (on your 4.3.4 system)

1. If your system had a standby master host configured, reinitialize your standby master using `gpinitstandby`:

```
$ gpinitstandby -s standby_hostname
```

2. If your system uses external tables with `gpfdist`, stop all `gpfdist` processes on your ETL servers and reinstall `gpfdist` using the compatible Greenplum Database 4.3.4 Load Tools package. Application Packages are available at [Pivotal Network](#).
3. Rebuild any custom modules against your 4.3.4 installation (for example, any shared library files for user-defined functions in `$GPHOME/lib`).
4. Use the Greenplum Database `gppkg` utility to install Greenplum Database extensions. If you were previously using any Greenplum Database extensions such as `pgcrypto`, `PL/R`, `PL/Java`, `PL/Perl`, and

PostGIS, download the corresponding packages from *Pivotal Network*, and install using this utility. See the *Greenplum Database 4.3 Utility Guide* for usage details.

5. If you want to utilize the Greenplum Command Center management tool, install the latest Command Center Console and update your environment variable to point to the latest Command Center binaries (source the `gpperfmon_path.sh` file from your new installation).

Note: The Greenplum Command Center management tool replaces Greenplum Performance Monitor.

Command Center Console packages are available from *Pivotal Network*.

6. Inform all database users of the completed upgrade. Tell users to update their environment to source the Greenplum Database 4.3.4 installation (if necessary).

Upgrade Checklist

This checklist provides a quick overview of all the steps required for an upgrade from 4.2.x.x to 4.3.4. Detailed upgrade instructions are provided in the *Known Issues in Greenplum Database 4.3.4* section.

Pre-Upgrade Preparation (on your current system)
* 4.2.x.x system is up and available
<input type="checkbox"/> Log in to your master host as the <code>gpadmin</code> user (your Greenplum superuser).
<input type="checkbox"/> (Optional) Run <code>VACUUM</code> on all databases,
<input type="checkbox"/> (Optional) Remove old server log files from <code>pg_log</code> in your master and segment data directories.
<input type="checkbox"/> Check for and recover any failed segments (<code>gpstate</code> , <code>gprecoverseg</code>).
<input type="checkbox"/> Copy or preserve any additional folders or files (such as backup folders).
<input type="checkbox"/> Install the Greenplum Database 4.3 binaries on all Greenplum hosts.
<input type="checkbox"/> Inform all database users of the upgrade and lockout time frame.
Upgrade Execution
* The system will be locked down to all user activity during the upgrade process
<input type="checkbox"/> Backup your current databases.
<input type="checkbox"/> Remove the standby master (<code>gpinitstandby -r</code>).

<input type="checkbox"/> Do a clean shutdown of your current system (<code>gpstop</code>).
<input type="checkbox"/> Update your environment to source the new Greenplum Database 4.3.3 installation.
<input type="checkbox"/> Run the upgrade utility (<code>gpmigrator_mirror</code> if you have mirrors, <code>gpmigrator</code> if you do not).
<input type="checkbox"/> After the upgrade process finishes successfully, your 4.3.3 system will be up and running.
Post-Upgrade (on your 4.3 system)
* The 4.2.x.x system is up
<input type="checkbox"/> Reinitialize your standby master host (<code>gpinitstandby</code>).
<input type="checkbox"/> Upgrade <code>gpfdist</code> on all of your ETL hosts.
<input type="checkbox"/> Rebuild any custom modules against your 4.3.4 installation.
<input type="checkbox"/> Download and install any Greenplum Database extensions.
<input type="checkbox"/> (Optional) Install the latest Command Center Console and update your environment to point to the latest Command Center binaries.
<input type="checkbox"/> Inform all database users of the completed upgrade.

For Users Running Greenplum Database 4.1.x.x

Users on a release prior to 4.1.x.x cannot upgrade directly to 4.3.4.

1. Upgrade from your current release to 4.2.x.x (follow the upgrade instructions in the latest Greenplum Database 4.2.x.x release notes available at [Pivotal Documentation](#)).
2. Follow the upgrade instructions in these release notes for [Upgrading from 4.2.x.x to 4.3.4](#).

For Users Running Greenplum Database 4.0.x.x

Users on a release prior to 4.1.x.x cannot upgrade directly to 4.3.4.

1. Upgrade from your current release to 4.1.x.x (follow the upgrade instructions in the latest Greenplum Database 4.1.x.x release notes available on [Support Zone](#)).
2. Upgrade from the current release to 4.2.x.x (follow the upgrade instructions in the latest Greenplum Database 4.2.x.x release notes available at [Pivotal Documentation](#)).
3. Follow the upgrade instructions in these release notes for [Upgrading from 4.2.x.x to 4.3.4](#).

For Users Running Greenplum Database 3.3.x.x

Users on a release prior to 4.0.x.x cannot upgrade directly to 4.3.4.

1. Upgrade from your current release to the latest 4.0.x.x release (follow the upgrade instructions in the latest Greenplum Database 4.0.x.x release notes available on [Support Zone](#)).
2. Upgrade the 4.0.x.x release to the latest 4.1.x.x release (follow the upgrade instructions in the latest Greenplum Database 4.1.x.x release notes available on [Support Zone](#)).
3. Upgrade from the 4.1.1 release to the latest 4.2.x.x release (follow the upgrade instructions in the latest Greenplum Database 4.2.x.x release notes available at [Pivotal Documentation](#)).
4. Follow the upgrade instructions in these release notes for [Upgrading from 4.2.x.x to 4.3.4](#).

Troubleshooting a Failed Upgrade

If you experience issues during the migration process and have active entitlements for Greenplum Database that were purchased through Pivotal, contact Pivotal Support. Information for contacting Pivotal Support is at <https://support.pivotal.io>.

Be prepared to provide the following information:

- A completed [Upgrade Procedure](#).
- Log output from `gpmigrator` and `gpcheckcat` (located in `~/gpAdminLogs`)

Greenplum Database Tools Compatibility

Client Tools

Greenplum releases a number of client tool packages on various platforms that can be used to connect to Greenplum Database and the Greenplum Command Center management tool. The following table describes the compatibility of these packages with this Greenplum Database release.

Tool packages are available from [Pivotal Network](#).

Table 4: Greenplum Database Tools Compatibility

Client Package	Description of Contents	Client Version	Server Versions
Greenplum Clients	Greenplum Database Command-Line Interface (psql)	4.3	4.3
Greenplum Connectivity	Standard PostgreSQL Database Drivers (ODBC, JDBC) PostgreSQL Client C API (libpq)	4.3	4.3
Greenplum Loaders	Greenplum Database Parallel Data Loading Tools (gpfdist, gpload)	4.3	4.3
Greenplum Command Center	Greenplum Database management tool.	1.2.0.1	4.3

The Greenplum Database Client Tools, Load Tools, and Connectivity Tools are supported on the following platforms:

- AIX 5.3L (32-bit)
- AIX 5.3L and AIX 6.1 (64-bit)
- Apple OSX on Intel processors (32-bit)
- HP-UX 11i v3 (B.11.31) Intel Itanium (Client and Load Tools only)
- Red Hat Enterprise Linux i386 (RHEL 5)
- Red Hat Enterprise Linux x86_64 (RHEL 5 and RHEL 6)
- SUSE Linux Enterprise Server x86_64 (SLES 10 and SLES 11)
- Solaris 10 SPARC32
- Solaris 10 SPARC64
- Solaris 10 i386
- Solaris 10 x86_64
- Windows 7 (32-bit and 64-bit)
- Windows Server 2003 R2 (32-bit and 64-bit)
- Windows Server 2008 R2 (64-bit)
- Windows XP (32-bit and 64-bit)

Greenplum Database Extensions Compatibility

Greenplum Database delivers an agile, extensible platform for in-database analytics, leveraging the system's massively parallel architecture. Greenplum Database enables turn-key in-database analytics with Greenplum extensions.

You can download Greenplum extensions packages from *Pivotal Network* and install them using the Greenplum Packager Manager (`gppkg`). See the *Greenplum Database Utility Guide* for details.

Note that Greenplum Package Manager installation files for extension packages may release outside of standard Database release cycles.

The following table provides information about the compatibility of the Greenplum Database Extensions and their components with this Greenplum Database release.

Note: The PL/Python database extension is already included with the standard Greenplum Database distribution.

Table 5: Greenplum Database Extensions Compatibility

Greenplum Database Extension	Extension Components	
	Name	Version
PostGIS 2.0.1 for Greenplum Database 4.3.x.x	PostGIS	2.0.3
	Proj	4.8.0
	Geos	3.3.8
PostGIS 1.0 for Greenplum Database	PostGIS	1.4.2
	Proj	4.7.0
	Geos	3.2.2

Greenplum Database Extension	Extension Components	
	Name	Version
PL/Java 1.2 for Greenplum Database 4.3.x.x	PL/Java	Based on 1.4.0
	Java JDK	1.6.0_26 Update 31
PL/R 2.1 for Greenplum Database 4.3.x.x	PL/R	8.3.0.15
	R	3.1.0
PL/R 1.0 for Greenplum Database 4.3.x.x	PL/R	8.3.0.12
	R	2.13.0
PL/Perl 1.2 for Greenplum Database 4.3.x.x	PL/Perl	Based on PostgreSQL 9.1
	Perl	5.12.4 on RHEL 6.x 5.5.8 on RHEL 5.x, SUSE 10
PL/Perl 1.1 for Greenplum Database	PL/Perl	Based on PostgreSQL 9.1
	Perl	5.12.4 on RHEL 5.x, SUSE 10
PL/Perl 1.0 for Greenplum Database	PL/Perl	Based on PostgreSQL 9.1
	Perl	5.12.4 on RHEL 5.x, SUSE 10
Pgcrypto 1.2 for Greenplum Database 4.3.x.x	Pgcrypto	Based on PostgreSQL 8.3
MADlib 1.5 for Greenplum Database 4.3.x.x	MADlib	Based on MADlib version 1.8

Greenplum Database 4.3 supports these minimum Greenplum Database extensions package versions.

Table 6: Greenplum Database 4.3 Package Version

Greenplum Database Extension	Minimum Package Version
PostGIS	2.0.1
PL/Java	1.1
PL/Perl	1.2
PL/R	1.0
Pgcrypto	1.1
MADlib	1.5

Package File Naming Convention

For Greenplum Database 4.3, this is the package file naming format.

```
pkgname-ver_pvpkg-version_gpdbrel-OS-version-arch.gppkg
```

This example is the package name for a postGIS package.

```
postgis-ossv2.0.3_pv2.0_gpdb4.3-rhel5-x86_64.gppkg
```

pkgname-ver - The package name and optional version of the software that was used to create the package extension. If the package is based on open source software, the version has format *ossvversion*. The version is the version of the open source software that the package is based on. For the postGIS package, *ossv2.0.3* specifies that the package is based on postGIS version 2.0.3.

pvpkg-version - The package version. The version of the Greenplum Database package. For the postGIS package, *pv2.0.1* specifies that the Greenplum Database package version is 2.0.1.

gpdbrel-OS-version-arch - The compatible Greenplum Database release. For the postGIS package, *gpdb4.3-rhel5-x86_64* specifies that package is compatible with Greenplum Database 4.3 on Red Hat Enterprise Linux version 5.x, x86 64-bit architecture.

Hadoop Distribution Compatibility

This table lists the supported Hadoop distributions:

Table 7: Supported Hadoop Distributions

Hadoop Distribution	Version	gp_hadoop_target_version
Pivotal HD	Pivotal HD 2.0 Pivotal HD 1.0 ¹	gphd-2.0
Greenplum HD	Greenplum HD 1.2	gphd-1.2
	Greenplum HD 1.1	gphd-1.1 (default)
Cloudera	CDH 5.0, 5.1	cdh4.1
	CDH 4.1 ² - CDH 4.7	cdh3u2
Hortonworks Data Platform	HDP 2.1	hdp2
MapR ³	MapR 4.x	gpmr-1.2
	MapR 1.x, 2.x, 3.x	gpmr-1.0

Notes:

1. Pivotal HD 1.0 is a distribution of Hadoop 2.0
2. For CDH 4.1, only CDH4 with MRv1 is supported
3. MapR requires the MapR client

Greenplum Database 4.3.4 Documentation

For the latest Greenplum Database documentation go to *Pivotal Documentation*. Greenplum documentation is provided in PDF format.

Table 8: Greenplum Database Documentation

Title	Revision
<i>Greenplum Database 4.3.4 Release Notes</i>	A02
<i>Greenplum Database 4.3 Installation Guide</i>	A05
<i>Greenplum Database 4.3 Administrator Guide</i>	A05
<i>Greenplum Database 4.3 Reference Guide</i>	A06
<i>Greenplum Database 4.3 Utility Guide</i>	A06
<i>Greenplum Database 4.3 Client Tools for UNIX</i>	A03
<i>Greenplum Database 4.3 Client Tools for Windows</i>	A03
<i>Greenplum Database 4.3 Connectivity Tools for UNIX</i>	A03
<i>Greenplum Database 4.3 Connectivity Tools for Windows</i>	A03
<i>Greenplum Database 4.3 Load Tools for UNIX</i>	A04
<i>Greenplum Database 4.3 Load Tools for Windows</i>	A03
<i>Greenplum Command Center 1.2.2 Administrator Guide</i>	A02