

Greenplum Database 4.3.7.0 Release Notes

Rev: A02

Updated: January, 2016

Welcome to Pivotal Greenplum Database 4.3.7.0

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 4.3.7.0. For previous versions of the release notes for Greenplum Database, go to [Pivotal Documentation](#) or EMC [Support Zone](#). For information about Greenplum Database end of life, see [Greenplum Database end of life policy](#).

Important: Pivotal Global Support Services (GSS) does **not** provide support for open source versions of Greenplum Database. Only Pivotal Greenplum Database is supported by Pivotal GSS.

About Greenplum Database 4.3.7.0

Greenplum Database 4.3.7.0 is a maintenance release that resolves known issues, adds new features, and includes changes to some features. Please refer to the following sections for more information about this release.

- [Product Enhancements](#)
- [Changed Features](#)
- [Supported Platforms](#)
- [Resolved Issues in Greenplum Database 4.3.7.0](#)
- [Known Issues in Greenplum Database 4.3.7.0](#)
- [Upgrading to Greenplum Database 4.3.7.0](#)
- [Greenplum Database Tools Compatibility](#)
- [Greenplum Database Extensions Compatibility](#)
- [Hadoop Distribution Compatibility](#)
- [Greenplum Database 4.3.7.0 Documentation](#)

Product Enhancements

Greenplum Database 4.3.7.0 includes these enhancements.

- *Support for Red Hat Enterprise Linux 7.x and CentOS 7.x*
- *gpcrondump Utility Enhancements*
- *gpdbrestore Utility Enhancements*
- *gptransfer Utility Can Transfer Specific Partitions of a Partitioned Table*
- *PL/Java classpath Enhancement*
- *Performance Enhancement for Queries with Distinct Qualified Aggregates*

Support for Red Hat Enterprise Linux 7.x and CentOS 7.x

Greenplum Database 4.3.7.0 supports Red Hat Enterprise Linux 64-bit 7.x and CentOS 64-bit 7.x.

Important: For Red Hat Enterprise Linux 7.2 or CentOS 7.2, the default `systemd` setting `RemoveIPC=yes` removes IPC connections when non-system users logout. This causes the Greenplum Database utility `gpinitssystem` to fail with semaphore errors. Perform one of the following to avoid this issue.

- Create the `gpadmin` user as a system account. For the `useradd` command, the `-r` option creates the user as a system user and the `-m` option creates a home directory for the user.
- Disable `RemoveIPC`. Set this parameter in `/etc/systemd/logind.conf` on the Greenplum Database host systems.

```
RemoveIPC=no
```

The setting takes effect after restarting the `systemd-login` service or rebooting the system. To restart the service, run this command as the root user.

```
service systemd-logind restart
```

Note: Pivotal supplies separate PL/Perl extension packages for Red Hat Enterprise Linux 7.x, 6.x, and 5.x. Ensure you install the correct package for your operating system.

For information about installing Greenplum Database server software, see the *Greenplum Database Installation Guide*. For information about supported platforms, see *Supported Platforms*.

gpcrondump Utility Enhancements

These are the Greenplum Database 4.3.7.0 enhancements for the `gpcrondump` utility

- You can back up database statistics. You can specify the option `--dump-stats` to back up statistics when performing a backup operation. The data is written to an SQL file and can be restored manually or with the `gpdbrestore` utility. The statistics are written in the master data directory to `db_dumps/YYYYMMDD/prefix_string_gp_statistics_1_1_timestamp`. If this option is specified with other options that include or exclude tables or schemas, the utility dumps only the statistics for the tables that are backed up.
- After a backup operation completes, the utility checks the `gpcrondump` status file for SQL execution errors and displays a warning if an error is found. The default location of the backup status files are in the `db_dumps/date/` directory.

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

gpdbrestore Utility Enhancements

These Greenplum Database 4.3.7.0 enhancements for the `gpdbrestore` utility

- You can specify the database schema for tables that are being restored. Specify the schema with the new option `--change-schema=schema_name`. The `schema_name` must exist in the database. If the schema does not exist, the utility returns an error. System catalog schemas are not supported.

You must specify tables to restore with the `-T` and `--table-file` options. If a table that is being restored exists in `schema-name`, the utility returns a warning and attempts to append the data to the table from the backup.

- You can restore global database information with the option `-G`. Global database metadata is not associated with any particular schema or table. You can restore the metadata that was backed up with the `gpcrondump` option `-G`. The `gpdbrestore` option `-G` option has been enhanced to support the keywords `include` and `only`.
 - The keyword `include` restores global objects in addition to performing a restore operation. This is the default if no keyword is specified.
 - The keyword `only` restores only global objects. No other database objects or table data are restored.

The `gpcrondump` option `-m` restores metadata that is associated with specific schemas or tables.

- You can restore only database metadata information with the new option `-m`. The utility restores database metadata information such as schema and table definitions and information created by `SET` statements. This option does not restore database table data. Database information that is not associated with a specific schema or table is always restored. All table and schema metadata is restored unless options are specified that include or exclude tables or schemas. If table or schema filters are specified, the utility restores the schema and table metadata only for the schemas and tables that are specified to be restored.

The `gpcrondump` option `-G` restores metadata that is not associated with a specific schema or table.

- You can restore database statistics that were backed up with the `gpcrondump` utility option `--dump-stats`. You can choose to restore only database statistics or include restoring database statistics as part of a restore operation. The `--restore-stats` option supports the keywords `include` and `only`.
 - The keyword `include` restores the statistics that were backed up in addition to performing a restore operation. This is the default if no keyword is specified
 - The keyword `only` restores only the statistics that were backed up. No other database objects or table data are restored.

If this option is specified with options that include or exclude tables or schemas, the utility restores only the statistics for the tables that are specified to be restored. If statistics would be restored for a table that does not exist in the database, the utility displays a warning.

- After a restore operation completes, the utility checks the `gpdbrestore` status file for SQL execution errors and displays a warning if an error is found. The default location of the backup status files are in the `db_dumps/date/` directory.

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

gptransfer Utility Can Transfer Specific Partitions of a Partitioned Table

In Greenplum Database 4.3.7.0, the Greenplum Database utility `gptransfer` supports the option `-partition-transfer` to copy specific leaf child partitions of partitioned tables from a source database to a target database. The leaf child partitions are the lowest level partitions of a partitioned database. When copying a leaf child partition, the partitioned table must exist in the destination database and these characteristics must be the same for the partitioned table in the source and destination database.

- Number of table columns and the order of the column data types (the source and destination table names and table column names can be different)
- Partition level of the specified source and destination tables
- Partitioning criteria of the specified source and destination leaf child partitions and child partitions above them in the hierarchy (partition type and partition column)

If the child partition is not a leaf child partition, the utility returns an error.

For information about the utility, see the *Greenplum Database Utility Guide*. For information about the partitioned tables, see "Partitioning Large Tables" in the *Greenplum Database Administrator Guide*.

PL/Java classpath Enhancement

The `pljava_classpath` server configuration parameter has been enhanced in Greenplum Database 4.3.7.0 to support Java jar files anywhere on the file system.

Important: For Greenplum Database 4.3.7.0, setting the `pljava_classpath` parameter now requires superuser privilege. If the parameter is set in SQL code, the code fails when it is executed by a user without superuser privilege.

The classpath enhancement requires the Greenplum Database PL/Java extension package version `pv1.3`. For information about the PL/Java package extension change, see *PL/Java Extension Package Change*.

For information about the enhancement to `pljava_classpath`, see *Changed Parameter*.

Performance Enhancement for Queries with Distinct Qualified Aggregates

The Pivotal Query Optimizer performance has been improved for queries that contain distinct qualified aggregates (DQA) without a grouping column when the table is not distributed on the columns used by the DQA. In Greenplum Database 4.3.7.0, the Pivotal Query Optimizer considers a plan alternative that evaluates the aggregate functions in three stages (local, intermediate, and global aggregations).

For information about queries that contain aggregates, see the *Greenplum Database Administrator Guide*.

Changed Features

- *Changed Features in Greenplum Database 4.3.7.0*
- *Changed Parameter*

Changed Features in Greenplum Database 4.3.7.0

These feature changes have been introduced in Greenplum Database 4.3.7.0:

- *Enhanced Support for gphdfs Protocol*
- *Enhanced Pivotal Query Optimizer Plan Generation*
- *pgbouncer Utility Configuration Change*
- *psql Utility Displays Partition Key*
- *JDBC Drivers Updated in Connectivity Tools*
- *PL/Java Extension Package Change*
- *PL/Perl Extension Support for Red Hat 7.x*

Enhanced Support for gphdfs Protocol

With Greenplum Database external tables created with the `CREATE EXTERNAL TABLE` command, you can specify the `gphdfs` protocol to access external files on an Hadoop file system (HDFS) as if they are regular database tables. For Greenplum Database 4.3.7.0, the `gphdfs` protocol includes these enhancements:

- Support of Hive decimal data type in Parquet files generated by Hive
- Support for Unicode characters in Parquet and Avro files
- Support for Greenplum Database smallint data type with Parquet and Avro files
- Support for Cloudera CDH 5.5.x Hadoop Distributions

For information about supported Hadoop distributions, see *Hadoop Distribution Compatibility*. For information about external tables, see "Loading and Unloading Data" in the *Greenplum Database Administrator Guide*.

Enhanced Pivotal Query Optimizer Plan Generation

In Greenplum Database 4.3.7.0, the Pivotal Query Optimizer includes this enhancement:

- The performance was poor for queries that contain distinct qualified aggregates (DQA) without a grouping column when the table is not distributed on the columns used by the DQA when they were executed by the Pivotal Query Optimizer.

The Pivotal Query Optimizer now considers a plan alternative that evaluates the aggregate functions in three stages (local, intermediate, and global aggregations).

For information about the Pivotal Query Optimizer, see the *Greenplum Database Administrator Guide*.

pgbouncer Utility Configuration Change

In the `pgbouncer` configuration file, the default value has changed for the parameter `change_server_reset_query`. The parameter specifies the query sent to the server when a connection release occurs. For Greenplum Database 4.3.7.0, the value for the parameter is `RESET ALL; SET SESSION AUTHORIZATION DEFAULT;`. In previous releases, the default value is `DISCARD ALL`.

PgBouncer in session pooling mode resets the state of the database connection by issuing a query before returning the connection to the pool. The `server_reset_query` parameter specifies the query that PgBouncer issues to reset the connection.

For information about the `pgbouncer` parameters and values, see the *Greenplum Database Utility Guide*. For information about using `pgbouncer`, see the *Greenplum Database Administrator Guide*.

psql Utility Displays Partition Key

When you run the `psql` command `\d` or `\d+` on a partitioned table or a child partition table of partitioned table, the utility displays partition keys on the current partition level. The key is displayed after the `Distributed by` clause information. This is an example of partition information:

```
Partition by: (partition_key_1, ..., partition_key_n)
```

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

JDBC Drivers Updated in Connectivity Tools

The JDBC 4 drivers have been updated in the Greenplum Database Connectivity Tools. These are the JDBC 4 drivers in the Greenplum Database 4.3.7.0 Connectivity Tools:

- `postgresql-9.4-1207.jdbc4.jar`
- `postgresql-9.4-1207.jdbc41.jar`
- `postgresql-9.4-1207.jdbc42.jar`

The PostgreSQL JDBC drivers are installed by the client tools installer into `greenplum-connectivity-4.3.x.x/drivers/jdbc`. To use a JDBC 4 driver, you use the JAR file based on the Java version being used:

- For Java 1.6, use the JDBC4 driver `postgresql-9.4-1207.jdbc4.jar`.
- For Java 1.7, use the JDBC41 driver `postgresql-9.4-1207.jdbc41.jar`.
- For Java 1.8, use the JDBC42 driver `postgresql-9.4-1207.jdbc42.jar`.

For information about JDBC drivers, see the *Greenplum Database Connectivity Tools*.

PL/Java Extension Package Change

The Greenplum Database PL/Java extension package has been updated to support the *PL/Java classpath Enhancement*. The PL/Java Extension Package version has been incremented to `pv1.3` from `pv1.2`.

For information about Greenplum Database extension compatibility, see *Greenplum Database Extensions Compatibility*.

PL/Perl Extension Support for Red Hat 7.x

For Greenplum Database 4.3.7.0, Pivotal supplies a PL/Perl extension package for Red Hat Enterprise Linux 7.x.

For information Red Hat Enterprise Linux 7.x support, see *Product Enhancements*.

Changed Parameter

The `pljava_classpath` server configuration parameter has been enhanced in Greenplum Database 4.3.7.0 to support Java jar files anywhere on the host file system. You specify the full path to the jar file.

For jar files that are in `$GPHOME/lib/postgresql/java`, the jar files do not require a path. For example, this can be the `pljava_classpath` for `examples.jar`. The file `examples.jar` is installed in `$GPHOME/lib/postgresql/java` when you install the PL/Java extension package with the `gppkg` utility.

```
pljava_classpath=examples.jar
```

Important: Setting the `pljava_classpath` parameter now requires superuser privilege. If the parameter is set in SQL code, the code fails when it is executed by a user without superuser privilege.

For information about Greenplum Database server configuration parameters, see the *Greenplum Database Reference Guide*.

pljava_classpath

A colon (:) separated list of jar files or directories containing jar files needed for PL/Java functions. The full path to the jar file or directory must be specified, except the path can be omitted for jar files in the `$GPHOME/lib/postgresql/java` directory. The jar files must be installed in the same locations on all Greenplum hosts and readable by the `gpadmin` user.

The `pljava_classpath` parameter is used to assemble the PL/Java classpath at the beginning of each user session. Jar files added after a session has started are not available to that session.

If the full path to a jar file is specified in `pljava_classpath` it is added to the PL/Java classpath. When a directory is specified, any jar files the directory contains are added to the PL/Java classpath. The search does not descend into subdirectories of the specified directories. If the name of a jar file is included in `pljava_classpath` with no path, the jar file must be in the `$GPHOME/lib/postgresql/java` directory.

Note: Performance can be affected if there are many directories to search or a large number of jar files.

Setting the `pljava_classpath` parameter requires superuser privilege. Setting the classpath in SQL code will fail when the code is executed by a user without superuser privilege. The `pljava_classpath` parameter must have been set previously by a superuser or in the `postgresql.conf` file. Changing the classpath in the `postgresql.conf` file requires a reload (`gpstop -u`).

Value Range	Default	Set Classifications
string		master session reload superuser

Downloading Greenplum Database

These are the locations of the Greenplum Database software and documentation:

- 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 4.3.7.0 runs on the following platforms:

- Red Hat Enterprise Linux 64-bit 7.x
- Red Hat Enterprise Linux 64-bit 6.x
- Red Hat Enterprise Linux 64-bit 5.x
- SuSE Linux Enterprise Server 64-bit 10 SP4, 11 SP1, 11 SP2
- Oracle Unbreakable Linux 64-bit 5.5
- CentOS 64-bit 7.x
- CentOS 64-bit 6.x
- CentOS 64-bit 5.x

Greenplum Database 4.3.x supports these Java versions:

- 8.xxx
- 7.xxx
- 6.xxx

Greenplum Database 4.3.x 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
4.3.7.0	3.0.0.3	5.6 (all versions) 5.5.0.x 5.4 (all versions) 5.3 (all versions)

Greenplum Database	Data Domain Boost	DDOS
4.3.6.2 4.3.6.1 4.3.6.0	3.0.0.3	5.6 (all versions) 5.5.0.x 5.4 (all versions) 5.3 (all versions)
4.3.5.3 4.3.5.2 4.3.5.1 4.3.5.0	3.0.0.3	5.5.0.x 5.4 (all versions) 5.3 (all versions)
4.3.4.2 4.3.4.1 4.3.4.0	3.0.0.3	5.5.0.x 5.4 (all versions) 5.3 (all versions)
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

Note: In addition to the DDOS versions listed in the previous table, Greenplum Database 4.3.4.0 and later supports all minor patch releases (fourth digit releases) later than the certified version.

Greenplum Database support on DCA:

- Greenplum Database 4.3.x, all versions, is supported on DCA V3.
- 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.

Supported Platform Notes

The following notes describe platform support for Greenplum Database. Please send any questions or comments about the Greenplum Database platform support to gpdb@pivotal.io.

- The only file system supported for running Greenplum Database is the XFS file system. All other file systems are explicitly *not* supported by Pivotal.
- Greenplum Database is supported on all 1U and 2U commodity servers with local storage. Special purpose hardware that is not commodity *may* be supported at the full discretion of Pivotal Product Management based on the general similarity of the hardware to commodity servers.

- Greenplum Database is supported on network or shared storage if the shared storage is presented as a block device to the servers running Greenplum Database and the XFS file system is mounted on the block device. Network file systems are *not* supported. When using network or shared storage, Greenplum Database mirroring must be used in the same way as with local storage, and no modifications may be made to the mirroring scheme or the recovery scheme of the segments. Other features of the shared storage such as de-duplication and/or replication are not directly supported by Pivotal, but may be used with support of the storage vendor as long as they do not interfere with the expected operation of Greenplum Database at the discretion of Pivotal.
- Greenplum Database is supported when running on virtualized systems, as long as the storage is presented as block devices and the XFS file system is mounted for the storage of the segment directories.
- A minimum of 10-gigabit network is required for a system configuration to be supported by Pivotal.
- Greenplum Database is supported on Amazon Web Services (AWS) servers using either Amazon instance store (Amazon uses the volume names `ephemeral[0-20]`) or Amazon Elastic Block Store (Amazon EBS) storage. If using Amazon EBS storage the storage should be RAID of Amazon EBS volumes and mounted with the XFS file system for it to be a supported configuration.

Resolved Issues in Greenplum Database 4.3.7.0

The table below lists issues that are now resolved in Greenplum Database 4.3.7.0

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

Table 2: Resolved Issues in 4.3.7.0

Issue Number	Category	Resolved In	Description
26141	Management Scripts: gpssh	4.3.7.0	<p>In some cases, the Greenplum Database utility <code>gpssh</code> did not wait long enough for a connection to a Greenplum Database host and returned an error when attempting to connect to a host.</p> <p>This issue has been resolved. To minimize connection failures, the amount of time that <code>gpssh</code> waits for a connection has been increased.</p>
26123	DML	4.3.7.0	<p>In some situations, an SQL query that required data from multiple segments did not return results when executed. The same query returned results when executed at a later time.</p> <p>This issue has been resolved. During query execution, the communication between Greenplum Database segments has been improved.</p>
26122	Query Optimizer	4.3.7.0	<p>When the Pivotal Query Optimizer (PQO) is enabled (<code>optimizer=on</code>), PQO did not distribute table data for <code>ALTER TABLE</code> commands that contained a distribution clause.</p> <p>The issue has been resolved. The PQO falls back to the legacy planner for <code>ALTER TABLE</code> commands that contain a distribution clause to ensure table data is rebalanced across the Greenplum Database segments.</p>

Issue Number	Category	Resolved In	Description
26116	GPHDFS	4.3.7.0	<p>The <code>gphdfs</code> protocol returned an exception when writing a <code>NULL</code> value to a Parquet format file.</p> <p>This issue has been resolved. The <code>gphdfs</code> protocol writes <code>NULL</code> values to a Parquet file without errors.</p>
26104	Storage: Access Methods	4.3.7.0	<p>Greenplum Database memory increased while inserting a large amount data into an append-optimized table because Greenplum Database did not properly release memory.</p> <p>Greenplum Database memory management has been improved when inserting data into append-optimized tables.</p>
26091	Query Execution	4.3.7.0	<p>In some cases, the <code>mregr_*</code> functions returned incorrect results due to incorrect initialization.</p> <p>This issue has been resolved. Now, newly created matrix values are reset to zero before performing the calculation.</p>
26076	Query Optimizer	4.3.7.0	<p>When the Pivotal Query Optimizer (PQO) is enabled (<code>optimizer=on</code>), Greenplum Database generated a PANIC for some queries that contain the <code>FULL OUTER JOIN</code> clause on partitioned tables and predicates on partitioning columns.</p> <p>This issue has been resolved. The PQO falls back to the legacy planner for the specified queries.</p>
26075	Loaders: gpload	4.3.7.0	<p>The Greenplum Database Loaders for Red Hat Enterprise 5 and 6 did not contain the library <code>libyaml-0.so.1</code>.</p> <p>This issue has been resolved.</p>
26063	Query Optimizer	4.3.7.0	<p>The Pivotal Query Optimizer performance was poor for queries that contain distinct qualified aggregates (DQA) without a grouping column when the table is not distributed on the columns used by the DQA.</p> <p>Performance has been enhanced, see <i>Performance Enhancement for Queries with Distinct Qualified Aggregates</i>.</p>
26050	GPHDFS	4.3.7.0	<p>A Greenplum Database PANIC occurred when reading data from an Avro file that contains Unicode characters with the <code>gphdfs</code> protocol.</p> <p>See <i>Enhanced Support for gphdfs Protocol</i>.</p>
26028	Query Planner	4.3.7.0	<p>In some situations, a query that was running only on the Greenplum Database master host could not be terminated with the <code>pg_cancel_backend</code> or <code>pg_terminate_backend</code> functions.</p>

Issue Number	Category	Resolved In	Description
26023	Backup and Restore	4.3.7.0	In some cases, the Greenplum Database <code>gpdbrestore</code> utility incorrectly restored tables in the default schema instead of their expected schema. The incorrect restore was caused by the Greenplum Database <code>gpcrondump</code> utility incorrectly backing up the database schema information. This issue has been resolved.
26016	Storage: Transaction Management	4.3.7.0	Greenplum Database dropped all user connections due to a Greenplum Database PANIC when a user with a default tablespace defined attempted to log in to Greenplum Database and an error occurred during Greenplum Database tablespace assignment for the user.
26014	Backup and Restore	4.3.7.0	The Greenplum Database utility <code>gpdbrestore</code> returned an error and did not restore a table from a backup if the <code>--truncate</code> option was specified and the table to be restored did not exist in the target database. This issue has been resolved. The utility restores the table from the backup and returns a warning message.
26007 25972	Query Execution	4.3.7.0	For some SQL commands that contain grouping operations such as <code>CUBE</code> and <code>ROLLUP</code> and <code>GROUPING SETS</code> , Greenplum Database did not handle memory properly when executing the command and caused a Greenplum Database segmentation fault.
25988	Backup and Restore	4.3.7.0	In some cases, the Greenplum Database utility <code>gpdbrestore</code> did not return a warning or error when a restore operation did not successfully insert data. This issue has been resolved. See also <i>gpdbrestore Utility Enhancements</i> .
25904	Backup and Restore	4.3.7.0	The Greenplum Database utility <code>pg_dumpall</code> could not dump tablespaces. The <code>-f</code> option was used for both providing an output file name and for dumping tablespaces, and the option could only be used to provide an output file name. This issue has been resolved. Now the utility uses the <code>-F</code> option for dumping tablespaces. The <code>-f</code> option is used to provide an output file name.
25844	Query Planner	4.3.7.0	When a Greenplum Database system is running multiple segments, the Greenplum Database legacy planner estimate for the number of rows for the table <code>pg_partition_rule</code> did not account for multiple segments.

Issue Number	Category	Resolved In	Description
25584	Query Execution	4.3.7.0	In some situations, a running Greenplum Database query could not be terminated with the functions <code>pg_cancel_backend</code> or <code>pg_terminate_backend</code> . Now, <code>pg_terminate_backend</code> will cancel a running query.
25385	Client Access Methods and Tools	4.3.7.0	In some situations when Greenplum Database hung when sending data to a client application, the connection to the application could not be terminated with the <code>pg_terminate_backend</code> function. The function has been improved.
25264	Backup and Restore	4.3.7.0	The Greenplum Database utility <code>gpdbrestore</code> performance was poor when append-optimized tables were restored. The utility updated tuple count information for all append-optimized tables in the database, not just the tables that were restored. The performance has been improved. The utility updates only append-optimized tables that are restored.

Known Issues in Greenplum Database 4.3.7.0

This section lists the known issues in Greenplum Database 4.3.7.0. 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 4.3.7.0

Issue	Category	Description
25934 25936	Query Optimizer Query Planner	<p>For queries that compare data from columns of different character types, for example a join comparing a columns of data types <code>CHAR(n)</code> and <code>VARCHAR(m)</code>, the returned results might not be as expected depending the padding added to the data (space characters added after the last non-space character).</p> <p>For example, this comparison returns false.</p> <pre>select 'A '::char(2) ='A '::text ;</pre> <p>This comparison returns true.</p> <pre>select 'A '::char(2) ='A '::varchar(5) ;</pre> <p>Workaround: Pivotal recommends specifying character column types to be of data type <code>VARCHAR</code> or <code>TEXT</code> so that comparisons include padding added to the data.</p> <p>For information about how the character data types <code>CHAR</code>, <code>VARCHAR</code>, and <code>TEXT</code> handle padding added to the data see the <code>CREATE TABLE</code> command in the <i>Greenplum Database Reference Guide</i>.</p>
25924	Query Execution	<p>For Pivotal Query Optimizer, generating plans with bitmap index/scans has been disabled to avoid possible crashes in the query execution engine. Disabling bitmap index/scan might affect performance of certain queries executed by the Pivotal Query Optimizer.</p> <p>Note: For queries executed by the legacy query optimizer, generating plans with bitmap index/scans is enabled.</p>
25829	Dispatch	<p>If the name of a Greenplum Database contains a quote character, SQL commands that are run in the database return an error if the commands access Greenplum Database segments.</p>
25737	Catalog and Metadata	<p>Greenplum Database does not support the <code>FILTER</code> clause within aggregate expressions.</p>
25754	Management Scripts: expansion	<p>The Greenplum Database <code>gpexpand</code> utility fails to create an input file for system expansion if the Greenplum Database system define different TCP/IP port numbers on different hosts for Greenplum Database internal communication.</p> <p>Workaround: Create the input file manually.</p>
25833	Management Scripts: gpexpand	<p>The Greenplum Database utility <code>gpexpand</code> fails when expanding a Greenplum Database system and in the system a database table column name contains a tab character. The utility does not support database names, table names, or column names that contain a tab character.</p>

Issue	Category	Description
15835	DDL and Utility Statements	<p>For multi-level partitioned tables that have these characteristics:</p> <ul style="list-style-type: none"> The top level partition is partitioned by range. The lowest level partition (the leaf child partitions) are partitioned by list. <p>Splitting a subpartition with the <code>ALTER TABLE SPLIT PARTITION</code> command returns an error and rolls back the transaction.</p>
25147	Query Optimizer	When changing a table definition with the <code>ALTER TABLE</code> command, the <code>REORGANIZE</code> clause cannot be specified when the distribution policy of the table is being changed to random distribution (with the <code>DISTRIBUTED RANDOMLY</code> clause).
12019	Management Scripts: checkperf	<p>When the Greenplum Database <code>gpcheckperf</code> utility is run with the option <code>-f host_file</code> and the host that is running <code>gpcheckperf</code> is listed in <code>host_file</code>, processes that were started <code>gpcheckperf</code> might not be cleaned up after the utility completes.</p> <p>Workaround: Manually stop the processes that were started by <code>gpcheckperf</code>.</p>
24870	Query Optimizer	The Pivotal Query Optimizer might terminate all sessions if a query attempts to cast to a timestamp a date with year greater than 200,000.
23571	Query Optimizer	For queries that contain inequality conditions such as <code>!=</code> , <code><</code> and <code>></code> , the Pivotal Query Optimizer does not consider table indexes when generating a query plan. For those queries, indexes are not used and the query might run slower than expected.
21508	Query Optimizer	The Pivotal Query Optimizer does not support GiST indexes.
20241	Query Optimizer	For partitioned tables with indexes, the Pivotal Query Optimizer does not use indexes the if a child partition is queried directly.
20030	Query Optimizer	The Pivotal Query Optimizer does not support partition elimination when the query contains functions that are applied to the partition key.
20360	Query Execution	The Pivotal Query Optimizer does not enforce different access rights in different parts of a partition table. Pivotal recommends that you set the same access privileges for the partitioned table and all its parts (child tables).
20241	Query Optimizer	The Pivotal Query Optimizer does not consider indices when querying parts/child tables of partitioned tables directly.

Issue	Category	Description
25326	Interconnect	<p>Setting the Greenplum Database server configuration parameter <code>log_hostname</code> to <code>on</code> on Greenplum Database segment hosts causes an Interconnect Error that states that the listeneraddress name or service not known.</p> <p>The parameter should be set to <code>on</code> only on the Greenplum Database master.</p>
25280	Management Scripts: gpstart/gpstop	<p>The Greenplum Database utility <code>gpstop</code>, the utility returns an error if it is run and the system environment variable <code>LANG</code> is set, for example, <code>export LANG=ja_JP.UTF-8</code>.</p> <p>Workaround: Unset the environment variable <code>LANG</code> before running the <code>gpstop</code> utility. For example:</p> <pre>\$ unset LANG</pre>
25246	Management Scripts: gpconfig	<p>When you set the server configuration parameters <code>gp_email_to</code> and <code>gp_email_from</code> with the Greenplum Database utility <code>gpconfig</code>, the utility removes the single quotes from the values.</p> <pre>\$ gpconfig -c gp_email_to -v 'test@my-email.com'</pre> <p>The improperly set parameter causes Greenplum Database to fail when it is restarted.</p> <p>Workaround: Enclose the value for <code>gp_email_to</code> or <code>gp_email_from</code> with double quotes.</p> <pre>\$ gpconfig -c gp_email_to -v "'test@my-email.com'"</pre>
25168	Locking, Signals, Processes	<p>When the server configuration parameter <code>client_min_messages</code> is set to either <code>PANIC</code> or <code>FATAL</code> and a <code>PANIC</code> or <code>FATAL</code> level message is encountered, Greenplum Database hangs.</p> <p>The <code>client_min_messages</code> parameter should not be set a value higher than <code>ERROR</code>.</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>
24031	gphdfs	<p>If a readable external table is created with <code>FORMAT 'CSV'</code> and uses the <code>gphdfs</code> 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>

Issue	Category	Description
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>
23366	Resource Management	<p>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.</p>
23492	Backup and Restore,	<p>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>gpdrestore</code>.</p>
23521	Client Access Methods and Tools	<p>Hadoop YARN based on Hadoop 2.2 or later does not work with Greenplum Database.</p> <p>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.</p>
20453	Query Planner	<p>For SQL queries of either of the following forms:</p> <pre>SELECT columns FROM table WHERE table.column NOT IN subquery; SELECT columns FROM table WHERE table.column = ALL subquery;</pre> <p>tuples that satisfy both of the following conditions are not included in the result set:</p> <ul style="list-style-type: none"> • <code>table.column</code> is NULL. • <code>subquery</code> returns the empty result.

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).
17285	Backup and Restore	<p>NFS backup with <code>gpcrondump -c</code> can fail.</p> <p>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.</p> <p>Workaround: Do not run <code>gpcrondump</code> with the <code>-c</code> option the first time a backup is invoked from a system.</p>
17837	Upgrade/Downgrade	<p>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.</p> <p>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>.</p>
17513	Management Scripts Suite	<p>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.</p> <p>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.</p>
17780	DDL/DML: Partitioning	<p><code>ALTER TABLE ADD PARTITION</code> inheritance issue</p> <p>When performing an <code>ALTER TABLE ADD PARTITION</code> 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 <code>ADD PARTITION</code> invocation. For leaf partitions that are already afflicted, the issue can be rectified through use of <code>EXCHANGE PARTITION</code>.</p>
17795	Management Scripts Suite	<p>Under some circumstances, <code>gppkg</code> on SUSE is unable to correctly interpret error messages returned by <code>rpm</code>.</p> <p>On SUSE, <code>gppkg</code> is unable to operate correctly under circumstances that require a non-trivial interpretation of underlying <code>rpm</code> commands. This includes scenarios that result from overlapping packages, partial installs, and partial uninstalls.</p>

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>
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 /etc/motd 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 filesystem be done in an environment with an uninterrupted power supply.</p>
17189	Loaders: gpfdist	<p>gpfdist shows the error "Address already in use" after successfully binding to socket IPv6.</p> <p>Greenplum supports IPv4 and IPv6. However, gpfdist fails to bind to socket IPv4, and shows the message "Address already in use", but binds successfully to socket IPv6.</p>
16064	Backup and Restore	<p>Restoring a compressed dump with the --ddboost option displays incorrect dump parameter information.</p> <p>When using gpdbrestore --ddboost to restore a compressed dump, the restore parameters incorrectly show "Restore compressed dump = Off". This error occurs even if gpdbrestore passes the --gp-c option to use gunzip for in-line de-compression.</p>
15899	Backup and Restore	<p>When running gpdbrestore with the list (-L) option, external tables do not appear; this has no functional impact on the restore job.</p>

Upgrading to Greenplum Database 4.3.7.0

The upgrade path supported for this release is Greenplum Database 4.2.x.x to Greenplum Database 4.3.7.0. 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 performing the following checks.

- 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.
- If upgrading from Greenplum Database 4.2.x.x, Pivotal recommends running the `gpcheckcat` utility to check for Greenplum Database catalog inconsistencies.

Note: If you need to run the `gpcheckcat` utility, Pivotal recommends running it a few weeks before the upgrade and that you run `gpcheckcat` during a maintenance period. If necessary, you can resolve any issues found by the utility before the scheduled upgrade.

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>.

- During the migration process from Greenplum Database 4.2.x.x, a backup is made of some files and directories in `$MASTER_DATA_DIRECTORY`. Pivotal recommends that files and directories that are not used by Greenplum Database be backed up, if necessary, and removed from the `$MASTER_DATA_DIRECTORY` before migration. For information about the Greenplum Database migration utilities, see the *Greenplum Database Utility Guide*.

Important: If you intend to use an extension package with Greenplum Database 4.3.7.0, you must install and use a Greenplum Database extension packages (gppkg files and contrib modules) that are built for Greenplum Database 4.3.5.0 or later. For custom modules that were used with Greenplum Database 4.3.4.x and earlier, you must rebuild any modules that were built against the provided C language header files for use with Greenplum Database 4.3.5.0 or later.

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

- [Upgrading from 4.3.x to 4.3.7.0](#)
- [Upgrading from 4.3.x to 4.3.7.0 on Pivotal DCA Systems](#)
- [Upgrading from 4.2.x.x to 4.3.7.0](#)
- [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.x.x 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 4.3.7.0

An upgrade from 4.3.x to 4.3.7.0 involves stopping Greenplum Database, updating the Greenplum Database software binaries, upgrading and restarting Greenplum Database. If you are using Greenplum Extension packages, you must install and use Greenplum Database 4.3.5.0 or later extension packages. If you are using custom modules with the extensions, you must also use modules that were built for use with Greenplum Database 4.3.5.0 or later.

Important: If you are upgrading from Greenplum Database 4.3.x on a Pivotal DCA system, see *Upgrading from 4.3.x to 4.3.7.0 on Pivotal DCA Systems*. This section is for upgrading to Greenplum Database 4.3.7.0 on non-DCA systems.

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 11). The utility is in this Greenplum Database directory: `$GPHOME/share/postgresql/upgrade`

For information about the utility, see *fix_ao_upgrade.py Utility*.

Note: If your database contains append-optimized tables that were converted from Greenplum Database 4.2.x append-only tables, and you are upgrading from a 4.3.x release earlier than 4.3.6.0, run the `fix_visimap_owner.sql` script to fix a Greenplum Database append-optimized table issue (See step 12). The utility is in this Greenplum Database directory: `$GPHOME/share/postgresql/upgrade`

For information about the script, see *fix_visimap_owner.sql Script*.

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). This example uses the `-a` option to disable confirmation prompts:

```
$ gpstop -a
```

4. Run the installer for 4.3.7.0 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-4.3.7.0
```

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-4.3.7.0/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-4.3.7.0 /usr/local/greenplum-db
```

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

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

7. Run the `gpsegininstall` utility to install the 4.3.7.0 binaries on all the segment hosts specified in the *hostfile*. For example:

```
$ gpsegininstall -f hostfile
```

8. Rebuild any modules that were built against the provided C language header files for use with Greenplum Database 4.3.5.0 or later (for example, any shared library files for user-defined functions in `$GPHOME/lib`). See your operating system documentation and your system administrator for information about rebuilding and compiling modules such as shared libraries.
9. 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 `gppkg` usage details.
10. After all segment hosts have been upgraded, you can log in as the `gpadmin` user and restart your Greenplum Database system:

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

11. 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.

12. For databases that contain append-optimized tables that were created from Greenplum Database 4.2.x append-only tables, run the `fix_visimap_owner.sql` script. The script resolves an issue associated

with relations associated with append-optimized tables. For example, this command runs the script on the database `testdb`.

```
$ psql -d testdb1 -f $GPHOME/share/postgresql/upgrade/
fix_visimap_owner.sql
```

The script displays this prompt that allows you to display changes to the affected relations without performing the operation.

```
Dry run, without making any modifications (y/n)?
```

- Enter `y` to list ownership changes that would have been made. The owner of the relation is not changed.
- Enter `n` make the ownership changes and display the changes to relation ownership.

Note: Pivotal recommends that you run the script during low activity period. Heavy workloads do not affect database functionality but might affect performance.

13. 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 4.3.7.0 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_visimap_owner.sql Script

The SQL script `fix_visimap_owner.sql` resolves ownership issues related to visimap relations that are associated with append-optimized tables.

When upgrading from Greenplum Database 4.2.x to 4.3.x, the 4.2.x append-only tables are converted to 4.3 append-optimized tables. When upgrading from 4.2.x to Greenplum Database 4.3.x earlier than 4.3.6.0, the upgrade process incorrectly assigned the owner of visimap relations to `gpadmin`, not the owner of the associated append-optimized table.

If you are migrating to this release Greenplum Database from a 4.3.x release earlier than 4.3.6.0, run this SQL script as the `gpadmin` superuser to fix the incorrect assignment issue for a database.

```
$GPHOME/share/postgresql/upgrade/fix_visimap_owner.sql
```

When you run the script, it temporarily creates two functions that update the visimap relations ownership and displays this message that lets you perform a test run without changing ownership.

```
Dry run, without making any modifications (y/n)?
```

If you enter `y`, the script displays the changes that would have been made. The owner of the relation is not changed.

If you enter `n`, the script changes the owner of the relations and displays the changes that are made.

Before exiting, the script deletes the functions it created.

Note: If you are migrating from Greenplum Database 4.2.x directly to Greenplum Database 4.3.7.0 you do not need to run the `fix_visimap_owner.sql` script. Also, you can run this script on Greenplum Database 4.3.x earlier than 4.3.6.0 to fix the incorrect ownership assignment of visimap relations.

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.

Dropping Orphan Tables on Greenplum Database Segments

If you upgraded to Greenplum Database 4.3.6.0 and a user dropped a table, in some cases, the table would be dropped only on the Greenplum Database master, not on the Greenplum Database segments. This created orphan tables on Greenplum Database segments. See resolved issue [25926](#). This issue occurs only with Greenplum Database 4.3.6.0. However, the orphan tables remain in Greenplum Database after upgrading to 4.3.7.0.

For Greenplum Database 4.3.6.2 and later, the installation contains this Python script to check for and drop orphan tables on segments.

```
$GPHOME/share/postgresql/upgrade/fix_orphan_segment_tables.py
```

You can run this script on Greenplum Database 4.3.7.0 to check for and drop orphan tables.

The script performs these operations:

- Checks for orphan tables on segments and generates file that contains a list of the orphan tables.
- Deletes orphan tables specified in a text file.

You run the script as a Greenplum Database administrator. The script attempts to log into Greenplum Database as user who runs the script.

To check all databases in the Greenplum Database instance, run this command on the Greenplum Database master. Specify the *port* to connect to Greenplum Database.

```
$GPHOME/share/postgresql/upgrade/fix_orphan_segment_tables.py -p port
```

To check a single database, specify the option `-d database`.

The command generates a list of orphan tables in the text file `orphan_tables_file_timestamp`. You can review the list and, if needed, modify it.

To delete orphan tables on the Greenplum Database segments, run this command on the Greenplum Database master. Specify the *port* to connect to Greenplum Database and the file containing the orphan tables to delete.

```
$GPHOME/share/postgresql/upgrade/fix_orphan_segment_tables.py -p port -f
orphan_tables_file_timestamp
```

The script connects only to the databases required to drop orphan tables.

Note: Pivotal recommends that you run the script during a period of low activity to prevent any issues that might occur due to concurrent drop operations.

Upgrading from 4.3.x to 4.3.7.0 on Pivotal DCA Systems

Upgrading Greenplum Database from 4.3.x to 4.3.7.0 on a Pivotal DCA system involves stopping Greenplum Database, updating the Greenplum Database software binaries, and restarting Greenplum Database. If you are using Greenplum Extension packages, you must install and use Greenplum Database 4.3.5.0 or later extension packages. If you are using custom modules with the extensions, you must also use modules that were built for use with Greenplum Database 4.3.5.0 or later.

Important: Skip this section if you are *not* installing Greenplum Database 4.3.7.0 on DCA systems. This section is only for installing Greenplum Database 4.3.7.0 on DCA systems.

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 8). The utility is in this Greenplum Database directory: `$GPHOME/share/postgresql/upgrade`

For information about the utility, see [fix_ao_upgrade.py Utility](#).

1. Log in to your Greenplum Database master host as the Greenplum administrative user (`gpadmin`):

```
# su - gpadmin
```

2. Download or copy the installer file to the Greenplum Database master host.
3. 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`.

4. Perform a smart shutdown of your current Greenplum Database 4.3.x system (there can be no active connections to the database). This example uses the `-a` option to disable confirmation prompts:

```
$ gpstop -a
```

5. As root, run the Pivotal DCA installer for 4.3.7.0 on the Greenplum Database master host and specify the file `hostfile` that lists all hosts in the cluster. If necessary, copy `hostfile` to the directory containing the installer before running the installer.

This example command runs the installer for Greenplum Database 4.3.7.0 for Redhat Enterprise Linux 5.x.

```
# ./greenplum-db-appliance-4.3.7.0-build-1-RHEL5-x86_64.bin hostfile
```

The file `hostfile` is a text file that lists all hosts in the cluster, one host name per line.

6. Install Greenplum Database extension packages.

Important: Rebuild any modules that were built against the provided C language header files for use with Greenplum Database 4.3.5.0 or later (for example, any shared library files for user-defined functions in `$GPHOME/lib`). See your operating system documentation and your system administrator for information about rebuilding and compiling modules such as shared libraries.

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

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

8. 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.

9. 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 4.3.4.1 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 4.2.x.x to 4.3.7.0

This section describes how you can upgrade from Greenplum Database 4.2.x.x or later to Greenplum Database 4.3.7.0. 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.

Important: If you intend to use extension packages with Greenplum Database 4.3.5.0 or later, you must install and use a Greenplum Database extension packages (gppkg files and contrib modules) that are built for Greenplum Database 4.3.5.0 or later. For custom modules that were used with Greenplum Database 4.3.4.x and earlier, you must rebuild any modules that were built against the provided C language header files for use with Greenplum Database 4.3.5.0 or later.

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 (non-DCA)

Important: If you are installing Greenplum Database 4.3 on a Pivotal DCA system, see [Install the Greenplum Database 4.3 Software Binaries on DCA Systems](#). This section is for installing Greenplum Database 4.3 on non-DCA systems.

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

```
# unzip greenplum-db-4.3.7.0-PLATFORM.zip
```

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

```
# /bin/bash greenplum-db-4.3.7.0-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-4.3.7.0`), 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 4.3.7.0 installation. This example changes to the `gpadmin` user before sourcing the file:

```
# su - gpadmin
$ source /usr/local/greenplum-db-4.3.7.0/greenplum_path.sh
```

8. Run the `gpsegininstall` utility to install the 4.3.7.0 binaries on all the segment hosts specified in the `hostfile`. For example:

```
$ gpsegininstall -f hostfile
```

Install the Greenplum Database 4.3 Software Binaries on DCA Systems

Important: Skip this section if you are *not* installing Greenplum Database 4.3 on DCA systems. This section is only for installing Greenplum Database 4.3 on DCA systems.

1. Download or copy the installer file to the Greenplum Database master host.
2. As root, run the Pivotal DCA installer for 4.3.7.0 on the Greenplum Database master host and specify the file `hostfile` that lists all hosts in the cluster. If necessary, copy `hostfile` to the directory containing the installer before running the installer.

This example command runs the installer for Greenplum Database 4.3.7.0.

```
# ./greenplum-db-appliance-4.3.7.0-build-1-RHEL5-x86_64.bin hostfile
```

The file `hostfile` is a text file that lists all hosts in the cluster, one host name per line.

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. As `gpadmin`, source the path file from your old 4.2.x.x installation. For example:

```
$ source /usr/local/greenplum-db-4.2.7.0/greenplum_path.sh
```

On a DCA system, the path to the might be similar to `/usr/local/GP-4.2.8.1/greenplum_path.sh` depending on the installed version.

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. This example uses the `-a` option to disable confirmation prompts:

```
$ gpstop -a
```

5. Source the path file from your new 4.3.7.0 installation. For example:

```
$ source /usr/local/greenplum-db-4.3.7.0/greenplum_path.sh
```

On a DCA system, the path to the file would be similar to `/usr/local/GP-4.3.7.0/greenplum_path.sh`.

6. Update the Greenplum Database environment so it is referencing your new 4.3.7.0 installation.
 - a. For example, update the `greenplum-db` symbolic link on the master and standby master to point to the new 4.3.7.0 installation directory. For example (as root):

```
# rm -rf /usr/local/greenplum-db
# ln -s /usr/local/greenplum-db-4.3.7.0 /usr/local/greenplum-db
# chown -R gpadmin /usr/local/greenplum-db
```

On a DCA system, the `ln` command would specify the install directory created by the DCA installer. For example:

```
# ln -s /usr/local/GP-4.3.7.0 /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-4.3.7.0 /usr/local/greenplum-db
=> chown -R gpadmin /usr/local/greenplum-db
=> exit
```

On a DCA system, the `ln` command would specify the install directory created by the DCA installer. For example:

```
=> ln -s /usr/local/GP-4.3.7.0 /usr/local/greenplum-db
```

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. The command is run as `gpadmin`. This example runs the `gpmigrator_mirror` utility as `gpadmin`:

```
$ gpmigrator_mirror --check-only
  /usr/local/greenplum-db-4.2.6.3
  /usr/local/greenplum-db-4.3.7.0
```

On a DCA system, the old `GPHOME` location might be similar to `/usr/local/GP-4.2.8.1` (depending on the old installed version) and the new `GPHOME` location would be similar to `/usr/local/GP-4.3.7.0`.

8. As `gpadmin`, run the 4.3.7.0 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:

```
$ gpmigrator_mirror /usr/local/greenplum-db-4.2.7.0
  /usr/local/greenplum-db-4.3.6.2
```

On a DCA system, the old `GPHOME` location might be similar to `/usr/local/GP-4.2.8.1` (depending on the old installed version) and the new `GPHOME` location would be similar to `/usr/local/GP-4.3.7.0`.

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 4.3.7.0 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 4.3.7.0 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.x Load Tools package. Application Packages are available at [Pivotal Network](#). For information about `gpfdist`, see the *Greenplum Database 4.3 Administrator Guide*.
3. Rebuild any modules that were built against the provided C language header files for use with Greenplum Database 4.3.5.0 or later. (for example, any shared library files for user-defined functions in `$GPHOME/lib`). See your operating system documentation and your system administrator for information about rebuilding and compiling modules such as shared libraries.
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 Utility Guide* for `gppkg` 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). See the Greenplum Command Center documentation for information about installing and configuring Greenplum Command Center.

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

Command Center Console packages are available from [Pivotal Network](#).

6. (optional) Check the status of Greenplum Database. For example, you can run the Greenplum Database utility `gpstate` to display status information of a running Greenplum Database.

```
$ gpstate
```

7. Inform all database users of the completed upgrade. Tell users to update their environment to source the Greenplum Database 4.3.7.0 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 4.3.7.0. Detailed upgrade instructions are provided in [Upgrading from 4.2.x.x to 4.3.7.0](#).

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.x 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.x system will be up and running.
Post-Upgrade (on your 4.3 system)	
* The 4.3.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.x installation.
<input type="checkbox"/>	Download and install any Greenplum Database extensions.
<input type="checkbox"/>	(Optional) Install the latest Greenplum Command Center 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 4.3.7.0.

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 4.3.7.0*.

For Users Running Greenplum Database 4.0.x.x

Users on a release prior to 4.1.x.x cannot upgrade directly to 4.3.7.0.

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 4.3.7.0*.

For Users Running Greenplum Database 3.3.x.x

Users on a release prior to 4.0.x.x cannot upgrade directly to 4.3.7.0.

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 4.3.7.0*.

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

Client Package	Description of Contents	Client Version	Server Versions
Greenplum Connectivity	Standard PostgreSQL Database Drivers (ODBC, JDBC ¹) 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.	1.3.0.2	4.3

Note: ¹The JDBC drivers that are shipped with the Greenplum Connectivity Tools are official PostgreSQL JDBC drivers built by the PostgreSQL JDBC Driver team (<https://jdbc.postgresql.org>).

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 6.x (RHEL 6)
- Red Hat Enterprise Linux x86_64 (RHEL 5)
- 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.

Pivotal supplies separate PL/Perl extension packages for Red Hat Enterprise Linux 7.x, 6.x and 5.x. Ensure you install the correct package for your operating system.

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
PL/Java 1.3 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	5.16.3 on RHEL 7.x 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

Note: Greenplum Database 4.3.7.0 does not support the PostGIS 1.0 extension package.

Greenplum Database 4.3.7.0 supports these minimum Greenplum Database extensions package versions.

Table 6: Greenplum Database 4.3.7.0 Package Version

Greenplum Database Extension	Minimum Package Version
PostGIS	2.0.1 and release <code>gpdb4.3orca</code>
PL/Java	1.3 and release <code>gpdb4.3orca</code>
PL/Perl	1.2 and release <code>gpdb4.3orca</code>
PL/R	2.1 and release <code>gpdb4.3orca</code>
Pgcrypto	1.2 and release <code>gpdb4.3orca</code>
MADlib	1.9.3 and release <code>gpdb4.3orca</code>

Note: Extension packages for Greenplum Database 4.3.4.x and earlier are not compatible with Greenplum Database 4.3.5.0 and later due to the introduction of Pivotal Query Optimizer. Also, extension packages for Greenplum Database 4.3.5.0 and later are not compatible with Greenplum Database 4.3.4.x and earlier.

To use extension packages with Greenplum Database 4.3.7.0, you must install and use Greenplum Database extension packages (gppkg files and contrib modules) that are built for Greenplum Database 4.3.5.0 or later. For custom modules that were used with Greenplum Database 4.3.4.x and earlier, you must rebuild any modules that were built against the provided C language header files for use with Greenplum Database 4.3.7.0.

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_pv2.0.1_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 3.0	gphd-3.0
	Pivotal HD 2.0, 2.1	gphd-2.0
	Pivotal HD 1.0 ¹	
Greenplum HD	Greenplum HD 1.2	gphd-1.2
	Greenplum HD 1.1	gphd-1.1 (default)
Cloudera	CDH 5.2, 5.3, 5.4.x, 5.5.x	cdh5
	CDH 5.0, 5.1	cdh4.1
	CDH 4.1 ² - CDH 4.7	cdh4.1
Hortonworks Data Platform	HDP 2.1, 2.2, 2.3	hdp2
MapR ³	MapR 4.x	gpmr-1.2
	MapR 1.x, 2.x, 3.x	gpmr-1.0
Apache Hadoop	2.x	hadoop2

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 4.3.7.0 Documentation

For the latest Greenplum Database documentation go to [Pivotal Documentation](#). Greenplum documentation is provided in HTML and PDF formats.

Table 8: Greenplum Database Documentation

Title	Revision
<i>Greenplum Database 4.3.7.0 Release Notes</i>	A02
<i>Greenplum Database 4.3 Installation Guide</i>	A11
<i>Greenplum Database 4.3 Administrator Guide</i>	A13
<i>Greenplum Database 4.3 Reference Guide</i>	A13
<i>Greenplum Database 4.3 Utility Guide</i>	A14

Title	Revision
<i>Greenplum Database 4.3 Client Tools for UNIX</i>	A06
<i>Greenplum Database 4.3 Client Tools for Windows</i>	A05
<i>Greenplum Database 4.3 Connectivity Tools for UNIX</i>	A05
<i>Greenplum Database 4.3 Connectivity Tools for Windows</i>	A04
<i>Greenplum Database 4.3 Load Tools for UNIX</i>	A08
<i>Greenplum Database 4.3 Load Tools for Windows</i>	A08
<i>Greenplum Command Center 2.0 Administrator Guide</i> <i>Greenplum Workload Manager 1.0 User Guide</i>	A01 A01